



# **AWS FSX for NetApp ONTAP (FSxN) für MLOps**

NetApp Solutions

NetApp  
April 26, 2024

This PDF was generated from [https://docs.netapp.com/de-de/netapp-solutions/ai/mlops\\_fsxn\\_s3\\_integration.html](https://docs.netapp.com/de-de/netapp-solutions/ai/mlops_fsxn_s3_integration.html) on April 26, 2024. Always check docs.netapp.com for the latest.

# Inhalt

- AWS FSX for NetApp ONTAP (FSxN) für MLOps ..... 1
  - Teil 1 – Integration von AWS FSX for NetApp ONTAP (FSxN) als privater S3-Bucket in AWS SageMaker .. 1
  - Teil 2 – Nutzung von AWS FSX for NetApp ONTAP (FSxN) als Datenquelle für das Modelltraining in SageMaker ..... 15
  - Teil 3 – Aufbau Einer vereinfachten MLOPS-Pipeline (CI/CT/CD) ..... 24

# AWS FSX for NetApp ONTAP (FSxN) für MLOps

## Autor(en):

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

In diesem Abschnitt wird die praktische Anwendung der Entwicklung von KI-Infrastrukturen erläutert und der Aufbau einer MLOps-Pipeline mithilfe von FSxN durchgängig dargestellt. Sie umfasst drei umfassende Beispiele und hilft Ihnen, Ihre MLOps-Anforderungen über diese leistungsstarke Datenmanagement-Plattform zu erfüllen.

Die Schwerpunkte dieser Artikel liegen auf:

1. ["Teil 1 – Integration von AWS FSX for NetApp ONTAP \(FSxN\) als privater S3-Bucket in AWS SageMaker"](#)
2. ["Teil 2 – Nutzung von AWS FSX for NetApp ONTAP \(FSxN\) als Datenquelle für das Modelltraining in SageMaker"](#)
3. ["Teil 3 – Aufbau Einer vereinfachten MLOPS-Pipeline \(CI/CT/CD\)"](#)

Am Ende dieses Abschnitts haben Sie ein fundiertes Verständnis dafür gewonnen, wie Sie FSxN zur Optimierung von MLOPS-Prozessen nutzen können.

## Teil 1 – Integration von AWS FSX for NetApp ONTAP (FSxN) als privater S3-Bucket in AWS SageMaker

### Autor(en):

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

### Einführung

Mithilfe von SageMaker als Beispiel bietet diese Seite Anleitungen zur Konfiguration von FSxN als privaten S3-Bucket.

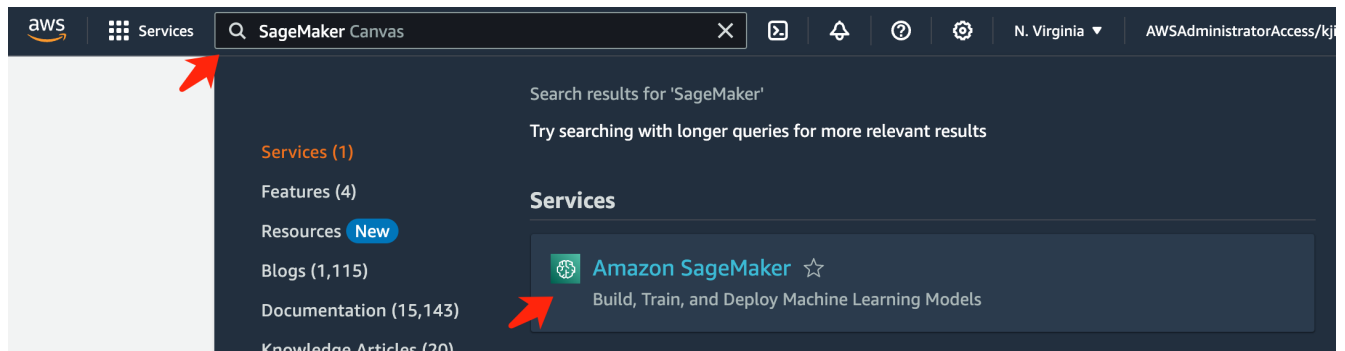
Weitere Informationen über FSxN finden Sie in dieser Präsentation (["Video-Link"](#))

### Benutzerhandbuch

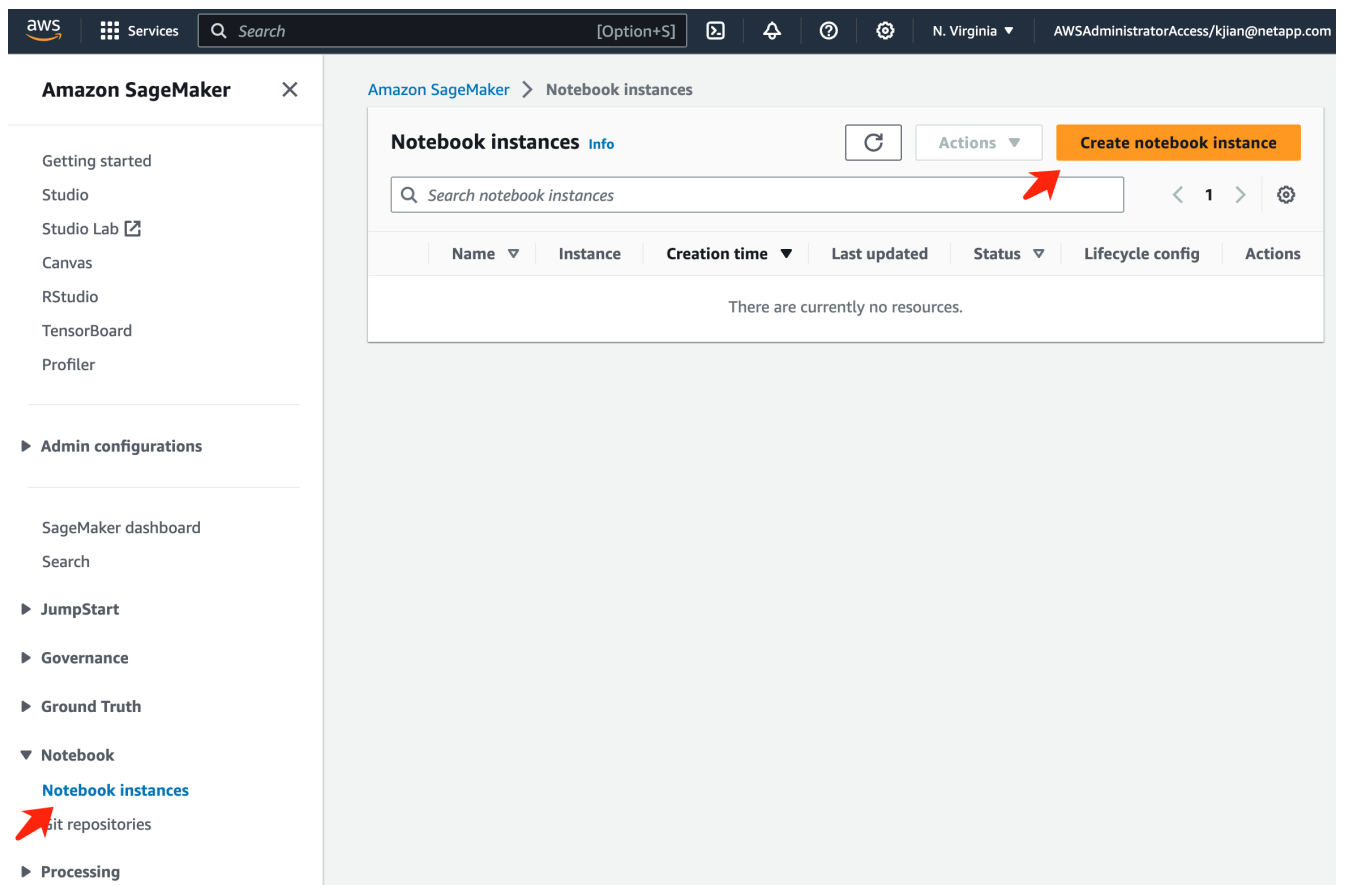
#### Server-Erstellung

Erstellen Sie eine SageMaker-Notebook-Instanz

1. Öffnen Sie die AWS-Konsole. Suchen Sie im Suchfeld nach SageMaker und klicken Sie auf den Dienst **Amazon SageMaker**.



- Öffnen Sie die **Notizbuchinstanzen** unter der Registerkarte Notizbuch, klicken Sie auf die orangefarbene Schaltfläche **Notizbuchinstanz erstellen**.



- Auf der Seite „Erstellung“  
Geben Sie den Namen der **Notebook-Instanz** ein  
Erweitern Sie das Fenster **Netzwerk**  
Lassen Sie andere Einträge standardmäßig und wählen Sie eine **VPC**, **Subnetz** und **Sicherheitsgruppe(n)** aus. (Diese **VPC** und **Subnetz** werden später zum Erstellen des FSxN-Dateisystems verwendet)  
Klicken Sie unten rechts auf den orangefarbenen Button **Notizbuchinstanz erstellen**.

## 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

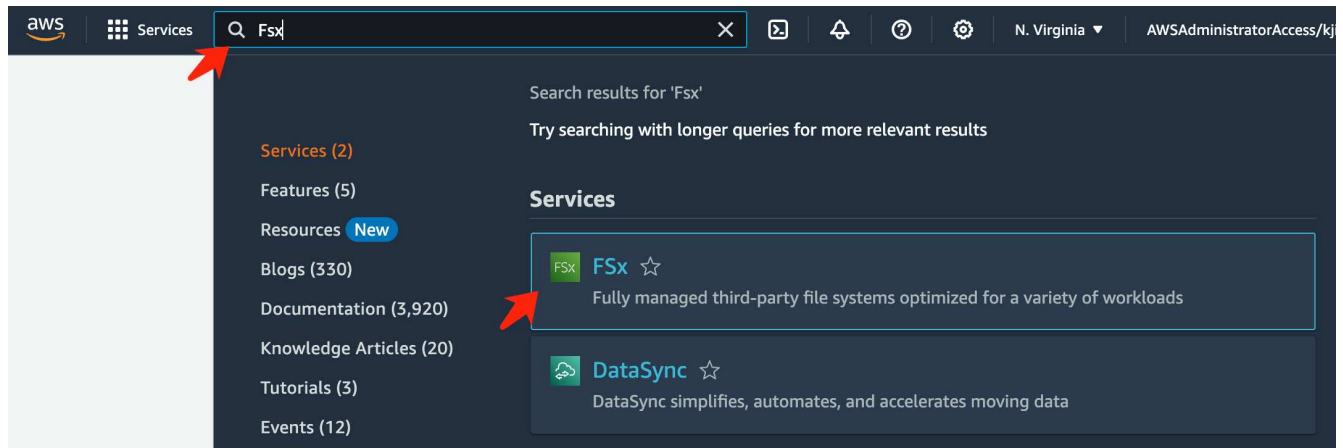
► Tags - optional

Cancel

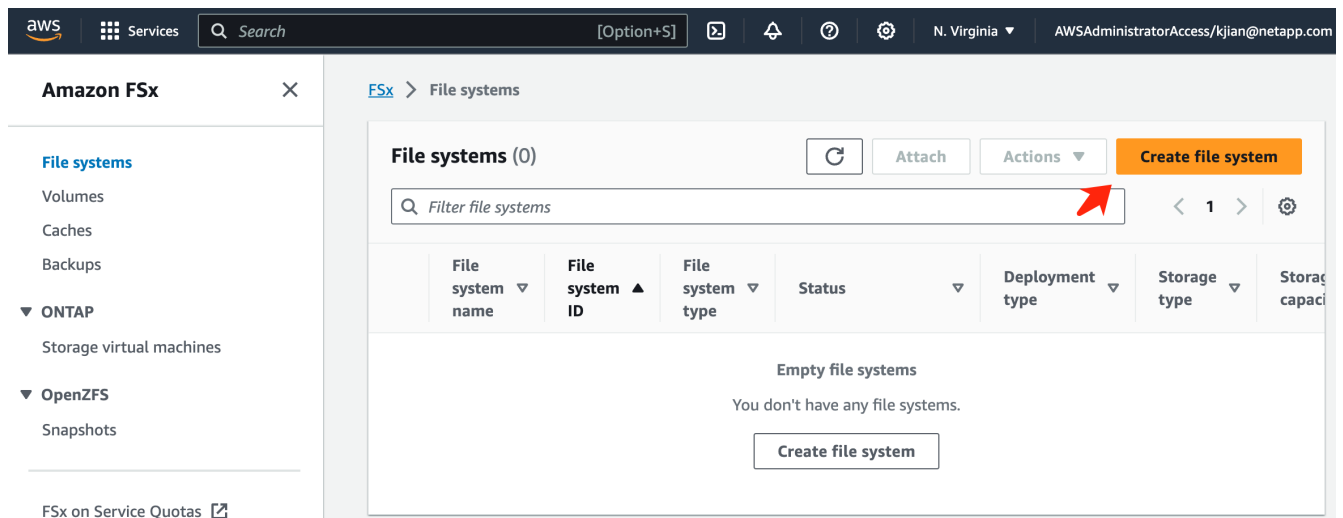
Create notebook instance

## Erstellen Sie ein FSxN-Dateisystem

1. Öffnen Sie die AWS-Konsole. Suchen Sie im Suchfeld FSX und klicken Sie auf den Dienst **FSX**.



2. Klicken Sie auf **Dateisystem erstellen**.



3. Wählen Sie die erste Karte **FSx für NetApp ONTAP** und klicken Sie auf **Weiter**.

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

FSx > File systems > Create file system

Step 1  
Select file system type

Step 2  
Specify file system details

Step 3  
Review and create

## Select file system type

### File system options

☒ Amazon FSx for NetApp ONTAP  
FSx  
Amazon FSx for NetApp ONTAP

☐ Amazon FSx for OpenZFS  
FSx  
Amazon FSx for OpenZFS

☐ Amazon FSx for Windows File Server  
FSx  
Amazon FSx for Windows File Server

☐ Amazon FSx for Lustre  
FSx  
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. Auf der Konfigurationsseite für Details.
- a. Wählen Sie die Option **Standard create**.

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

FSx > File systems > Create file system

Step 1  
[Select file system type](#)

Step 2  
Specify file system details

Step 3  
Review and create

## Specify file system details

### Creation method

☐ Quick create  
Use recommended best-practice configurations. Most configuration options can be changed after the file system is created.

☒ Standard create  
You set all of the configuration options, including specifying performance, networking, security, backups, and maintenance.

- b. Geben Sie den Namen des **Dateisystems** und die Kapazität des **SSD-Speichers** ein.

### 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. Stellen Sie sicher, dass Sie die **VPC** und **Subnetz** mit der **SageMaker Notebook**-Instanz identisch verwenden.



## 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. Geben Sie den Namen der **Storage Virtual Machine** und **Geben Sie ein Passwort** für Ihre SVM (Storage Virtual Machine) ein.

### 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. Lassen Sie andere Einträge standardmäßig und klicken Sie auf die orangefarbene Schaltfläche **Weiter** unten rechts.

► Backup and maintenance - optional

► Tags - optional

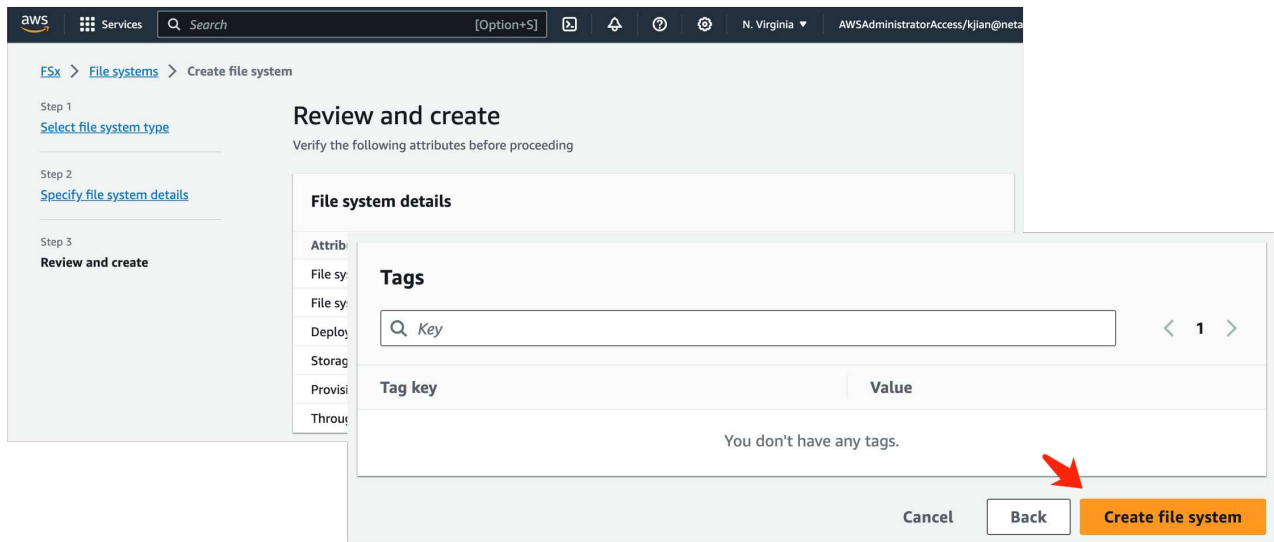
Cancel

Back

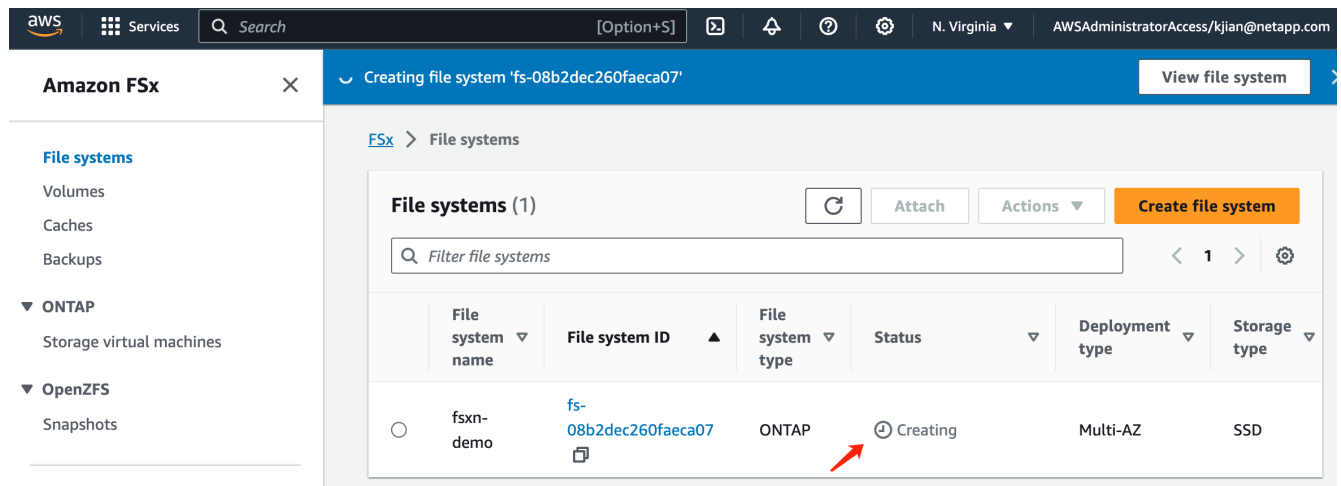
Next

f. Klicken Sie auf die orangefarbene Schaltfläche **Dateisystem erstellen** unten rechts auf der Überprüfungsseite.

8



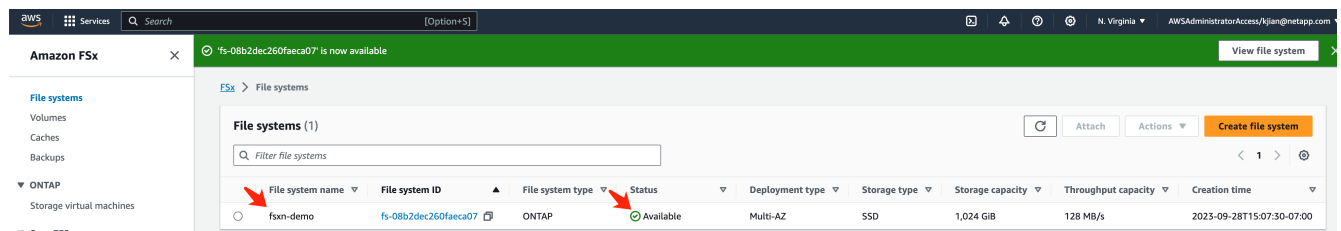
5. Es kann ungefähr 20-40 Minuten\* dauern, um das FSX Dateisystem hochzudrehen.



## Serverkonfiguration

### ONTAP-Konfiguration

1. Öffnen Sie das erstellte FSX Dateisystem. Bitte stellen Sie sicher, dass der Status **verfügbar** ist.



2. Wählen Sie die Registerkarte **Verwaltung** aus und behalten Sie die Optionen **Verwaltungsendpunkt - IP-Adresse** und **ONTAP-Administratorbenutzername** bei.

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

### 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	SSD storage capacity 1024 GiB <b>Update</b>	Availability Zones us-east-1a (Preferred) us-east-1b (Standby)
Lifecycle state Creating	Throughput capacity 128 MB/s <b>Update</b>	Creation time 2023-09-28T14:41:50-07:00
File system type ONTAP	Provisioned IOPS 3072 <b>Update</b>	
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	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 172.31.31.157 172.31.32.38	ONTAP administrator password <b>Update</b>

3. Öffnen Sie die erstellte **SageMaker Notebook-Instanz** und klicken Sie auf **JupyterLab öffnen**.

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

### Amazon SageMaker

- Getting started
- Studio
- Studio Lab
- Canvas
- RStudio
- TensorBoard

Amazon SageMaker > Notebook instances

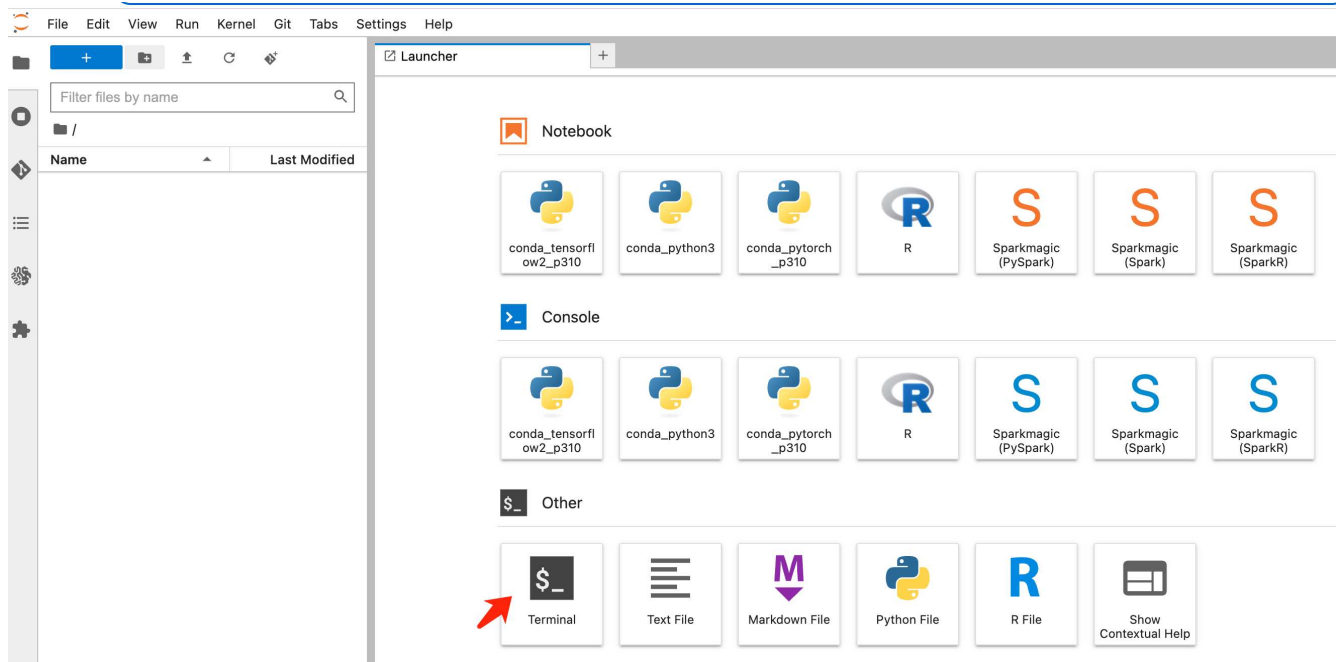
## Notebook instances

Search notebook instances

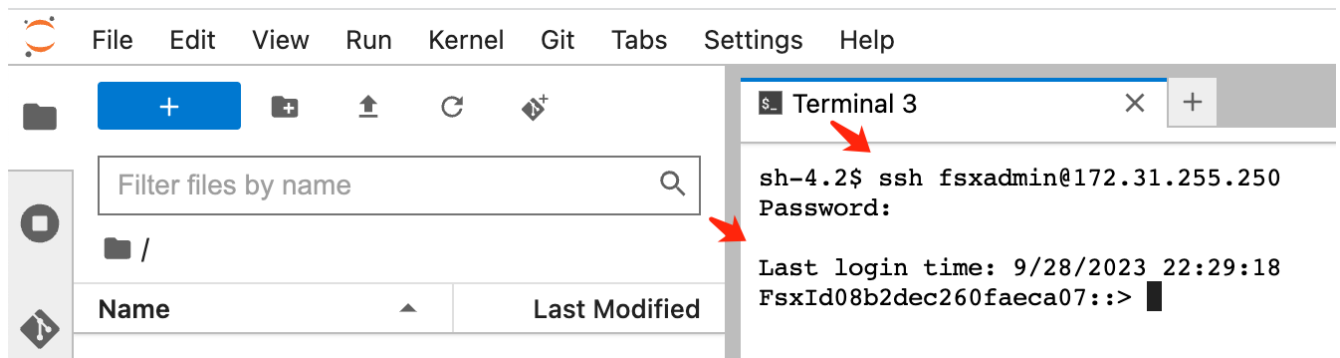
**Create notebook instance**

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. Öffnen Sie auf der Seite Jupyter Lab ein neues **Terminal**.



5. Geben Sie zum Anmelden beim FSxN ONTAP-Dateisystem den ssh-Befehl `ssh@<admin user Name> <ONTAP Server IP>` ein. (Benutzername und IP-Adresse werden aus Schritt 2 abgerufen)  
Bitte verwenden Sie das Passwort, das beim Erstellen der **Storage Virtual Machine** verwendet wird.



6. Führen Sie die Befehle in der folgenden Reihenfolge aus.  
Wir verwenden **fsxn-ontap** als Namen für den **FSxN privaten S3-Bucket-Namen**.  
Bitte verwenden Sie für das Argument **-vserver** den Namen der virtuellen Speichermaschine\*.

```
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:~> vserver object-store-server create -vserver fsxn-svm-demo -object-store-server fsx_s3 -is-http-enabled true -is-https-enabled false
FsxId08b2dec260faeca07:~> vserver object-store-server user create -vserver fsxn-svm-demo -user s3user
FsxId08b2dec260faeca07:~> vserver object-store-server group create -name s3group -users s3user -policies FullAccess
FsxId08b2dec260faeca07:~> vserver object-store-server bucket create fsxn-ontap -vserver fsxn-svm-demo -type nas -nas-path /voll
FsxId08b2dec260faeca07:~>

```

7. Führen Sie die folgenden Befehle aus, um die Endpunkt-IP und die Zugangsdaten für FSxN private S3 abzurufen.

```

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

set adv

vserver object-store-server user show

```

8. Die IP-Adresse des Endpunkts und die Anmeldeinformationen bleiben für die zukünftige Verwendung erhalten.

```

sh-4.2$ ssh fsxadmin@172.31.255.250
Password:
Last login time: 9/28/2023 22:32:42
FsxId08b2dec260faeca07:~> network interface show -vserver 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:~> vserver 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

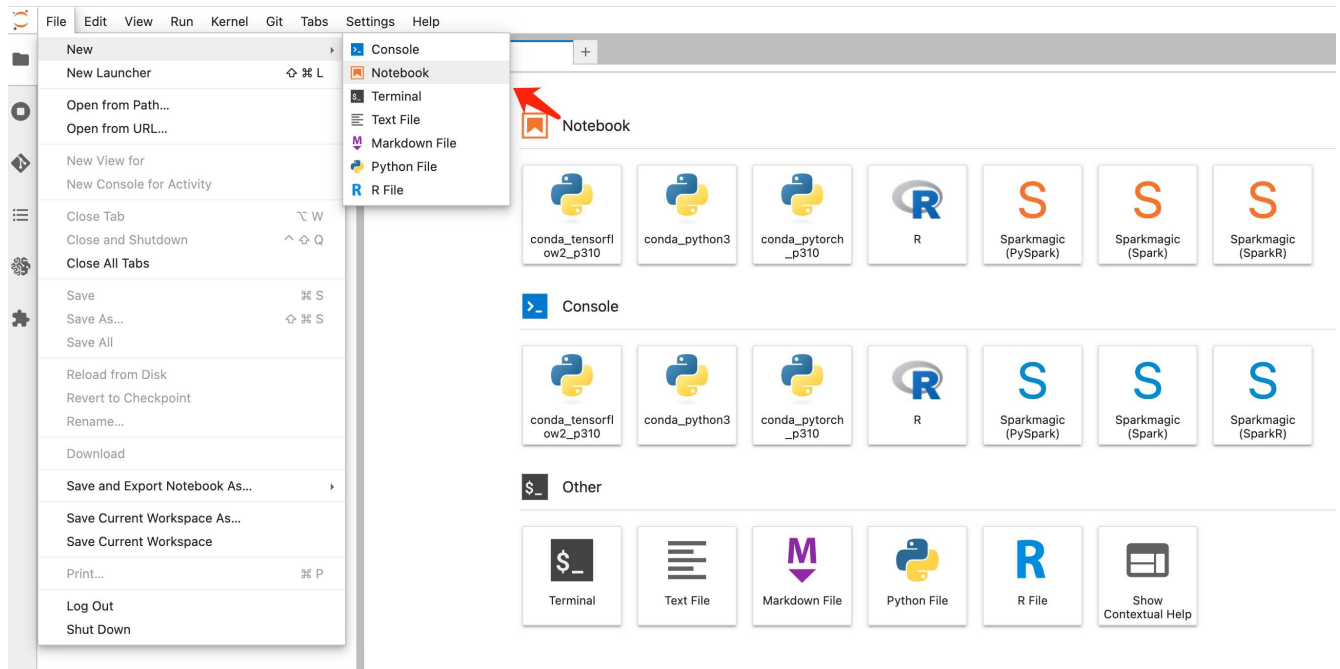
2 entries were displayed.

FsxId08b2dec260faeca07:~>

```

## Client-Konfiguration

1. Erstellen Sie in der SageMaker-Notebook-Instanz ein neues Jupyter-Notebook.



2. Verwenden Sie den unten stehenden Code als Umgehung, um Dateien in einen FSxN privaten S3-Bucket hochzuladen.

Ein umfangreiches Codebeispiel finden Sie in diesem Notizbuch.

["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']

```

Damit ist die Integration zwischen FSxN und der SageMaker-Instanz abgeschlossen.

## Nützliche Debugging-Checkliste

- Stellen Sie sicher, dass sich die SageMaker-Notebook-Instanz und das FSxN-Dateisystem im gleichen VPC befinden.



- Denken Sie daran, den Befehl **set dev** auf ONTAP auszuführen, um die Berechtigungsebene auf **dev** zu setzen.

## FAQ (Stand 27. September 2023)

F: Warum erhalte ich den Fehler **"ein Fehler ist aufgetreten (NotImplemented) beim Aufruf der CreateMultipartUpload Operation: Der von Ihnen angeforderte s3 Befehl ist nicht implementiert"** beim Hochladen von Dateien in FSxN?

A: Als privater S3-Bucket unterstützt FSxN das Hochladen von Dateien mit bis zu 100 MB. Bei Verwendung des S3-Protokolls werden Dateien mit einer Größe von mehr als 100 MB in 100-MB-Blöcke unterteilt, und die Funktion „CreateMultipartUpload“ wird aufgerufen. Die aktuelle Implementierung von FSxN private S3 unterstützt diese Funktion jedoch nicht.

F: Warum erhalte ich den Fehler **"ein Fehler ist aufgetreten (AccessDenied) beim Aufruf der PutObject-Operationen: Zugriff verweigert"** beim Hochladen von Dateien in FSxN?

A: Um von einer SageMaker-Notebook-Instanz auf den privaten FSxN-S3-Bucket zuzugreifen, wechseln Sie die AWS-Anmeldeinformationen zu den FSxN-Anmeldeinformationen. Die Gewährung von Schreibberechtigungen für die Instanz erfordert jedoch eine Problemumkehrungslösung, bei der der Bucket gemountet und der Shell-Befehl 'chmod' ausgeführt wird, um die Berechtigungen zu ändern.

F: Wie kann ich den FSxN privaten S3-Eimer mit anderen SageMaker ML-Diensten integrieren?

A: Leider bietet das SageMaker Services SDK keine Möglichkeit, den Endpunkt für den privaten S3-Bucket anzugeben. Daher ist FSxN S3 nicht kompatibel mit SageMaker-Diensten wie SageMaker Data Wrangler, SageMaker Clarify, SageMaker Glue, SageMaker Athena, SageMaker AutoML, und andere.

## Teil 2 – Nutzung von AWS FSX for NetApp ONTAP (FSxN) als Datenquelle für das Modelltraining in SageMaker

### Autor(en):

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

### Einführung

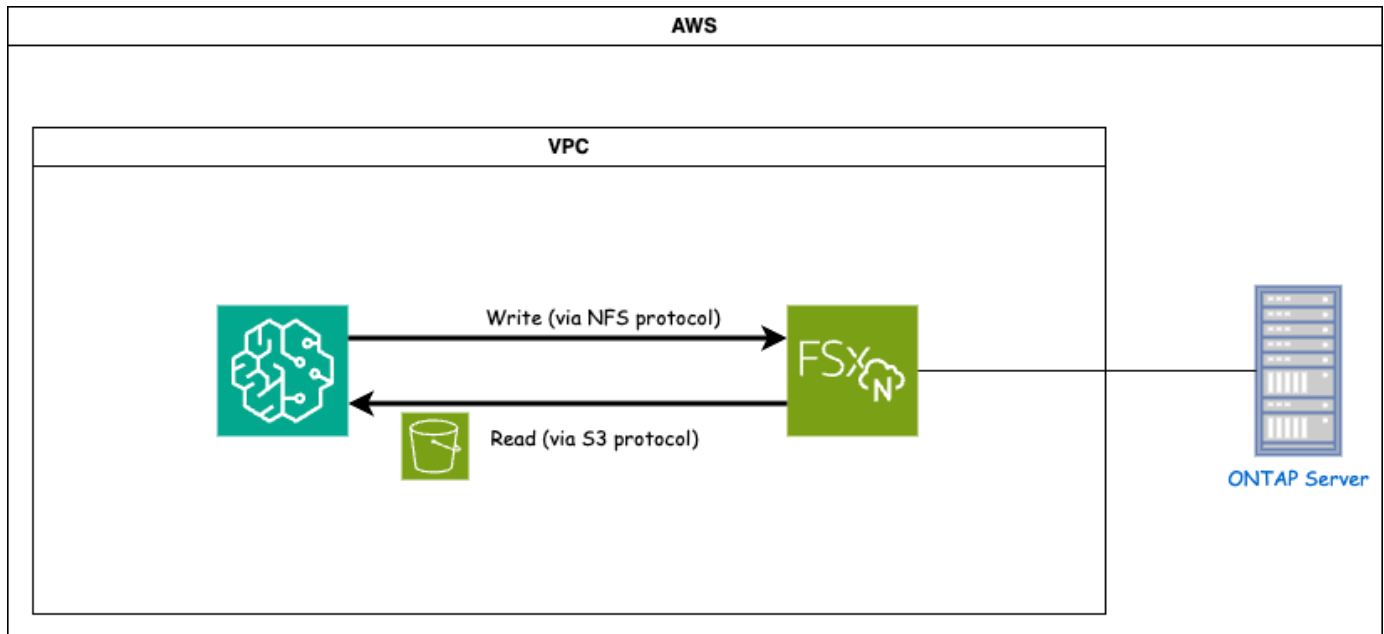
Dieses Tutorial bietet ein praktisches Beispiel für ein Computer-Vision-Klassifizierungsprojekt und bietet praktische Erfahrungen beim Aufbau von ML-Modellen, die FSxN als Datenquelle in der SageMaker-Umgebung verwenden. Der Schwerpunkt des Projekts liegt auf der Verwendung von PyTorch, einem Deep-Learning-Framework, um die Reifenqualität anhand von Reifenbildern zu klassifizieren. Er betont die Entwicklung von Machine-Learning-Modellen, die FSxN als Datenquelle in Amazon SageMaker verwenden.

### Was ist FSxN

Amazon FSX for NetApp ONTAP ist eine wirklich vollständig gemanagte Storage-Lösung von AWS. Es nutzt das ONTAP Filesystem von NetApp, um zuverlässigen und hochperformanten Storage bereitzustellen. Durch die Unterstützung von Protokollen wie NFS, SMB und iSCSI ermöglicht sie einen nahtlosen Zugriff von verschiedenen Computing-Instanzen und Containern. Der Service bietet eine außergewöhnliche Performance, die schnelle und effiziente Datenvorgänge ermöglicht. Es bietet zudem eine hohe Verfügbarkeit und Langlebigkeit, wodurch sichergestellt wird, dass die Daten zugänglich und geschützt bleiben. Darüber hinaus ist die Storage-Kapazität von Amazon FSX for NetApp ONTAP skalierbar, sodass Sie sie einfach an Ihre Anforderungen anpassen können.

## Voraussetzung

### Netzwerkumgebung



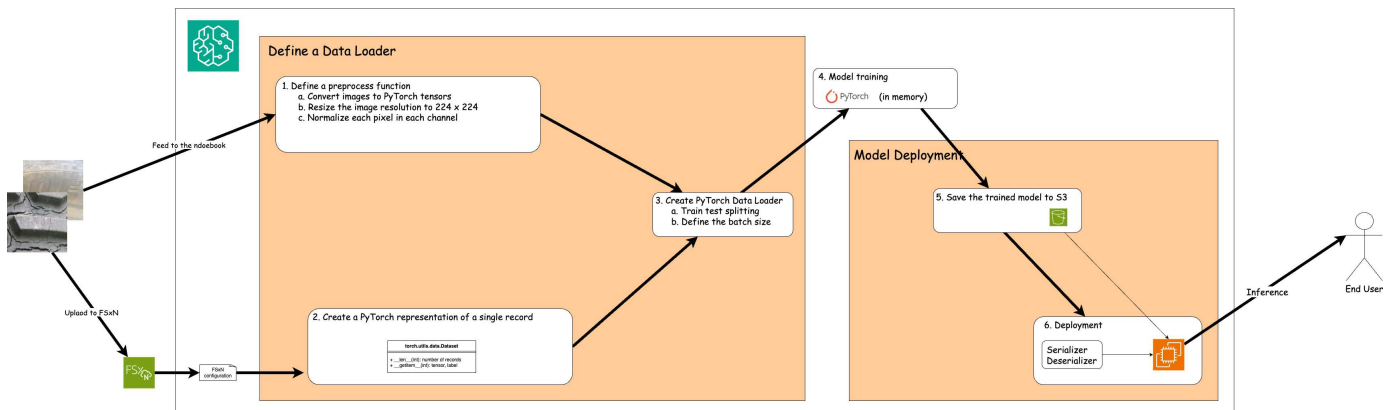
FSxN (Amazon FSX for NetApp ONTAP) ist ein AWS Storage-Service. Sie umfasst ein Filesystem, das auf dem NetApp ONTAP System ausgeführt wird, und eine AWS-gemanagte System Virtual Machine (SVM), die eine Verbindung mit ihr herstellt. Im angegebenen Diagramm befindet sich der von AWS gemanagte NetApp ONTAP-Server außerhalb der VPC. Die SVM dient als Vermittler zwischen SageMaker und dem NetApp ONTAP-System, empfängt Betriebsanforderungen von SageMaker und leitet sie an den zugrunde liegenden Speicher weiter. Für den Zugriff auf FSxN muss SageMaker innerhalb derselben VPC wie die FSxN-Bereitstellung platziert werden. Diese Konfiguration gewährleistet die Kommunikation und den Datenzugriff zwischen SageMaker und FSxN.

### Datenzugriff

In realen Szenarien verwenden Data Scientists in der Regel die in FSxN gespeicherten Daten, um ihre Modelle für maschinelles Lernen zu erstellen. Da das FSxN-Dateisystem nach der Erstellung jedoch zunächst leer ist, müssen die Trainingsdaten für Demonstrationszwecke manuell hochgeladen werden. Dies kann erreicht werden, indem FSxN als Volumen auf SageMaker montiert wird. Nachdem das Dateisystem erfolgreich gemountet wurde, können Sie den Datensatz an den gemounteten Speicherort hochladen, sodass er für das Training Ihrer Modelle in der SageMaker-Umgebung verfügbar ist. Dieser Ansatz ermöglicht es Ihnen, die Speicherkapazität und Funktionen von FSxN zu nutzen, während Sie mit SageMaker für die Modellentwicklung und das Training arbeiten.

Beim Lesen von Daten wird FSxN als privater S3-Bucket konfiguriert. Weitere Informationen zur Konfiguration finden Sie unter ["Teil 1 – Integration von AWS FSX for NetApp ONTAP \(FSxN\) als privater S3-Bucket in AWS SageMaker"](#)

### Integrationsübersicht



Der Workflow, in dem Trainingsdaten in FSxN für den Aufbau eines Deep-Learning-Modells in SageMaker verwendet werden, lässt sich in drei Hauptschritte zusammenfassen: Definition des Datenladers, Modelltraining und Bereitstellung. Auf hoher Ebene bilden diese Schritte die Grundlage einer MLOPS-Pipeline. Jeder Schritt umfasst jedoch mehrere detaillierte Teilschritte für eine umfassende Implementierung. Diese Teilschritte umfassen verschiedene Aufgaben wie Datenvorverarbeitung, Dataset-Splitting, Modellkonfiguration, Hyperparameter-Tuning, Modellbewertung, und Modellimplementierung. Diese Schritte gewährleisten einen gründlichen und effektiven Prozess für die Erstellung und Implementierung von Deep-Learning-Modellen unter Verwendung von Trainingsdaten von FSxN in der SageMaker-Umgebung.

## Schritt-für-Schritt-Integration

### Datenlader

Um ein PyTorch Deep-Learning-Netzwerk mit Daten zu trainieren, wird ein Datenlader erstellt, um die Dateneinspeisung zu erleichtern. Der Data Loader definiert nicht nur die Batch-Größe, sondern bestimmt auch den Ablauf zum Lesen und Vorverarbeiten jedes Datensatzes innerhalb des Stapels. Durch die Konfiguration des Data Loader können wir die Verarbeitung von Daten in Batches übernehmen und so das Training des Deep-Learning-Netzwerks ermöglichen.

Der Datenlader besteht aus 3 Teilen.

### Vorverarbeitungsfunktion

```
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]
    )
])
```

Der oben genannte Code-Snippet demonstriert die Definition von Bildvorverarbeitungstransformationen mit dem Modul **torchvision.transforms**. In dieser funktoriellen Funktion wird das vorprozessierende Objekt erstellt, um eine Reihe von Transformationen anzuwenden. Erstens wandelt die **ToTensor()** Transformation das Bild in eine Tensor-Darstellung um. Anschließend passt die Umwandlung **Resize 224,224** das Bild auf eine feste

Größe von 24x224 Pixeln an. Schließlich normalisiert die **Normalisieren()** Transformation die Tensor-Werte, indem sie den Mittelwert subtrahiert und durch die Standardabweichung entlang jedes Kanals dividiert. Die für die Normalisierung verwendeten Werte für Mittelwert und Standardabweichung werden häufig in vortrainierten neuronalen Netzmodellen verwendet. Insgesamt bereitet dieser Code die Bilddaten für die weitere Verarbeitung oder Eingabe in ein vortrainiertes Modell vor, indem er sie in einen Tensor konvertiert, die Größe verändert und die Pixelwerte normalisiert.

### Die PyTorch-Dataset-Klasse

```
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
```

Diese Klasse bietet Funktionen zum Abrufen der Gesamtzahl der Datensätze im Datensatz und definiert die Methode zum Lesen von Daten für jeden Datensatz. Innerhalb der Funktion **getitem** verwendet der Code das Bucket-Objekt boto3 S3, um die Binärdaten aus FSxN abzurufen. Der Code-Stil für den Zugriff auf Daten aus FSxN ähnelt dem Lesen von Daten aus Amazon S3. Die nachfolgende Erklärung geht auf den Erstellungsprozess des privaten S3-Objekts **bucket** ein.

## FSxN als privates S3-Repository

```
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 ---
```

Um Daten aus FSxN in SageMaker zu lesen, wird ein Handler erstellt, der auf den FSxN-Speicher mit dem S3-Protokoll verweist. Auf diese Weise kann FSxN als privater S3-Bucket behandelt werden. Die Handler-Konfiguration umfasst die Angabe der IP-Adresse der FSxN SVM, des Bucket-Namens und der erforderlichen Anmeldedaten. Eine umfassende Erklärung zum Bezug dieser Konfigurationselemente finden Sie in dem Dokument unter ["Teil 1 – Integration von AWS FSX for NetApp ONTAP \(FSxN\) als privater S3-Bucket in AWS SageMaker"](#).

In dem oben genannten Beispiel wird das Bucket-Objekt verwendet, um das PyTorch-Datensatzobjekt zu instanziierten. Das Datensatzobjekt wird im nachfolgenden Abschnitt näher erläutert.

## Der 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)
```

Im angegebenen Beispiel wird eine Batch-Größe von 64 angegeben, was darauf hinweist, dass jeder Batch 64 Datensätze enthält. Durch die Kombination der PyTorch **Datensatz** Klasse, der Vorverarbeitungsfunktion und der Training Batch Größe erhalten wir den Data Loader für das Training. Dieser Daten-Loader erleichtert den Prozess, den Datensatz während der Trainingsphase in Batches zu durchlaufen.

## Modelltraining

```
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')

```

Dieser Code implementiert einen standardmäßigen PyTorch-Trainingsprozess. Es definiert ein neuronales Netzmodell mit dem Namen **TyreQualityClassifier**, das konvolutionelle Schichten und eine lineare Schicht verwendet, um die Reifenqualität zu klassifizieren. Die Trainingsschleife iteriert Daten-Batches, berechnet den Verlust und aktualisiert die Parameter des Modells mittels Rückverbreitung und Optimierung. Außerdem werden die aktuelle Zeit, die aktuelle Epoche, der Stapel und der Verlust für Überwachungszwecke gedruckt.

## Modellbereitstellung

### Einsatz

```

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())

```

Der Code speichert das PyTorch-Modell in **Amazon S3**, da SageMaker das Modell für die Bereitstellung in S3 speichern muss. Durch das Hochladen des Modells auf **Amazon S3** wird es für SageMaker zugänglich, was die Bereitstellung und Inferenz auf dem bereitgestellten Modell ermöglicht.

```

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
)

```

Dieser Code erleichtert die Bereitstellung eines PyTorch-Modells auf SageMaker. Es definiert einen benutzerdefinierten Serialisator, **TyreQualitySerializer**, der Eingabedaten als PyTorch-Tensor vorverarbeitet und serialisiert. Die Klasse **TyreQualityPredictor** ist ein benutzerdefinierter Prädiktor, der den definierten Serialisator und einen **JSONDeserializer** verwendet. Der Code erstellt außerdem ein **PyTorchModel**-Objekt, um den S3-Standort des Modells, die IAM-Rolle, die Framework-Version und den Eintrittspunkt für die Inferenz

festzulegen. Der Code generiert einen Zeitstempel und erstellt einen Endpunktnamen basierend auf dem Modell und dem Zeitstempel. Schließlich wird das Modell mithilfe der Bereitstellungsmethode bereitgestellt, wobei die Anzahl der Instanzen, der Instanztyp und der Name des generierten Endpunkts angegeben werden. Dadurch kann das PyTorch-Modell auf SageMaker bereitgestellt und für Inferenz zugänglich sein.

## Inferenz

```
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)
```

Dies ist das Beispiel für die Verwendung des implementierten Endpunkts zur Inferenz.

# Teil 3 – Aufbau Einer vereinfachten MLOPS-Pipeline (CI/CT/CD)

## Autor(en):

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

## Einführung

In diesem Tutorial erfahren Sie, wie Sie verschiedene AWS-Services für die Erstellung einer einfachen MLOPS-Pipeline nutzen können, die Continuous Integration (CI), Continuous Training (CT) und Continuous Deployment (CD) umfasst. Im Gegensatz zu herkömmlichen DevOps-Pipelines erfordert MLOps beim Abschluss des Betriebszyklus zusätzliche Überlegungen. Wenn Sie dieses Tutorial befolgen, erhalten Sie Einblicke in die Integration von CT in die MLOPS-Schleife, was ein kontinuierliches Training Ihrer Modelle und eine nahtlose Bereitstellung für die Inferenz ermöglicht. Das Tutorial führt Sie durch die Nutzung von AWS-Services zur Einrichtung dieser End-to-End-MLOPS-Pipeline.

## Manifest

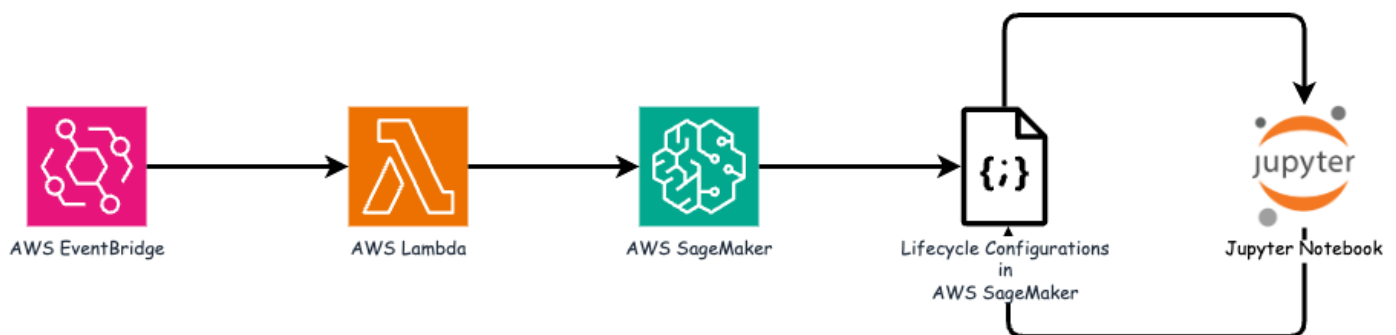
Funktionalität	Name	Kommentar
Datenspeicher	AWS FSxN	Siehe <a href="#">"Teil 1 – Integration von AWS FSX for NetApp ONTAP (FSxN) als privater S3-Bucket in AWS SageMaker"</a> .
Data Science IDE	AWS SageMaker	Dieses Tutorial basiert auf dem in vorgestellten Jupyter Notebook <a href="#">"Teil 2 – Nutzung von AWS FSX for NetApp ONTAP (FSxN) als Datenquelle für das Modelltraining in SageMaker"</a> .

Funktionalität	Name	Kommentar
Funktion zum Auslösen der MLOPS-Pipeline	AWS Lambda-Funktion	-
Cron-Job-Trigger	AWS EventBridge	-
Deep-Learning-Framework	PyTorch	-
AWS Python SDK	Boto3	-
Programmiersprache	Python	v3.10

## Voraussetzung

- Ein vorkonfiguriertes FSxN-Dateisystem. In diesem Tutorial werden die in FSxN gespeicherten Daten für den Trainingsprozess verwendet.
- Eine **SageMaker Notebook-Instanz**, die so konfiguriert ist, dass sie dieselbe VPC wie das oben erwähnte FSxN-Dateisystem verwendet.
- Bevor Sie die **AWS Lambda-Funktion** aktivieren, stellen Sie sicher, dass die **SageMaker Notebook-Instanz** den Status **angehalten** hat.
- Der Instanztyp **ml.g4dn.xlarge** wird benötigt, um die GPU-Beschleunigung zu nutzen, die für die Berechnung tiefer neuronaler Netzwerke notwendig ist.

## Der Netapp Architektur Sind



Diese MLOPS-Pipeline ist eine praktische Implementierung, die mithilfe eines Cron-Jobs eine serverlose Funktion auslöst, die wiederum einen AWS-Service ausführt, der mit einer Lifecycle-Callback-Funktion registriert ist. Die **AWS EventBridge** fungiert als Cron-Job. Es ruft regelmäßig eine **AWS Lambda-Funktion** auf, die für die Umschulung und Neuimplementierung des Modells verantwortlich ist. Bei diesem Vorgang wird die **AWS SageMaker Notebook-Instanz** hochgefahren, um die erforderlichen Aufgaben auszuführen.

## Schritt-für-Schritt-Konfiguration

### Lebenszykluskonfigurationen

Um die Lifecycle-Callback-Funktion für die AWS SageMaker Notebook-Instanz zu konfigurieren, würden Sie **Lifecycle-Konfigurationen** verwenden. Mit diesem Service können Sie die erforderlichen Aktionen definieren, die beim Starten der Notebook-Instanz ausgeführt werden müssen. Konkret kann ein Shell-Skript innerhalb der **Lifecycle-Konfigurationen** implementiert werden, um die Notebook-Instanz nach Abschluss der Trainings- und Bereitstellungsprozesse automatisch herunterzufahren. Dies ist eine erforderliche Konfiguration, da die Kosten eine der wichtigsten Überlegungen bei MLOPS sind.

Es ist wichtig zu beachten, dass die Konfiguration für **Lifecycle-Konfigurationen** im Voraus eingerichtet werden muss. Daher wird empfohlen, die Konfiguration dieses Aspekts zu priorisieren, bevor mit dem Setup der anderen MLOPS-Pipeline fortgefahren wird.

1. Um eine Lifecycle-Konfiguration einzurichten, öffnen Sie das Fenster **SageMaker** und navigieren Sie zu **Lifecycle-Konfigurationen** unter dem Abschnitt **Admin-Konfigurationen**.

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 endpoint.

► Domain structure diagram

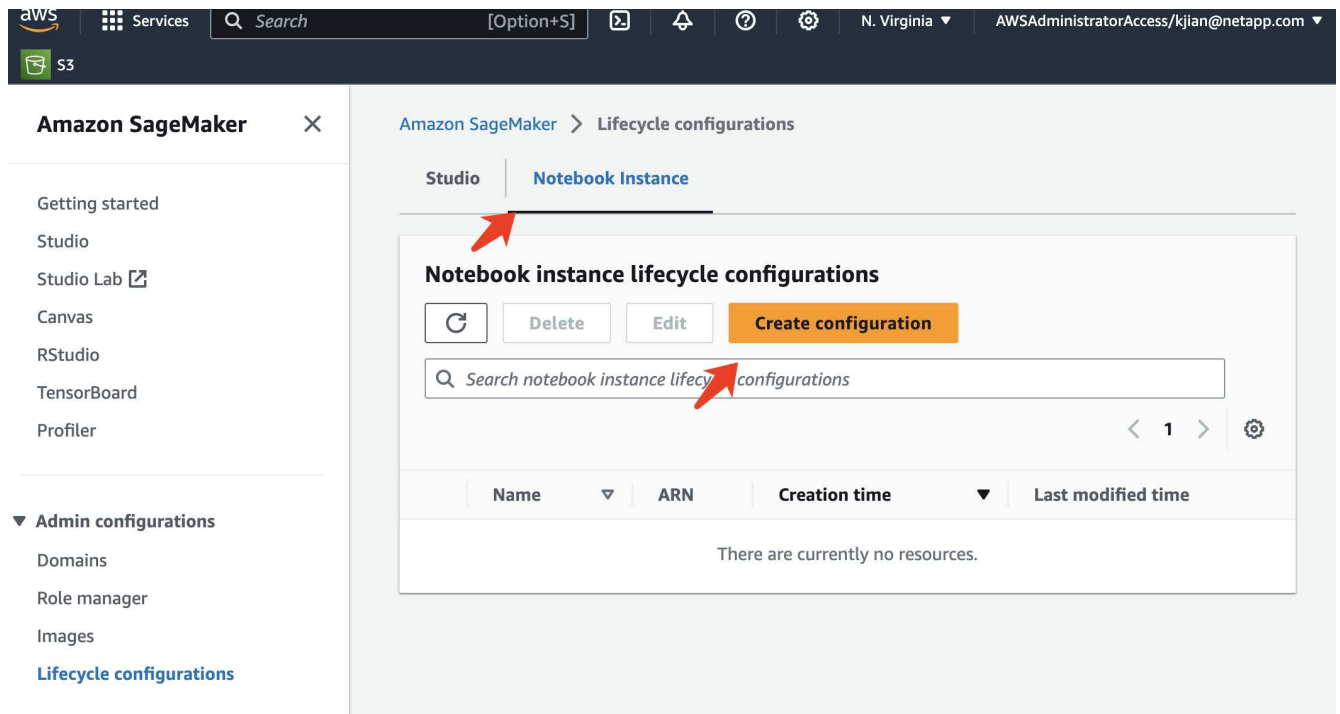
Domains (4)

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. Wählen Sie die Registerkarte **Notebook-Instanz** aus und klicken Sie auf die Schaltfläche **Konfiguration erstellen**



3. Fügen Sie den unten stehenden Code in den Eingabebereich ein.

```
#!/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. Dieses Skript führt das Jupyter Notebook aus, das das erneute Training und die erneute Bereitstellung des Modells für die Inferenz übernimmt. Nach Abschluss der Ausführung wird das Notebook automatisch innerhalb von 5 Minuten heruntergefahren. Weitere Informationen zur Problembeschreibung und zur Codeimplementierung finden Sie unter ["Teil 2 – Nutzung von AWS FSX for NetApp ONTAP \(FSxN\) als Datenquelle für das Modelltraining in 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. Navigieren Sie nach der Erstellung zu Notizbuchinstanzen, wählen Sie die Zielinstanz aus, und klicken Sie im Dropdown-Menü Aktionen auf **Einstellungen aktualisieren**.

Amazon SageMaker > Notebook instances

Notebook instances Info

Search notebook instances

Name	Instance	Creation time	Status	Actions
fsxn-ontap	ml.g4dn.xlarge	9/29/2020	Stopped	Start

Actions

- Open Jupyter
- Open JupyterLab
- Stop
- Start
- Update settings
- Add/Edit tags
- Delete

6. Wählen Sie die erstellte **Lifecycle-Konfiguration** aus und klicken Sie auf **Notebook-Instanz aktualisieren**.

Amazon SageMaker > Notebook instances > fsxn-ontap > Edit notebook instance

## Edit notebook instance

**Notebook instance settings**

Notebook instance name: fsxn-ontap

Notebook instance type: ml.g4dn.xlarge

Elastic Inference: none

Platform identifier: Amazon Linux 2, Jupyter Lab 3

**Additional configuration**

Lifecycle configuration - optional

Customize your notebook environment with default scripts and plugins.

fsxn-demo-lifecycle-callback

No configuration

Create a new lifecycle configuration

fsxn-demo-lifecycle-callback

2

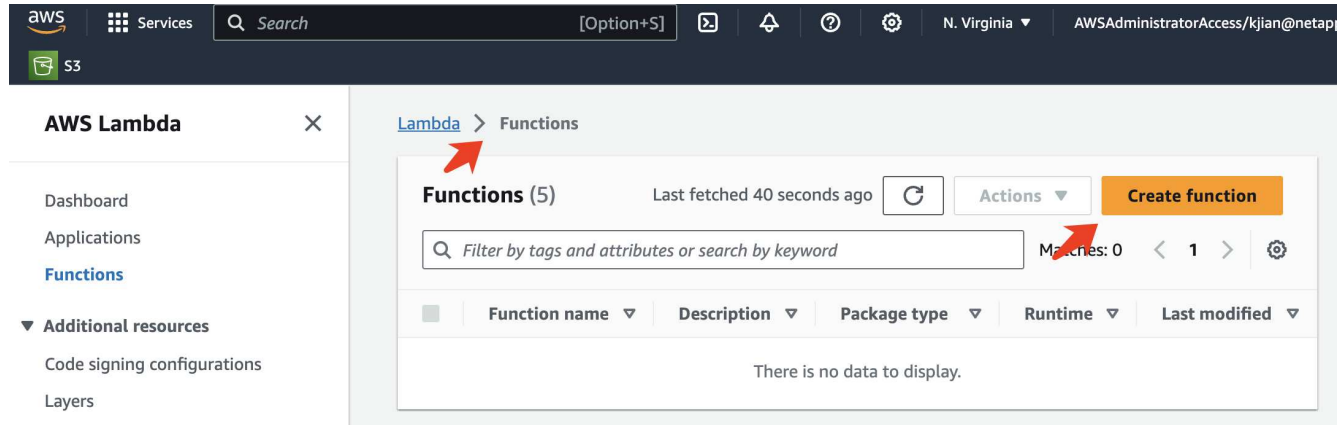
Permissions and encryption



## AWS Lambda serverlose Funktion

Wie bereits erwähnt, ist die **AWS Lambda-Funktion** für die Einrichtung der **AWS SageMaker Notebook-Instanz** zuständig.

1. Um eine **AWS Lambda-Funktion** zu erstellen, navigieren Sie zum entsprechenden Panel, wechseln Sie zum Reiter **Funktionen** und klicken Sie auf **Create Function**.



2. Bitte legen Sie alle erforderlichen Einträge auf der Seite ab und denken Sie daran, die Runtime auf **Python 3.10** umzuschalten.

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.

- Bitte überprüfen Sie, ob die vorgesehene Rolle die erforderliche Berechtigung hat **AmazonSageMakerFullAccess** und klicken Sie auf den Button **Funktion erstellen**.

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. Wählen Sie die erstellte Lambda-Funktion aus. Kopieren Sie auf der Registerkarte Code den folgenden Code, und fügen Sie ihn in den Textbereich ein. Dieser Code startet die Notebook-Instanz mit dem Namen **fsxn-ontap**.

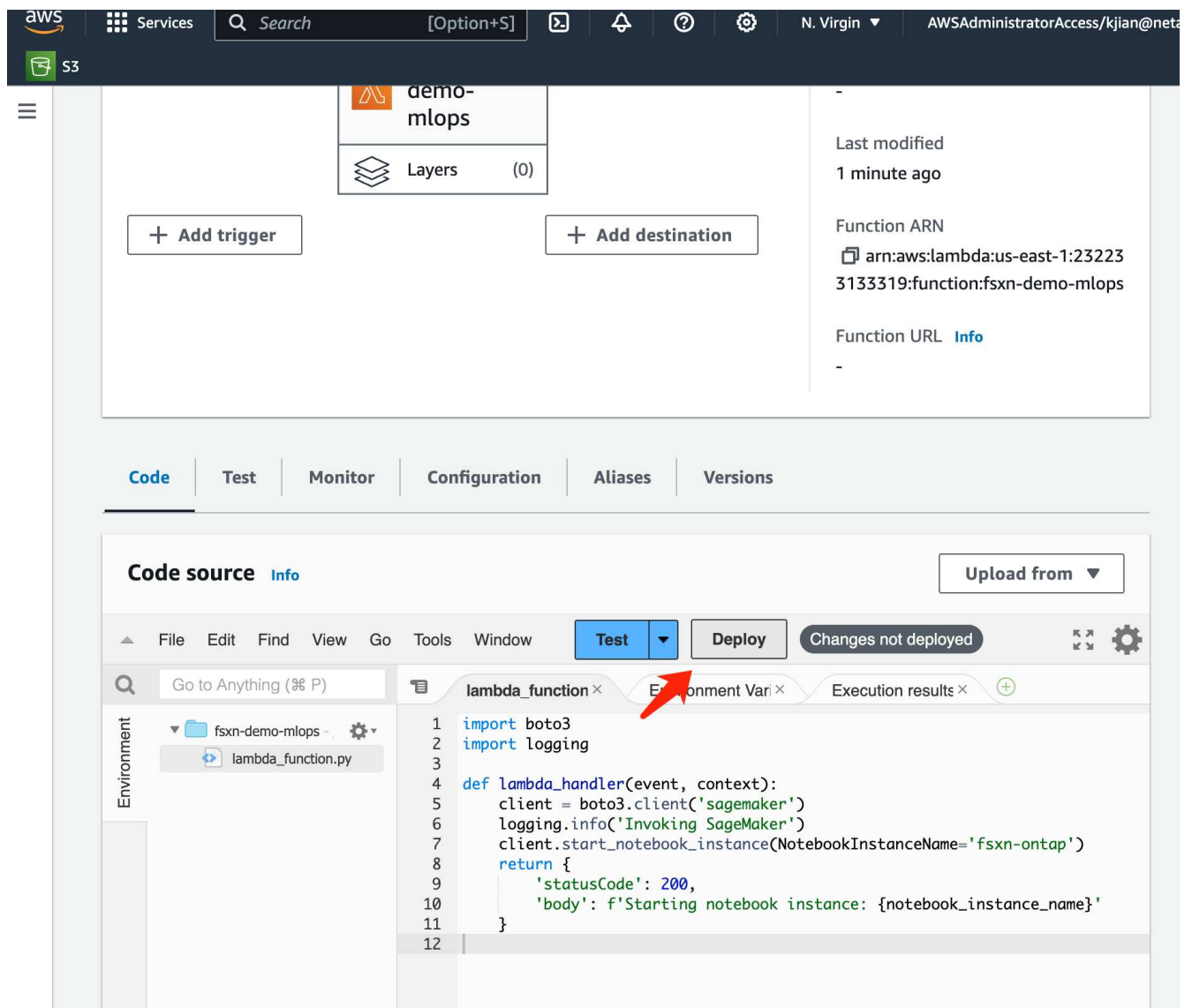
```

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. Klicken Sie auf die Schaltfläche **deploy**, um diese Codeänderung anzuwenden.



6. Um anzugeben, wie diese AWS Lambda-Funktion ausgelöst werden soll, klicken Sie auf die Schaltfläche **Add Trigger**.

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 account information. The breadcrumb trail indicates the current location: [Lambda](#) > [Functions](#) > fsxn-demo-mlops. The function name 'fsxn-demo-mlops' is prominently displayed, with buttons for 'Throttle', 'Copy ARN', and 'Actions' to its right. Below this, 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. To the right of the function card, a sidebar provides 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. Wählen Sie EventBridge aus dem Dropdown-Menü aus, und klicken Sie dann auf das Optionsfeld Neue Regel erstellen. Geben Sie im Feld Ausdruck Zeitplan ein `rate(1 day)`, und klicken Sie auf die Schaltfläche Hinzufügen, um diese neue Cron-Job-Regel auf die AWS Lambda-Funktion zu erstellen und anzuwenden.

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

Nach Abschluss der zweistufigen Konfiguration startet die **AWS Lambda-Funktion** täglich das **SageMaker Notebook**, führt Modellumschulungen mit den Daten aus dem **FSxN-Repository** durch, stellt das aktualisierte Modell in der Produktionsumgebung wieder bereit und fährt die **SageMaker Notebook-Instanz** automatisch herunter, um die Kosten zu optimieren. Damit bleibt das Modell auf dem neuesten Stand.

Damit ist das Tutorial zur Entwicklung einer MLOPS-Pipeline abgeschlossen.

## Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

## Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.