



FSx ONTAP pour MLOps

NetApp artificial intelligence solutions

NetApp
August 18, 2025

Sommaire

- FSx ONTAP pour MLOps 1
 - Amazon FSx for NetApp ONTAP (FSx ONTAP) pour MLOps 1
- Partie 1 - Intégration Amazon FSx for NetApp ONTAP (FSx ONTAP) en tant que compartiment S3 privé dans AWS SageMaker 1
 - Introduction 1
 - Guide de l'utilisateur 1
 - Liste de contrôle de débogage utile 14
 - FAQ (au 27 septembre 2023). 15
- Partie 2 - Exploiter AWS Amazon FSx for NetApp ONTAP (FSx ONTAP) comme source de données pour la formation de modèles dans SageMaker 15
 - Introduction 15
 - Qu'est-ce que FSx ONTAP 15
 - Condition préalable 16
 - Présentation de l'intégration. 16
 - Intégration étape par étape 17
- Partie 3 - Création d'un pipeline MLOps simplifié (CI/CT/CD) 24
 - Introduction 24
 - Manifeste 24
 - Condition préalable 25
 - Architecture 25
 - Configuration étape par étape 25

FSx ONTAP pour MLOps

Amazon FSx for NetApp ONTAP (FSx ONTAP) pour MLOps

Cette section se penche sur l'application pratique du développement de l'infrastructure d'IA, en fournissant une procédure pas à pas de bout en bout de la construction d'un pipeline MLOps à l'aide de FSx ONTAP. Composé de trois exemples complets, il vous guide pour répondre à vos besoins MLOps via cette puissante plateforme de gestion de données.

Ces articles portent sur :

1. ["Partie 1 - Intégration Amazon FSx for NetApp ONTAP \(FSx ONTAP\) en tant que compartiment S3 privé dans AWS SageMaker"](#)
2. ["Partie 2 - Exploiter Amazon FSx for NetApp ONTAP \(FSx ONTAP\) comme source de données pour la formation de modèles dans SageMaker"](#)
3. ["Partie 3 - Création d'un pipeline MLOps simplifié \(CI/CT/CD\)"](#)

À la fin de cette section, vous aurez acquis une solide compréhension de la manière d'utiliser FSx ONTAP pour rationaliser les processus MLOps.

Partie 1 - Intégration Amazon FSx for NetApp ONTAP (FSx ONTAP) en tant que compartiment S3 privé dans AWS SageMaker

Cette section fournit un guide sur la configuration de FSx ONTAP en tant que bucket S3 privé à l'aide d'AWS SageMaker.

Introduction

En utilisant SageMaker comme exemple, cette page fournit des conseils sur la configuration de FSx ONTAP en tant que bucket S3 privé.

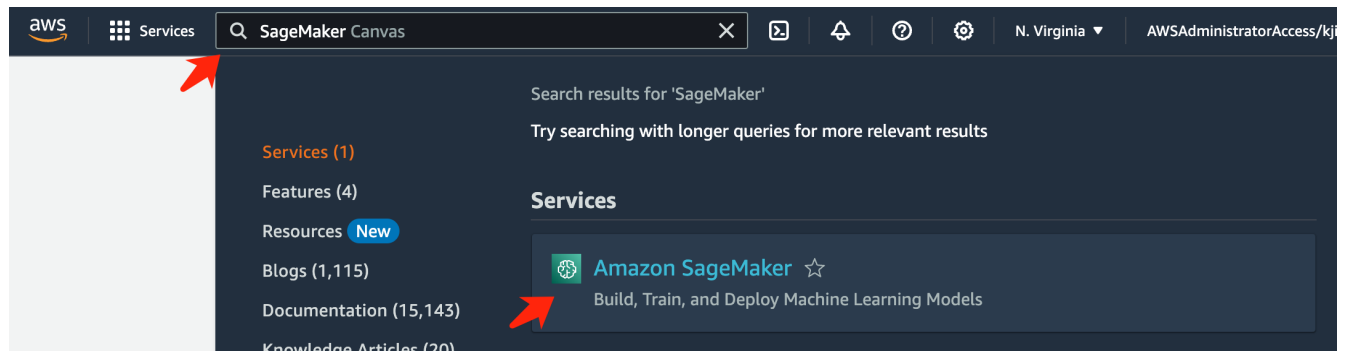
Pour plus d'informations sur FSx ONTAP, veuillez consulter cette présentation (["Lien vidéo"](#))

Guide de l'utilisateur

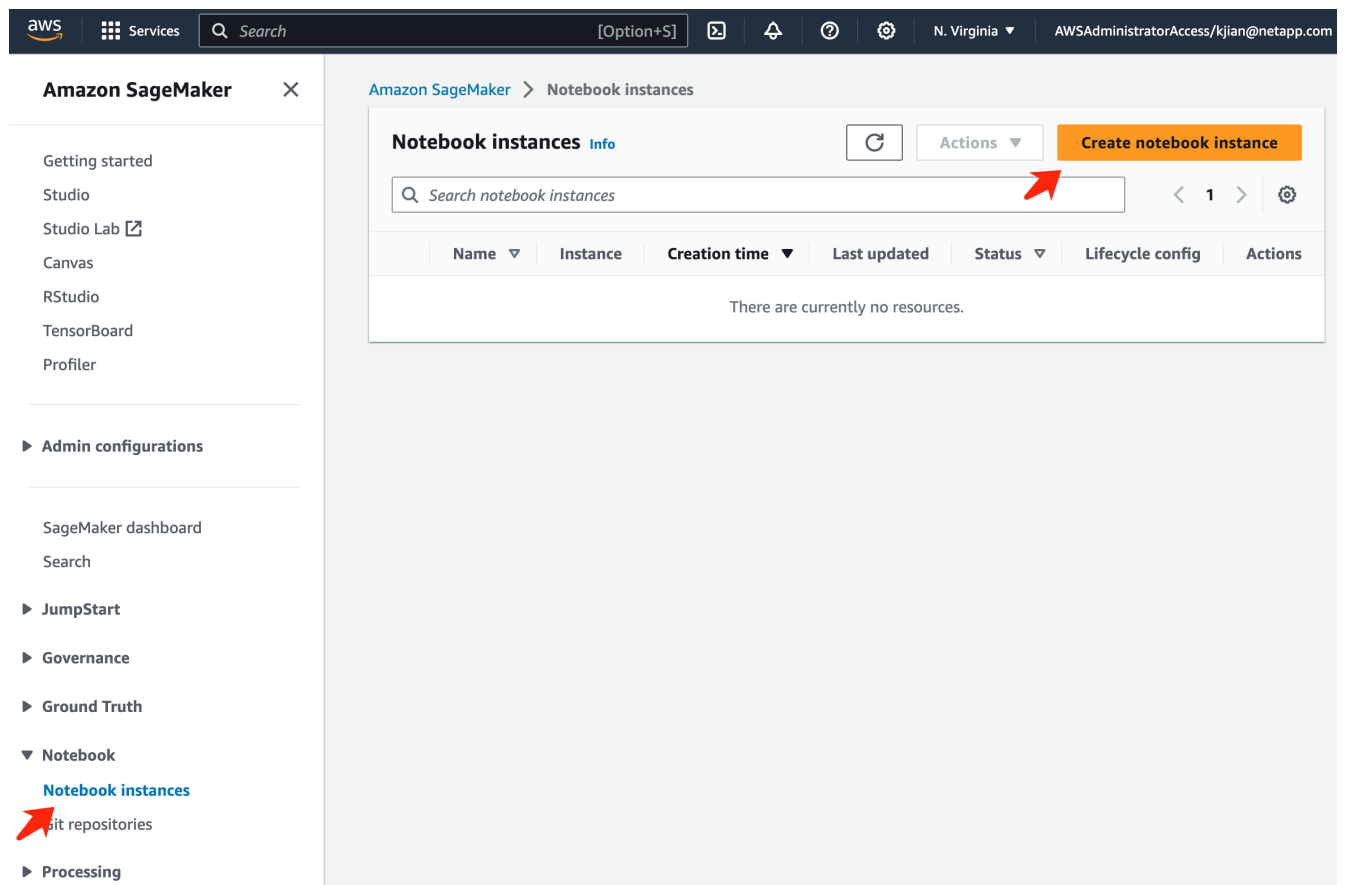
Création de serveur

Créer une instance de bloc-notes SageMaker

1. Ouvrez la console AWS. Dans le panneau de recherche, recherchez SageMaker et cliquez sur le service **Amazon SageMaker**.



2. Ouvrez les **Instances de bloc-notes** sous l'onglet Bloc-notes, cliquez sur le bouton orange **Créer une instance de bloc-notes**.



3. Dans la page de création, saisissez le **Nom de l'instance du notebook**. Développez le panneau **Réseau**. Laissez les autres entrées par défaut et sélectionnez un **VPC**, un **Sous-réseau** et un ou plusieurs **Groupe de sécurité**. (Ce **VPC** et **Sous-réseau** seront utilisés pour créer ultérieurement le système de fichiers FSx ONTAP) Cliquez sur le bouton orange **Créer une instance de notebook** en bas à droite.

Amazon SageMaker > Notebook instances > Create notebook instance

Create notebook instance

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

Notebook instance settings

Notebook instance name
fsxn-demo

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-00060df0d9f562672 (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](#)

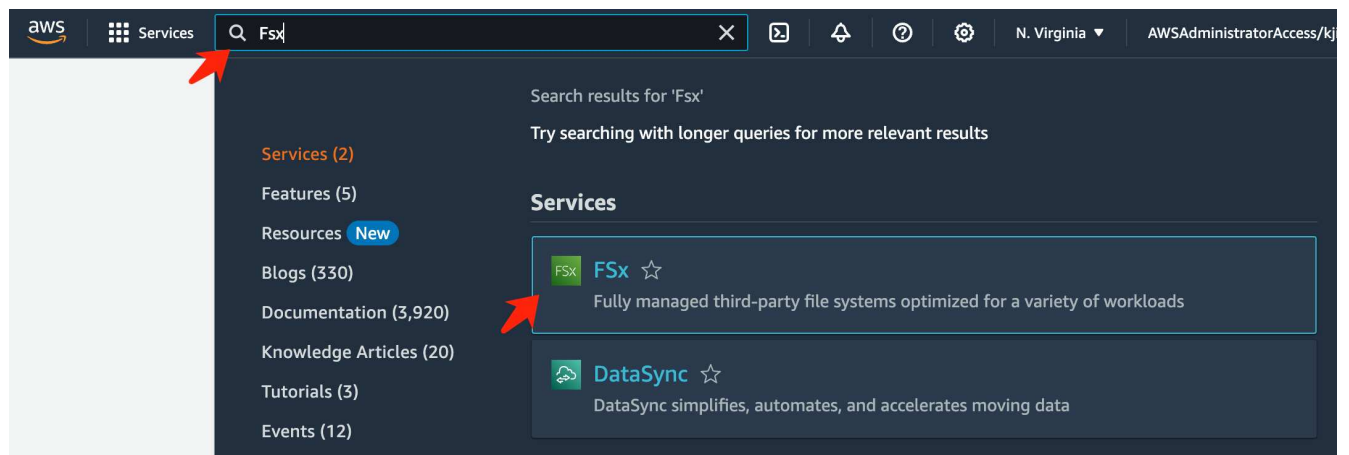
► Git repositories - optional

► Tags - optional

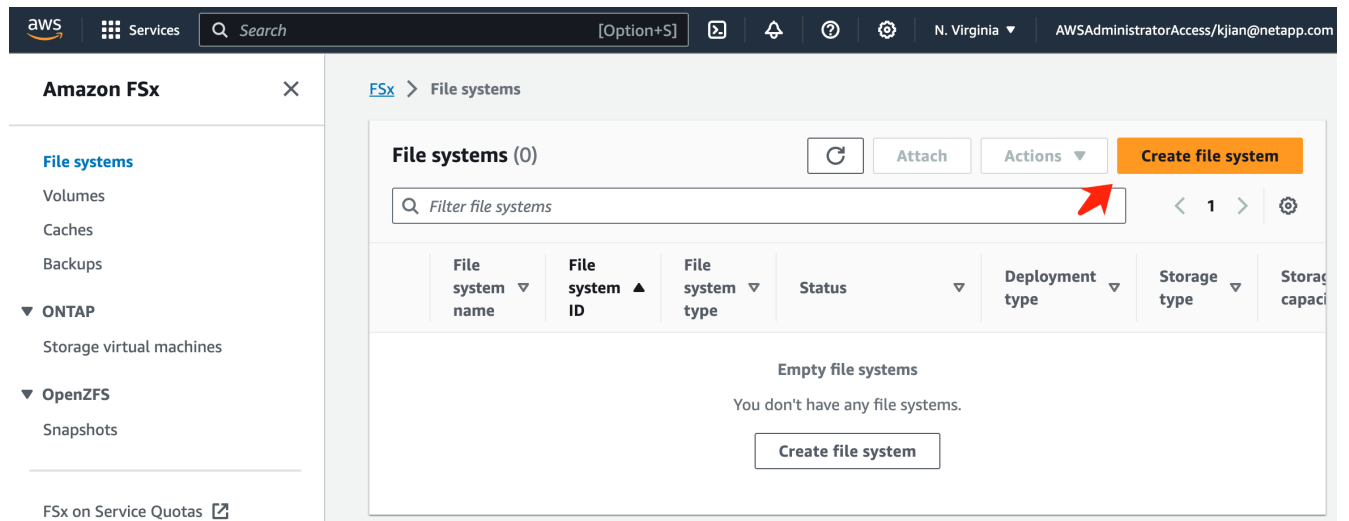
Cancel Create notebook instance

Créer un système de fichiers FSx ONTAP

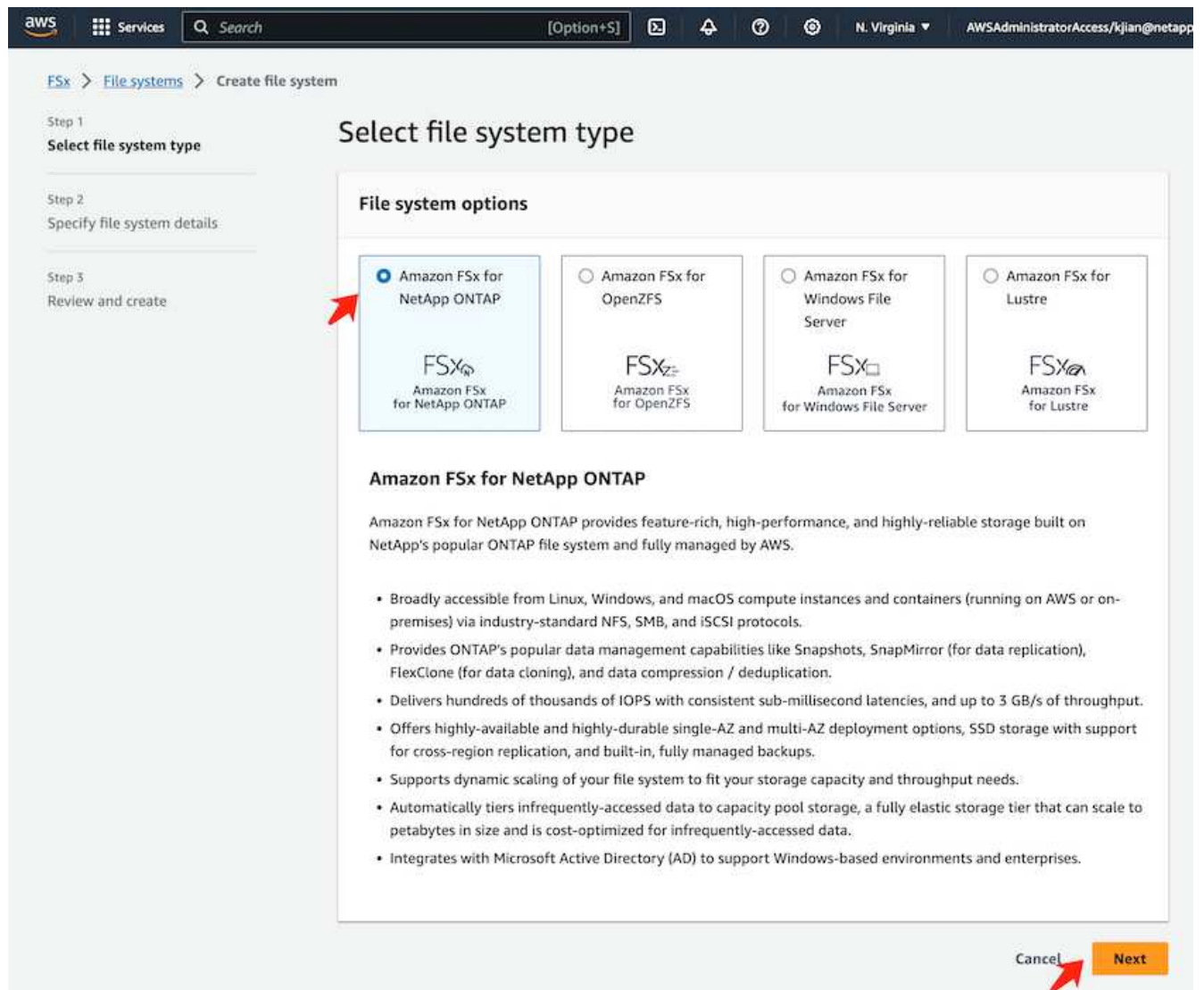
1. Ouvrez la console AWS. Dans le panneau de recherche, recherchez Fsx et cliquez sur le service **FSx**.



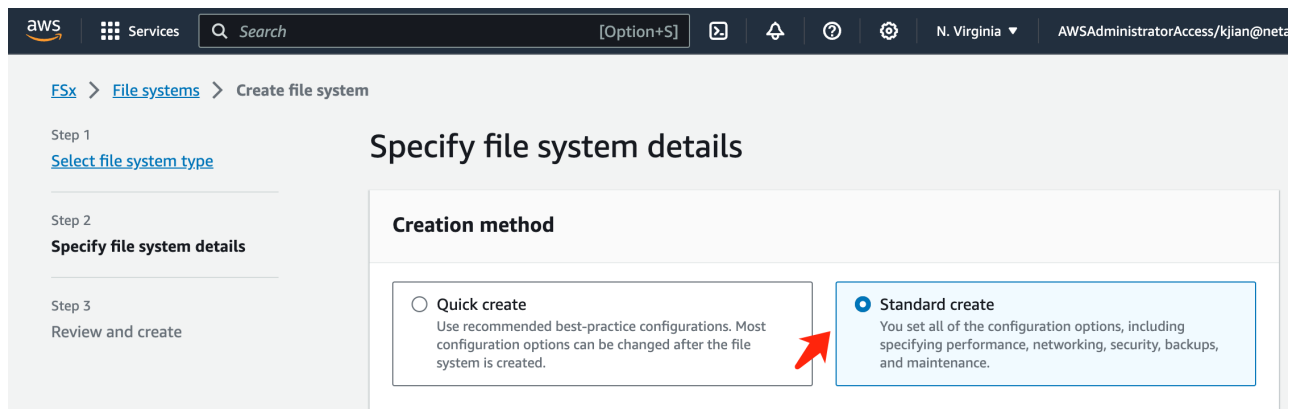
2. Cliquez sur **Créer un système de fichiers**.



3. Sélectionnez la première carte **FSx ONTAP** et cliquez sur **Suivant**.



4. Dans la page de configuration des détails.
- a. Sélectionnez l'option **Création standard**.



- b. Saisissez le **Nom du système de fichiers** et la **Capacité de stockage SSD**.

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. Assurez-vous d'utiliser le **VPC** et le **sous-réseau** identiques à l'instance **SageMaker Notebook**.

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. Saisissez le nom de la **machine virtuelle de stockage** et **Spécifiez un mot de passe** pour votre SVM (machine virtuelle de stockage).

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. Laissez les autres entrées par défaut et cliquez sur le bouton orange **Suivant** en bas à droite.

► Backup and maintenance - optional

► Tags - optional

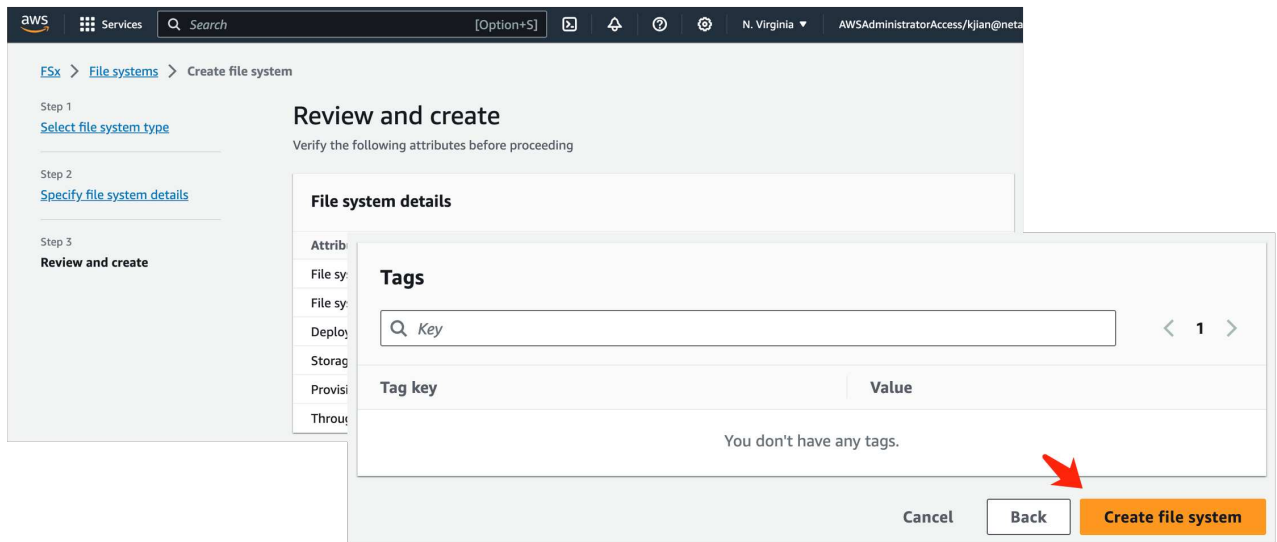
Cancel

Back

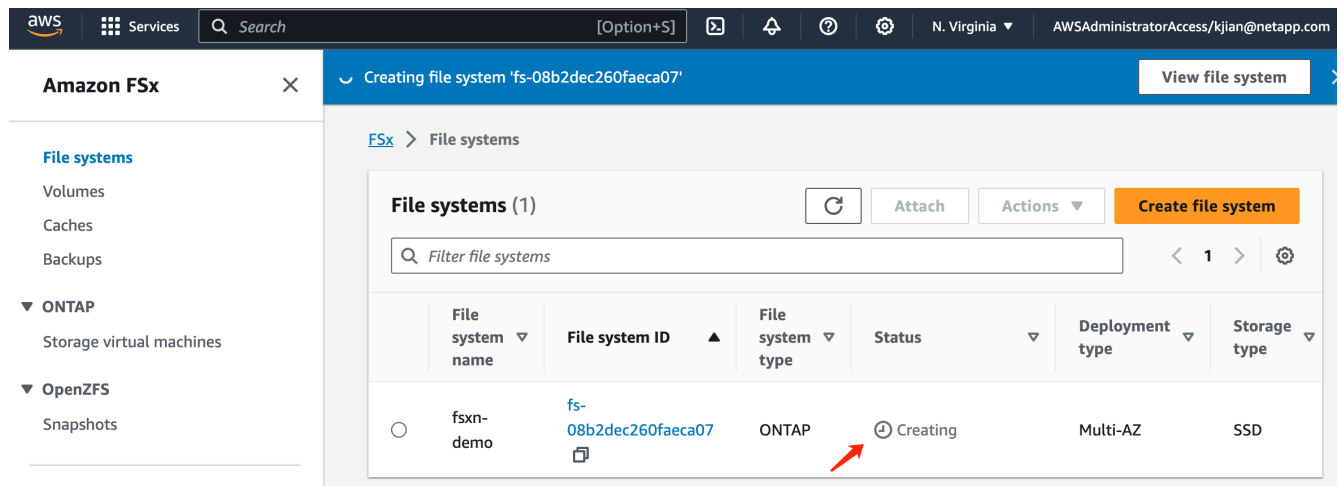
Next

f. Cliquez sur le bouton orange **Créer un système de fichiers** en bas à droite de la page de révision.

8



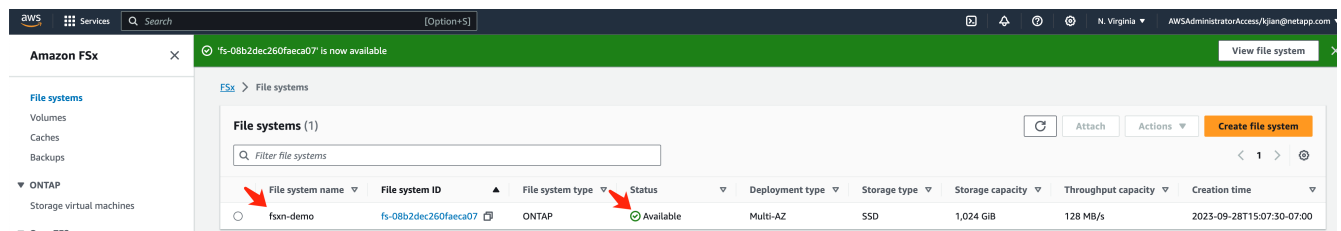
5. Le démarrage du système de fichiers FSx peut prendre environ **20 à 40 minutes**.



Configuration du serveur

Configuration ONTAP

1. Ouvrez le système de fichiers FSx créé. Veuillez vous assurer que le statut est **Disponible**.



2. Sélectionnez l'onglet **Administration** et conservez le **Point de terminaison de gestion - Adresse IP** et le ***Nom d'utilisateur de l'administrateur ONTAP ***.

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
		Update

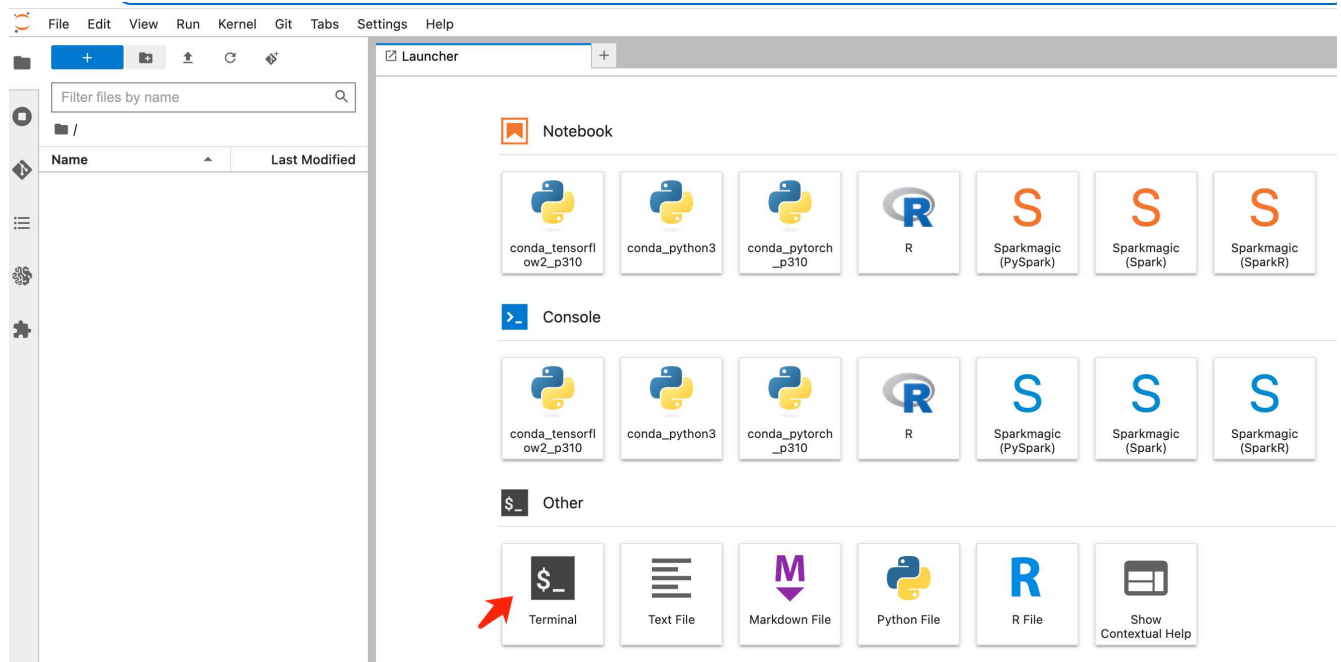
3. Ouvrez l'instance **SageMaker Notebook** créée et cliquez sur **Ouvrir JupyterLab**.

Amazon SageMaker Notebook instances

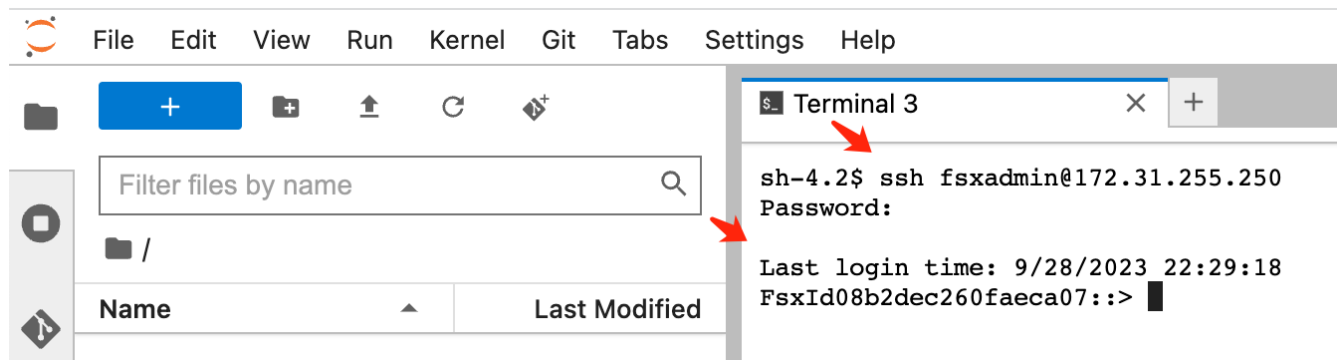
Notebook instances

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. Dans la page Jupyter Lab, ouvrez un nouveau **Terminal**.



- Entrez la commande `ssh <nom d'utilisateur administrateur>@<IP du serveur ONTAP >` pour vous connecter au système de fichiers FSx ONTAP . (Le nom d'utilisateur et l'adresse IP sont récupérés à l'étape 2) Veuillez utiliser le mot de passe utilisé lors de la création de la **machine virtuelle de stockage**.



- Exécutez les commandes dans l'ordre suivant. Nous utilisons **fsxn-ontap** comme nom pour le **nom du bucket S3 privé FSx ONTAP ***. Veuillez utiliser le ***nom de la machine virtuelle de stockage** pour l'argument **-vserver**.

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

vserver object-store-server user create -vserver fsxn-svm-demo -user
s3user

vserver object-store-server group create -name s3group -users s3user
-policies FullAccess

vserver object-store-server bucket create fsxn-ontap -vserver fsxn-svm-
demo -type nas -nas-path /vol1
```

```

sh-4.2$ ssh fsxadmin@172.31.255.250
Password:
Last login time: 9/28/2023 22:29:34
FsxId08b2dec260faeca07:~> vsserver object-store-server create -vsserver fsxn-svm-demo -object-store-server fsx_s3 -is-http-enabled true -is-https-enabled false
FsxId08b2dec260faeca07:~> vsserver object-store-server user create -vsserver fsxn-svm-demo -user s3user
FsxId08b2dec260faeca07:~> vsserver object-store-server group create -name s3group -users s3user -policies FullAccess
FsxId08b2dec260faeca07:~> vsserver object-store-server bucket create fsxn-ontap -vsserver fsxn-svm-demo -type nas -nas-path /vol1
FsxId08b2dec260faeca07:~>

```

- Exécutez les commandes ci-dessous pour récupérer l'adresse IP et les informations d'identification du point de terminaison pour FSx ONTAP privé S3.

```

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

set adv

vsserver object-store-server user show

```

- Conservez l'adresse IP et les informations d'identification du point de terminaison pour une utilisation ultérieure.

```

sh-4.2$ ssh fsxadmin@172.31.255.250
Password:
Last login time: 9/28/2023 22:32:42
FsxId08b2dec260faeca07:~> network interface show -vsserver fsxn-svm-demo -lif nfs_smb_management_1

    Vserver Name: fsxn-svm-demo
    Logical Interface Name: nfs_smb_management_1
    Service Policy: default-data-files
    Service List: data-core, data-nfs, data-cifs,
                  management-ssh, management-https,
                  data-s3-server, data-dns-server
    (DEPRECATED)-Role: data
    Data Protocol: nfs, cifs, s3
    Network Address: FSx IP Address
    Netmask: 255.255.255.192
    Bits in the Netmask: 26
    Is VIP LIF: false
    Subnet Name: -
    Home Node: FsxId08b2dec260faeca07-01
    Home Port: e0e
    Current Node: FsxId08b2dec260faeca07-01
    Current Port: e0e
    Operational Status: up
    Extended Status: -
    Is Home: true
    Administrative Status: up
    Failover Policy: system-defined
    (DEPRECATED)-Firewall Policy: data
    Auto Revert: true
    Fully Qualified DNS Zone Name: none
    DNS Query Listen Enable: false
    Failover Group Name: Fsxn
    FCP WFPN: -
    Address family: ipv4
    Comment: -
    IPspace of LIF: Default
    Is Dynamic DNS Update Enabled?: true
    Probe-port for Cloud Load Balancer: -
    Broadcast Domain: Fsxn
    Vserver Type: data
    Required RDMA offload protocols: -

FsxId08b2dec260faeca07:~> set adv

Warning: These advanced commands are potentially dangerous; use them only when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y

FsxId08b2dec260faeca07:~> vsserver object-store-server user show

-----
Vserver  User      ID      Access Key      Secret Key
-----
fsxn-svm-demo
  Comment: Root User
fsxn-svm-demo
  s3user      1      AWS Access Key ID      AWS Secret Access Key
-----

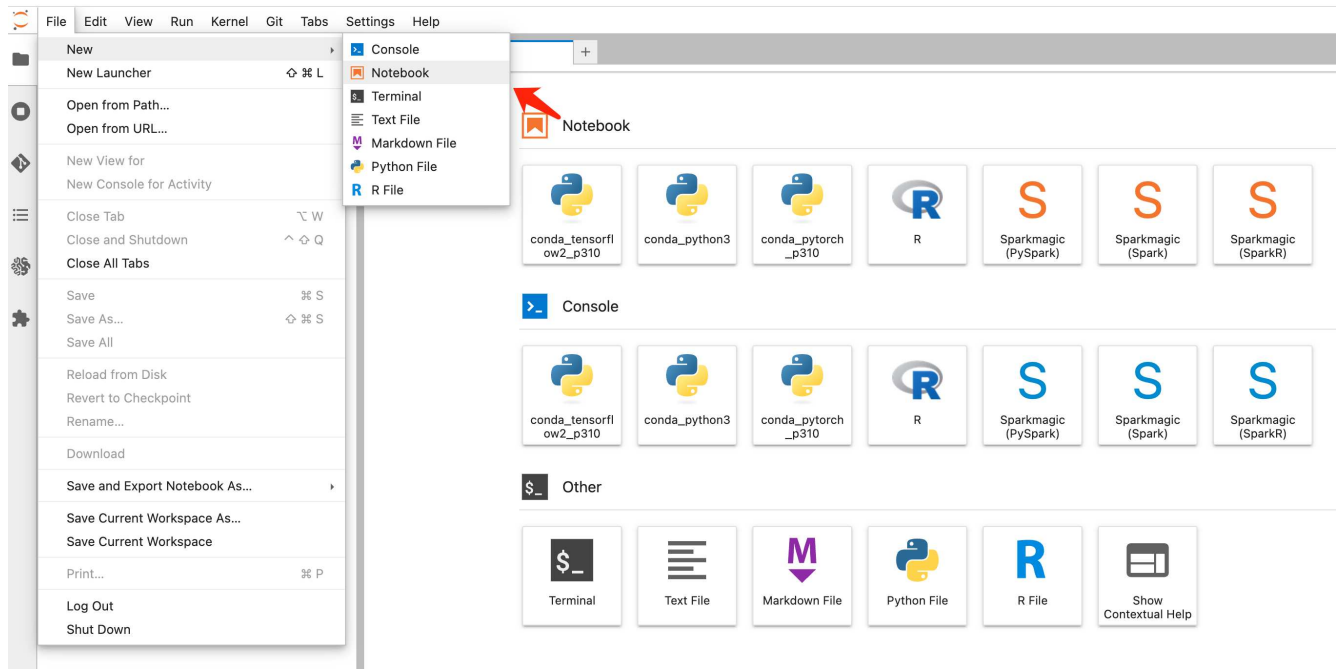
2 entries were displayed.

FsxId08b2dec260faeca07:~>

```

Configuration du client

1. Dans l'instance SageMaker Notebook, créez un nouveau notebook Jupyter.



2. Utilisez le code ci-dessous comme solution de contournement pour télécharger des fichiers vers le bucket S3 privé FSx ONTAP . Pour un exemple de code complet, veuillez vous référer à ce bloc-notes."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 Fsx 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']

```

Ceci conclut l'intégration entre FSx ONTAP et l'instance SageMaker.

Liste de contrôle de débogage utile

- Assurez-vous que l'instance SageMaker Notebook et le système de fichiers FSx ONTAP se trouvent dans le même VPC.
- N'oubliez pas d'exécuter la commande **set dev** sur ONTAP pour définir le niveau de privilège sur **dev**.

FAQ (au 27 septembre 2023)

Q : Pourquoi est-ce que j'obtiens l'erreur « **Une erreur s'est produite (NotImplemented) lors de l'appel de l'opération CreateMultipartUpload : la commande s3 que vous avez demandée n'est pas implémentée** » lors du téléchargement de fichiers vers FSx ONTAP ?

R : En tant que bucket S3 privé, FSx ONTAP prend en charge le téléchargement de fichiers jusqu'à 100 Mo. Lors de l'utilisation du protocole S3, les fichiers de plus de 100 Mo sont divisés en morceaux de 100 Mo et la fonction « CreateMultipartUpload » est appelée. Cependant, l'implémentation actuelle de FSx ONTAP private S3 ne prend pas en charge cette fonction.

Q : Pourquoi est-ce que j'obtiens l'erreur « **Une erreur s'est produite (AccessDenied) lors de l'appel des opérations PutObject : Accès refusé** » lors du téléchargement de fichiers vers FSx ONTAP ?

R : Pour accéder au compartiment S3 privé FSx ONTAP à partir d'une instance SageMaker Notebook, remplacez les informations d'identification AWS par les informations d'identification FSx ONTAP . Cependant, l'octroi d'une autorisation d'écriture à l'instance nécessite une solution de contournement qui implique le montage du bucket et l'exécution de la commande shell « chmod » pour modifier les autorisations.

Q : Comment puis-je intégrer le bucket S3 privé FSx ONTAP avec d'autres services SageMaker ML ?

R : Malheureusement, le SDK des services SageMaker ne fournit pas de moyen de spécifier le point de terminaison du bucket S3 privé. Par conséquent, FSx ONTAP S3 n'est pas compatible avec les services SageMaker tels que Sagemaker Data Wrangler, Sagemaker Clarify, Sagemaker Glue, Sagemaker Athena, Sagemaker AutoML et autres.

Partie 2 - Exploiter AWS Amazon FSx for NetApp ONTAP (FSx ONTAP) comme source de données pour la formation de modèles dans SageMaker

Cet article est un didacticiel sur l'utilisation Amazon FSx for NetApp ONTAP (FSx ONTAP) pour la formation de modèles PyTorch dans SageMaker, en particulier pour un projet de classification de la qualité des pneus.

Introduction

Ce didacticiel propose un exemple pratique d'un projet de classification de vision par ordinateur, offrant une expérience pratique dans la création de modèles ML qui utilisent FSx ONTAP comme source de données dans l'environnement SageMaker. Le projet se concentre sur l'utilisation de PyTorch, un framework d'apprentissage en profondeur, pour classer la qualité des pneus en fonction des images de pneus. Il met l'accent sur le développement de modèles d'apprentissage automatique utilisant FSx ONTAP comme source de données dans Amazon SageMaker.

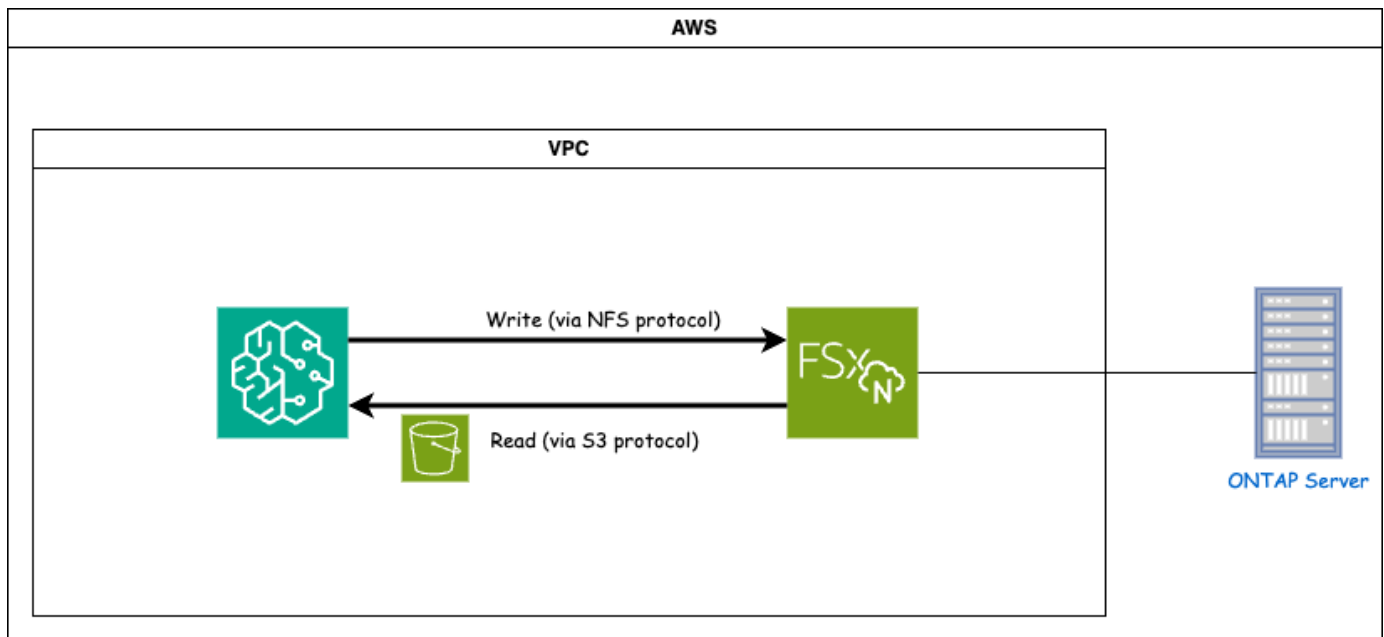
Qu'est-ce que FSx ONTAP

Amazon FSx ONTAP est en effet une solution de stockage entièrement gérée proposée par AWS. Il exploite le système de fichiers ONTAP de NetApp pour fournir un stockage fiable et hautes performances. Avec la prise en charge de protocoles tels que NFS, SMB et iSCSI, il permet un accès transparent à partir de différentes instances de calcul et conteneurs. Le service est conçu pour offrir des performances exceptionnelles, garantissant des opérations de données rapides et efficaces. Il offre également une haute disponibilité et une durabilité élevée, garantissant que vos données restent accessibles et protégées. De plus, la capacité de stockage d'Amazon FSx ONTAP est évolutive, ce qui vous permet de l'ajuster facilement en fonction de vos

besoins.

Condition préalable

Environnement réseau



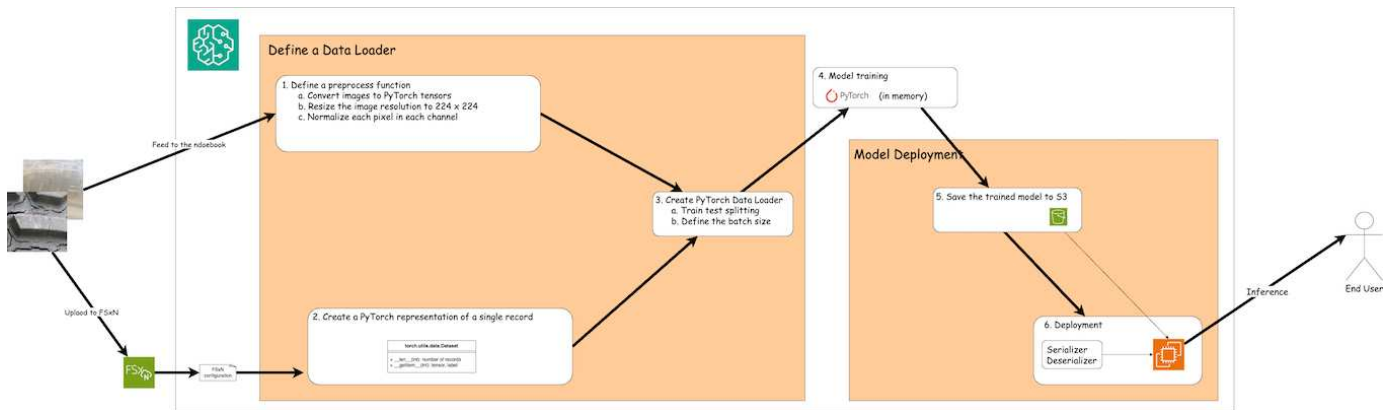
FSx ONTAP (Amazon FSx ONTAP) est un service de stockage AWS. Il comprend un système de fichiers exécuté sur le système NetApp ONTAP et une machine virtuelle système gérée par AWS (SVM) qui s'y connecte. Dans le diagramme fourni, le serveur NetApp ONTAP géré par AWS est situé en dehors du VPC. Le SVM sert d'intermédiaire entre SageMaker et le système NetApp ONTAP, recevant les demandes d'opération de SageMaker et les transmettant au stockage sous-jacent. Pour accéder à FSx ONTAP, SageMaker doit être placé dans le même VPC que le déploiement FSx ONTAP. Cette configuration assure la communication et l'accès aux données entre SageMaker et FSx ONTAP.

Accès aux données

Dans les scénarios réels, les scientifiques des données utilisent généralement les données existantes stockées dans FSx ONTAP pour créer leurs modèles d'apprentissage automatique. Cependant, à des fins de démonstration, étant donné que le système de fichiers FSx ONTAP est initialement vide après sa création, il est nécessaire de télécharger manuellement les données de formation. Cela peut être réalisé en montant FSx ONTAP en tant que volume sur SageMaker. Une fois le système de fichiers monté avec succès, vous pouvez télécharger votre ensemble de données vers l'emplacement monté, le rendant ainsi accessible pour la formation de vos modèles dans l'environnement SageMaker. Cette approche vous permet d'exploiter la capacité de stockage et les capacités de FSx ONTAP tout en travaillant avec SageMaker pour le développement et la formation de modèles.

Le processus de lecture des données implique la configuration de FSx ONTAP en tant que bucket S3 privé. Pour connaître les instructions de configuration détaillées, veuillez vous référer à ["Partie 1 - Intégration Amazon FSx for NetApp ONTAP \(FSx ONTAP\) en tant que compartiment S3 privé dans AWS SageMaker"](#)

Présentation de l'intégration



Le flux de travail d'utilisation des données de formation dans FSx ONTAP pour créer un modèle d'apprentissage en profondeur dans SageMaker peut être résumé en trois étapes principales : définition du chargeur de données, formation du modèle et déploiement. À un niveau élevé, ces étapes constituent la base d'un pipeline MLOps. Cependant, chaque étape implique plusieurs sous-étapes détaillées pour une mise en œuvre complète. Ces sous-étapes englobent diverses tâches telles que le prétraitement des données, le fractionnement des ensembles de données, la configuration du modèle, le réglage des hyperparamètres, l'évaluation du modèle et le déploiement du modèle. Ces étapes garantissent un processus complet et efficace pour la création et le déploiement de modèles d'apprentissage en profondeur à l'aide de données de formation de FSx ONTAP dans l'environnement SageMaker.

Intégration étape par étape

Loader de données

Afin de former un réseau d'apprentissage profond PyTorch avec des données, un chargeur de données est créé pour faciliter l'alimentation des données. Le chargeur de données définit non seulement la taille du lot, mais détermine également la procédure de lecture et de prétraitement de chaque enregistrement au sein du lot. En configurant le chargeur de données, nous pouvons gérer le traitement des données par lots, permettant ainsi la formation du réseau d'apprentissage en profondeur.

Le chargeur de données se compose de 3 parties.

Fonction de prétraitement

```
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]
    )
])
```

L'extrait de code ci-dessus illustre la définition des transformations de prétraitement d'image à l'aide du module **torchvision.transforms**. Dans ce tutoriel, l'objet de prétraitement est créé pour appliquer une série de transformations. Tout d'abord, la transformation **ToTensor()** convertit l'image en une représentation tensorielle.

Par la suite, la transformation **Resize 224,224** redimensionne l'image à une taille fixe de 224x224 pixels. Enfin, la transformation **Normalize()** normalise les valeurs du tenseur en soustrayant la moyenne et en divisant par l'écart type le long de chaque canal. Les valeurs moyennes et d'écart type utilisées pour la normalisation sont couramment utilisées dans les modèles de réseaux neuronaux pré-entraînés. Dans l'ensemble, ce code prépare les données d'image pour un traitement ultérieur ou une entrée dans un modèle pré-entraîné en les convertissant en tenseur, en les redimensionnant et en normalisant les valeurs des pixels.

La classe de jeu de données PyTorch

```
import torch
from io import BytesIO
from PIL import Image

class FSxNImageDataset(torch.utils.data.Dataset):
    def __init__(self, bucket, prefix='', preprocess=None):
        self.image_keys = [
            s3_obj.key
            for s3_obj in list(bucket.objects.filter(Prefix=prefix).all())
        ]
        self.preprocess = preprocess

    def __len__(self):
        return len(self.image_keys)

    def __getitem__(self, index):
        key = self.image_keys[index]
        response = bucket.Object(key)

        label = 1 if key[13:].startswith('defective') else 0

        image_bytes = response.get()['Body'].read()
        image = Image.open(BytesIO(image_bytes))
        if image.mode == 'L':
            image = image.convert('RGB')

        if self.preprocess is not None:
            image = self.preprocess(image)
        return image, label
```

Cette classe fournit des fonctionnalités permettant d'obtenir le nombre total d'enregistrements dans l'ensemble de données et définit la méthode de lecture des données pour chaque enregistrement. Dans la fonction **getitem**, le code utilise l'objet bucket S3 boto3 pour récupérer les données binaires de FSx ONTAP. Le style de code permettant d'accéder aux données de FSx ONTAP est similaire à la lecture des données d'Amazon S3. L'explication suivante approfondit le processus de création de l'objet S3 privé **bucket**.

```
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 ---
```

Pour lire les données de FSx ONTAP dans SageMaker, un gestionnaire est créé qui pointe vers le stockage FSx ONTAP à l'aide du protocole S3. Cela permet à FSx ONTAP d'être traité comme un bucket S3 privé. La configuration du gestionnaire inclut la spécification de l'adresse IP du SVM FSx ONTAP, du nom du bucket et des informations d'identification nécessaires. Pour une explication complète sur l'obtention de ces éléments de configuration, veuillez vous référer au document à l'adresse ["Partie 1 - Intégration Amazon FSx for NetApp ONTAP \(FSx ONTAP\) en tant que compartiment S3 privé dans AWS SageMaker"](#).

Dans l'exemple mentionné ci-dessus, l'objet bucket est utilisé pour instancier l'objet de jeu de données PyTorch. L'objet de jeu de données sera expliqué plus en détail dans la section suivante.

```
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)
```

Dans l'exemple fourni, une taille de lot de 64 est spécifiée, indiquant que chaque lot contiendra 64 enregistrements. En combinant la classe PyTorch **Dataset**, la fonction de prétraitement et la taille du lot d'entraînement, nous obtenons le chargeur de données pour l'entraînement. Ce chargeur de données facilite le processus d'itération de l'ensemble de données par lots pendant la phase de formation.

Formation de modèle

```
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')

```

Ce code implémente un processus de formation PyTorch standard. Il définit un modèle de réseau neuronal appelé **TyreQualityClassifier** utilisant des couches convolutives et une couche linéaire pour classer la qualité des pneus. La boucle de formation parcourt les lots de données, calcule la perte et met à jour les paramètres du modèle à l'aide de la rétropropagation et de l'optimisation. De plus, il imprime l'heure actuelle, l'époque, le lot et la perte à des fins de surveillance.

Déploiement du modèle

Déploiement

```

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())

```

Le code enregistre le modèle PyTorch sur **Amazon S3** car SageMaker exige que le modèle soit stocké dans S3 pour le déploiement. En téléchargeant le modèle sur **Amazon S3**, il devient accessible à SageMaker, permettant le déploiement et l'inférence sur le modèle déployé.

```

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
)

```

Ce code facilite le déploiement d'un modèle PyTorch sur SageMaker. Il définit un sérialiseur personnalisé, **TyreQualitySerializer**, qui prétraite et sérialise les données d'entrée sous forme de tenseur PyTorch. La classe **TyreQualityPredictor** est un prédicteur personnalisé qui utilise le sérialiseur défini et un **JSONDeserializer**. Le code crée également un objet **PyTorchModel** pour spécifier l'emplacement S3 du modèle, le rôle IAM, la version du framework et le point d'entrée pour l'inférence. Le code génère un

horodatage et construit un nom de point de terminaison basé sur le modèle et l'horodatage. Enfin, le modèle est déployé à l'aide de la méthode `deploy`, en spécifiant le nombre d'instances, le type d'instance et le nom du point de terminaison généré. Cela permet au modèle PyTorch d'être déployé et accessible pour l'inférence sur SageMaker.

Inférence

```
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)
```

Voici l'exemple d'utilisation du point de terminaison déployé pour effectuer l'inférence.

Partie 3 - Création d'un pipeline MLOps simplifié (CI/CT/CD)

Cet article fournit un guide pour créer un pipeline MLOps avec les services AWS, en se concentrant sur le recyclage automatisé des modèles, le déploiement et l'optimisation des coûts.

Introduction

Dans ce didacticiel, vous apprendrez à exploiter divers services AWS pour créer un pipeline MLOps simple qui englobe l'intégration continue (CI), la formation continue (CT) et le déploiement continu (CD). Contrairement aux pipelines DevOps traditionnels, MLOps nécessite des considérations supplémentaires pour terminer le cycle opérationnel. En suivant ce tutoriel, vous obtiendrez des informations sur l'intégration de CT dans la boucle MLOps, permettant une formation continue de vos modèles et un déploiement transparent pour l'inférence. Le didacticiel vous guidera tout au long du processus d'utilisation des services AWS pour établir ce pipeline MLOps de bout en bout.

Manifeste

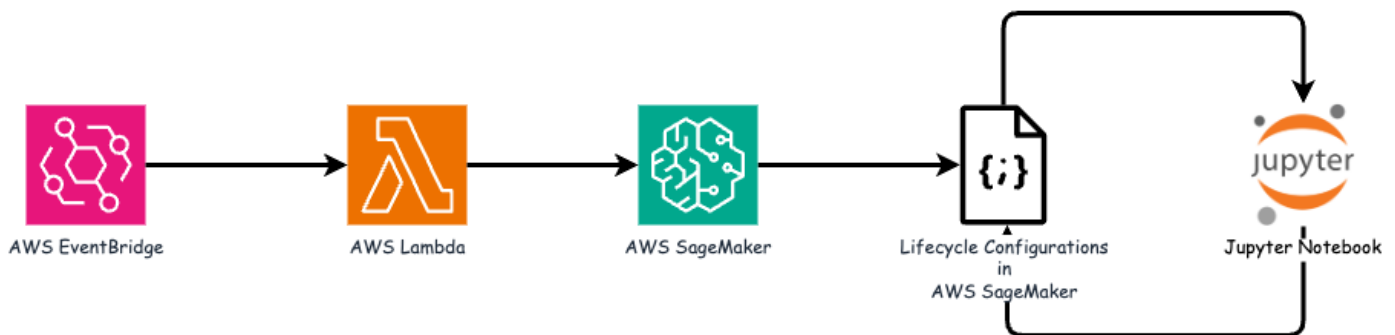
Fonctionnalité	Nom	Commentaire
Stockage de données	AWS FSx ONTAP	"Partie 1 - Intégration Amazon FSx for NetApp ONTAP (FSx ONTAP) en tant que compartiment S3 privé dans AWS SageMaker" .
IDE de science des données	AWS SageMaker	Ce tutoriel est basé sur le notebook Jupyter présenté dans "Partie 2 - Exploiter Amazon FSx for NetApp ONTAP (FSx ONTAP) comme source de données pour la formation de modèles dans SageMaker" .

Fonctionnalité	Nom	Commentaire
Fonction pour déclencher le pipeline MLOps	Fonction AWS Lambda	-
Déclencheur de tâche Cron	AWS EventBridge	-
Cadre d'apprentissage profond	PyTorch	-
Kit de développement logiciel (SDK) AWS Python	boto3	-
Langage de programmation	Python	v3.10

Condition préalable

- Un système de fichiers FSx ONTAP préconfiguré. Ce didacticiel utilise les données stockées dans FSx ONTAP pour le processus de formation.
- Une **instance SageMaker Notebook** configurée pour partager le même VPC que le système de fichiers FSx ONTAP mentionné ci-dessus.
- Avant de déclencher la **fonction AWS Lambda**, assurez-vous que l'**instance SageMaker Notebook** est à l'état **arrêté**.
- Le type d'instance **ml.g4dn.xlarge** est requis pour tirer parti de l'accélération GPU nécessaire aux calculs des réseaux neuronaux profonds.

Architecture



Ce pipeline MLOps est une implémentation pratique qui utilise une tâche cron pour déclencher une fonction sans serveur, qui à son tour exécute un service AWS enregistré auprès d'une fonction de rappel de cycle de vie. **AWS EventBridge** agit comme une tâche cron. Il appelle périodiquement une **fonction AWS Lambda** chargée de recycler et de redéployer le modèle. Ce processus implique de lancer l'instance **AWS SageMaker Notebook** pour effectuer les tâches nécessaires.

Configuration étape par étape

Configurations du cycle de vie

Pour configurer la fonction de rappel du cycle de vie pour l'instance AWS SageMaker Notebook, vous devez utiliser **Configurations du cycle de vie**. Ce service vous permet de définir les actions nécessaires à effectuer lors du démarrage de l'instance du notebook. Plus précisément, un script shell peut être implémenté dans les **configurations du cycle de vie** pour arrêter automatiquement l'instance du notebook une fois les processus de formation et de déploiement terminés. Il s'agit d'une configuration obligatoire car le coût est l'une des

principales considérations dans MLOps.

Il est important de noter que la configuration des **configurations du cycle de vie** doit être définie à l'avance. Par conséquent, il est recommandé de prioriser la configuration de cet aspect avant de procéder à l'autre configuration du pipeline MLOps.

1. Pour configurer un cycle de vie, ouvrez le panneau **Sagemaker** et accédez à **Configurations du cycle de vie** sous la section **Configurations d'administration**.

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 Studio notebook instance. Each domain receives a personal and private Amazon SageMaker Studio notebook instance.

► Domain structure diagram

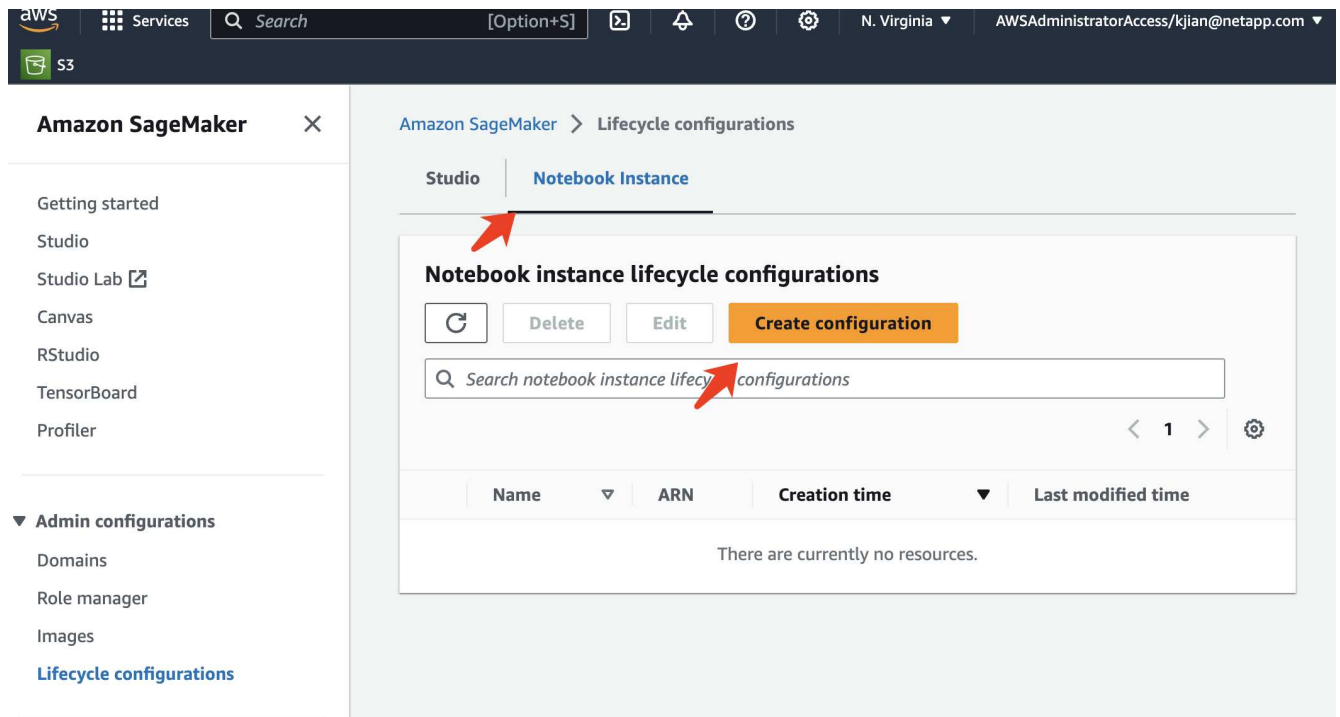
Domains (4)

Info

Q Find domain name

	Name	
☐	rdsml-east-1	
☐	rdsml-east-2	
☐	rdsml-east-3	
☐	rdsml-east-4	

2. Sélectionnez l'onglet **Instance de bloc-notes** et cliquez sur le bouton **Créer une configuration**



3. Collez le code ci-dessous dans la zone de saisie.

```
#!/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. Ce script exécute le Jupyter Notebook, qui gère le recyclage et le redéploiement du modèle pour l'inférence. Une fois l'exécution terminée, le notebook s'éteindra automatiquement dans les 5 minutes. Pour en savoir plus sur l'énoncé du problème et l'implémentation du code, veuillez vous référer à "[Partie 2 - Exploiter Amazon FSx for NetApp ONTAP \(FSx ONTAP\) comme source de données pour la formation de modèles dans SageMaker](#)".

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

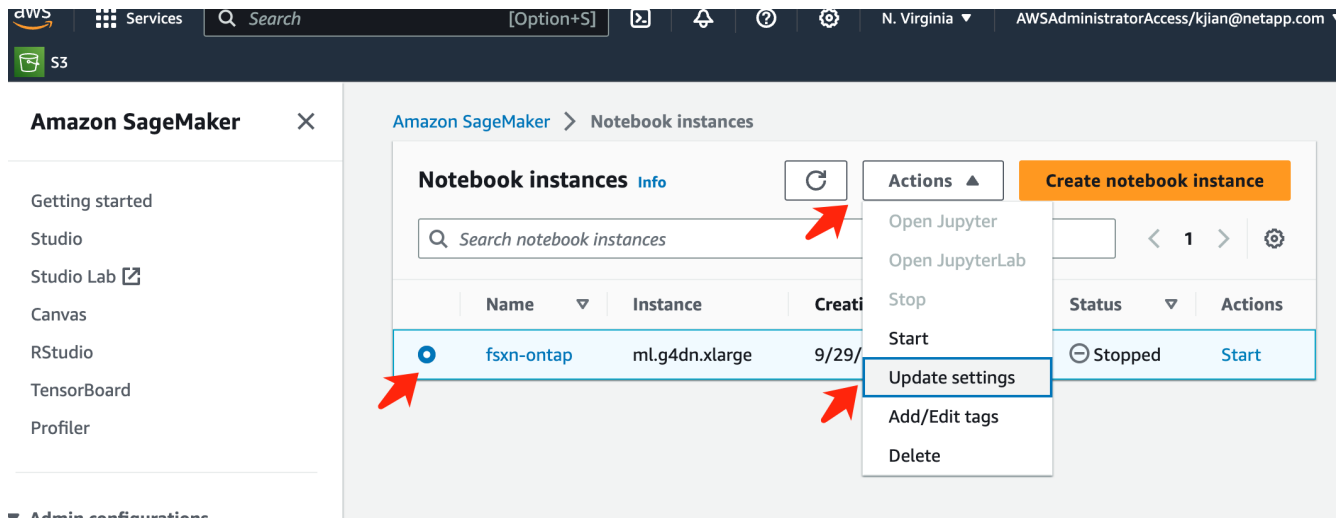
This script will be run each time an associated notebook instance is started, including during initial creation. If the associated notebook instance is already started, it will be run the next time it is stopped and started. [a curated list of sample scripts](#)

```
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 torch_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
```

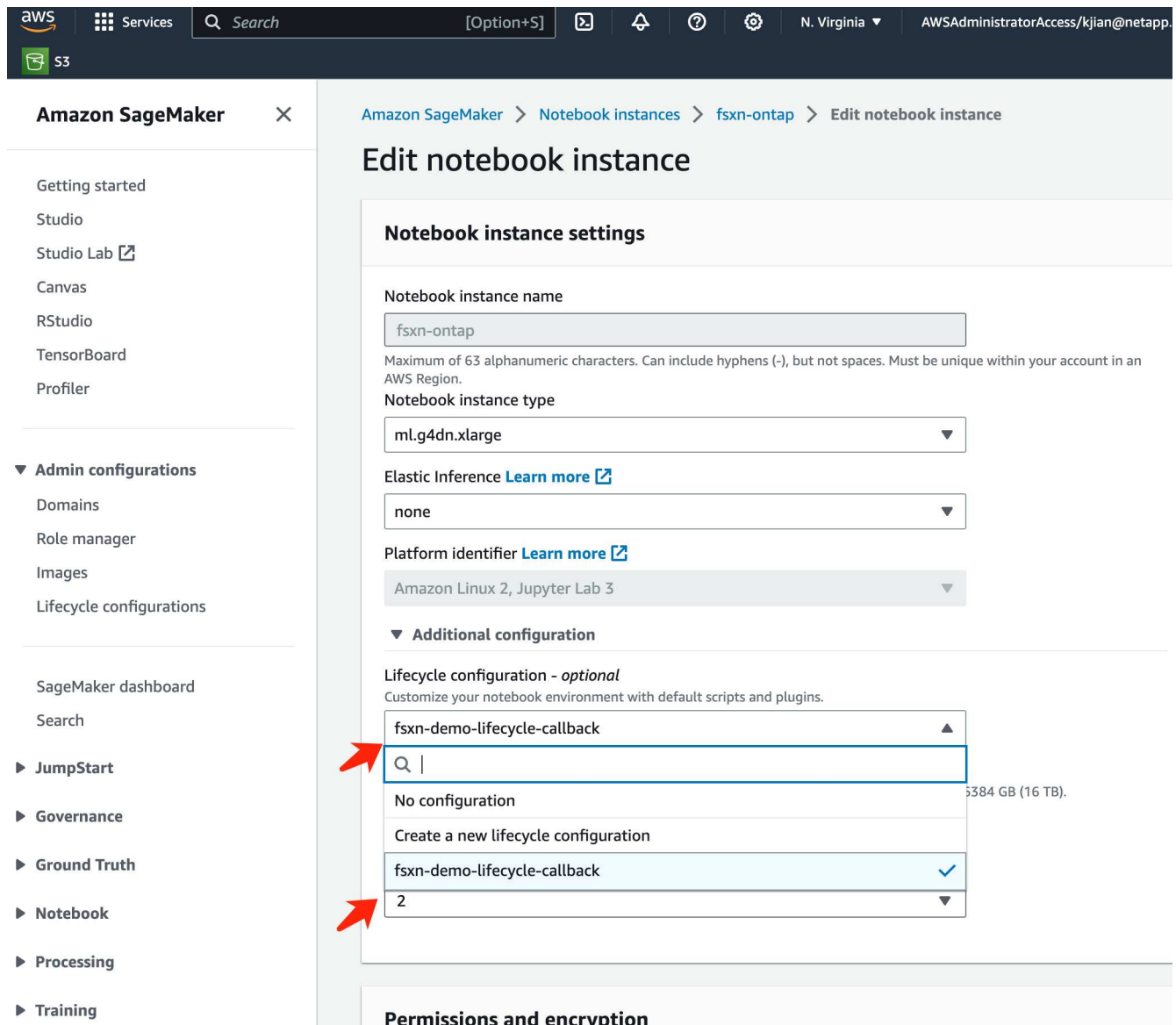
Cancel Create configuration

CloudShell Feedback

5. Après la création, accédez aux instances de Notebook, sélectionnez l'instance cible et cliquez sur **Mettre à jour les paramètres** sous la liste déroulante Actions.



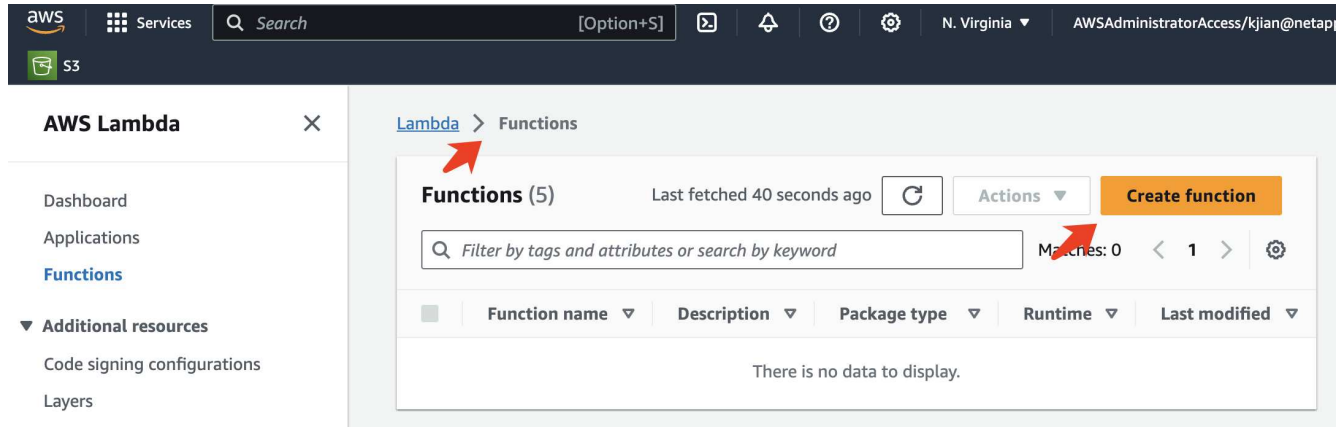
6. Sélectionnez la **Configuration du cycle de vie** créée et cliquez sur **Mettre à jour l'instance du bloc-notes**.



Fonction sans serveur AWS Lambda

Comme mentionné précédemment, la **fonction AWS Lambda** est responsable du démarrage de l'**instance AWS SageMaker Notebook**.

1. Pour créer une **fonction AWS Lambda**, accédez au panneau correspondant, passez à l'onglet **Fonctions** et cliquez sur **Créer une fonction**.



2. Veuillez déposer toutes les entrées requises sur la page et n'oubliez pas de passer le Runtime à **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. Veuillez vérifier que le rôle désigné dispose de l'autorisation requise **AmazonSageMakerFullAccess** et cliquez sur le bouton **Créer une fonction**.

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**

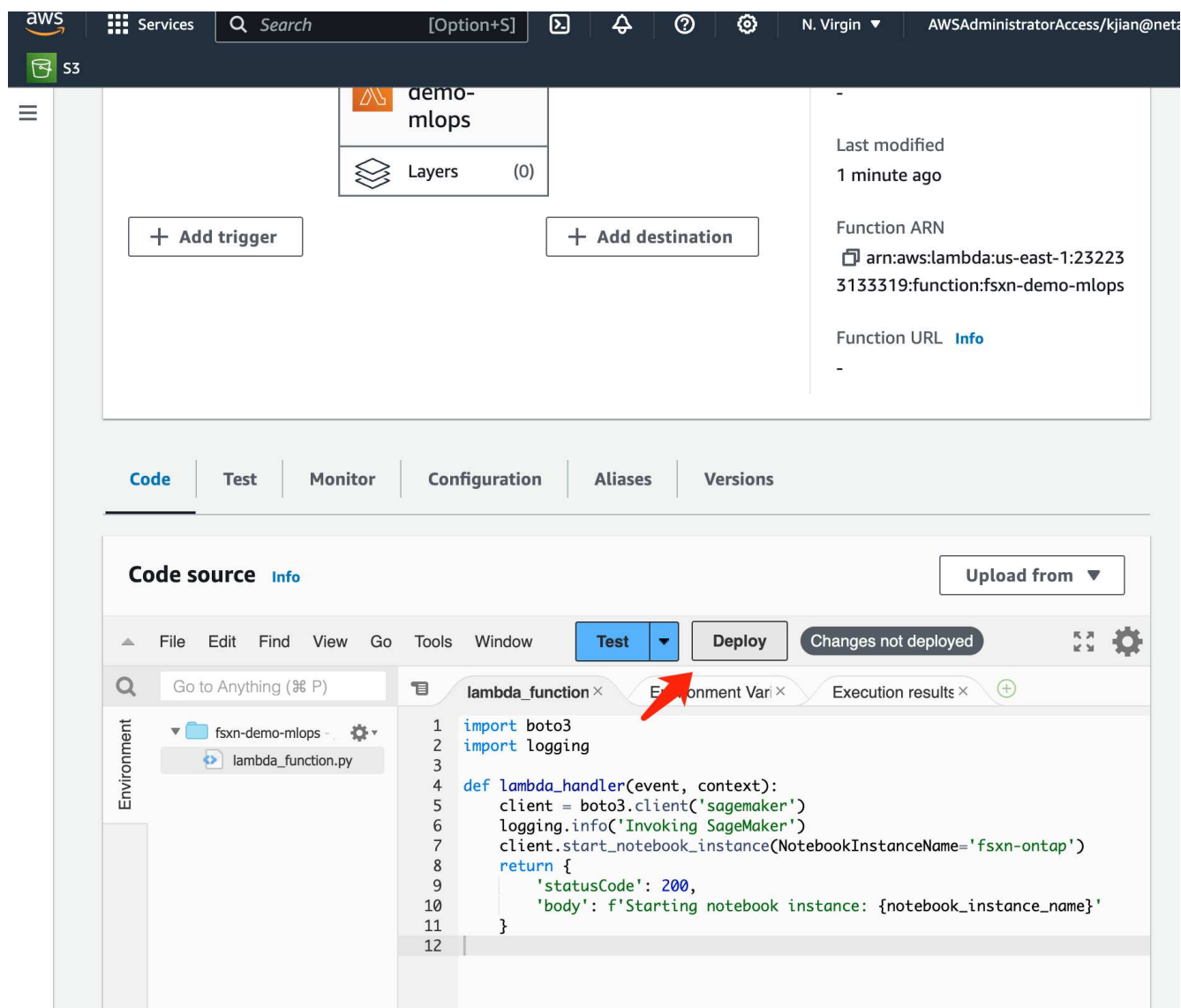
Cancel Create function

4. Sélectionnez la fonction Lambda créée. Dans l'onglet code, copiez et collez le code suivant dans la zone de texte. Ce code démarre l'instance de notebook nommée **fsxn-ontap**.

```
import boto3
import logging

def lambda_handler(event, context):
    client = boto3.client('sagemaker')
    logging.info('Invoking SageMaker')
    client.start_notebook_instance(NotebookInstanceName='fsxn-ontap')
    return {
        'statusCode': 200,
        'body': f'Starting notebook instance: {notebook_instance_name}'
    }
```

5. Cliquez sur le bouton **Déployer** pour appliquer cette modification de code.



6. Pour spécifier comment déclencher cette fonction AWS Lambda, cliquez sur le bouton Ajouter un déclencheur.

The screenshot shows the AWS Lambda console interface. At the top, the navigation bar includes the AWS logo, 'Services', a search bar, and the user's profile. 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 an 'Actions' dropdown menu. Below the function name, the 'Function overview' section is expanded, showing a card for the function with its icon and a 'Layers (0)' section. 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 details panel lists: '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. Sélectionnez EventBridge dans le menu déroulant, puis cliquez sur le bouton radio intitulé Créer une nouvelle règle. Dans le champ d'expression de planification, saisissez `rate(1 day)`, et cliquez sur le bouton Ajouter pour créer et appliquer cette nouvelle règle de tâche cron à la fonction AWS Lambda.

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

Une fois la configuration en deux étapes terminée, la fonction **AWS Lambda** lancera quotidiennement le **SageMaker Notebook**, effectuera un recyclage du modèle à l'aide des données du référentiel **FSx ONTAP**, redéploiera le modèle mis à jour dans l'environnement de production et arrêtera automatiquement l'instance **SageMaker Notebook** pour optimiser les coûts. Cela garantit que le modèle reste à jour.

Ceci conclut le tutoriel sur le développement d'un pipeline MLOps.

Informations sur le copyright

Copyright © 2025 NetApp, Inc. Tous droits réservés. Imprimé aux États-Unis. Aucune partie de ce document protégé par copyright ne peut être reproduite sous quelque forme que ce soit ou selon quelque méthode que ce soit (graphique, électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système de récupération électronique) sans l'autorisation écrite préalable du détenteur du droit de copyright.

Les logiciels dérivés des éléments NetApp protégés par copyright sont soumis à la licence et à l'avis de non-responsabilité suivants :

CE LOGICIEL EST FOURNI PAR NETAPP « EN L'ÉTAT » ET SANS GARANTIES EXPRESSES OU TACITES, Y COMPRIS LES GARANTIES TACITES DE QUALITÉ MARCHANDE ET D'ADÉQUATION À UN USAGE PARTICULIER, QUI SONT EXCLUES PAR LES PRÉSENTES. EN AUCUN CAS NETAPP NE SERA TENU POUR RESPONSABLE DE DOMMAGES DIRECTS, INDIRECTS, ACCESSOIRES, PARTICULIERS OU EXEMPLAIRES (Y COMPRIS L'ACHAT DE BIENS ET DE SERVICES DE SUBSTITUTION, LA PERTE DE JOUISSANCE, DE DONNÉES OU DE PROFITS, OU L'INTERRUPTION D'ACTIVITÉ), QUELLES QU'EN SOIENT LA CAUSE ET LA DOCTRINE DE RESPONSABILITÉ, QU'IL S'AGISSE DE RESPONSABILITÉ CONTRACTUELLE, STRICTE OU DÉLICTELLE (Y COMPRIS LA NÉGLIGENCE OU AUTRE) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL, MÊME SI LA SOCIÉTÉ A ÉTÉ INFORMÉE DE LA POSSIBILITÉ DE TELS DOMMAGES.

NetApp se réserve le droit de modifier les produits décrits dans le présent document à tout moment et sans préavis. NetApp décline toute responsabilité découlant de l'utilisation des produits décrits dans le présent document, sauf accord explicite écrit de NetApp. L'utilisation ou l'achat de ce produit ne concède pas de licence dans le cadre de droits de brevet, de droits de marque commerciale ou de tout autre droit de propriété intellectuelle de NetApp.

Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevets américains, étrangers ou par une demande en attente.

LÉGENDE DE RESTRICTION DES DROITS : L'utilisation, la duplication ou la divulgation par le gouvernement sont sujettes aux restrictions énoncées dans le sous-paragraphe (b)(3) de la clause Rights in Technical Data-Noncommercial Items du DFARS 252.227-7013 (février 2014) et du FAR 52.227-19 (décembre 2007).

Les données contenues dans les présentes se rapportent à un produit et/ou service commercial (tel que défini par la clause FAR 2.101). Il s'agit de données propriétaires de NetApp, Inc. Toutes les données techniques et tous les logiciels fournis par NetApp en vertu du présent Accord sont à caractère commercial et ont été exclusivement développés à l'aide de fonds privés. Le gouvernement des États-Unis dispose d'une licence limitée irrévocable, non exclusive, non cessible, non transférable et mondiale. Cette licence lui permet d'utiliser uniquement les données relatives au contrat du gouvernement des États-Unis d'après lequel les données lui ont été fournies ou celles qui sont nécessaires à son exécution. Sauf dispositions contraires énoncées dans les présentes, l'utilisation, la divulgation, la reproduction, la modification, l'exécution, l'affichage des données sont interdits sans avoir obtenu le consentement écrit préalable de NetApp, Inc. Les droits de licences du Département de la Défense du gouvernement des États-Unis se limitent aux droits identifiés par la clause 252.227-7015(b) du DFARS (février 2014).

Informations sur les marques commerciales

NETAPP, le logo NETAPP et les marques citées sur le site <http://www.netapp.com/TM> sont des marques déposées ou des marques commerciales de NetApp, Inc. Les autres noms de marques et de produits sont des marques commerciales de leurs propriétaires respectifs.