



데이터 파이프라인, 데이터 레이크 및 관리 NetApp Solutions

NetApp
April 20, 2024

목차

데이터 파이프라인, 데이터 레이크 및 관리	1
MLOps용 AWS FSx for NetApp ONTAP(FSxN)	1
Domino Data Lab 및 NetApp의 하이브리드 멀티 클라우드 MLOps	36
NetApp 및 VMware를 사용하는 NVIDIA AI Enterprise	50
TR-4851: 자율 주행 워크로드를 위한 NetApp StorageGRID 데이터 레이크 - 솔루션 설계	60
NetApp AI Control Plane	60
Iguazio를 사용한 MLRun 파이프라인	113
TR-4915: AI 및 분석 워크플로우를 위해 E-Series 및 BeeGFS로 데이터 이동	140

데이터 파이프라인, 데이터 레이크 및 관리

MLOps용 AWS FSx for NetApp ONTAP(FSxN)

• 작성자: *

Jian Jian(Ken), NetApp 수석 데이터 및 응용 과학자

이 섹션에서는 FSxN을 사용한 MLOps 파이프라인 구축에 대한 종합적인 단계별 안내를 제공하여 AI 인프라 개발의 실제 적용 사례를 자세히 설명합니다. 세 가지 포괄적인 예시로 구성된 이 강력한 데이터 관리 플랫폼을 통해 MLOps 요구 사항을 충족할 수 있습니다.

이 문서에서는 다음 사항에 중점을 둡니다.

1. "1부 - AWS FSx for NetApp ONTAP(FSxN)를 프라이빗 S3 버킷으로 AWS SageMaker에 통합"
2. "2부 - AWS FSx for NetApp ONTAP(FSxN)를 SageMaker에서 모델 훈련을 위한 데이터 소스로 활용"
3. "3부 - 간소화된 MLOps 파이프라인 구축(CI/CT/CD)"

이 섹션을 마치면 FSxN을 사용하여 MLOps 프로세스를 간소화하는 방법을 확실히 이해할 수 있게 됩니다.

1부 - AWS FSx for NetApp ONTAP(FSxN)를 프라이빗 S3 버킷으로 AWS SageMaker에 통합

• 작성자: *

Jian Jian(Ken), NetApp 수석 데이터 및 응용 과학자

소개

SageMaker를 예로 사용하는 이 페이지에서는 FSxN을 전용 S3 버킷으로 구성하는 방법에 대한 지침을 제공합니다.

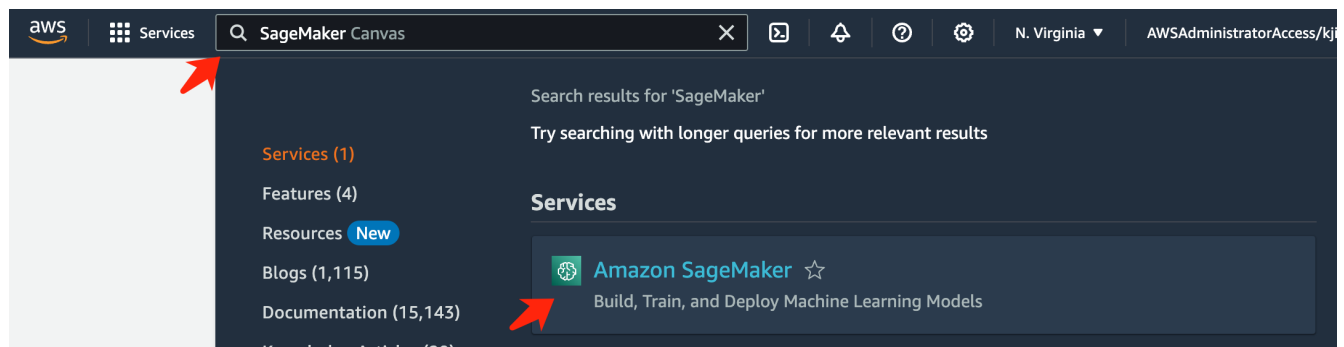
FSxN에 대한 자세한 내용은 이 프레젠테이션(["비디오 링크"](#))

사용 설명서

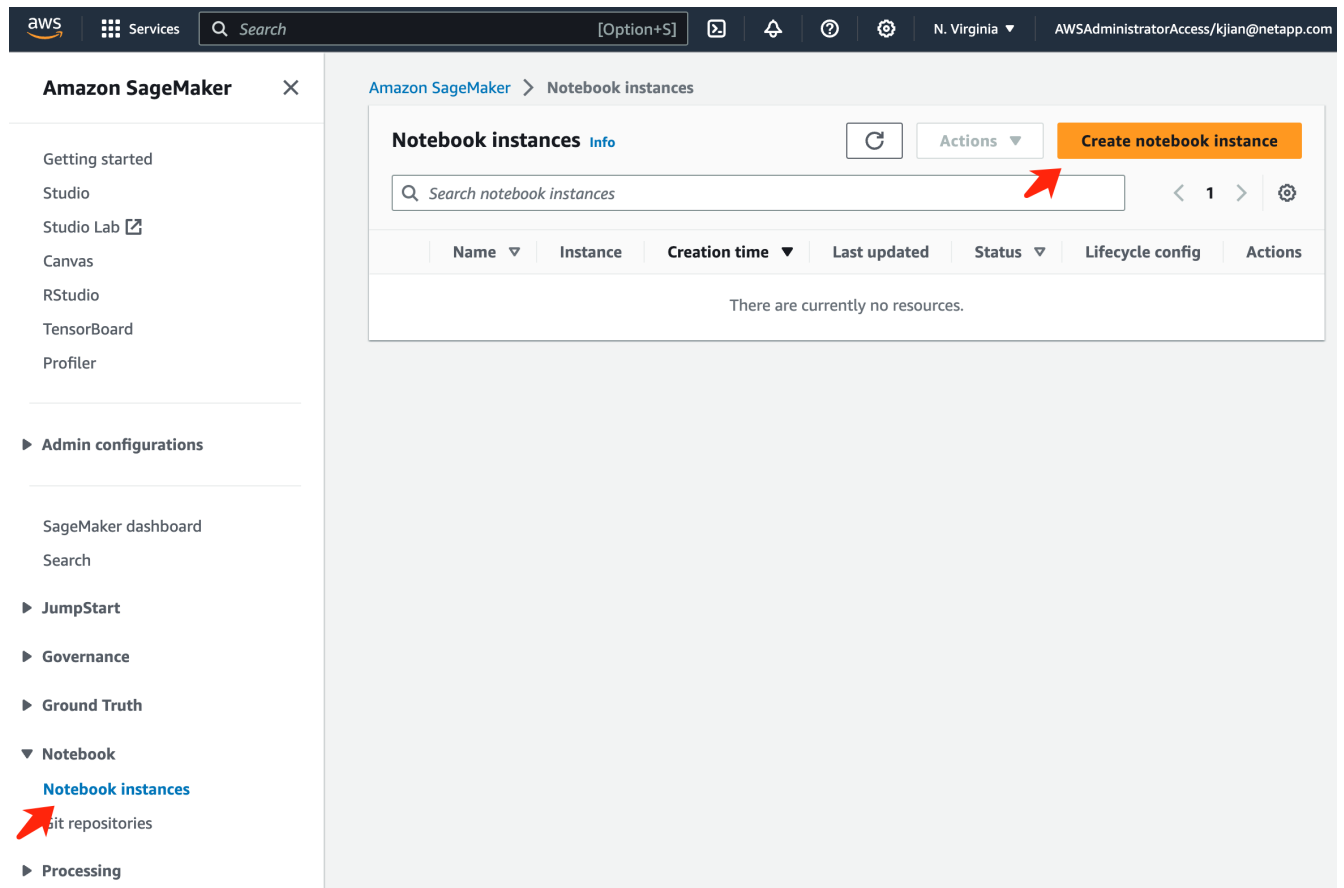
서버 생성

SageMaker 전자 필기장 인스턴스를 만듭니다

1. AWS 콘솔을 엽니다. 검색 패널에서 SageMaker를 검색하고 서비스 * Amazon SageMaker * 를 클릭합니다.



2. 노트북 탭에서 * 노트북 인스턴스 * 를 열고 주황색 버튼 * 노트북 인스턴스 만들기 * 를 클릭합니다.



3. 생성 페이지에서
노트북 인스턴스 이름 * 을 입력합니다
네트워크 * 패널을 확장합니다
다른 항목을 기본값으로 두고 * VPC *, * 서브넷 * 및 * 보안 그룹 * 을 선택합니다. (이 * VPC * 및 * 서브넷 * 은 나중에 FSxN 파일 시스템을 생성하는 데 사용됩니다.)
오른쪽 아래에 있는 주황색 버튼 * 노트북 인스턴스 만들기 * 를 클릭합니다.

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

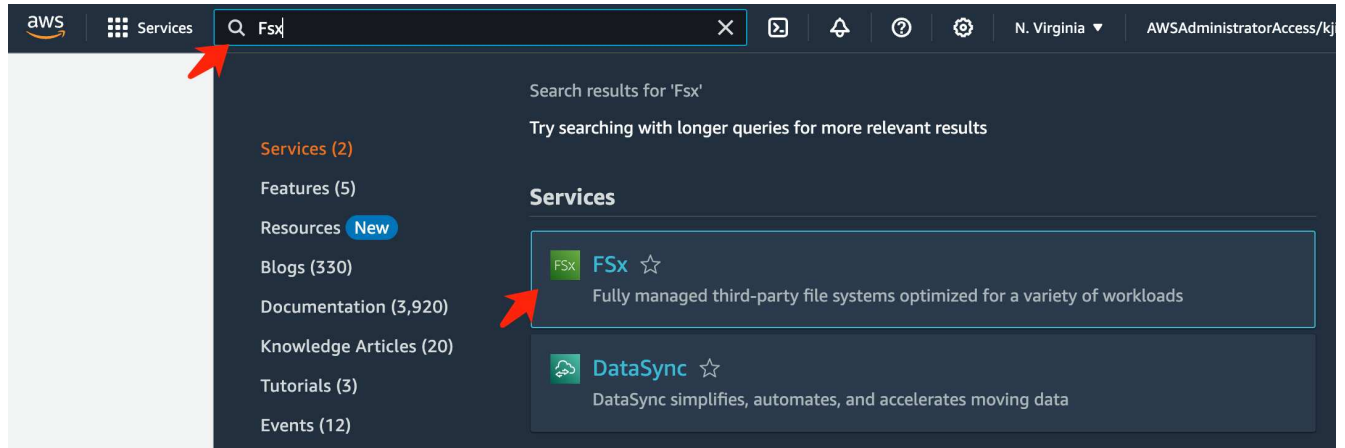
► Tags - optional

Cancel

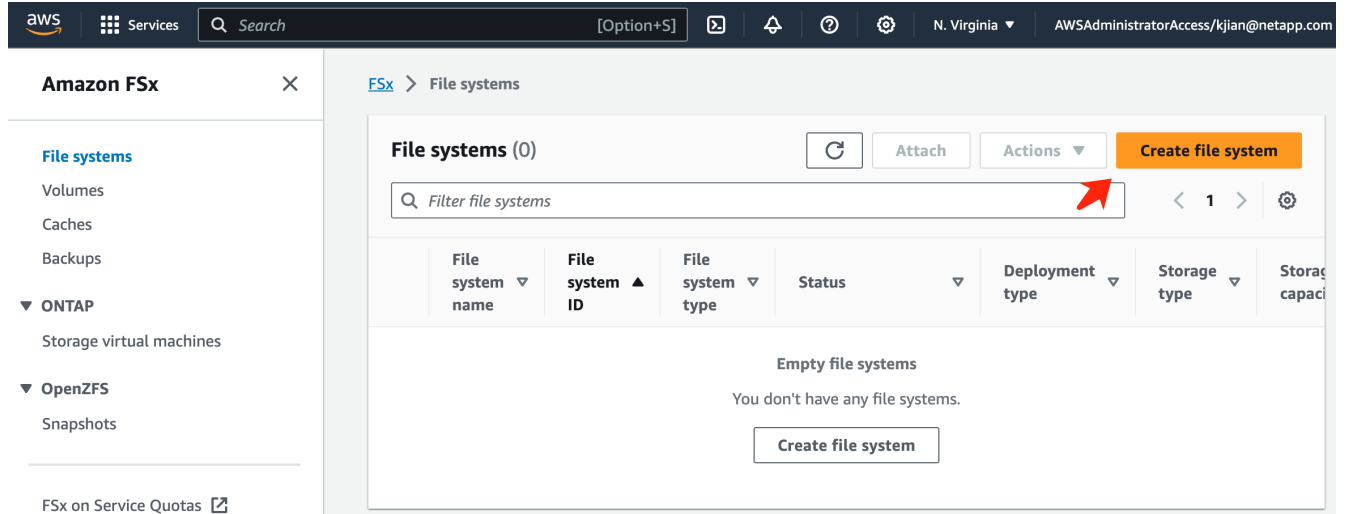
Create notebook instance

FSxN 파일 시스템을 생성합니다

1. AWS 콘솔을 엽니다. 검색 패널에서 FSX를 검색하고 서비스 * FSx * 를 클릭합니다.



2. Create file system * 을 클릭합니다.



3. 첫 번째 카드 * FSx for NetApp ONTAP * 를 선택하고 * Next * 를 클릭합니다.

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

FSx > File systems > Create file system

Step 1
Select file system type

Step 2
Specify file system details

Step 3
Review and create

Select file system type

File system options

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

Amazon FSx for NetApp ONTAP

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

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

Cancel Next

4. 세부 정보 구성 페이지

a. 표준 생성 * 옵션을 선택합니다.

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. 파일 시스템 이름 * 과 * SSD 스토리지 용량 * 을 입력합니다.

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. SageMaker Notebook * 인스턴스와 동일한 * VPC * 및 * 서브넷 * 을 사용해야 합니다.

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. 스토리지 가상 머신 * 이름을 입력하고 * SVM(스토리지 가상 머신)에 대한 암호 * 를 지정합니다.

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

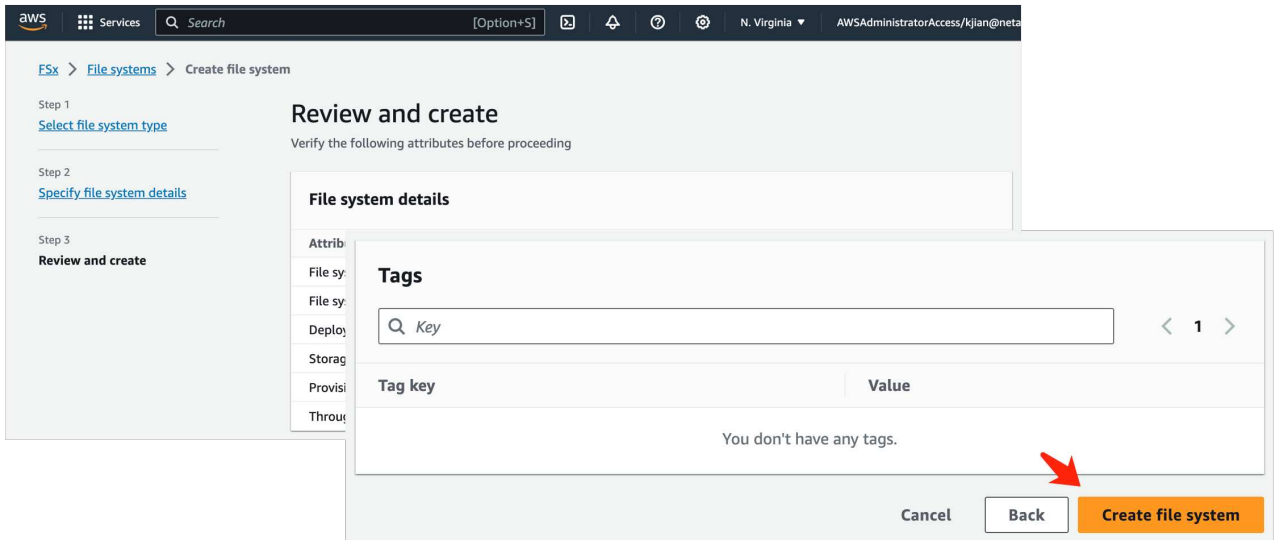
e. 다른 항목은 기본값으로 두고 오른쪽 하단에 있는 주황색 버튼 * Next * 를 클릭합니다.

► **Backup and maintenance - optional**

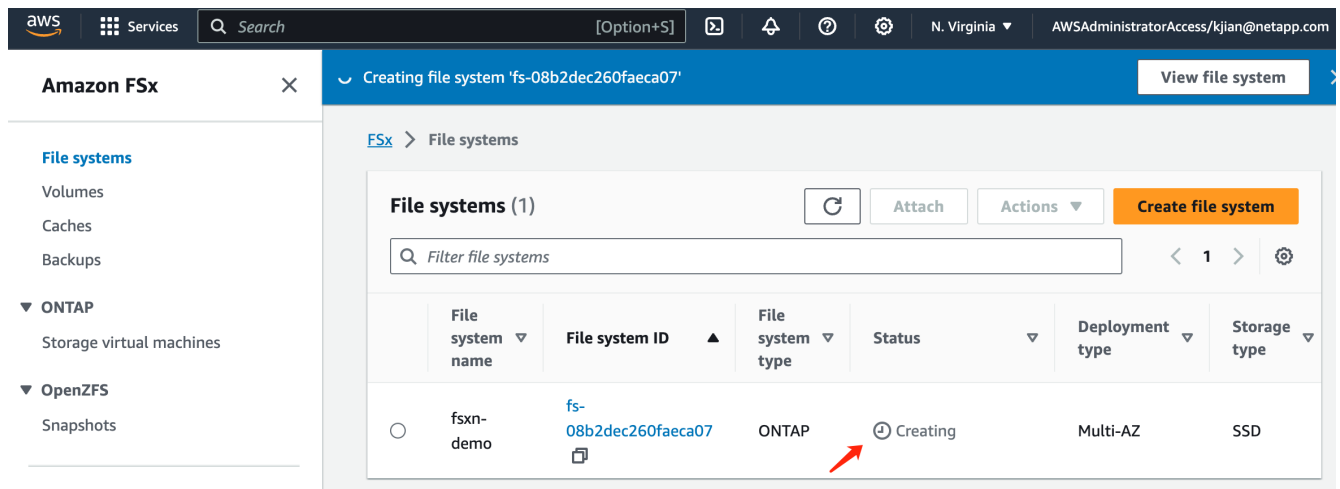
► **Tags - optional**

Cancel Back Next

f. 검토 페이지 오른쪽 아래에 있는 주황색 버튼 * 파일 시스템 생성 * 을 클릭합니다.



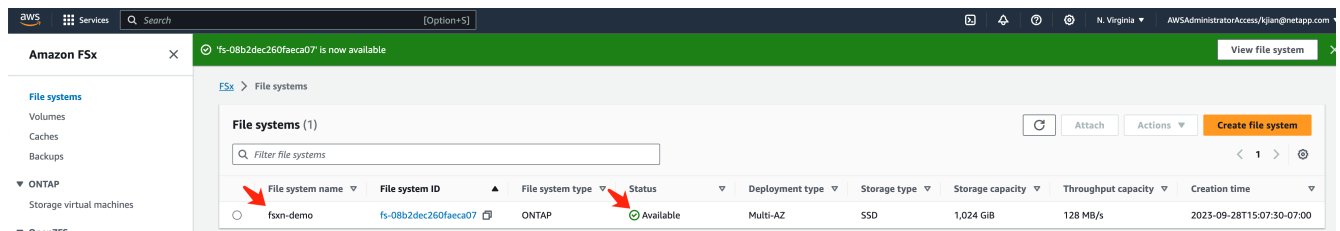
5. FSx 파일 시스템을 구동하는 데 약 * 20-40분 * 정도 걸릴 수 있습니다.



서버 구성

ONTAP 구성

1. 생성된 FSx 파일 시스템을 엽니다. 상태가 * 사용 가능 * 인지 확인하십시오.



2. 관리 * 탭을 선택하고 * 관리 끝점 - IP 주소 * 및 * ONTAP 관리자 사용자 이름 * 을 유지합니다.

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 172.31.31.157 172.31.32.38	ONTAP administrator password Update

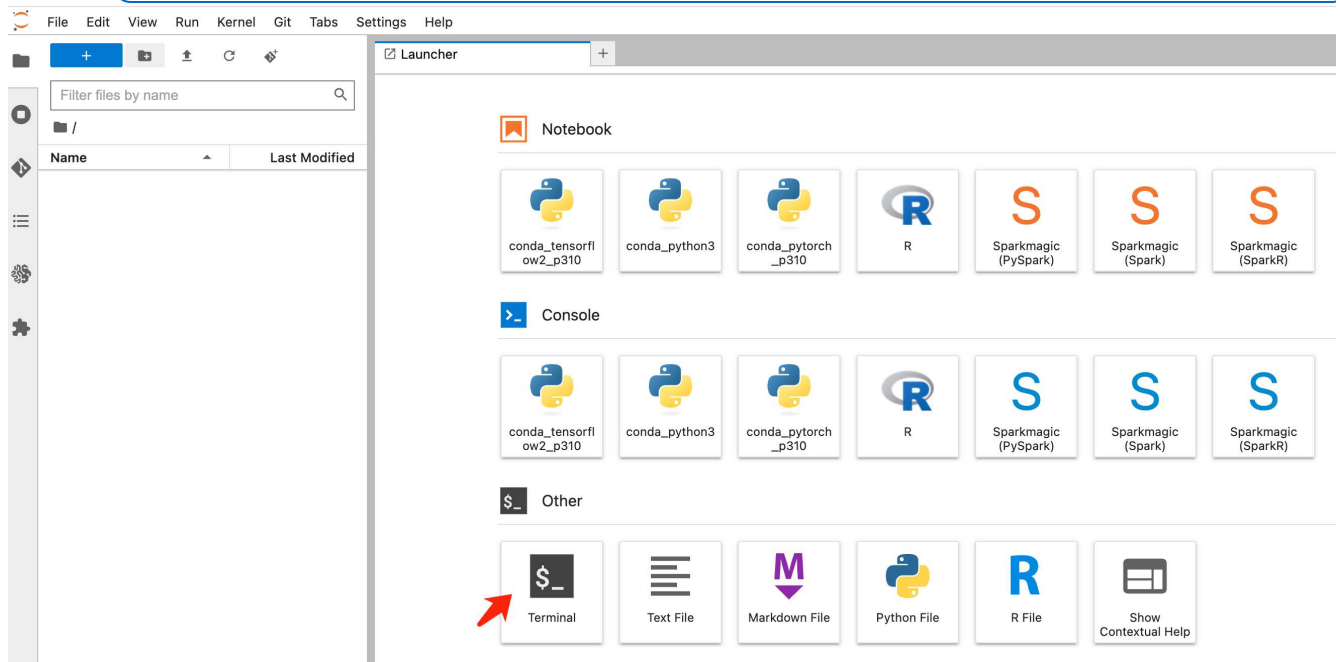
3. 생성된 * SageMaker Notebook Instance * 를 열고 * Open JupyterLab * 을 클릭합니다.

Amazon SageMaker Notebook instances

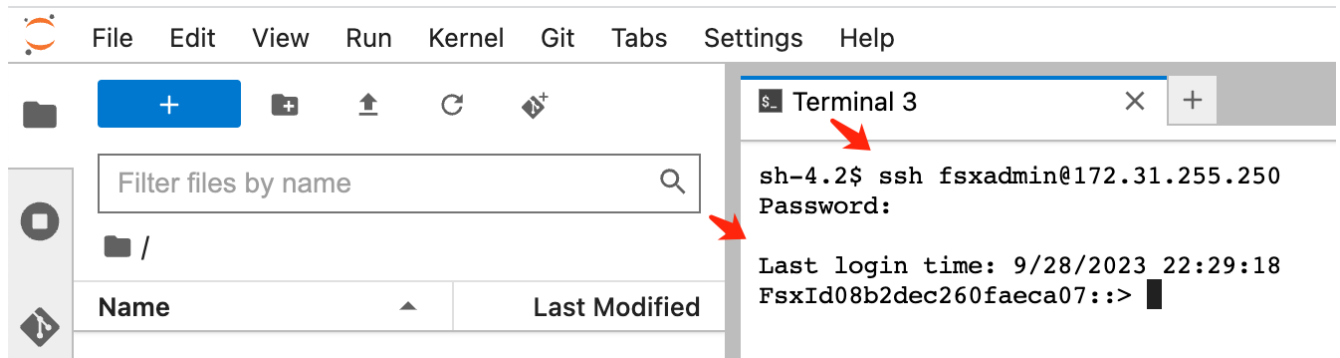
Notebook instances

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. Jupyter Lab 페이지에서 새 * Terminal * 을 엽니다.



5. ssh 명령 `ssh <admin 사용자 이름>@<ONTAP 서버 IP>` 를 입력하여 FSxN ONTAP 파일 시스템에 로그인합니다. (2단계에서 사용자 이름과 IP 주소를 조회한다.)
스토리지 가상 머신 * 을 생성할 때 사용한 암호를 사용하십시오.



6. 명령을 다음 순서로 실행합니다.
FSxN 개인 S3 버킷 이름 * 의 이름으로 * fsxn-ONTAP * 을 사용합니다.
vserver * 인수에 * 스토리지 가상 머신 이름 * 을 사용하십시오.

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

7. FSxN private S3에 대한 엔드포인트 IP 및 자격 증명을 검색하려면 다음 명령을 실행합니다.

```
network interface show -vsverer fsxn-svm-demo -lif nfs_smb_management_1
set adv
vsverer object-store-server user show
```

8. 나중에 사용할 수 있도록 끝점 IP 및 자격 증명을 유지합니다.

```
sh-4.2$ ssh fsxadmin@172.31.255.250
Password:
Last login time: 9/28/2023 22:32:42
FsxId08b2dec260faeca07:~> network interface show -vsverer 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: Fsx
FCP WWP: -
Address family: ipv4
Comment: -
IPspace of LIF: Default
Is Dynamic DNS Update Enabled?: true
Probe-port for Cloud Load Balancer: -
Broadcast Domain: Fsx
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

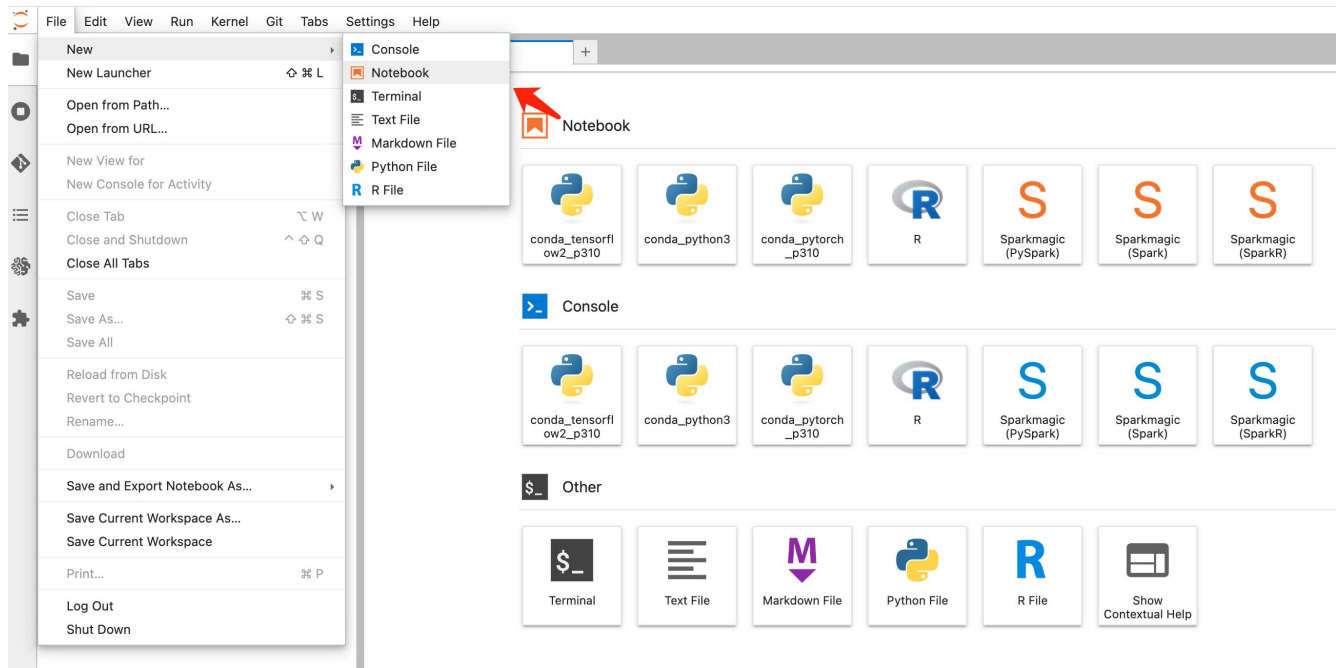
FsxId08b2dec260faeca07:~> vsverer object-store-server user show
Vserver  User      ID      Access Key      Secret Key
-----  -
fsxn-svm-demo
  root      0      -              -
  Comment: Root User
fsxn-svm-demo
  s3user    1      AWS Access Key Id AWS Secret Access Key

2 entries were displayed.

FsxId08b2dec260faeca07:~>
```

클라이언트 구성

1. SageMaker Notebook 인스턴스에서 새 Jupyter 노트북을 만듭니다.



2. FSxN 프라이빗 S3 버킷에 파일을 업로드하는 해결 방법으로 아래 코드를 사용하십시오.
포괄적인 코드 예제는 이 노트북을 참조하십시오.
["fsxn_demo.ipynb"를 참조하십시오](#)

```
# 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']

```

이제 FSxN 및 SageMaker 인스턴스 간의 통합을 마치겠습니다.

유용한 디버깅 체크리스트

- SageMaker Notebook 인스턴스와 FSxN 파일 시스템이 동일한 VPC에 있는지 확인합니다.

- ONTAP에서 * set dev * 명령을 실행하여 권한 수준을 * dev * 로 설정해야 합니다.

FAQ(2023년 9월 27일 기준)

Q: FSxN에 파일을 업로드할 때 "** CreateMultipartUpload 작업을 호출할 때 오류가 발생했습니다(NotImplemented). 요청한 S3 명령이 구현되지 않았습니다 *" 오류가 발생하는 이유는 무엇입니까?

A: FSxN은 전용 S3 버킷으로 최대 100MB의 파일 업로드를 지원합니다. S3 프로토콜을 사용할 때는 100MB 이상의 파일을 100MB 청크로 나누고 'CreateMultipartUpload' 기능을 호출한다. 그러나 현재 FSxN 프라이빗 S3의 구현에서는 이 기능이 지원되지 않습니다.

Q: FSxN에 파일을 업로드할 때 "** PutObject 작업: 액세스 거부 *" 를 호출할 때 오류가 발생했습니다(AccessDenied)" 오류가 발생하는 이유는 무엇입니까?

A: SageMaker Notebook 인스턴스에서 FSxN 전용 S3 버킷에 액세스하려면 AWS 자격 증명을 FSxN 자격 증명으로 전환합니다. 그러나 인스턴스에 쓰기 권한을 부여하려면 버킷을 마운트하고 'chmod' 셸 명령을 실행하여 권한을 변경하는 해결 방법이 필요합니다.

Q: FSxN 전용 S3 버킷을 다른 SageMaker ML 서비스와 어떻게 통합할 수 있습니까?

A: 안타깝게도 SageMaker 서비스 SDK는 전용 S3 버킷의 끝점을 지정하는 방법을 제공하지 않습니다. 따라서 FSxN S3는 Sagemaker Data Wrangler, Sagemaker Clarify, Sagemaker Glue, Sagemaker Athena, Sagemaker AutoML 등의 SageMaker 서비스와 호환되지 않습니다. 있습니다.

2부 - AWS FSx for NetApp ONTAP(FSxN)를 SageMaker에서 모델 훈련을 위한 데이터 소스로 활용

- 작성자: *
Jian Jian(Ken), NetApp 수석 데이터 및 응용 과학자

소개

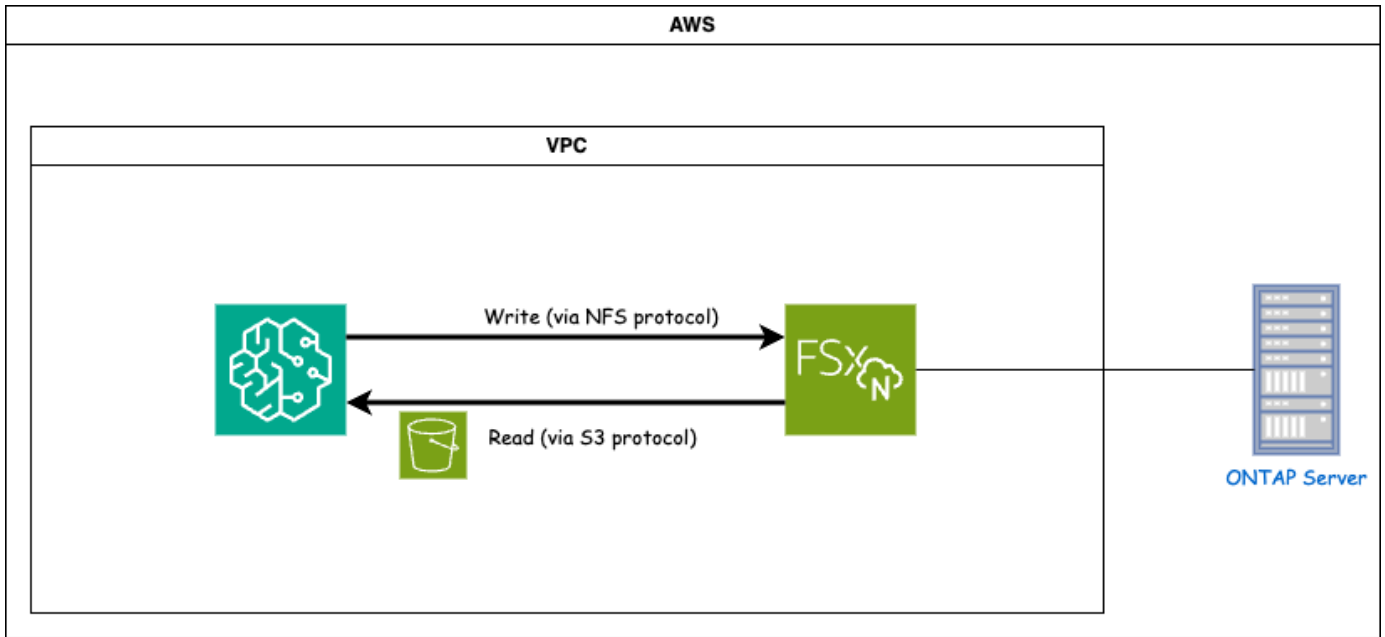
이 자습서는 SageMaker 환경 내에서 FSxN을 데이터 소스로 사용하는 ML 모델을 직접 구축하는 경험을 제공하는 컴퓨터 비전 분류 프로젝트의 실제 예를 제공합니다. 이 프로젝트는 딥 러닝 프레임워크인 PyTorch를 사용하여 타이어 이미지를 기준으로 타이어 품질을 분류하는 데 초점을 맞추고 있습니다. FSxN을 Amazon SageMaker의 데이터 원본으로 사용하는 머신 러닝 모델의 개발을 강조합니다.

FSxN이란

Amazon FSx for NetApp ONTAP은 AWS에서 제공하는 완전 관리형 스토리지 솔루션입니다. NetApp의 ONTAP 파일 시스템을 활용하여 안정적인 고성능 스토리지를 제공합니다. NFS, SMB, iSCSI와 같은 프로토콜을 지원하여 다양한 컴퓨팅 인스턴스 및 컨테이너에서 원활하게 액세스할 수 있습니다. 이 서비스는 뛰어난 성능을 제공하여 빠르고 효율적인 데이터 운영을 보장하도록 설계되었습니다. 또한 높은 가용성과 내구성을 제공하여 데이터에 계속 액세스하고 보호할 수 있습니다. 또한 Amazon FSx for NetApp ONTAP의 스토리지 용량은 확장되므로 필요에 따라 쉽게 조정할 수 있습니다.

필수 구성 요소

네트워크 환경



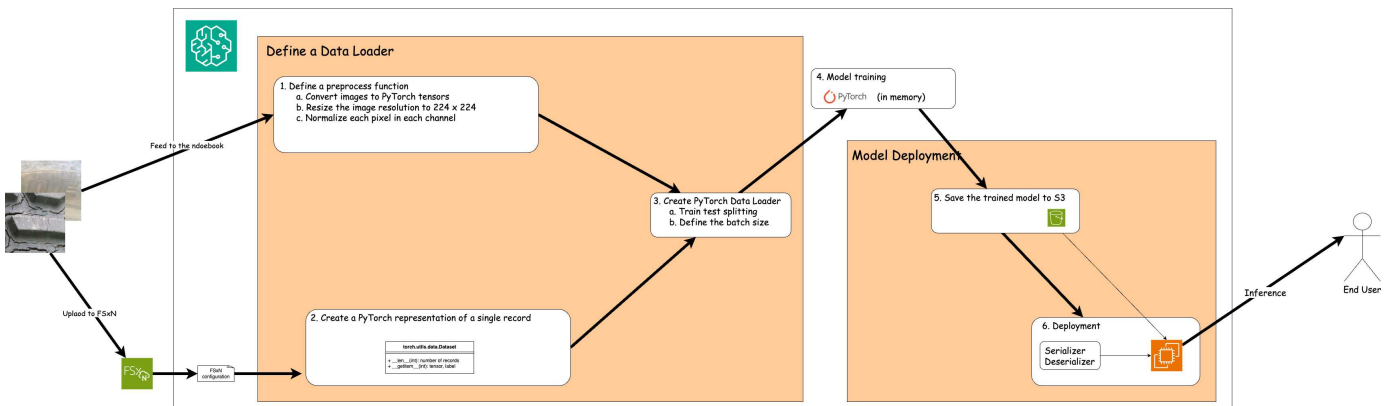
FSxN(Amazon FSx for NetApp ONTAP)은 AWS 스토리지 서비스입니다. NetApp ONTAP 시스템에서 실행 중인 파일 시스템과 이 시스템에 연결하는 AWS 관리형 SVM(시스템 가상 머신)이 포함됩니다. 제공된 다이어그램에서 AWS가 관리하는 NetApp ONTAP 서버는 VPC 외부에 있습니다. SVM은 SageMaker와 NetApp ONTAP 시스템 간의 매개 변수 역할을 하여 SageMaker로부터 작업 요청을 받아 기본 스토리지로 전달합니다. FSxN에 액세스하려면 SageMaker를 FSxN 구축과 동일한 VPC 내에 배치해야 합니다. 이 구성은 SageMaker와 FSxN 간의 통신 및 데이터 액세스를 보장합니다.

데이터 액세스

실제 시나리오에서 데이터 과학자는 일반적으로 FSxN에 저장된 기존 데이터를 활용하여 머신 러닝 모델을 구축합니다. 그러나 데모용으로 FSxN 파일 시스템은 생성 후 처음 비어 있으므로 교육 데이터를 수동으로 업로드해야 합니다. FSxN을 SageMaker에 볼륨으로 마운트하면 이 작업을 수행할 수 있습니다. 파일 시스템이 성공적으로 마운트되면 데이터 세트를 마운트된 위치에 업로드하여 SageMaker 환경 내에서 모델을 교육하기 위해 액세스할 수 있습니다. 이러한 접근 방식을 통해 FSxN의 스토리지 용량과 기능을 활용하는 동시에 SageMaker와 협력하여 모델 개발 및 교육을 수행할 수 있습니다.

데이터 읽기 프로세스에는 FSxN을 전용 S3 버킷으로 구성하는 작업이 포함됩니다. 자세한 구성 지침은 [참조하십시오 "1부 - AWS FSx for NetApp ONTAP\(FSxN\)를 프라이빗 S3 버킷으로 AWS SageMaker에 통합"](#)

통합 개요



FSxN의 교육 데이터를 사용하여 SageMaker에서 딥 러닝 모델을 구축하는 워크플로는 데이터 로더 정의, 모델 교육 및

배포의 세 가지 주요 단계로 요약할 수 있습니다. 개략적으로 보면 이러한 단계는 MLOps 파이프라인의 토대가 됩니다. 그러나 각 단계에는 포괄적인 구현을 위한 몇 가지 세부 하위 단계가 포함됩니다. 이러한 하위 단계에는 데이터 전처리, 데이터 세트 분할, 모델 구성, 하이퍼파라미터 조정, 모델 평가, 모델 구축을 지원합니다. 이러한 단계를 통해 SageMaker 환경 내에서 FSxN의 교육 데이터를 사용하여 딥 러닝 모델을 구축 및 배포할 수 있는 완벽하고 효과적인 프로세스를 보장할 수 있습니다.

단계별 통합

데이터 로더

데이터를 사용하여 PyTorch 딥 러닝 네트워크를 훈련하기 위해 데이터 피드를 용이하게 하기 위한 데이터 로더가 생성됩니다. 데이터 로더는 배치 크기를 정의할 뿐만 아니라 배치 내에서 각 레코드를 읽고 사전 처리하는 절차를 결정합니다. 데이터 로더를 구성하면 데이터 처리를 일괄 처리할 수 있어 딥 러닝 네트워크에 대한 교육이 가능합니다.

데이터 로더는 세 부분으로 구성됩니다.

전처리 기능

```
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]
    )
])
```

위의 코드 조각은 * torchvision.transforms * 모듈을 사용한 이미지 전처리 변환의 정의를 보여 줍니다. 이 튜토리얼에서는 일련의 변환을 적용하기 위해 전처리 개체가 만들어집니다. 첫째, * ToTensor() * 변환은 이미지를 텐서 표현으로 변환합니다. 그 후 * 크기 조정((224,224) * 변환은 고정 크기 224x224 픽셀로 이미지 크기를 조정합니다. 마지막으로 * 정규화() * 변환은 평균을 빼고 각 채널을 따라 표준 편차로 나누어 텐서 값을 정규화합니다. 정규화에 사용되는 평균 및 표준 편차 값은 일반적으로 사전 훈련된 신경망 모델에 사용됩니다. 전반적으로 이 코드는 이미지 데이터를 텐서로 변환하고 크기를 조정하며 픽셀 값을 정규화함으로써 사전 훈련된 모델로 추가 처리 또는 입력을 위해 준비합니다.

PyTorch 데이터 집합 클래스

```

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

```

이 클래스는 데이터 집합의 총 레코드 수를 가져오는 기능을 제공하며 각 레코드에 대한 데이터를 읽는 방법을 정의합니다. `__getitem__` 함수 내에서 코드는 boto3 S3 Bucket 객체를 사용하여 FSxN에서 이진 데이터를 검색합니다. FSxN에서 데이터에 액세스하기 위한 코드 스타일은 Amazon S3에서 데이터를 읽는 것과 비슷합니다. 다음 설명은 전용 S3 객체 * Bucket *의 생성 프로세스에 대해 자세히 설명합니다.

FSxN은 프라이빗 **S3** 저장소로 사용됩니다

```

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 ---

```

SageMaker에서 FSxN의 데이터를 읽으려면 S3 프로토콜을 사용하여 FSxN 스토리지를 가리키는 처리기가 만들어집니다. 따라서 FSxN을 전용 S3 버킷으로 처리할 수 있습니다. 핸들러 구성에는 FSxN SVM의 IP 주소, 버킷 이름 및 필요한 자격 증명을 지정하는 작업이 포함됩니다. 이러한 구성 항목을 얻는 방법에 대한 자세한 설명은 의 문서를 참조하십시오 ["1부 - AWS FSx for NetApp ONTAP\(FSxN\)를 프라이빗 S3 버킷으로 AWS SageMaker에 통합"](#).

위에서 언급한 예제에서 버킷 객체는 PyTorch 데이터 집합 객체를 인스턴스화하는 데 사용됩니다. 데이터세트 객체에 대해서는 다음 섹션에서 자세히 설명합니다.

PyTorch 데이터 로더

```
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)
```

제공된 예에서는 배치 크기가 64로 지정되어 각 배치에 64개의 레코드가 포함됨을 나타냅니다. PyTorch * Dataset * 클래스, 전처리 기능 및 훈련 배치 크기를 결합하여 훈련을 위한 데이터 로더를 얻습니다. 이 데이터 로더는 교육 단계에서 데이터 세트를 일괄적으로 반복하는 프로세스를 지원합니다.

모델 교육

```
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')

```

이 코드는 표준 PyTorch 교육 프로세스를 구현합니다. 타이어의 품질을 분류하기 위해 선형과 선층을 사용하여 * TyreQualityClassifier * 라는 신경망 모델을 정의합니다. 훈련 루프는 데이터 배치를 반복하고 손실을 계산하며 역전파와 최적화를 사용하여 모델의 파라미터를 업데이트합니다. 또한 모니터링을 위해 현재 시간, 에포치, 배치 및 손실을 인쇄합니다.

모델 구축

구축

```

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())

```

SageMaker는 배포를 위해 모델을 S3에 저장해야 하기 때문에 코드는 PyTorch 모델을 * Amazon S3 * 에 저장합니다. 모델을 * Amazon S3 * 에 업로드하면 SageMaker에 액세스할 수 있으므로 배포된 모델에 대한 구축 및 추론이 가능합니다.

```

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
)

```

이 코드를 사용하면 SageMaker에서 PyTorch 모델을 쉽게 배포할 수 있습니다. 또한 입력 데이터를 PyTorch 텐서로 미리 처리하고 serialize하는 사용자 지정 serializer * TyreQualitySerializer * 를 정의합니다. TyreQualityPredictor* 클래스는 정의된 serializer와 * JSONDeserializer*를 사용하는 사용자 지정 예측자입니다. 또한 이 코드는 * PyTorchModel * 개체를 만들어 모델의 S3 위치, IAM 역할, 프레임워크 버전 및 추론을 위한 진입점을 지정합니다. 이 코드에서는 타임스탬프를 생성하고 모델 및 타임스탬프를 기반으로 끝점 이름을 생성합니다. 마지막으로, 모델은

Deploy 메서드를 사용하여 배포되며 인스턴스 수, 인스턴스 유형 및 생성된 끝점 이름을 지정합니다. 이를 통해 PyTorch 모델을 구축하고 SageMaker에서 추론을 위해 액세스할 수 있습니다.

추론

```
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)
```

다음은 배포된 끝점을 사용하여 추론을 수행하는 예제입니다.

3부 - 간소화된 MLOps 파이프라인 구축(CI/CT/CD)

- 작성자: *
Jian Jian(Ken), NetApp 수석 데이터 및 응용 과학자

소개

이 자습서에서는 다양한 AWS 서비스를 활용하여 지속적인 통합(CI), 지속적인 교육(CT) 및 지속적인 배포(CD)를 포함한 간단한 MLOps 파이프라인을 구축하는 방법을 알아봅니다. 기존의 DevOps 파이프라인과 달리 MLOps는 운영 주기를 완료하는 데 추가 사항을 고려해야 합니다. 이 자습서를 따르면 CT를 MLOps 루프에 통합하는 방법에 대한 통찰력을 얻고 모델을 지속적으로 훈련하고 추론을 위한 원활한 배포를 수행할 수 있습니다. 이 자습서는 AWS 서비스를 활용하여 이 엔드 투 엔드 MLOps 파이프라인을 구축하는 프로세스를 안내합니다.

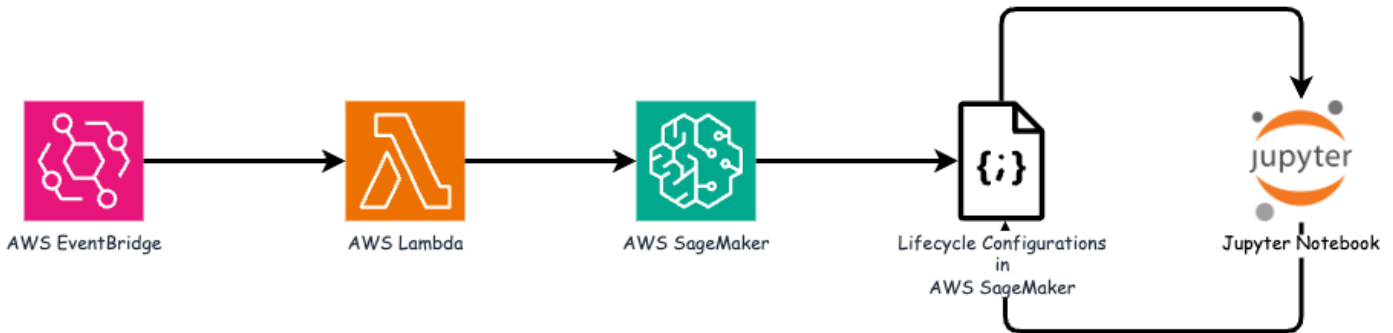
매니페스트

기능	이름	설명
데이터 스토리지	AWS FSxN	을 참조하십시오 "1부 - AWS FSx for NetApp ONTAP(FSxN)를 프라이빗 S3 버킷으로 AWS SageMaker에 통합" .
데이터 과학 IDE	AWS SageMaker를 참조하십시오	이 자습서는 에서 제공하는 Jupyter 노트북을 기반으로 합니다 "2부 - AWS FSx for NetApp ONTAP(FSxN)를 SageMaker에서 모델 훈련을 위한 데이터 소스로 활용" .
MLOps 파이프라인을 트리거하는 기능입니다	AWS Lambda 함수	-
cron 작업 트리거	AWS 이벤트 브리지	-
딥 러닝 프레임워크	PyTorch	-
AWS Python SDK에서 지원됩니다	봇3	-
프로그래밍 언어	파이썬	v3.10

필수 구성 요소

- 사전 구성된 FSxN 파일 시스템입니다. 이 자습서에서는 교육 프로세스에 FSxN에 저장된 데이터를 활용합니다.
- 위에서 언급한 FSxN 파일 시스템과 동일한 VPC를 공유하도록 구성된 * SageMaker 노트북 인스턴스 *.
- AWS Lambda 함수 * 를 트리거하기 전에 * SageMaker Notebook Instance * 가 * Stopped * 상태인지 확인하십시오.
- 딥 신경 네트워크의 계산에 필요한 GPU 가속을 활용하려면 * ml.g4dn.xlarge * 인스턴스 유형이 필요합니다.

있습니다



이 MLOps 파이프라인은 cron 작업을 사용하여 서버리스 기능을 트리거하는 실용적인 구현으로, 수명주기 콜백 기능으로 등록된 AWS 서비스를 실행합니다. AWS EventBridge * 는 cron 작업 역할을 합니다. 주기적으로 모델을 재교육하고 재배포할 책임이 있는 * AWS Lambda 함수 * 를 호출합니다. 이 프로세스에서는 * AWS SageMaker Notebook * 인스턴스를 회전하여 필요한 작업을 수행합니다.

단계별 구성

문서 수정 상태 설정

AWS SageMaker Notebook 인스턴스에 대한 Lifecycle 콜백 기능을 구성하려면 * Lifecycle configuration * 을 사용합니다. 이 서비스를 사용하면 노트북 인스턴스를 회전할 때 수행해야 하는 필요한 작업을 정의할 수 있습니다. 특히 교육 및 배포 프로세스가 완료되면 노트북 인스턴스를 자동으로 종료하도록 * Lifecycle 구성 * 내에서 셸 스크립트를 구현할 수 있습니다. 비용은 MLOps의 주요 고려 사항 중 하나이므로 이 구성은 필수 구성입니다.

반드시 * Lifecycle 설정 * 에 대한 구성을 미리 설정해야 합니다. 따라서 다른 MLOps 파이프라인 설정을 진행하기 전에 이 구성 우선 순위를 지정하는 것이 좋습니다.

1. Lifecycle 구성을 설정하려면 * Sagemaker * 패널을 열고 * Admin configuration * 섹션의 * Lifecycle configuration * 으로 이동합니다.

aws

Services

Search

S3

Amazon SageMaker

×

Getting started

Studio

Studio Lab

Canvas

RStudio

TensorBoard

Profiler

▼ Admin configurations

Domains

Role manager

Images

Lifecycle configurations

SageMaker dashboard

Search

► JumpStart

Amazon SageMaker > Domains

Domains

Info

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

► Domain structure diagram

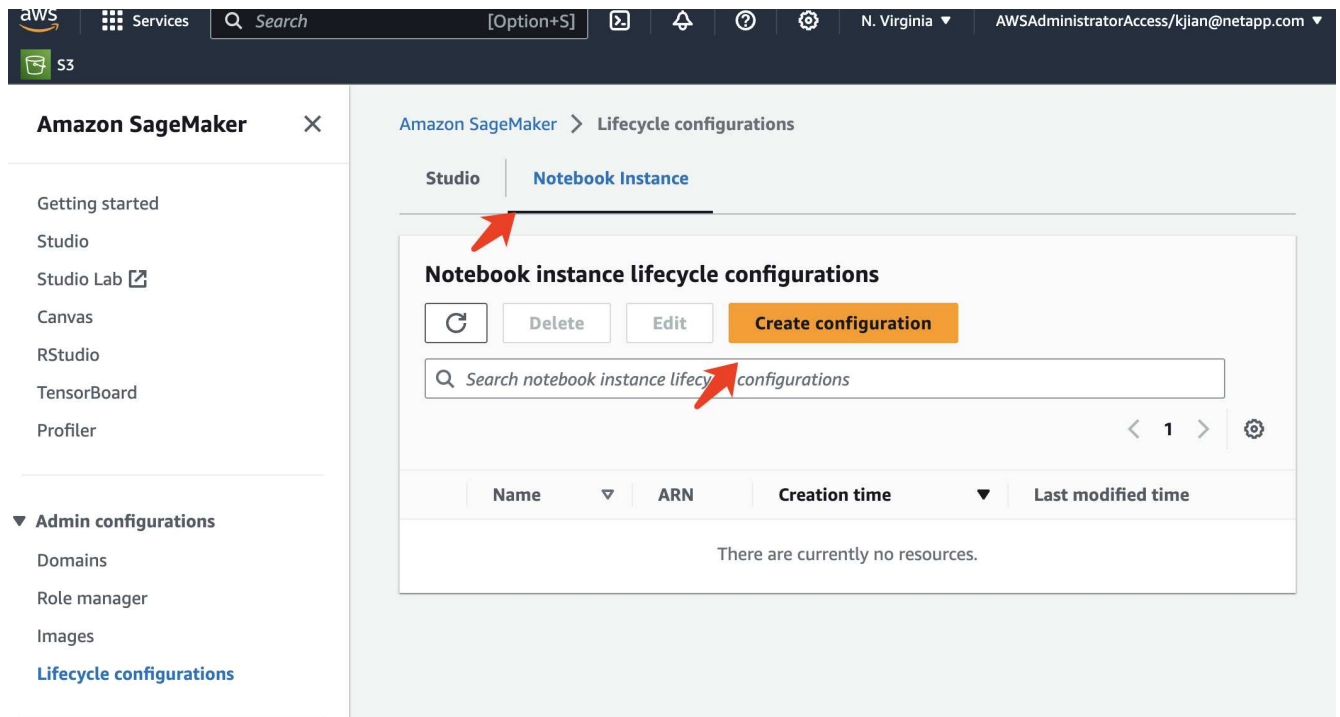
Domains (4)

Info

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. Notebook Instance * 탭을 선택하고 * Create configuration * 버튼을 클릭합니다



3. 아래 코드를 입력 영역에 붙여 넣습니다.

```
#!/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. 이 스크립트는 추론을 위해 모델의 재훈련 및 재배포를 처리하는 Jupyter Notebook을 실행합니다. 실행이 완료되면 노트북이 5분 내에 자동으로 종료됩니다. 문제 설명 및 코드 구현에 대한 자세한 내용은 을 참조하십시오 ["2부 - AWS FSx for NetApp ONTAP\(FSxN\)를 SageMaker에서 모델 훈련을 위한 데이터 소스로 활용"](#).

aws Services Search [Option+S]

S3

Amazon SageMaker > Lifecycle configurations > Create lifecycle configuration

Create lifecycle configuration

Configuration setting

Name

fsxn-demo-lifecycle-callback

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

Scripts

Start notebook Create notebook

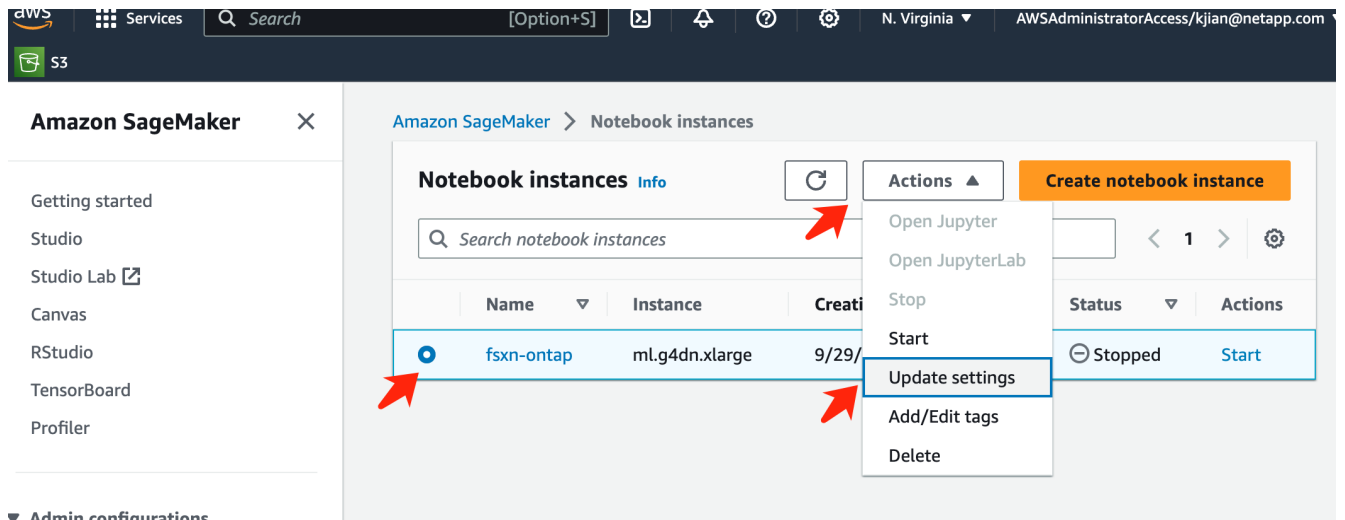
This script will be run each time an associated notebook instance is started, including during initial creation. If the associated notebook instance is already started, it will be run the next time it is stopped and started. [a curated list of sample scripts](#)

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

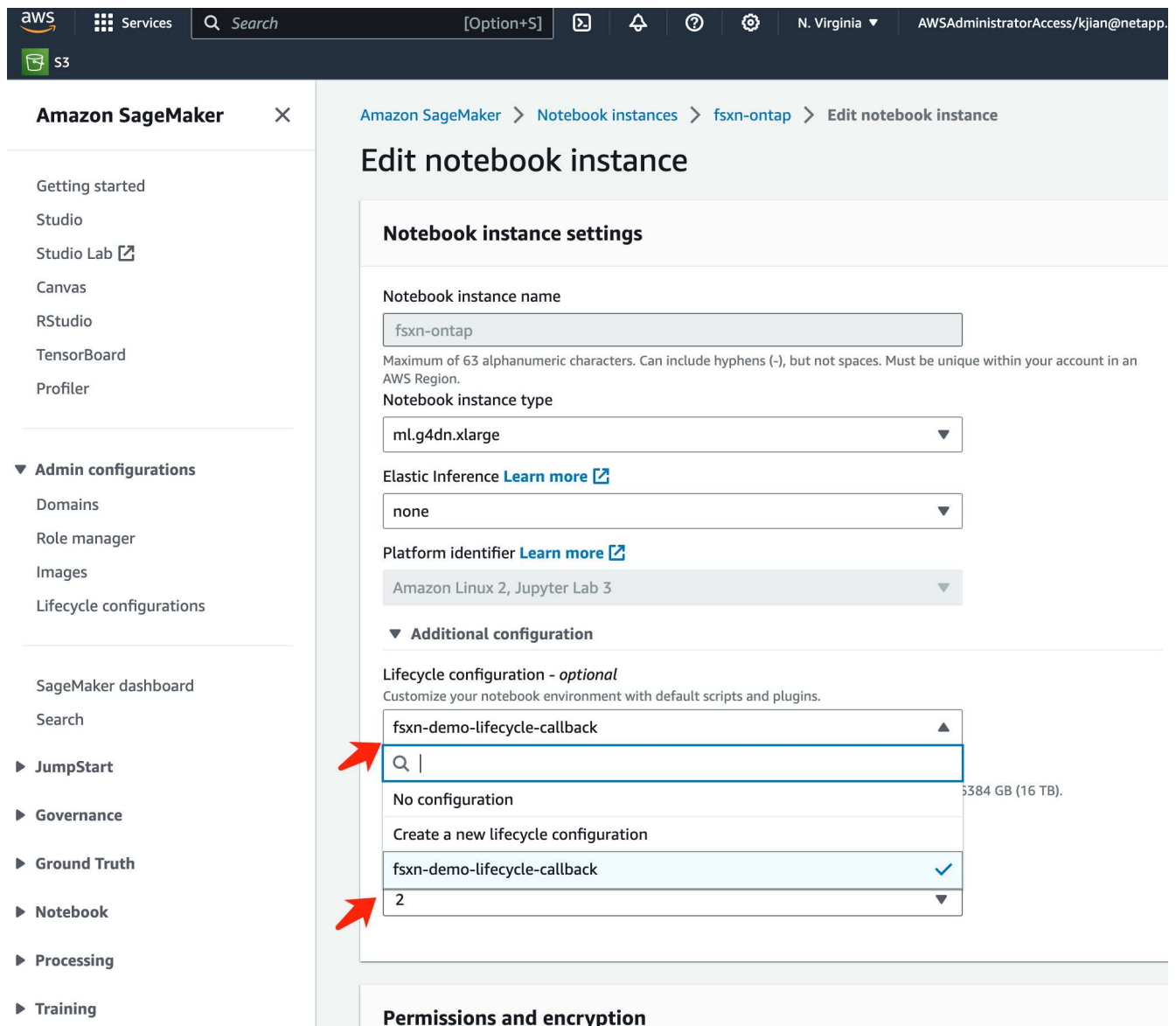
Cancel Create configuration

CloudShell Feedback

5. 만든 후 노트북 인스턴스로 이동하여 대상 인스턴스를 선택하고 작업 드롭다운 아래의 * 설정 업데이트 * 를 클릭합니다.



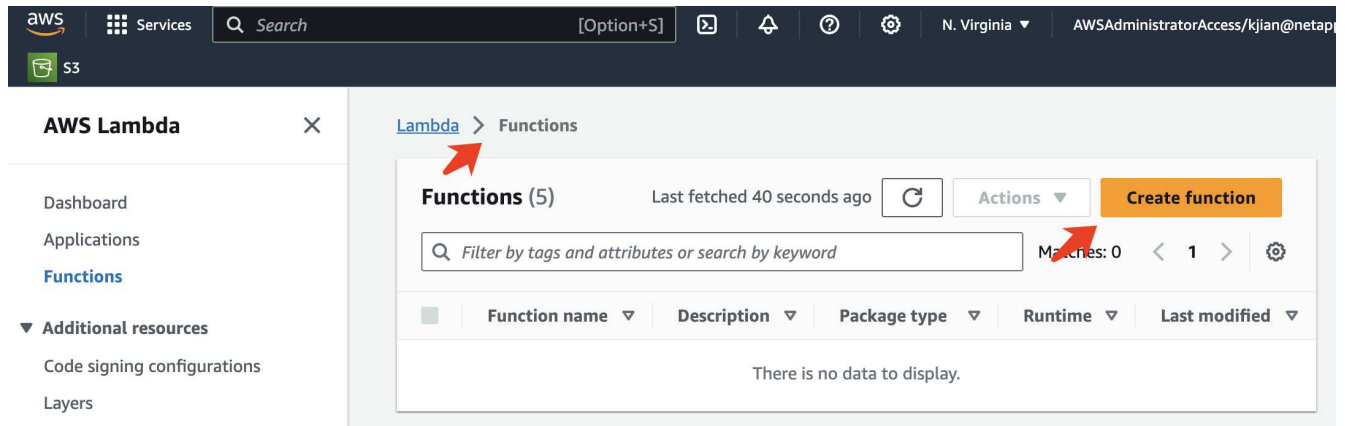
6. 생성된 * 문서 수정 상태 구성 * 을 선택하고 * 노트북 인스턴스 업데이트 * 를 클릭합니다.



AWS Lambda 서버리스 함수입니다

앞서 언급한 것처럼 * AWS Lambda 함수 * 는 * AWS SageMaker Notebook 인스턴스 * 를 회전하는 역할을 합니다.

1. AWS Lambda 함수 * 를 만들려면 해당 패널로 이동하여 * Functions * 탭으로 이동한 다음 * Create Function * 을 클릭합니다.



2. 페이지에 필요한 모든 항목을 기록하고 Runtime을 * Python 3.10 * 으로 전환해야 합니다.

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. 지정된 역할에 필요한 권한 * AmazonSageMakerFullAccess * 가 있는지 확인하고 * 기능 생성 * 버튼을 클릭하십시오.

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**

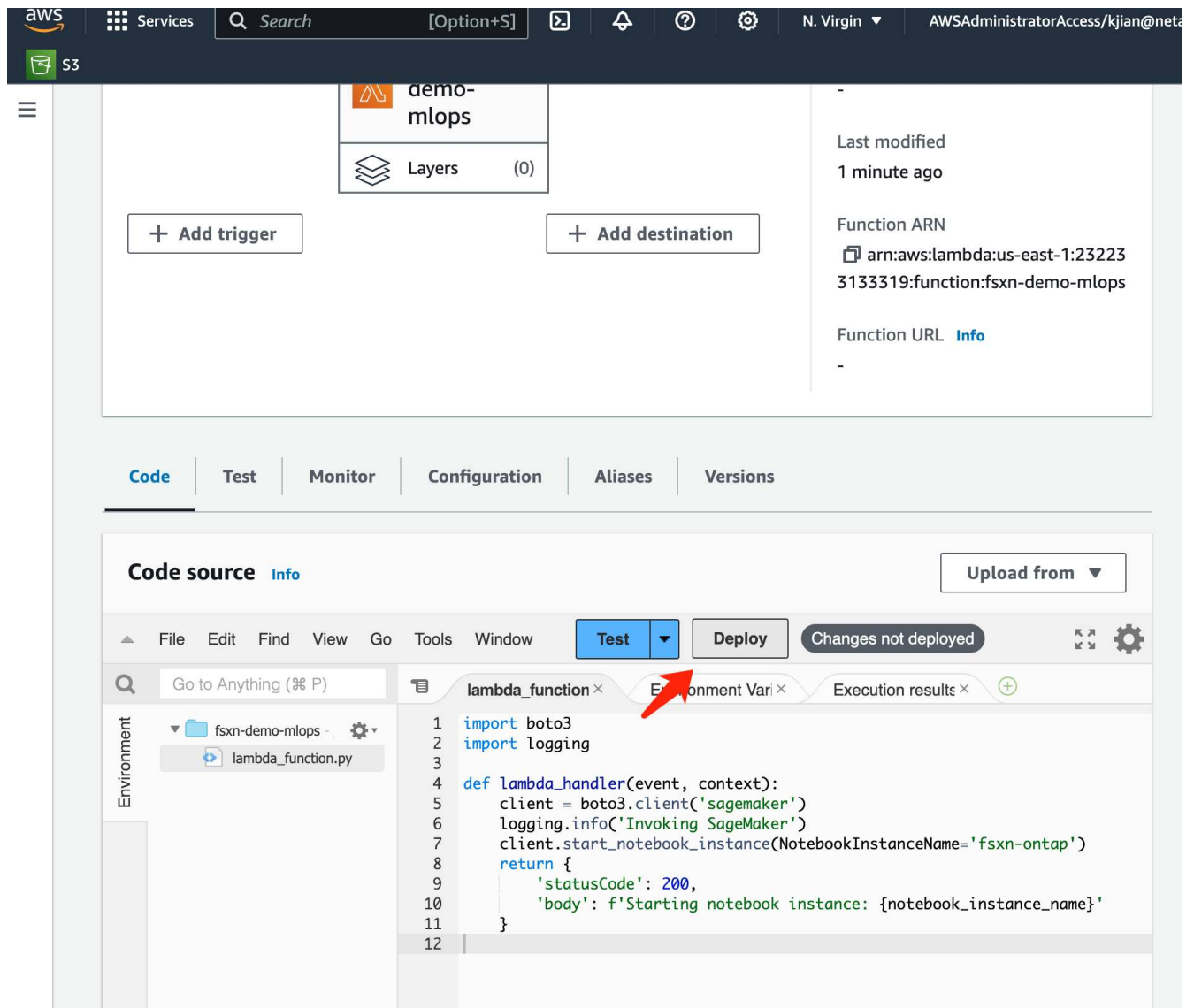
Cancel Create function

4. 생성된 Lambda 함수를 선택합니다. 코드 탭에서 다음 코드를 복사하여 텍스트 영역에 붙여 넣습니다. 이 코드는 * fsxn-ONTAP * 이라는 노트북 인스턴스를 시작합니다.

```
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. 이 코드 변경 사항을 적용하려면 * 배포 * 버튼을 클릭하십시오.



6. 이 AWS Lambda 함수를 트리거하는 방법을 지정하려면 Add Trigger(트리거 추가) 단추를 클릭합니다.

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


S3


Lambda > Functions > fsxn-demo-mlops

fsxn-demo-mlops

Throttle Copy ARN Actions

▼ Function overview Info

 fsxn-demo-mlops

 Layers (0)

+ Add trigger + Add destination

Description -

Last modified 2 minutes ago

Function ARN
arn:aws:lambda:us-east-1:232233133319:function:fsxn-demo-mlops

Function URL Info

7. 드롭다운 메뉴에서 EventBridge 를 선택한 다음 새 규칙 만들기 라디오 단추를 클릭합니다. 스케줄 표현식 필드에 을 입력합니다 `rate(1 day)`를 클릭하고 추가 버튼을 클릭하여 이 새 cron 작업 규칙을 생성하고 AWS Lambda 함수에 적용합니다.

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

2단계 구성을 완료하면 * AWS Lambda 함수 * 가 매일 * SageMaker 노트북 * 을 시작하고, * FSxN * 리포지토리의 데이터를 사용하여 모델 재학습을 수행하고, 업데이트된 모델을 프로덕션 환경으로 재배포한 후 * SageMaker 노트북 인스턴스 * 를 자동으로 종료하여 비용을 최적화합니다. 이렇게 하면 모델이 최신 상태로 유지됩니다.

이것으로 MLOps 파이프라인 개발을 위한 자습서를 마칩니다.

Domino Data Lab 및 NetApp의 하이브리드 멀티 클라우드 MLOps

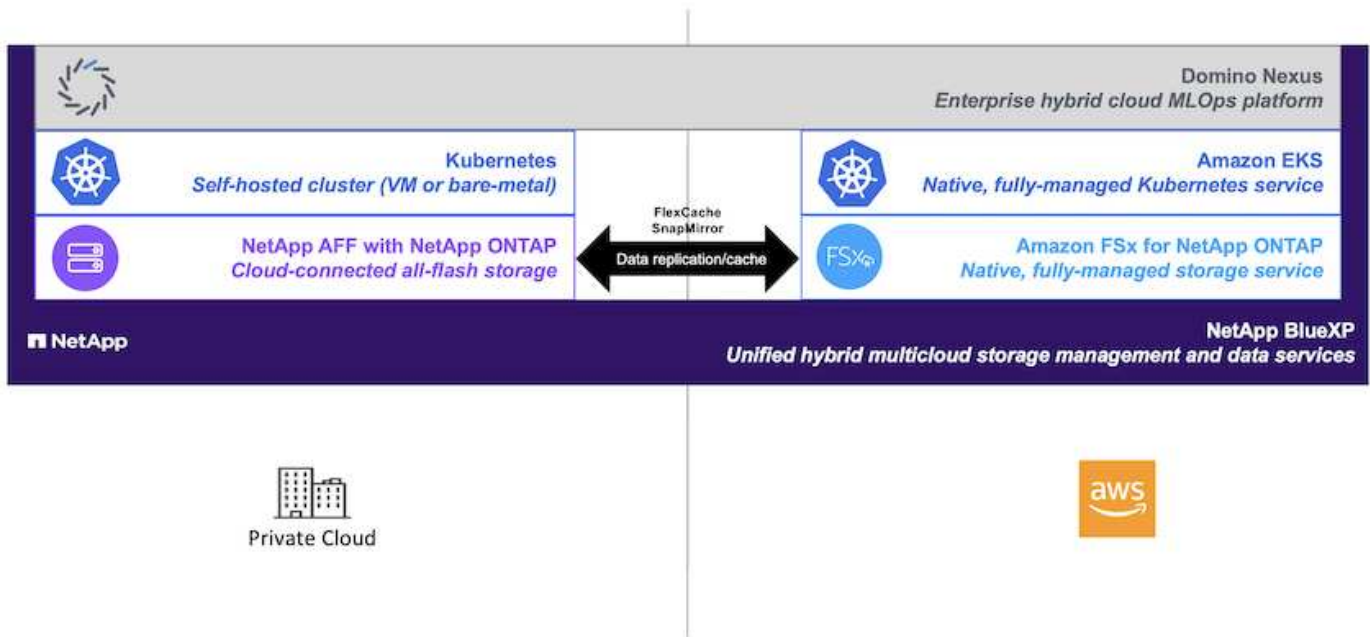
Domino Data Lab 및 NetApp의 하이브리드 멀티 클라우드 MLOps

Mike Oglesby, NetApp에서 직접 지원합니다

현재 전 세계 모든 조직이 AI를 채택하여 비즈니스와 프로세스를 혁신하고 있습니다. 따라서 AI 지원 컴퓨팅 인프라는 부족한 경우가 많습니다. 기업들은 다양한 지역, 데이터 센터 및 클라우드 간에 비용, 가용성 및 성능을 균형 있게 활용하기 위해 하이브리드 멀티 클라우드 MLOps 아키텍처를 채택하고 있습니다.

Domino Data Lab의 Domino Nexus는 통합 MLOps 제어 플레인으로서, 클라우드, 지역 또는 온프레미스의 모든 컴퓨팅 클러스터 전반에서 데이터 과학 및 머신 러닝 워크로드를 실행할 수 있습니다. 기업 전체의 데이터 과학 사일로를 통합하므로 모델을 한 곳에서 구축, 배포 및 모니터링할 수 있습니다. 마찬가지로, NetApp의 하이브리드 클라우드 데이터 관리 기능은 데이터를 실행 중인 위치와 상관없이 작업 공간과 작업 공간에 데이터를 가져올 수 있도록 지원합니다. Domino Nexus와 NetApp을 페어링하면 데이터 가용성에 대해 걱정할 필요 없이 여러 환경에서 워크로드를 유연하게 예약할 수 있습니다. 다시 말해, 워크로드와 데이터를 적절한 컴퓨팅 환경으로 전송함으로써 데이터 개인 정보 보호 및 주권에 관한 규정을 준수하면서 AI 구현을 앞당길 수 있습니다.

이 솔루션은 온프레미스 Kubernetes 클러스터와 AWS(Amazon Web Services)에서 실행되는 EKS(Elastic Kubernetes Service) 클러스터를 통합하는 통합 MLOps 컨트롤 플레인을 배포하는 것을 보여줍니다.



기술 개요

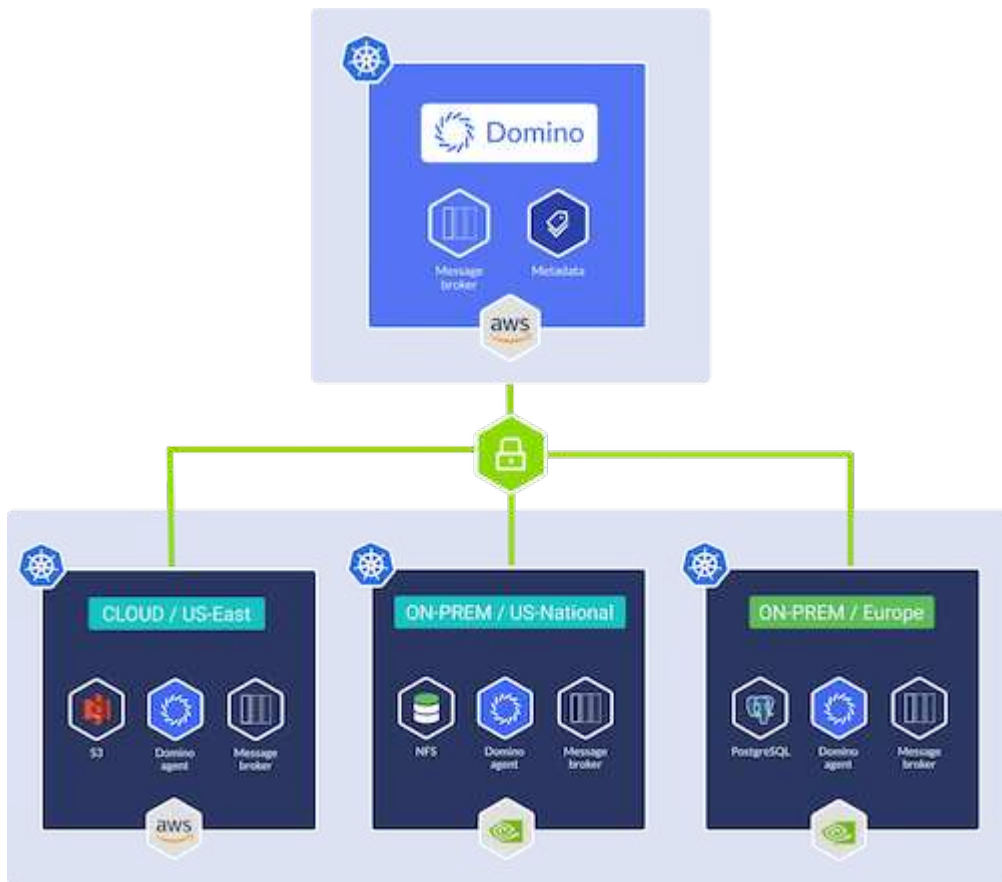
Domino 데이터 랩

Domino Data Lab은 Fortune 100대 기업의 20% 이상이 신뢰하는 최고의 엔터프라이즈 AI 플랫폼을 통해 모델 중심의 비즈니스를 지원합니다. Domino는 데이터 과학 작업의 개발 및 구축을 가속화하는 동시에 협업과 거버넌스를 강화합니다. Domino를 사용하면 전 세계 기업들이 더 나은 약품을 개발하고, 더 생산적인 작물을 재배하고, 더 나은 자동차를 만드는 등 다양한 작업을 수행할 수 있습니다. 2013년에 설립된 Domino는 Coatue Management, Great Hill

Partners, Highland Capital, Sequoia Capital 및 기타 주요 투자자들의 지원을 받고 있습니다.

Domino를 통해 기업과 데이터 과학자는 통합된 엔드 투 엔드 플랫폼에서 AI를 빠르고 책임적이며 비용 효율적으로 구축, 배포 및 관리할 수 있습니다. IT 팀은 모든 환경에서 필요한 모든 데이터, 툴, 컴퓨팅, 모델 및 프로젝트에 액세스할 수 있으므로 협업, 이전 작업을 재사용하고 운영 중인 모델을 추적하여 정확성을 높이고, 모범 사례에 따라 표준화하고, AI를 책임 및 관리할 수 있습니다.

- * 개방적이고 유연한 : * 오픈 소스 및 상용 툴, 인프라 등 가장 광범위한 에코시스템에 액세스하여 최고의 혁신을 실현하고 공급업체에 종속되지 않습니다.
- * 기록 시스템: * 기업 전체의 AI 운영 및 지식을 위한 중앙 허브로 모범 사례, 다기능 협업, 빠른 혁신 및 효율성을 지원합니다.
- * 통합: * 엔터프라이즈 프로세스, 제어 및 거버넌스를 위해 설계된 통합 워크플로우와 자동화는 규정 준수 및 규제 요건을 충족합니다.
- * 하이브리드 멀티 클라우드: * 온프레미스, 하이브리드, 모든 클라우드 또는 멀티 클라우드 등 어디서나 데이터에 가까운 위치에서 AI 워크로드를 실행하여 비용 절감, 최적의 성능 및 규정 준수를 지원할 수 있습니다.



도미노 넥서스

Domino Nexus는 모든 클라우드, 지역 또는 온프레미스의 컴퓨팅 클러스터 전반에서 데이터 과학 및 머신 러닝 워크로드를 실행할 수 있는 단일 창입니다. 기업 전체의 데이터 과학 사일로를 통합하므로 모델을 한 곳에서 구축, 배포 및 모니터링할 수 있습니다.

NetApp BlueXP

NetApp BlueXP는 NetApp의 모든 스토리지 및 데이터 서비스를 하이브리드 멀티 클라우드 데이터 자산을 구축, 보호 및 관리할 수 있는 단일 툴에 통합합니다. 또한 온프레미스 및 클라우드 환경에서 스토리지 및 데이터 서비스를 위한

통합된 경험을 제공하고 AI/ML의 기능을 통해 운영을 간소화하며 오늘날의 클라우드 주도 환경에 필요한 유연한 소비 매개 변수와 통합 보호를 제공합니다.

NetApp ONTAP를 참조하십시오

NetApp의 최신 세대 스토리지 관리 소프트웨어인 ONTAP 9는 기업이 인프라를 현대화하고 클라우드 지원 데이터 센터로 전환할 수 있도록 지원합니다. ONTAP는 업계 최고 수준의 데이터 관리 기능을 활용하여 데이터가 상주하는 위치와 상관없이 단일 톨셋으로 데이터를 관리하고 보호할 수 있습니다. 필요에 따라 에지, 코어, 클라우드 등 어느 위치로도 데이터를 자유롭게 이동할 수 있습니다. ONTAP 9에는 데이터 관리를 단순화하고, 중요 데이터를 더 빨리 처리하고, 보호하며, 하이브리드 클라우드 아키텍처 전체에서 차세대 인프라 기능을 지원하는 다양한 기능이 포함되어 있습니다.

데이터 관리를 단순화하십시오

데이터 관리는 AI 애플리케이션에 적합한 리소스를 사용하고 AI/ML 데이터 세트를 교육할 수 있도록 엔터프라이즈 IT 운영 및 데이터 과학자에게 매우 중요합니다. NetApp 기술에 대한 다음 추가 정보는 이 검증의 범위에 포함되지 않지만, 배포에 따라 달라질 수 있습니다.

ONTAP 데이터 관리 소프트웨어에는 운영을 간소화 및 단순화하고 총 운영 비용을 절감하는 다음과 같은 기능이 있습니다.

- 인라인 데이터 컴팩션 및 확대된 중복제거: 데이터 컴팩션은 스토리지 블록 내부의 낭비되는 공간을 줄이고, 중복제거는 실제 용량을 상당히 늘려줍니다. 이는 로컬에 저장된 데이터와 클라우드로 계층화된 데이터에 적용됩니다.
- 최소, 최대 및 적응형 서비스 품질(AQoS): 세부적인 서비스 품질(QoS) 제어로 고도의 공유 환경에서 중요 애플리케이션의 성능 수준을 유지할 수 있습니다.
- NetApp FabricPool을 참조하십시오. AWS(Amazon Web Services), Azure, NetApp StorageGRID 스토리지 솔루션을 포함한 퍼블릭 클라우드 및 프라이빗 클라우드 스토리지에 콜드 데이터를 자동으로 계층화합니다. FabricPool에 대한 자세한 내용은 ["TR-4598: FabricPool 모범 사례"](#)를 참조하십시오.

데이터 가속화 및 보호

ONTAP는 탁월한 수준의 성능과 데이터 보호를 제공하며 다음과 같은 방법으로 이러한 기능을 확장합니다.

- 성능 및 짧은 지연 시간: ONTAP는 가장 짧은 지연 시간으로 가장 높은 처리량을 제공합니다.
- 데이터 보호: ONTAP는 모든 플랫폼에서 공통 관리를 지원하는 내장 데이터 보호 기능을 제공합니다.
- NVE(NetApp 볼륨 암호화). ONTAP는 온보드 및 외부 키 관리를 모두 지원하는 기본 볼륨 레벨 암호화를 제공합니다.
- 멀티테넌시 및 다단계 인증. ONTAP를 사용하면 인프라 리소스를 최고 수준의 보안으로 공유할 수 있습니다.

미래 지향형 인프라

ONTAP은 다음과 같은 기능을 통해 끊임없이 변화하는 까다로운 비즈니스 요구사항을 충족할 수 있도록 지원합니다.

- 원활한 확장 및 무중단 운영: ONTAP은 운영 중단 없이 기존 컨트롤러 및 스케일아웃 클러스터에 용량을 추가할 수 있도록 지원합니다. 고객은 고비용이 따르는 데이터 마이그레이션이나 운영 중단 없이 NVMe 및 32Gb FC와 같은 최신 기술로 업그레이드할 수 있습니다.
- 클라우드 연결: ONTAP은 주요 클라우드와 연결되는 스토리지 관리 소프트웨어로, 모든 퍼블릭 클라우드에서 소프트웨어 정의 스토리지 및 클라우드 네이티브 인스턴스 옵션이 제공됩니다.

- 새로운 애플리케이션과 통합: ONTAP은 기존 엔터프라이즈 앱을 지원하는 인프라와 동일한 인프라를 사용하여 자율주행 차량, 스마트 시티, Industry 4.0과 같은 차세대 플랫폼 및 애플리케이션을 위한 엔터프라이즈급 데이터 서비스를 제공합니다.

NetApp ONTAP용 Amazon FSx

Amazon FSx for NetApp ONTAP는 완전 관리형 AWS 서비스로서, NetApp의 인기 ONTAP 파일 시스템을 기반으로 구축되어 매우 안정적이고 확장 가능하며 성능이 우수하며 풍부한 파일 스토리지를 제공합니다. ONTAP용 FSx는 NetApp 파일 시스템의 친숙한 기능, 성능, 기능 및 API 작업을 완벽하게 관리되는 AWS 서비스의 민첩성, 확장성 및 간편성과 결합합니다.

NetApp Astra Trident

Astra Trident를 사용하면 ONTAP(AFF, NetApp FAS, Select, 클라우드, NetApp ONTAP용 Amazon FSx), Element 소프트웨어(NetApp HCI, SolidFire), Azure NetApp Files 서비스 및 Google Cloud 기반 Cloud Volumes Service Astra Trident는 Kubernetes와 기본적으로 통합되는 CSI(컨테이너 스토리지 인터페이스) 호환 동적 스토리지 오케스트레이터입니다.

쿠버네티스

Kubernetes는 Google에서 원래 설계한 개방형 소스, 분산형 컨테이너 오케스트레이션 플랫폼으로, 현재 CNCF(Cloud Native Computing Foundation)에서 관리하고 있습니다. Kubernetes는 컨테이너화된 애플리케이션의 구축, 관리 및 크기 조정 기능을 자동화할 수 있으며 엔터프라이즈 환경에서 지배적인 컨테이너 오케스트레이션 플랫폼입니다.

Amazon Elastic Kubernetes Service(EKS)

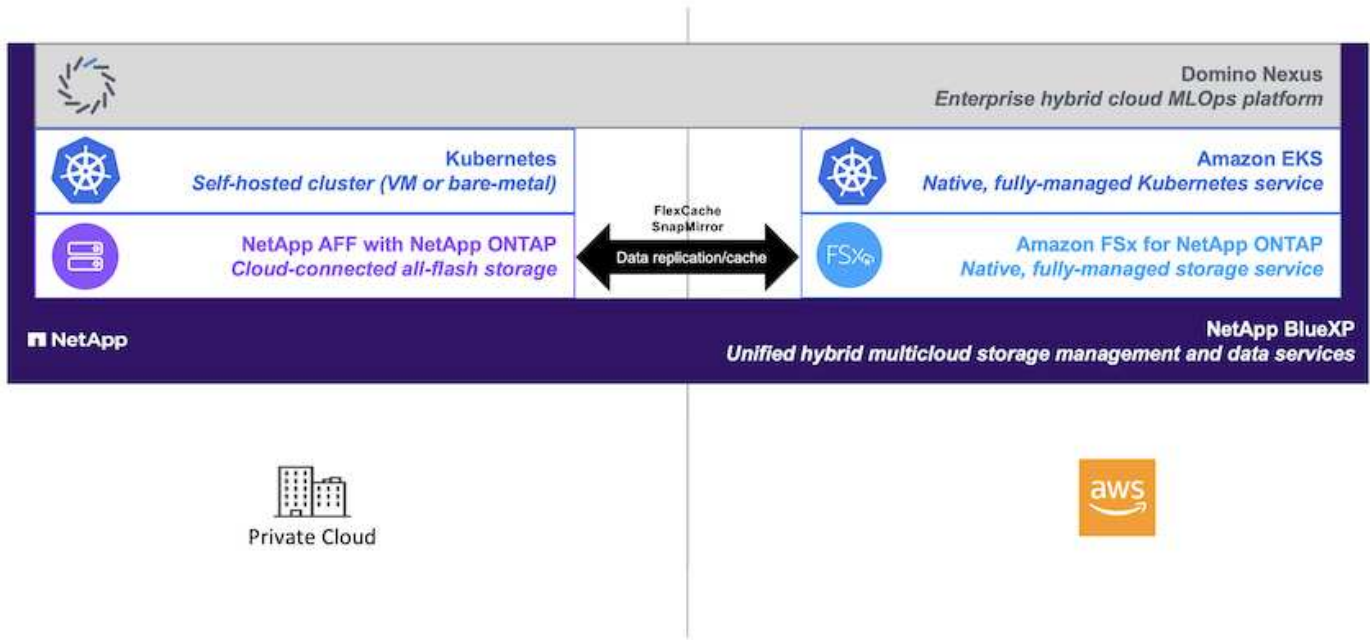
Amazon EKS(Amazon Elastic Kubernetes Service)는 AWS 클라우드의 관리형 Kubernetes 서비스입니다. Amazon EKS는 컨테이너 예약, 애플리케이션 가용성 관리, 클러스터 데이터 저장 및 기타 주요 작업을 담당하는 Kubernetes 컨트롤 플레인 노드의 가용성과 확장성을 자동으로 관리합니다. Amazon EKS를 사용하면 AWS 인프라의 모든 성능, 확장성, 안정성 및 가용성과 AWS 네트워킹 및 보안 서비스와의 통합을 활용할 수 있습니다.

있습니다

이 솔루션은 Domino Nexus의 하이브리드 멀티 클라우드 워크로드 스케줄링 기능을 NetApp 데이터 서비스와 결합하여 통합된 하이브리드 클라우드 MLOps 플랫폼을 구축합니다. 자세한 내용은 다음 표를 참조하십시오.

구성 요소	이름	방법입니다
MLOps 제어 평면	"Domino Nexus 탑재 Domino Enterprise AI 플랫폼"	설치하고
MLOps 플랫폼 컴퓨팅 환경	"Domino Nexus 데이터 플레인"	AWS, 온프레미스 데이터 센터에 있습니다
온프레미스 컴퓨팅 플랫폼	"쿠버네티스" 와 함께 "NetApp Astra Trident"	온프레미스 데이터 센터
클라우드 컴퓨팅 플랫폼	"Amazon Elastic Kubernetes Service(EKS)" 와 함께 "NetApp Astra Trident"	설치하고

구성 요소	이름	방법입니다
온프레미스 데이터 플랫폼	"NetApp 스토리지 어플라이언스"에서 제공 "NetApp ONTAP를 참조하십시오"	온프레미스 데이터 센터
클라우드 데이터 플랫폼	"NetApp ONTAP용 Amazon FSx"	설치하고



초기 설정

이 섹션에서는 사내 데이터 센터와 AWS가 통합된 하이브리드 환경에서 Domino Nexus와 NetApp 데이터 서비스를 활용하기 위해 수행해야 하는 초기 설정 작업에 대해 설명합니다.

필수 구성 요소

이 섹션에서 설명하는 단계를 수행하기 전에 다음 작업을 이미 수행한 것으로 가정합니다.

- 온프레미스 NetApp ONTAP 스토리지 플랫폼을 이미 구축하고 구성했습니다. 자세한 내용은 ["NetApp 제품 설명서"](#)를 참조하십시오.
- AWS에서 이미 Amazon FSx for NetApp ONTAP 인스턴스를 프로비저닝했습니다. 자세한 내용은 ["Amazon FSx for NetApp ONTAP 제품 페이지를 참조하십시오"](#)를 참조하십시오.
- 온프레미스 데이터 센터에서 이미 Kubernetes 클러스터를 프로비저닝했습니다. 자세한 내용은 ["Domino 관리 가이드"](#)를 참조하십시오.
- AWS에 이미 Amazon EKS 클러스터를 프로비저닝했습니다. 자세한 내용은 ["Domino 관리 가이드"](#)를 참조하십시오.
- 온프레미스 Kubernetes 클러스터에 NetApp Astra Trident를 설치했습니다. 또한 스토리지 리소스를 프로비저닝 및 관리할 때 온프레미스 NetApp ONTAP 스토리지 플랫폼을 사용하도록 이 Trident 인스턴스를 구성했습니다. 자세한 내용은 ["NetApp Astra Trident 문서"](#)를 참조하십시오.
- Amazon EKS 클러스터에 NetApp Astra Trident를 설치했습니다. 또한 스토리지 리소스를 프로비저닝하고 관리할 때 Amazon FSx for NetApp ONTAP 인스턴스를 사용하도록 이 Trident 인스턴스를 구성했습니다. 자세한 내용은 ["NetApp Astra Trident 문서"](#)를 참조하십시오.

를 참조하십시오 ["NetApp Astra Trident 문서"](#).

- 사내 데이터 센터와 AWS의 VPC(Virtual Private Cloud) 간에 양방향 네트워크 연결이 있어야 합니다. 이를 구현하는 다양한 옵션에 대한 자세한 내용은 를 참조하십시오 ["Amazon VPN\(가상 사설망\) 설명서"](#).

AWS에 Domino Enterprise AI 플랫폼을 설치합니다

AWS에 Domino Enterprise MLOps Platform을 설치하려면 에 설명된 지침을 따르십시오 ["Domino 관리 가이드"](#). 이전에 프로비저닝한 동일한 Amazon EKS 클러스터에 Domino를 구축해야 합니다. 또한 NetApp Astra Trident가 이 EKS 클러스터에 이미 설치되어 있고 구성되어 있어야 하며, domino.yml 설치 구성 파일에서 Trident로 관리하는 스토리지 클래스를 공유 스토리지 클래스로 지정해야 합니다.



을 참조하십시오 ["Domino 설치 구성 참조 가이드"](#) domino.yml 설치 구성 파일에서 공유 저장소 클래스를 지정하는 방법에 대한 자세한 내용은 을 참조하십시오.



["기술 보고서 TR-4952"](#) Amazon FSx for NetApp ONTAP를 사용하여 AWS에서 Domino를 배포하는 과정을 안내하며, 발생하는 문제를 해결하는 데 유용한 참고 자료가 될 수 있습니다.

Domino Nexus를 활성화합니다

다음으로 Domino Nexus를 활성화해야 합니다. 을 참조하십시오 ["Domino 관리 가이드"](#) 를 참조하십시오.

온프레미스 데이터 센터에 Domino Data Plane을 구축하십시오

그런 다음 사내 데이터 센터에 Domino Data Plane을 구축해야 합니다. 이전에 프로비저닝한 온프레미스 Kubernetes 클러스터에 이 데이터 플레인을 구축해야 합니다. 또한 NetApp Astra Trident가 이 Kubernetes 클러스터에 이미 설치되어 구성되어 있어야 합니다. 을 참조하십시오 ["Domino 관리 가이드"](#) 를 참조하십시오.

기존 NetApp 볼륨을 Domino에 표시합니다

이 섹션에서는 기존 NetApp ONTAP NFS 볼륨을 Domino MLOps 플랫폼에 노출하기 위해 수행해야 하는 작업에 대해 설명합니다. 이와 같은 단계는 온프레미스와 AWS 모두에 적용됩니다.

NetApp ONTAP 볼륨을 Domino에 노출하는 이유

NetApp 볼륨을 Domino와 함께 사용하면 다음과 같은 이점이 있습니다.

- NetApp ONTAP의 스케일아웃 기능을 활용하여 매우 큰 데이터 세트에서 워크로드를 실행할 수 있습니다.
- 개별 노드에 데이터를 복사하지 않고도 여러 컴퓨팅 노드에서 워크로드를 실행할 수 있습니다.
- NetApp의 하이브리드 멀티 클라우드 데이터 이동과 동기화 기능을 활용하여 여러 데이터 센터 및/또는 클라우드에서 데이터에 액세스할 수 있습니다.
- 다른 데이터 센터 또는 클라우드에 데이터 캐시를 쉽고 빠르게 생성할 수 있기를 원합니다.

Astra Trident가 프로비저닝하지 않은 기존 NFS 볼륨을 노출합니다

기존 NetApp ONTAP NFS 볼륨을 Astra Trident에 의해 프로비저닝하지 않은 경우 이 하위 섹션에 설명된 단계를 따르십시오.



온프레미스 볼륨의 경우 온프레미스 Kubernetes 클러스터에 PV 및 PVC를 생성합니다. Amazon FSx for NetApp ONTAP 볼륨의 경우 Amazon EKS에서 PV 및 PVC를 생성합니다.

먼저 Kubernetes 클러스터에 영구 볼륨(PV) 및 영구 볼륨 클레임(PVC)을 생성해야 합니다. PV 및 PVC를 생성하려면 ["NFS PV/PVC 예"](#) Domino 관리 가이드에서 값을 업데이트하여 환경에 맞게 업데이트합니다. 예 올바른 값을 지정해야 합니다 namespace, nfs.path, 및 nfs.server 필드를 선택합니다. 또한 해당 ONTAP NFS 볼륨에 저장된 데이터의 특성을 나타내는 PV 및 PVC 고유 이름을 지정하는 것이 좋습니다. 예를 들어, 볼륨에 제조 결함의 이미지가 포함되어 있으면 PV, pv-mfg-defect-images 및 PVC, pvc-mfg-defect-images.

Domino에서 외부 데이터 볼륨을 등록합니다

그런 다음 Domino에 외부 데이터 볼륨을 등록해야 합니다. 외부 데이터 볼륨을 등록하려면 을 참조하십시오 ["지침"](#) Domino 관리 가이드 볼륨을 등록할 때 'Volume Type(볼륨 유형)' 드롭다운 메뉴에서 "NFS"를 선택해야 합니다. "NFS"를 선택하면 '사용 가능한 볼륨' 목록에 PVC가 표시됩니다.

The screenshot shows a 'Register an External Volume' window. On the left, a sidebar lists three steps: '1 Volume NFS', '2 Configuration Read-Only', and '3 Access Everyone'. The main content area has a 'Volume Type' dropdown menu set to 'NFS'. Below it, under the heading 'Available Volumes', there is a list box containing one item: 'chatbot-data-cache'. At the bottom right of the window are two buttons: 'Cancel' and 'Next >'.

Astra Trident가 프로비저닝한 기존 볼륨을 노출합니다

기존 볼륨을 Astra Trident에 의해 프로비저닝한 경우 이 하위 섹션에 설명된 단계를 따르십시오.

기존 PVC 편집

Astra Trident에 의해 볼륨을 프로비저닝한 경우 해당 볼륨에 해당하는 영구 볼륨 클레임(PVC)이 이미 존재합니다. 이 볼륨을 Domino에 노출하려면 PVC를 편집하고 의 레이블 목록에 다음 레이블을 추가해야 합니다
metadata.labels 필드:

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

Domino에서 외부 데이터 볼륨을 등록합니다

그런 다음 Domino에 외부 데이터 볼륨을 등록해야 합니다. 외부 데이터 볼륨을 등록하려면 을 참조하십시오 ["지침"](#) Domino 관리 가이드 볼륨을 등록할 때 'Volume Type(볼륨 유형)' 드롭다운 메뉴에서 "Generic(일반)"을 선택해야 합니다. "Generic(일반)"을 선택하면 'Available Volumes(사용 가능한 볼륨)' 목록에 PVC가 표시됩니다.

서로 다른 환경에서 동일한 데이터에 액세스합니다

이 섹션에서는 다양한 컴퓨팅 환경에서 동일한 데이터에 액세스하기 위해 수행해야 하는 작업에 대해 설명합니다. Domino MLOps 플랫폼에서 컴퓨팅 환경을 "데이터 플레인"이라고 합니다. 데이터가 한 데이터 플레인의 NetApp 볼륨에 있지만 다른 데이터 플레인에서 데이터에 액세스해야 하는 경우 이 섹션에 요약된 작업을 따르십시오. 이러한 유형의 시나리오를 "버스팅(bursting)" 또는 대상 환경이 클라우드인 경우 "클라우드 버스팅(Cloud Bursting)"이라고 합니다. 이 기능은 제한되거나 초과 할당된 컴퓨팅 리소스를 처리할 때 필요한 경우가 많습니다. 예를 들어, 온프레미스 컴퓨팅 클러스터가 초과 할당된 경우 워크로드를 즉시 시작할 수 있는 클라우드로 워크로드를 예약할 수 있습니다.

다른 데이터 플레인에 상주하는 NetApp 볼륨에 액세스하기 위한 두 가지 권장 옵션이 있습니다. 이러한 옵션은 아래 하위 섹션에 설명되어 있습니다. 특정 요구 사항에 따라 다음 옵션 중 하나를 선택합니다. 다음 표에는 두 가지 옵션의 이점과 단점이 설명되어 있습니다.

옵션을 선택합니다	이점	단점
옵션 1 - 캐시	<ul style="list-style-type: none">- 간단한 워크플로- 필요에 따라 데이터의 하위 집합을 캐시하는 기능- 소스에 데이터를 다시 쓰는 기능입니다- 관리할 원격 복제본이 없습니다	<ul style="list-style-type: none">- 캐시 하이드레이션에 따른 초기 데이터 액세스의 지연 시간이 증가합니다.
옵션 2 - 거울 이미지	<ul style="list-style-type: none">- 소스 볼륨의 전체 복사본입니다- 캐시 수화로 인한 지연 시간 증가 없음(미러 작업 완료 후)	<ul style="list-style-type: none">- 데이터를 액세스하기 전에 미러 작업이 완료될 때까지 기다려야 합니다- 원격 복제본을 관리해야 합니다- 소스에 다시 쓸 수 없습니다

옵션 1 - 다른 데이터 계층에 있는 볼륨의 캐시를 생성합니다

와 함께 ["NetApp FlexCache 기술"](#) 다른 데이터 플레인에서 상주하는 NetApp 볼륨의 캐시를 생성할 수 있습니다. 예를 들어, 온프레미스 데이터 플레인에서 NetApp 볼륨이 있는 경우 AWS 데이터 플레인에서 해당 볼륨에 액세스해야 하는 경우 AWS에 볼륨의 캐시를 생성할 수 있습니다. 이 섹션에서는 다른 데이터 플레인에서 상주하는 NetApp 볼륨의 캐시를 생성하기 위해 수행해야 하는 작업에 대해 간략하게 설명합니다.

대상 환경에 **FlexCache** 볼륨을 생성합니다



대상 환경이 온프레미스 데이터 센터인 경우 온프레미스 ONTAP 시스템에 FlexCache 볼륨을 생성합니다. 대상 환경이 AWS인 경우 Amazon FSx for NetApp ONTAP 인스턴스에 FlexCache 볼륨을 생성합니다.

먼저 대상 환경에 FlexCache 볼륨을 생성해야 합니다.

BlueXP를 사용하여 FlexCache 볼륨을 생성하는 것이 좋습니다. BlueXP로 FlexCache 볼륨을 생성하려면 [에 요약된 지침을 따르십시오 "BlueXP 볼륨 캐싱 설명서"](#).

BlueXP를 사용하지 않으려는 경우, ONTAP System Manager 또는 ONTAP CLI를 사용하여 FlexCache 볼륨을 생성할 수 있습니다. System Manager로 FlexCache 볼륨을 생성하려면 [에 나와 있는 지침을 참조하십시오 "ONTAP 설명서"](#). ONTAP CLI로 FlexCache 볼륨을 생성하려면 [에 나와 있는 지침을 참조하십시오 "ONTAP 설명서"](#).

이 프로세스를 자동화하려면 [을 사용할 수 있습니다 "BlueXP API"](#), ["ONTAP REST API를 참조하십시오"](#) 또는 [을 누릅니다 "ONTAP Ansible 컬렉션"](#).



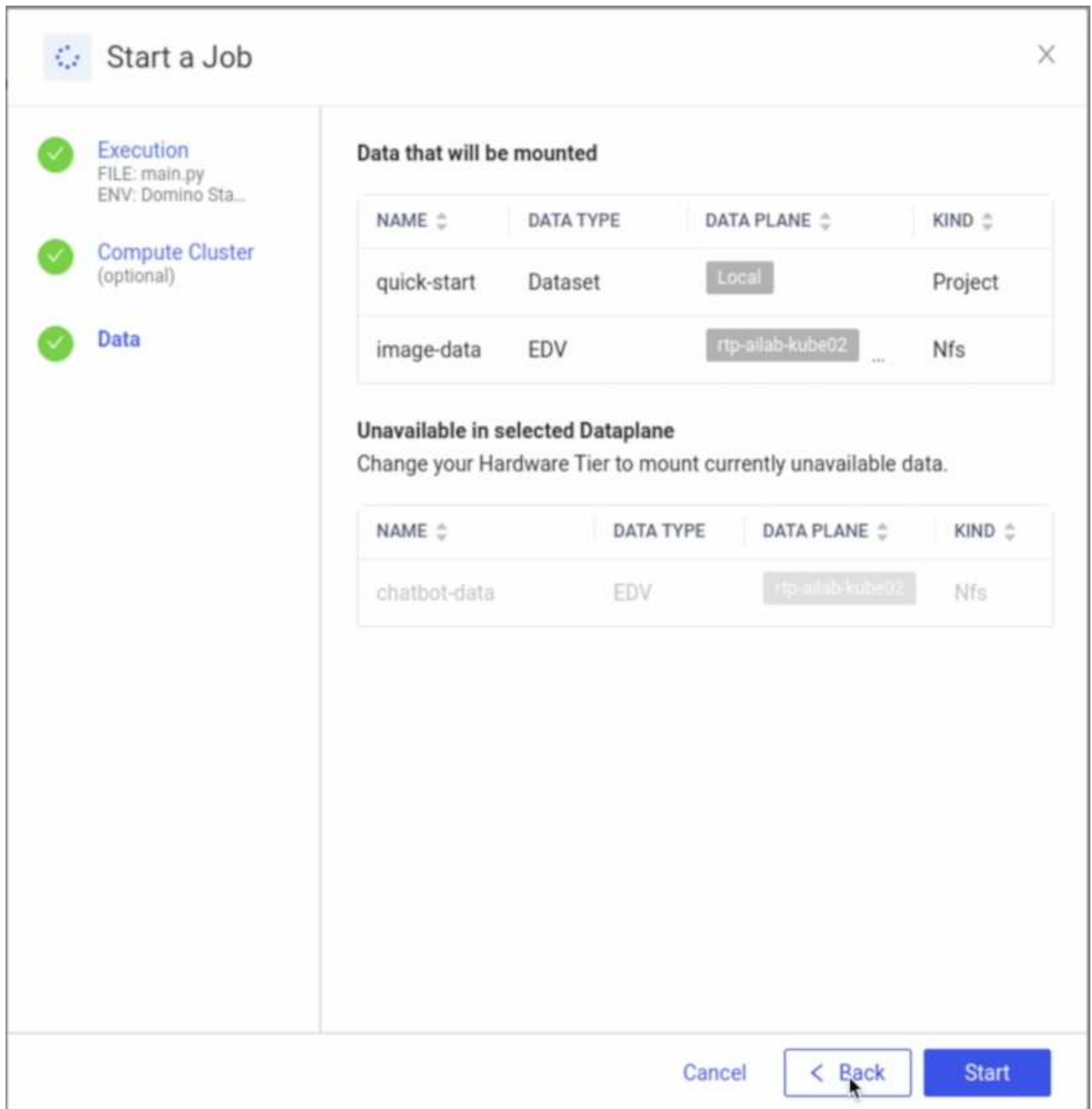
Amazon FSx for NetApp ONTAP에서는 System Manager를 사용할 수 없습니다.

FlexCache 볼륨을 **Domino**에 표시합니다

다음으로 FlexCache 볼륨을 Domino MLOps 플랫폼에 노출해야 합니다. FlexCache 볼륨을 Domino에 표시하려면 [의 'Astra Trident에서 프로비저닝하지 않은 기존 NFS 볼륨 노출' 하위 섹션에 설명된 지침을 따르십시오 "기존 NetApp 볼륨을 Domino에 노출" 섹션을 참조하십시오](#) 있습니다.

이제 다음 스크린샷과 같이 대상 데이터 평면에서 작업 및 작업 공간을 시작할 때 FlexCache 볼륨을 마운트할 수 있습니다.

FlexCache 볼륨을 생성하기 전에



도미노에 FlexCache 볼륨을 노출 한 후

Start a Job

Execution

FILE: model.py

ENV: Domino Sta...

Compute Cluster

(optional)

3

Data

Data that will be mounted

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

Unavailable in selected Dataplane

Change your Hardware Tier to mount currently unavailable data.

NAME	DATA TYPE	DATA PLANE	KIND
No data found			

Cancel

< Back

Start

옵션 2 - 다른 데이터 평면에 상주하는 볼륨을 복제합니다

SnapMirror 관계 작성

BlueXP를 사용하여 SnapMirror 관계를 생성하는 것이 좋습니다. BlueXP와 SnapMirror 관계를 생성하려면 [에 요약된 지침을 따르십시오 "BlueXP 복제 설명서"](#).

BlueXP를 사용하지 않으려는 경우 ONTAP System Manager 또는 ONTAP CLI를 사용하여 SnapMirror 관계를 생성할 수 있습니다. System Manager와 SnapMirror 관계를 생성하려면 [에 나와 있는 지침을 참조하십시오 "ONTAP 설명서"](#). ONTAP CLI와 SnapMirror 관계를 생성하려면 [에 나와 있는 지침을 참조하십시오 "ONTAP 설명서"](#).

이 프로세스를 자동화하려면 [을 사용할 수 있습니다 "BlueXP API"](#), ["ONTAP REST API를 참조하십시오"](#) 또는 [을 누릅니다 "ONTAP Ansible 컬렉션"](#).



Amazon FSx for NetApp ONTAP에서는 System Manager를 사용할 수 없습니다.

SnapMirror 관계 끊기

그런 다음 데이터 액세스를 위해 타겟 볼륨을 활성화하려면 SnapMirror 관계를 해제해야 합니다. 초기 복제가 완료될 때까지 기다린 후 이 단계를 수행하십시오.



BlueXP, ONTAP System Manager 또는 ONTAP CLI에서 미리 상태를 확인하여 복제가 완료되었는지 여부를 확인할 수 있습니다. 복제가 완료되면 미리 상태가 "snapmusled"가 됩니다.

BlueXP를 사용하여 SnapMirror 관계를 해소하는 것이 좋습니다. SnapMirror와 BlueXP의 관계를 해제하려면 [에 설명된 지침을 따르십시오 "BlueXP 복제 설명서"](#).

BlueXP를 사용하지 않으려는 경우 ONTAP System Manager 또는 ONTAP CLI를 사용하여 SnapMirror 관계를 해제할 수 있습니다. System Manager와 SnapMirror의 관계를 해제하려면 [에 설명된 지침을 참조하십시오 "ONTAP 설명서"](#). ONTAP CLI와의 SnapMirror 관계를 해제하려면 [에 설명된 지침을 참조하십시오 "ONTAP 설명서"](#).

이 프로세스를 자동화하려면 [을 사용할 수 있습니다 "BlueXP API"](#), ["ONTAP REST API를 참조하십시오"](#) 또는 [을 누릅니다 "ONTAP Ansible 컬렉션"](#).

Domino에 대상 볼륨을 표시합니다

다음으로 대상 볼륨을 Domino MLOps 플랫폼에 노출해야 합니다. 대상 볼륨을 Domino에 노출하려면 [의 'Astra Trident에서 프로비저닝하지 않은 기존 NFS 볼륨 노출' 하위 섹션에 설명된 지침을 따르십시오 "기존 NetApp 볼륨을 Domino에 노출" 섹션을 참조하십시오](#) 있습니다.

이제 다음 스크린샷과 같이 대상 데이터 평면에서 작업 및 작업 공간을 시작할 때 대상 볼륨을 마운트할 수 있습니다.

SnapMirror 관계를 생성하기 전에

Start a Job

Execution

FILE: model.py

ENV: Domino Sta...

Compute Cluster

(optional)

3

Data

Data that will be mounted

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

Unavailable in selected Dataplane

Change your Hardware Tier to mount currently unavailable data.

NAME	DATA TYPE	DATA PLANE	KIND
No data found			

Cancel

< Back

Start

추가 정보를 찾을 수 있는 위치

이 문서에 설명된 정보에 대한 자세한 내용은 다음 문서 및/또는 웹 사이트를 참조하십시오.

- Domino 데이터 랩

"https://domino.ai"

- 도미노 넥서스

"https://domino.ai/platform/nexus"

- NetApp BlueXP

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

- NetApp ONTAP 데이터 관리 소프트웨어

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

- NetApp AI 솔루션

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

감사의 말

- Josh Mineroff, Domino Data Lab 기술 제휴 SA 책임자
- Nicholas Jablonski, Domino Data Lab의 현장 CTO
- Prabu Arjunan, NetApp 솔루션 설계자
- Brian Young, NetApp 기술 제휴 파트너 글로벌 제휴 책임자

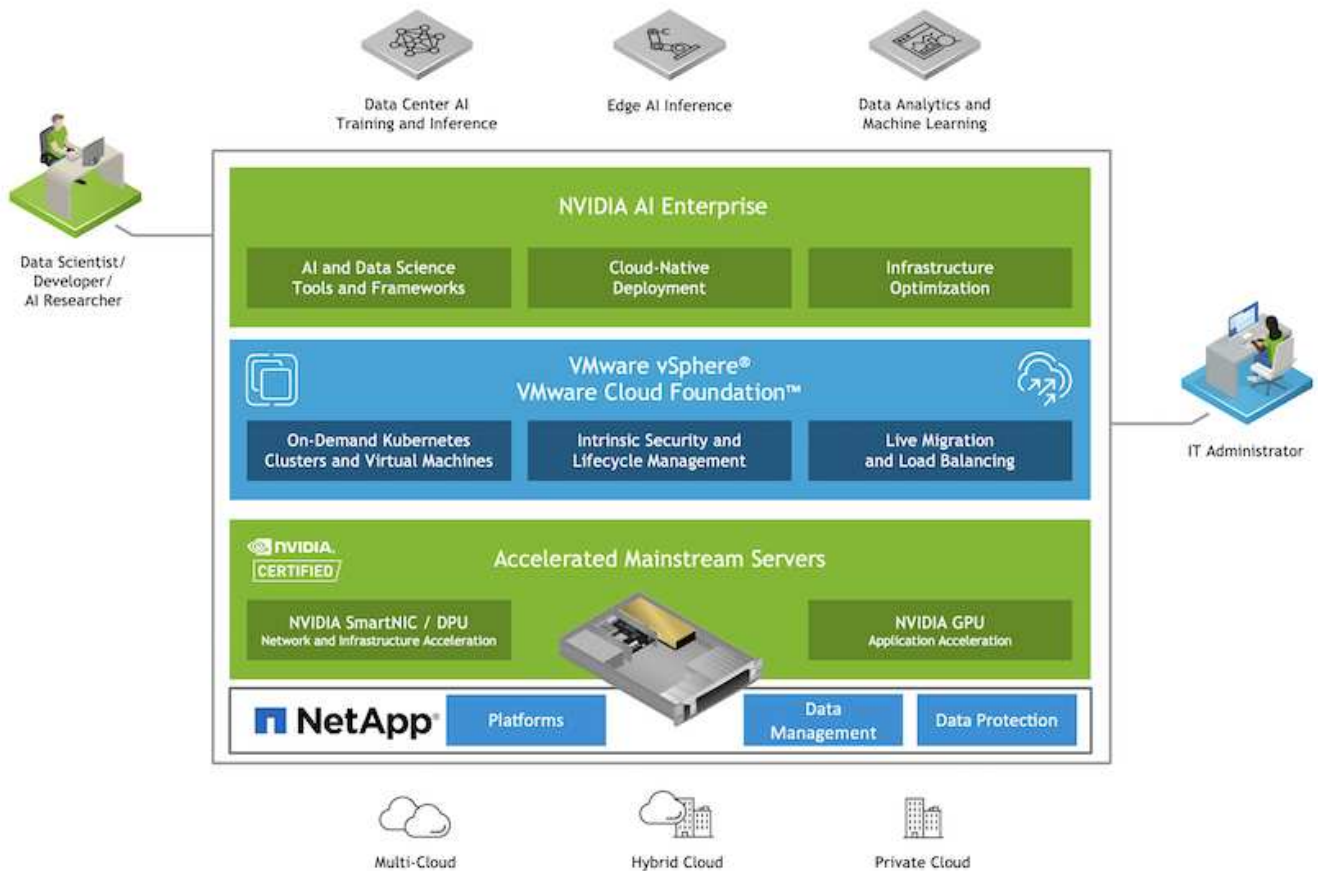
NetApp 및 VMware를 사용하는 NVIDIA AI Enterprise

NetApp 및 VMware를 사용하는 NVIDIA AI Enterprise

Mike Oglesby, NetApp에서 직접 지원합니다

IT 설계자 및 관리자의 경우 AI 툴링은 복잡하고 익숙하지 않을 수 있습니다. 또한 많은 AI 플랫폼은 엔터프라이즈급이 아닙니다. NetApp과 VMware이 지원하는 NVIDIA AI Enterprise는 간소화된 엔터프라이즈급 AI 아키텍처를 제공하도록 제작되었습니다.

NVIDIA AI Enterprise는 NVIDIA 인증 시스템과 함께 VMware vSphere에서 실행하도록 NVIDIA에서 최적화, 인증 및 지원하는 엔드 투 엔드 클라우드 네이티브 AI 및 데이터 분석 소프트웨어 제품군입니다. 이 소프트웨어를 사용하면 최신 하이브리드 클라우드 환경에서 AI 워크로드를 쉽고 빠르게 구축, 관리, 확장할 수 있습니다. NetApp 및 VMware를 기반으로 하는 NVIDIA AI Enterprise는 단순하고 친숙한 패키지로 엔터프라이즈급 AI 워크로드 및 데이터 관리를 제공합니다.



기술 개요

NVIDIA AI 엔터프라이즈

NVIDIA AI Enterprise는 NVIDIA 인증 시스템과 함께 VMware vSphere에서 실행하도록 NVIDIA에서 최적화, 인증 및 지원하는 엔드 투 엔드 클라우드 네이티브 AI 및 데이터 분석 소프트웨어 제품군입니다. 이 소프트웨어를 사용하면 최신 하이브리드 클라우드 환경에서 AI 워크로드를 쉽고 빠르게 구축, 관리, 확장할 수 있습니다.

NGC(NVIDIA GPU Cloud)

NVIDIA NGC는 AI 전문가가 AI 솔루션을 개발할 수 있도록 GPU 최적화 소프트웨어 카탈로그를 호스팅합니다. 또한 모델 훈련을 위한 NVIDIA Base Command, 모델을 배포 및 모니터링하는 NVIDIA Fleet Command, 독점 AI 소프트웨어에 안전하게 액세스하고 관리하기 위한 NGC 프라이빗 레지스트리 등 다양한 AI 서비스에 대한 액세스를 제공합니다. 또한, NVIDIA AI 엔터프라이즈 고객은 NGC 포털을 통해 지원을 요청할 수 있습니다.

VMware vSphere를 참조하십시오

VMware vSphere는 데이터 센터를 CPU, 스토리지 및 네트워킹 리소스를 포함하는 집계된 컴퓨팅 인프라로 변환하는 VMware의 가상화 플랫폼입니다. vSphere는 이러한 인프라스트럭처를 통합 운영 환경으로 관리하고 관리자에게 해당 환경에 참여하는 데이터 센터를 관리할 수 있는 툴을 제공합니다.

vSphere의 두 가지 핵심 구성 요소는 ESXi와 vCenter Server입니다. ESXi는 관리자가 가상 머신 및 가상 어플라이언스를 생성 및 실행하는 가상화 플랫폼입니다. vCenter Server는 관리자가 네트워크 및 풀 호스트 리소스에 연결된 여러 호스트를 관리하는 서비스입니다.

NetApp ONTAP를 참조하십시오

NetApp의 최신 세대 스토리지 관리 소프트웨어인 ONTAP 9는 기업이 인프라를 현대화하고 클라우드 지원 데이터 센터로 전환할 수 있도록 지원합니다. ONTAP는 업계 최고 수준의 데이터 관리 기능을 활용하여 데이터가 상주하는 위치와 상관없이 단일 톨셋으로 데이터를 관리하고 보호할 수 있습니다. 필요에 따라 에지, 코어, 클라우드 등 어느 위치로도 데이터를 자유롭게 이동할 수 있습니다. ONTAP 9에는 데이터 관리를 단순화하고, 중요 데이터를 더 빨리 처리하고, 보호하며, 하이브리드 클라우드 아키텍처 전체에서 차세대 인프라 기능을 지원하는 다양한 기능이 포함되어 있습니다.

데이터 관리를 단순화하십시오

데이터 관리는 AI 애플리케이션에 적합한 리소스를 사용하고 AI/ML 데이터 세트를 교육할 수 있도록 엔터프라이즈 IT 운영 및 데이터 과학자에게 매우 중요합니다. NetApp 기술에 대한 다음 추가 정보는 이 검증의 범위에 포함되지 않지만, 배포에 따라 달라질 수 있습니다.

ONTAP 데이터 관리 소프트웨어에는 운영을 간소화 및 단순화하고 총 운영 비용을 절감하는 다음과 같은 기능이 있습니다.

- 인라인 데이터 컴팩션 및 확대된 중복제거: 데이터 컴팩션은 스토리지 블록 내부의 낭비되는 공간을 줄이고, 중복제거는 실제 용량을 상당히 늘려줍니다. 이는 로컬에 저장된 데이터와 클라우드로 계층화된 데이터에 적용됩니다.
- 최소, 최대 및 적응형 서비스 품질(AQoS): 세부적인 서비스 품질(QoS) 제어로 고도의 공유 환경에서 중요 애플리케이션의 성능 수준을 유지할 수 있습니다.
- NetApp FabricPool을 참조하십시오. AWS(Amazon Web Services), Azure, NetApp StorageGRID 스토리지 솔루션을 포함한 퍼블릭 클라우드 및 프라이빗 클라우드 스토리지에 콜드 데이터를 자동으로 계층화합니다. FabricPool에 대한 자세한 내용은 [이를 참조하십시오 "TR-4598: FabricPool 모범 사례"](#).

데이터 가속화 및 보호

ONTAP는 탁월한 수준의 성능과 데이터 보호를 제공하며 다음과 같은 방법으로 이러한 기능을 확장합니다.

- 성능 및 짧은 지연 시간: ONTAP는 가장 짧은 지연 시간으로 가장 높은 처리량을 제공합니다.
- 데이터 보호: ONTAP는 모든 플랫폼에서 공통 관리를 지원하는 내장 데이터 보호 기능을 제공합니다.
- NVE(NetApp 볼륨 암호화). ONTAP는 온보드 및 외부 키 관리를 모두 지원하는 기본 볼륨 레벨 암호화를 제공합니다.
- 멀티테넌시 및 다단계 인증. ONTAP를 사용하면 인프라 리소스를 최고 수준의 보안으로 공유할 수 있습니다.

미래 지향형 인프라

ONTAP은 다음과 같은 기능을 통해 끊임없이 변화하는 까다로운 비즈니스 요구사항을 충족할 수 있도록 지원합니다.

- 원활한 확장 및 무중단 운영: ONTAP은 운영 중단 없이 기존 컨트롤러 및 스케일아웃 클러스터에 용량을 추가할 수 있도록 지원합니다. 고객은 고비용이 따르는 데이터 마이그레이션이나 운영 중단 없이 NVMe 및 32Gb FC와 같은 최신 기술로 업그레이드할 수 있습니다.
- 클라우드 연결: ONTAP은 클라우드에 가장 많이 연결된 스토리지 관리 소프트웨어로, 모든 퍼블릭 클라우드에서 ONTAP Select(소프트웨어 정의 스토리지) 및 NetApp Cloud Volumes Service(클라우드 네이티브 인스턴스)에 대한 옵션을 제공합니다.
- 새로운 애플리케이션과 통합: ONTAP은 기존 엔터프라이즈 앱을 지원하는 인프라와 동일한 인프라를 사용하여 자율주행 차량, 스마트 시티, Industry 4.0과 같은 차세대 플랫폼 및 애플리케이션을 위한 엔터프라이즈급 데이터 서비스를 제공합니다.

NetApp DataOps 툴킷

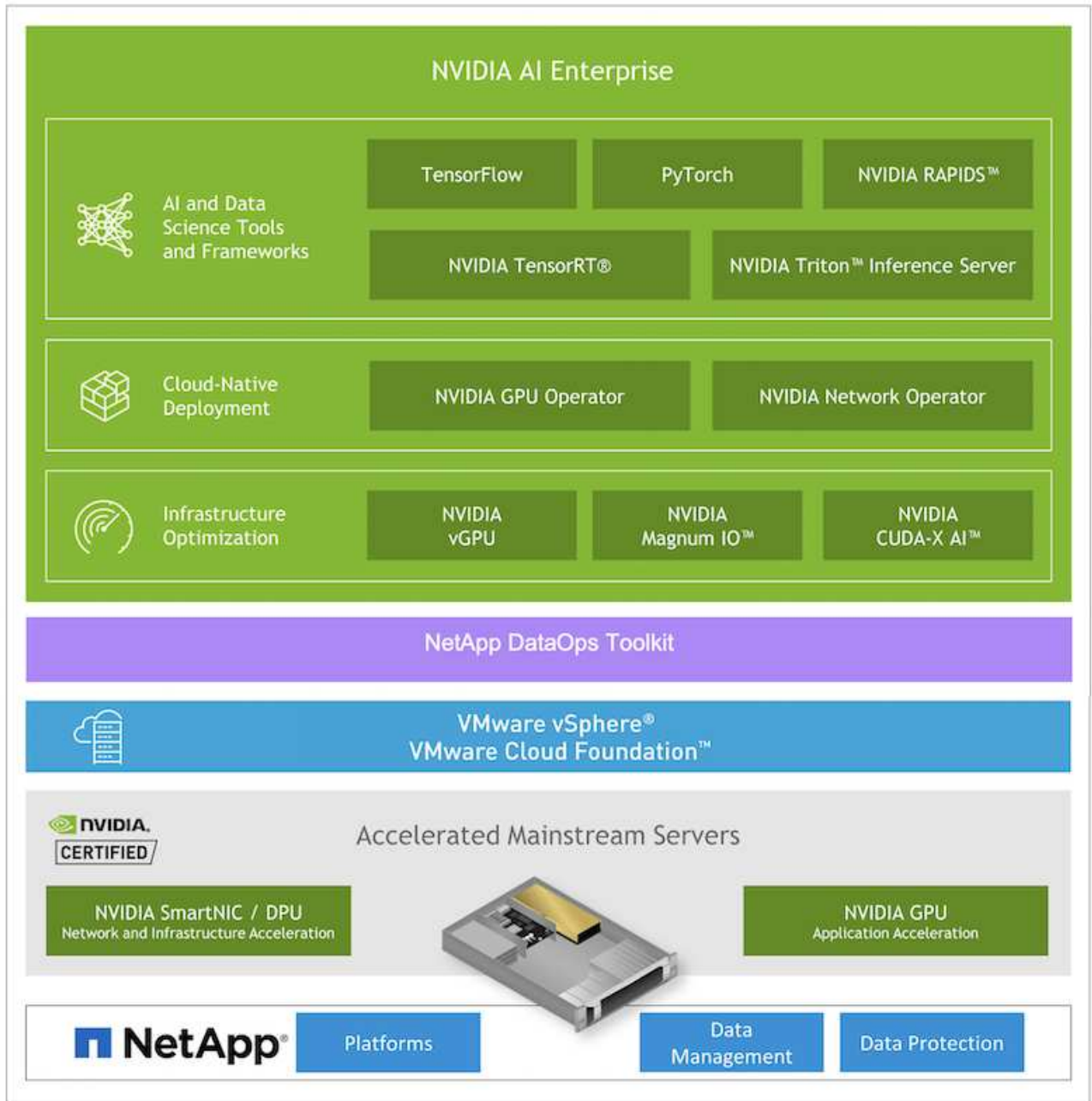
NetApp DataOps 툴킷은 고성능 스케일아웃 NetApp 스토리지가 지원하는 개발/교육 작업 공간 및 추론 서버의 관리를 단순화하는 Python 기반 툴입니다. 주요 기능은 다음과 같습니다.

- 고성능 스케일아웃 NetApp 스토리지를 기반으로 하는 새로운 고용량 JupyterLab 작업 공간을 빠르게 프로비저닝합니다.
- 엔터프라이즈급 NetApp 스토리지를 통해 지원되는 새로운 NVIDIA Triton Inference Server 인스턴스를 빠르게 프로비저닝합니다.
- 실험 또는 신속한 반복을 위해 고용량 JupyterLab 작업 공간을 거의 동시에 복제합니다.
- 백업 및/또는 추적 가능성/베이스라인 기능을 위해 고용량 JupyterLab 작업 공간의 스냅샷을 거의 동시에 저장합니다.
- 대용량 고성능 데이터 볼륨을 거의 동시에 프로비저닝, 복제 및 스냅샷으로 제공합니다.

있습니다

이 솔루션은 NetApp, VMware 및 NVIDIA 인증 시스템을 갖춘, 그 우수성이 입증된 친숙한 아키텍처를 기반으로 구축되었습니다. 자세한 내용은 다음 표를 참조하십시오.

구성 요소	세부 정보
AI 및 데이터 분석 소프트웨어	" VMware용 NVIDIA AI Enterprise "
가상화 플랫폼	" VMware vSphere를 참조하십시오 "
컴퓨팅 플랫폼	" NVIDIA 인증 시스템 "
데이터 관리 플랫폼	" NetApp ONTAP를 참조하십시오 "



초기 설정

이 섹션에서는 NetApp 및 VMware에서 NVIDIA AI Enterprise를 활용하기 위해 수행해야 하는 초기 설정 작업에 대해 설명합니다.

필수 구성 요소

이 섹션에 요약된 단계를 수행하기 전에 이미 VMware vSphere 및 NetApp ONTAP를 구축했다고 가정합니다. 을 참조하십시오 ["NVIDIA AI 엔터프라이즈 제품 지원 매트릭스"](#) 지원되는 vSphere 버전에 대한 자세한 내용은 를 참조하십시오. 을 참조하십시오 ["NetApp 및 VMware 솔루션 설명서"](#) NetApp ONTAP와 함께 VMware vSphere를 구축하는 방법에 대한 자세한 내용은 를 참조하십시오.

NVIDIA AI Enterprise Host Software를 설치합니다

NVIDIA AI Enterprise 호스트 소프트웨어를 설치하려면 이 섹션 1-4에 설명된 지침을 따르십시오 ["NVIDIA AI Enterprise 빠른 시작 가이드 를 참조하십시오"](#).

NVIDIA NGC 소프트웨어를 활용합니다

이 섹션에서는 NVIDIA AI 엔터프라이즈 환경에서 NVIDIA NGC 엔터프라이즈 소프트웨어를 활용하기 위해 수행해야 하는 작업에 대해 설명합니다.

설정

이 섹션에서는 NVIDIA AI Enterprise 환경에서 NVIDIA NGC 엔터프라이즈 소프트웨어를 활용하기 위해 수행해야 하는 초기 설정 작업에 대해 설명합니다.

필수 구성 요소

이 섹션에 요약된 단계를 수행하기 전에 이 나와 있는 지침에 따라 NVIDIA AI Enterprise 호스트 소프트웨어를 이미 구축했다고 가정합니다 ["초기 설정" 페이지](#).

vGPU를 사용하여 Ubuntu 게스트 VM을 생성합니다

먼저 vGPU를 사용하여 Ubuntu 20.04 게스트 VM을 만들어야 합니다. vGPU를 사용하여 Ubuntu 20.04 게스트 VM을 생성하려면 이 지침 개요를 따르십시오 ["NVIDIA AI Enterprise 구축 가이드 를 참조하십시오"](#).

NVIDIA 게스트 소프트웨어를 다운로드하고 설치합니다

그런 다음 이전 단계에서 생성한 게스트 VM 내에 필요한 NVIDIA 게스트 소프트웨어를 설치해야 합니다. 게스트 VM 내에서 필요한 NVIDIA 게스트 소프트웨어를 다운로드하여 설치하려면 이 섹션 5.1-5.4에 설명된 지침을 따르십시오 ["NVIDIA AI Enterprise 빠른 시작 가이드 를 참조하십시오"](#).



섹션 5.4에 설명된 검증 작업을 수행할 때, 가이드를 작성한 후 CUDA 컨테이너 이미지가 업데이트되었으므로 다른 CUDA 컨테이너 이미지 버전 태그를 사용해야 할 수 있습니다. 검증을 위해 'NVIDIA/CUDA:11.0.3-BASE-uubu20.04'를 사용했습니다.

AI/분석 프레임워크 컨테이너 다운로드

그런 다음, 게스트 VM 내에서 사용할 수 있도록 NVIDIA NGC에서 필요한 AI 또는 분석 프레임워크 컨테이너 이미지를 다운로드해야 합니다. 게스트 VM 내에서 프레임워크 컨테이너를 다운로드하려면 이 설명된 지침을 따르십시오 ["NVIDIA AI Enterprise 구축 가이드 를 참조하십시오"](#).

NetApp DataOps 툴킷을 설치하고 구성합니다

그런 다음 게스트 VM 내에 기존 환경을 위한 NetApp DataOps 툴킷을 설치해야 합니다. NetApp DataOps 툴킷은 게스트 VM 내의 터미널에서 ONTAP 시스템의 스케일아웃 데이터 볼륨을 직접 관리하는 데 사용할 수 있습니다. 게스트 VM에 NetApp DataOps 툴킷을 설치하려면 다음 작업을 수행하십시오.

1. PIP를 설치합니다.

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

2. 게스트 VM 터미널에서 로그아웃한 후 다시 로그인합니다.
3. NetApp DataOps 툴킷을 구성합니다. 이 단계를 완료하려면 ONTAP 시스템에 대한 API 액세스 세부 정보가 필요합니다. 스토리지 관리자로부터 이러한 정보를 얻어야 할 수 있습니다.

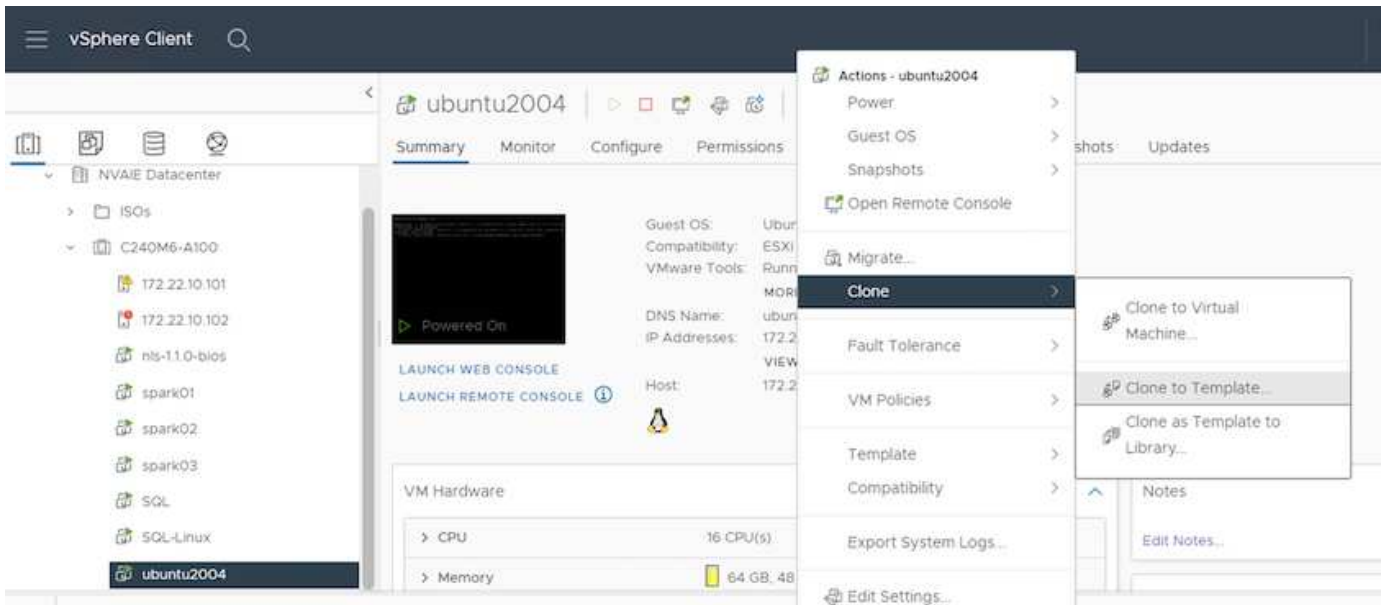
```
$ netapp_dataops_cli.py config

Enter ONTAP management LIF hostname or IP address (Recommendation: Use
SVM management interface): 172.22.10.10
Enter SVM (Storage VM) name: NVAIE-client
Enter SVM NFS data LIF hostname or IP address: 172.22.13.151
Enter default volume type to use when creating new volumes
(flexgroup/flexvol) [flexgroup]:
Enter export policy to use by default when creating new volumes
[default]:
Enter snapshot policy to use by default when creating new volumes
[none]:
Enter unix filesystem user id (uid) to apply by default when creating
new volumes (ex. '0' for root user) [0]:
Enter unix filesystem group id (gid) to apply by default when creating
new volumes (ex. '0' for root group) [0]:
Enter unix filesystem permissions to apply by default when creating new
volumes (ex. '0777' for full read/write permissions for all users and
groups) [0777]:
Enter aggregate to use by default when creating new FlexVol volumes:
aff_a400_01_NVME_SSD_1
Enter ONTAP API username (Recommendation: Use SVM account): admin
Enter ONTAP API password (Recommendation: Use SVM account):
Verify SSL certificate when calling ONTAP API (true/false): false
Do you intend to use this toolkit to trigger BlueXP Copy and Sync
operations? (yes/no): no
Do you intend to use this toolkit to push/pull from S3? (yes/no): no
Created config file: '/home/user/.netapp_dataops/config.json'.
```

게스트 VM 템플릿을 생성합니다

마지막으로, 게스트 VM을 기반으로 VM 템플릿을 생성해야 합니다. 이 템플릿을 사용하여 NVIDIA NGC 소프트웨어를 사용하기 위한 게스트 VM을 빠르게 생성할 수 있습니다.

게스트 VM을 기반으로 VM 템플릿을 생성하려면 VMware vSphere에 로그인하고 게스트 VM 이름을 마우스 오른쪽 단추로 클릭한 다음 '클론'을 선택하고 '템플릿으로 클론 복제...'를 선택한 다음 마법사를 따릅니다.



사용 사례 - TensorFlow 교육 작업 예

이 섹션에서는 NVIDIA AI Enterprise 환경 내에서 TensorFlow 교육 작업을 실행하기 위해 수행해야 하는 작업에 대해 설명합니다.

필수 구성 요소

이 섹션에 설명된 단계를 수행하기 전에 이 섹션에 설명된 지침에 따라 게스트 VM 템플릿을 이미 생성했다고 가정합니다 ["설정"](#) 페이지.

템플릿에서 게스트 **VM**을 생성합니다

먼저 이전 섹션에서 생성한 템플릿에서 새 게스트 VM을 생성해야 합니다. 템플릿에서 새 게스트 VM을 생성하려면 VMware vSphere에 로그인하고 템플릿 이름을 마우스 오른쪽 단추로 클릭한 다음 '이 템플릿에서 새 VM...'을 선택하고 마법사를 따릅니다.

vSphere Client

172.22.10.100

NVAIE Datacenter

Discovered virtual machine

vCLS

nls-1.1.0-bios

spark01

spark02

spark03

SQL

SQL-Linux

ubuntu2004

vgpu-client-ubuntu2004

Recent Tasks

Alarms

Task Name	Target
Delete virtual machine	
Clone virtual machine	

All

More Tasks

vgpu-client-ubuntu2004

SummaryMonitorConfiguration

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

Actions - vgpu-client-ubuntu2004

New VM from This Template...

Convert to Virtual Machine...

Clone to Template...

Clone to Library...

Move to folder...

Rename...

Edit Notes...

Tags & Custom Attributes

Add Permission...

Alarms

Remove from Inventory

Delete from Disk

vSAN

데이터 볼륨 생성 및 마운트

그런 다음, 교육 데이터 세트를 저장할 새 데이터 볼륨을 생성해야 합니다. NetApp DataOps 툴킷을 사용하여 새 데이터 볼륨을 빠르게 생성할 수 있습니다. 다음 예제 명령은 용량이 2TB인 'imagenet'이라는 이름의 볼륨을 생성하는 방법을 보여 줍니다.

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

데이터 볼륨을 데이터로 채우기 전에 게스트 VM 내에 마운트해야 합니다. NetApp DataOps 툴킷을 사용하여 데이터 볼륨을 빠르게 마운트할 수 있습니다. 다음 예제 명령은 이전 단계에서 생성한 볼륨의 모팅을 보여 줍니다.

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

데이터 볼륨을 채웁니다

새 볼륨을 프로비저닝하고 마운트한 후에는 소스 위치에서 교육 데이터 세트를 가져와서 새 볼륨에 배치할 수 있습니다. 일반적으로 이 작업은 S3 또는 Hadoop 데이터 레이크에서 데이터를 가져오는 작업을 수반하며, 경우에 따라 데이터 엔지니어의 도움을 받게 됩니다.

TensorFlow 교육 작업을 실행합니다

이제 TensorFlow 교육 작업을 실행할 준비가 되었습니다. TensorFlow 교육 작업을 실행하려면 다음 작업을 수행하십시오.

1. NVIDIA NGC 엔터프라이즈 TensorFlow 컨테이너 이미지를 가져옵니다.

```
$ sudo docker pull nvcr.io/nvaie/tensorflow-2-1:22.05-tf1-nvaie-2.1-py3
```

2. NVIDIA NGC 엔터프라이즈 TensorFlow 컨테이너의 인스턴스를 시작합니다. '-v' 옵션을 사용하여 데이터 볼륨을 컨테이너에 연결합니다.

```
$ sudo docker run --gpus all -v ~/imagenet:/imagenet -it --rm  
nvcr.io/nvaie/tensorflow-2-1:22.05-tf1-nvaie-2.1-py3
```

3. 컨테이너 내에서 TensorFlow 교육 프로그램을 실행합니다. 다음 예제 명령은 컨테이너 이미지에 포함된 ResNet-50 훈련 프로그램의 예를 보여 줍니다.

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

추가 정보를 찾을 수 있는 위치

이 문서에 설명된 정보에 대한 자세한 내용은 다음 문서 및/또는 웹 사이트를 참조하십시오.

- NetApp ONTAP 데이터 관리 소프트웨어 - ONTAP 정보 라이브러리

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

- NetApp DataOps 툴킷

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

- NVIDIA AI Enterprise 및 VMware

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

감사의 말

- Bobby Oommen, 선임 NetApp 선임 관리자
- Ramesh Isaac, NetApp 시스템 관리자
- Ronney Daniel, NetApp 기술 마케팅 엔지니어

TR-4851: 자율 주행 워크로드를 위한 NetApp StorageGRID 데이터 레이크 - 솔루션 설계

David Arnette, NetApp

TR-4851은 머신 러닝(ML) 및 딥 러닝(DL) 소프트웨어 개발을 위한 데이터 저장소 및 관리 시스템으로 NetApp StorageGRID 오브젝트 스토리지를 사용하는 방법을 보여줍니다. 이 문서에서는 자율주행 차량 소프트웨어 개발의 데이터 흐름과 요구 사항과 데이터 라이프사이클을 간소화하는 StorageGRID 기능에 대해 설명합니다. 이 솔루션은 ML 및 DL 개발 프로세스에서 일반적인 모든 다단계 데이터 파이프라인 워크플로에 적용됩니다.

["TR-4851: 자율 주행 워크로드를 위한 NetApp StorageGRID 데이터 레이크 - 솔루션 설계"](#)

NetApp AI Control Plane

TR-4798: NetApp AI Control Plane

Mike Oglesby, NetApp에서 직접 지원합니다

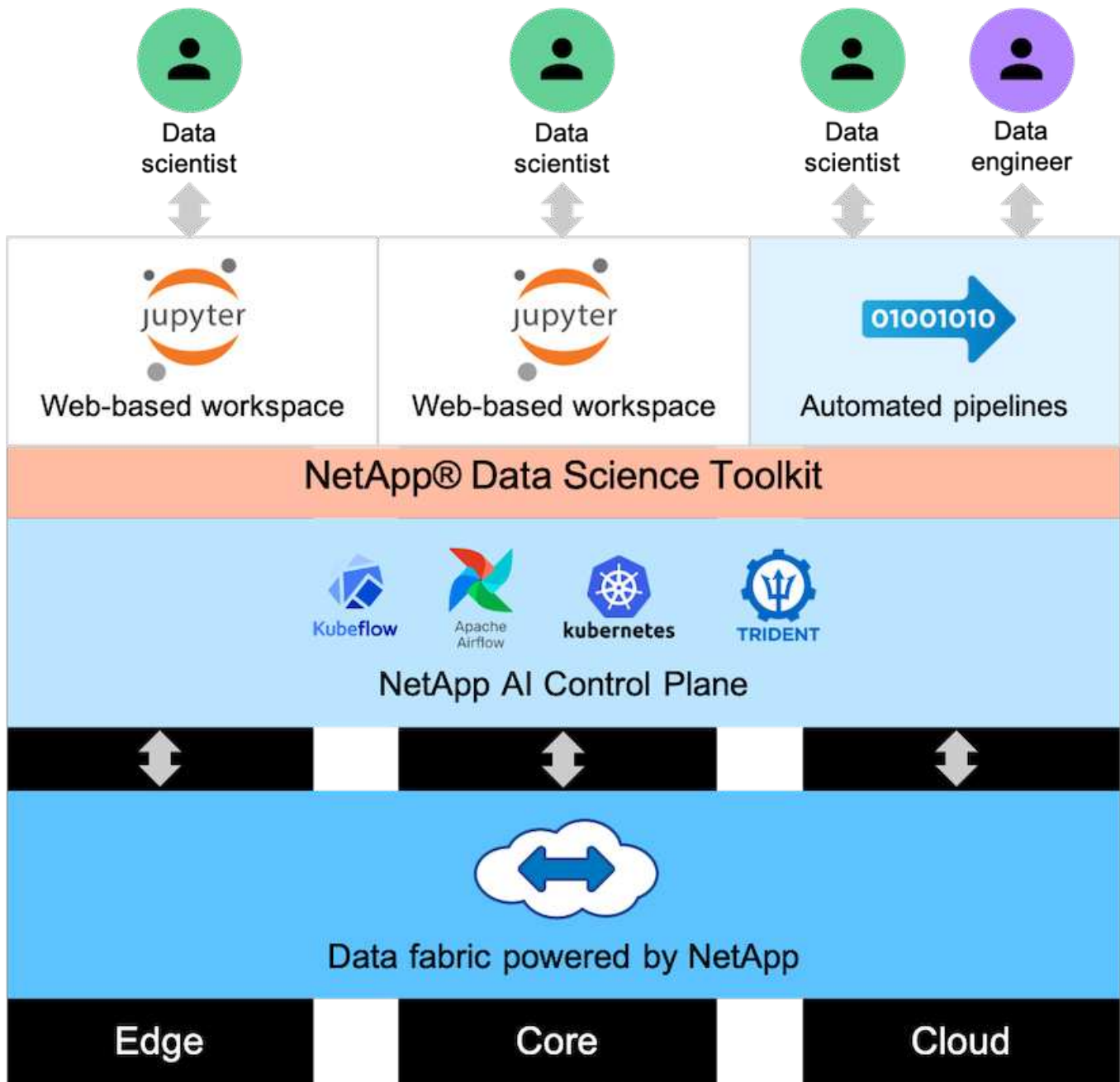
모든 규모와 업종에 상관없이 모든 기업과 조직은 실제 문제를 해결하고 혁신적인 제품과 서비스를 제공하며 경쟁이 갈수록 치열해지는 시장에서 경쟁 우위를 확보하기 위해 인공지능(AI), 머신 러닝(ML), 딥 러닝(DL)으로 눈을 돌리고 있습니다. AI, ML 및 DL의 사용이 증가함에 따라 워크로드 확장성 및 데이터 가용성을 비롯한 많은 과제에 직면하게 됩니다. 이 문서에서는 NetApp 데이터 관리 기능을 널리 사용되는 오픈 소스 툴 및 프레임워크와 결합하여 제공하는 솔루션인 NetApp AI Control Plane을 사용하여 이러한 과제를 해결하는 방법을 보여줍니다.

이 보고서는 데이터 네임스페이스를 빠르게 클론 복제하는 방법을 보여 줍니다. 또한, 사이트 및 지역 전반에서 데이터를 원활하게 복제하여 결합형 통합 AI/ML/DL 데이터 파이프라인을 생성하는 방법을 보여줍니다. 또한 추적 가능성 및 버전

관리를 위한 데이터 및 모델 기준선의 거의 즉각적인 생성을 통합하는 AI, ML 및 DL 교육 워크플로우를 정의하고 구현하는 방법을 안내합니다. 이 솔루션을 사용하면 모든 모델 훈련을 다시 모델을 훈련 및/또는 검증하는 데 사용된 정확한 데이터 세트로 추적할 수 있습니다. 마지막으로, 이 문서에서는 방대한 데이터 세트에 액세스하여 Jupyter Notebook 작업 공간을 신속하게 프로비저닝하는 방법을 설명합니다.

참고: 동일한 데이터 세트에 대한 공유 액세스가 필요한 다수의 GPU 서버가 포함된 대규모 HPC 스타일 분산 교육이거나 병렬 파일 시스템이 필요하거나 선호한다면, 확인해 보십시오 ["TR-4890"](#). 이 기술 보고서에서는 을 포함하는 방법을 설명합니다 ["NetApp의 완벽하게 지원되는 병렬 파일 시스템 솔루션인 BeeGFS"](#) NetApp AI Control Plane의 일부로, 이 솔루션은 소수의 NVIDIA DGX A100 시스템에서 완전한 140개 노드 SuperPOD까지 확장하도록 설계되었습니다.

NetApp AI Control Plane은 데이터 과학자 및 데이터 엔지니어를 대상으로 하므로 최소한의 NetApp 또는 NetApp ONTAP® 전문 지식이 필요합니다. 이 솔루션에서는 단순하고 친숙한 톨과 인터페이스를 사용하여 데이터 관리 기능을 실행할 수 있습니다. 귀사 환경에 NetApp 스토리지가 이미 구축되어 있다면 지금 바로 NetApp AI Control Plane을 시험 구동할 수 있습니다. 솔루션을 시험하고 싶지만 NetApp 스토리지가 없는 경우, 를 방문하십시오 ["cloud.netapp.com"](#)클라우드 기반 NetApp 스토리지 솔루션을 사용하여 몇 분 이내에 시스템을 구축하고 실행할 수 있습니다. 다음 그림은 솔루션을 시각적으로 보여 줍니다.



개념 및 구성 요소

인공 지능

AI는 컴퓨터가 인간의 마음의 인지 기능을 모방하도록 훈련되는 컴퓨터 과학 분야입니다. AI 개발자는 컴퓨터를 교육하여 사람과 유사하거나 훨씬 뛰어난 방식으로 문제를 배우고 해결합니다. 딥 러닝 및 머신 러닝은 AI의 하위 필드입니다. 조직은 중요한 비즈니스 요구사항을 지원하기 위해 AI, ML 및 DL을 점점 더 채택하고 있습니다. 몇 가지 예는 다음과 같습니다.

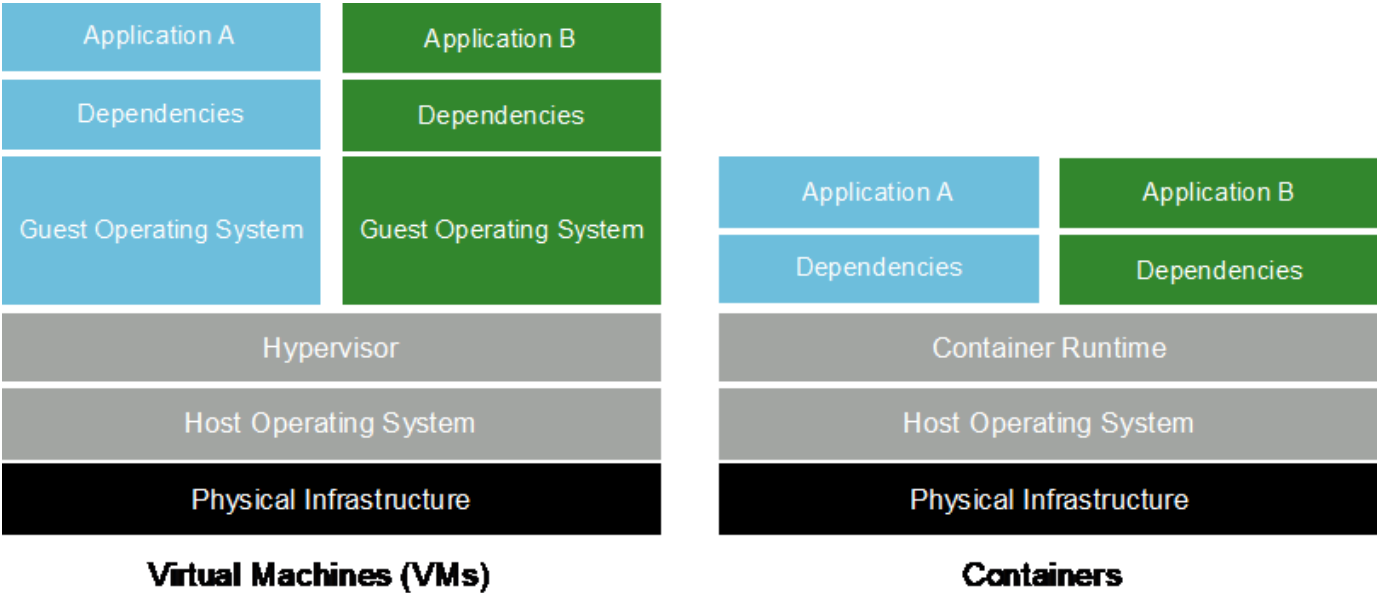
- 방대한 양의 데이터를 분석하여 이전에 알려지지 않은 비즈니스 인사이트를 도출합니다
- 자연어 처리를 사용하여 고객과 직접 상호 작용
- 다양한 비즈니스 프로세스 및 기능 자동화

최신 AI 훈련 및 추론 워크로드에는 대규모 병렬 컴퓨팅 기능이 필요합니다. 따라서 GPU의 병렬 처리 기능이 범용 CPU보다 훨씬 뛰어나기 때문에 AI 작업을 실행하는 데 GPU가 점점 더 많이 사용되고 있습니다.

컨테이너

컨테이너는 공유 호스트 운영 체제 커널 위에서 실행되는 격리된 사용자 공간 인스턴스입니다. 컨테이너 채택이 빠르게 증가하고 있습니다. 컨테이너는 가상 머신(VM)이 제공하는 것과 동일한 애플리케이션 샌드박스(sandbox)의 많은 이점을 제공합니다. 하지만 VM이 사용하는 하이퍼바이저 및 게스트 운영 체제 계층이 없어졌기 때문에 컨테이너는 훨씬 더 가볍습니다. 다음 그림에서는 가상 시스템과 컨테이너를 보여 줍니다.

또한 컨테이너를 사용하면 애플리케이션 종속성, 실행 시간 등을 애플리케이션과 직접 효율적으로 패키징할 수 있습니다. 가장 일반적으로 사용되는 컨테이너 패키징 형식은 Docker 컨테이너입니다. Docker 컨테이너 형식으로 컨테이너화된 애플리케이션은 Docker 컨테이너를 실행할 수 있는 모든 시스템에서 실행할 수 있습니다. 모든 종속성이 컨테이너 자체에 패키징되어 있기 때문에 응용 프로그램의 종속성이 컴퓨터에 없는 경우에도 마찬가지입니다. 자세한 내용은 를 참조하십시오 ["Docker 웹 사이트"](#).



쿠버네티스

Kubernetes는 Google에서 원래 설계한 개방형 소스, 분산형 컨테이너 오케스트레이션 플랫폼으로, 현재 CNCF(Cloud Native Computing Foundation)에서 관리하고 있습니다. Kubernetes는 컨테이너화된 애플리케이션의 구축, 관리, 확장 기능을 자동화할 수 있습니다. 최근 몇 년 동안 Kubernetes는 주요 컨테이너 오케스트레이션 플랫폼으로 부상했습니다. 다른 컨테이너 패키징 형식과 실행 시간이 지원되지만 Kubernetes는 Docker 컨테이너용 오케스트레이션 시스템으로 가장 많이 사용됩니다. 자세한 내용은 를 참조하십시오 ["Kubernetes 웹 사이트"](#).

NetApp 트라이던트

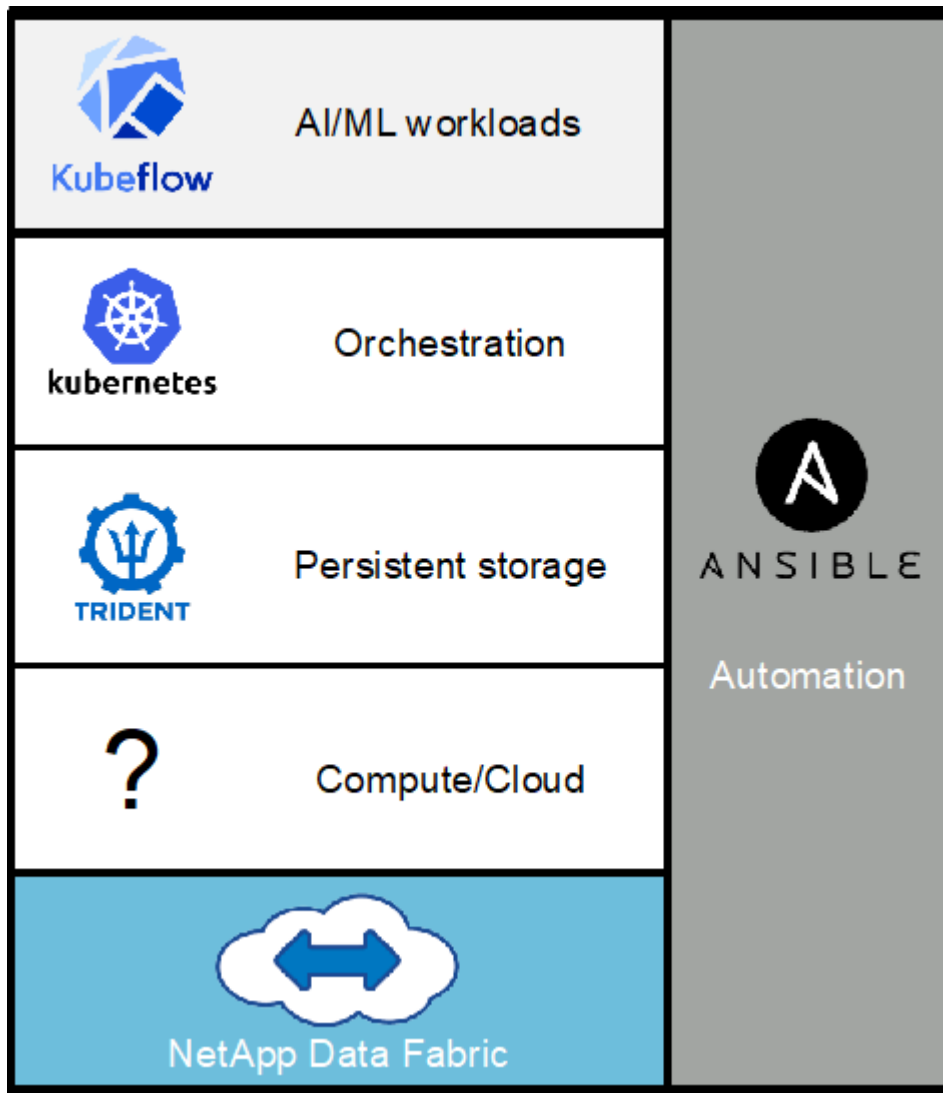
Trident는 NetApp에서 개발 및 유지 관리하는 오픈 소스 스토리지 오케스트레이터로서 Kubernetes 워크로드를 위한 영구 스토리지의 생성, 관리 및 사용을 크게 단순화합니다. Kubernetes 네이티브 애플리케이션인 Trident는 Kubernetes 클러스터 내에서 직접 실행됩니다. Trident를 사용하면 Kubernetes 사용자(개발자, 데이터 과학자, Kubernetes 관리자 등)가 이미 익숙한 표준 Kubernetes 형식으로 영구 스토리지 볼륨을 생성, 관리 및 상호 작용할 수 있습니다. 이와 동시에 NetApp 기술에서 제공하는 NetApp 고급 데이터 관리 기능과 Data Fabric을 활용할 수 있습니다. Trident는 영구 스토리지의 복잡성을 추상화하여 사용이 간편합니다. 자세한 내용은 를 참조하십시오 ["Trident 웹 사이트"](#).

NVIDIA DeepOps

DeepOps는 NVIDIA의 오픈 소스 프로젝트로, Ansible을 사용하여 GPU 서버 클러스터를 모범 사례에 따라 자동으로 구축합니다. DeepOps는 모듈식이며 다양한 배포 작업에 사용할 수 있습니다. 이 문서와 이 문서에서 설명하는 검증 연습에서는 GPU 서버 작업자 노드로 구성된 Kubernetes 클러스터를 배포하는 데 DeepOps를 사용합니다. 자세한 내용은 를 참조하십시오 ["DeepOps 웹 사이트"](#).

Kubeflow

Kubeflow는 Google에서 원래 개발한 Kubernetes용 오픈 소스 AI 및 ML 툴킷입니다. Kubeflow 프로젝트를 통해 Kubernetes에서 AI 및 ML 워크플로우를 간단하게 배포, 이식 및 확장할 수 있습니다. Kubeflow는 복잡한 Kubernetes를 추상화하여 데이터 과학자가 자신이 가장 잘 알고 있는 데이터 과학에 집중할 수 있도록 지원합니다. 시각화는 다음 그림을 참조하십시오. 쿠버플로는 엔터프라이즈 IT 부서가 Kubernetes에서 점점 더 표준화되고 있으므로 상당한 주목을 받고 있습니다. 자세한 내용은 를 참조하십시오 ["Kubeflow 웹 사이트"](#).



Kubeflow 파이프라인

Kubeflow 파이프라인은 Kubeflow의 핵심 구성 요소입니다. Kubeflow 파이프라인은 이식 가능하고 확장 가능한 AI 및 ML 워크플로우를 정의하고 배포하기 위한 플랫폼 및 표준입니다. 자세한 내용은 를 참조하십시오 ["Kubeflow 공식 문서"](#).

Jupyter 노트북 서버

Jupyter Notebook Server는 데이터 과학자가 실시간 코드와 설명이 포함된 Jupyter Notebooks라는 위키 형식의 문서를 만들 수 있는 오픈 소스 웹 애플리케이션입니다. Jupyter Notebooks는 AI 및 ML 프로젝트를 문서화, 저장, 공유하는 수단으로 AI 및 ML 커뮤니티에서 널리 사용되고 있습니다. Kubeflow는 Kubernetes에서 Jupyter Notebook Server의 프로비저닝 및 구축을 단순화합니다. Jupyter Notebooks에 대한 자세한 내용은 [를 참조하십시오 "Jupyter 웹 사이트"](#). Kubeflow와 관련하여 Jupyter Notebooks에 대한 자세한 내용은 [를 참조하십시오 "Kubeflow 공식 문서"](#).

아파치 기류

Apache Airflow는 복잡한 엔터프라이즈 워크플로우를 프로그래밍 방식으로 작성, 스케줄링 및 모니터링할 수 있는 오픈 소스 워크플로우 관리 플랫폼입니다. ETL 및 데이터 파이프라인 워크플로우를 자동화하는 데 주로 사용되지만, 이러한 유형의 워크플로우에만 국한되지 않습니다. Airbnb가 공기 흐름 프로젝트를 시작했지만 그 이후 업계에서 매우 인기를 끌며 현재는 Apache Software Foundation의 후원으로 자리 잡았습니다. Python으로 공기 흐름을 작성하고 Python 스크립트를 통해 공기 흐름을 생성하고 "코드로 구성"이라는 원칙에 따라 공기 흐름을 설계할 수 있습니다. 많은 엔터프라이즈 공기 흐름 사용자가 이제 Kubernetes에서 공기 흐름을 실행합니다.

유도된 DAG(Acclie Graphs)

공기 흐름에서 워크플로우는 DAG(Directed Acyclic Graphs)라고 합니다. DAG는 DAG 정의에 따라 순차적으로, 병렬로 또는 둘의 조합으로 실행되는 작업으로 구성됩니다. 공기 흐름 스케줄러는 DAG 정의에 지정된 작업 수준 종속성을 준수하여 일련의 작업자에 대해 개별 작업을 실행합니다. DAG는 Python 스크립트를 통해 정의 및 생성됩니다.

NetApp ONTAP 9

NetApp ONTAP 9은 NetApp의 최신 세대 스토리지 관리 소프트웨어로, 이 소프트웨어를 사용하여 귀사와 같은 기업에서 인프라를 현대화하고 클라우드 지원 데이터 센터로 전환할 수 있습니다. 업계 최고의 데이터 관리 기능을 갖춘 ONTAP은 데이터의 위치와 상관없이 단일 톨셋으로 데이터를 관리하고 보호할 수 있도록 지원합니다. 필요에 따라 에지, 코어, 클라우드 등 어느 위치로도 데이터를 자유롭게 이동할 수 있습니다. ONTAP 9에는 데이터 관리를 단순화하고, 중요 데이터를 더 빨리 처리하고 보호하는 다수의 기능이 포함되어 있으므로 하이브리드 클라우드 아키텍처 전체에 미래 지향형 인프라를 제공합니다.

데이터 관리 단순화

애플리케이션 및 데이터 세트에 적절한 리소스를 사용할 수 있도록 데이터 관리는 엔터프라이즈 IT 운영에 매우 중요합니다. ONTAP에는 운영을 간소화 및 단순화하고 총 운영 비용을 절감할 수 있는 다음과 같은 기능이 포함되어 있습니다.

- * 인라인 데이터 컴팩션 및 확대된 중복제거. * 데이터 컴팩션은 스토리지 블록 내부의 낭비되는 공간을 줄이고, 중복제거는 실제 용량을 크게 증가시킵니다.
- * 최소, 최대 및 적응형 서비스 품질(QoS). * 세분화된 QoS 제어로 고도의 공유 환경에서 중요 애플리케이션의 성능 수준을 유지할 수 있습니다.
- * ONTAP FabricPool. * 이 기능은 콜드 데이터를 AWS(Amazon Web Services), Azure, NetApp StorageGRID 오브젝트 기반 스토리지와 같은 퍼블릭 및 프라이빗 클라우드 스토리지 옵션으로 자동 계층화합니다.

데이터 가속화 및 보호

ONTAP는 탁월한 수준의 성능과 데이터 보호를 제공하며 다음과 같은 기능으로 이러한 기능을 확장합니다.

- * 고성능 및 낮은 지연 시간 * ONTAP는 가장 짧은 지연 시간으로 가장 높은 처리량을 제공합니다.
- * NetApp ONTAP FlexGroup 기술. * FlexGroup 볼륨은 최대 20PB 및 4천억 개 파일까지 선형적으로 확장할 수

있는 고성능 데이터 컨테이너로, 단일 네임스페이스를 제공하여 데이터 관리를 단순화합니다.

- * 데이터 보호. * ONTAP은 모든 플랫폼에서 공통 관리를 지원하는 내장 데이터 보호 기능을 제공합니다.
- * NetApp 볼륨 암호화. * ONTAP은 온보드 및 외부 키 관리를 모두 지원하는 기본 볼륨 레벨 암호화를 제공합니다.

미래 지향형 인프라

ONTAP 9은 지속적으로 변화하는 까다로운 요구사항을 충족할 수 있도록 지원합니다.

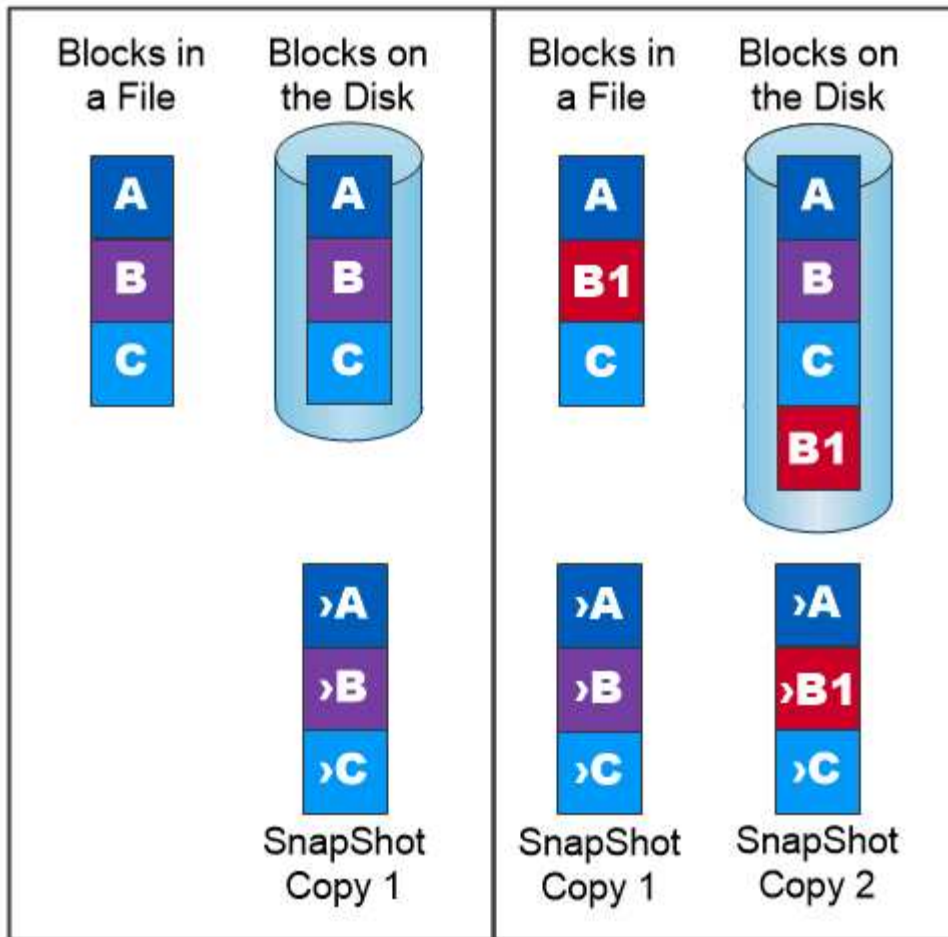
- * 원활한 확장 및 무중단 운영 * ONTAP은 기존 컨트롤러 및 스케일아웃 클러스터에 무중단으로 용량을 추가할 수 있도록 지원합니다. 고비용이 따르는 데이터 마이그레이션이나 운영 중단 없이 NVMe 및 32Gb FC와 같은 최신 기술로 업그레이드할 수 있습니다.
- * 클라우드 연결. * ONTAP은 클라우드에 가장 많이 연결된 스토리지 관리 소프트웨어 중 하나로, 모든 퍼블릭 클라우드에서 소프트웨어 정의 스토리지(ONTAP Select) 및 클라우드 네이티브 인스턴스(NetApp Cloud Volumes Service) 옵션을 제공합니다.
- 새로운 애플리케이션과의 통합 * 기존 엔터프라이즈 애플리케이션을 지원하는 인프라와 동일한 인프라를 사용하는 ONTAP은 OpenStack, Hadoop, MongoDB와 같은 차세대 플랫폼 및 애플리케이션을 위한 엔터프라이즈급 데이터 서비스를 제공합니다.

NetApp Snapshot 복사본

NetApp 스냅샷 복사본은 볼륨의 읽기 전용 시점 이미지입니다. 다음 그림과 같이 이미지는 스토리지 공간을 최소한으로 사용하고, 마지막 스냅샷 복사본 생성 이후 생성된 파일의 변경사항만 기록하므로 경미한 성능 오버헤드를 발생시킵니다.

스냅샷 복사본은 핵심 ONTAP 스토리지 가상화 기술인 WAFL(Write Anywhere File Layout)의 효율성을 높여줍니다. 데이터베이스와 마찬가지로 WAFL은 메타데이터를 사용하여 디스크의 실제 데이터 블록을 가리킵니다. 하지만 WAFL은 데이터베이스와 달리 기존 블록을 덮어쓰지 않습니다. 업데이트된 데이터를 새 블록에 쓰고 메타데이터를 변경합니다. ONTAP은 데이터 블록을 복사하는 대신 스냅샷 복사본을 생성할 때 메타데이터를 참조하므로 스냅샷 복사본이 매우 효율적입니다. 이렇게 하면 복사할 블록을 찾는 데 다른 시스템이 발생하는 탐색 시간과 복사본 자체를 만드는 비용이 제거됩니다.

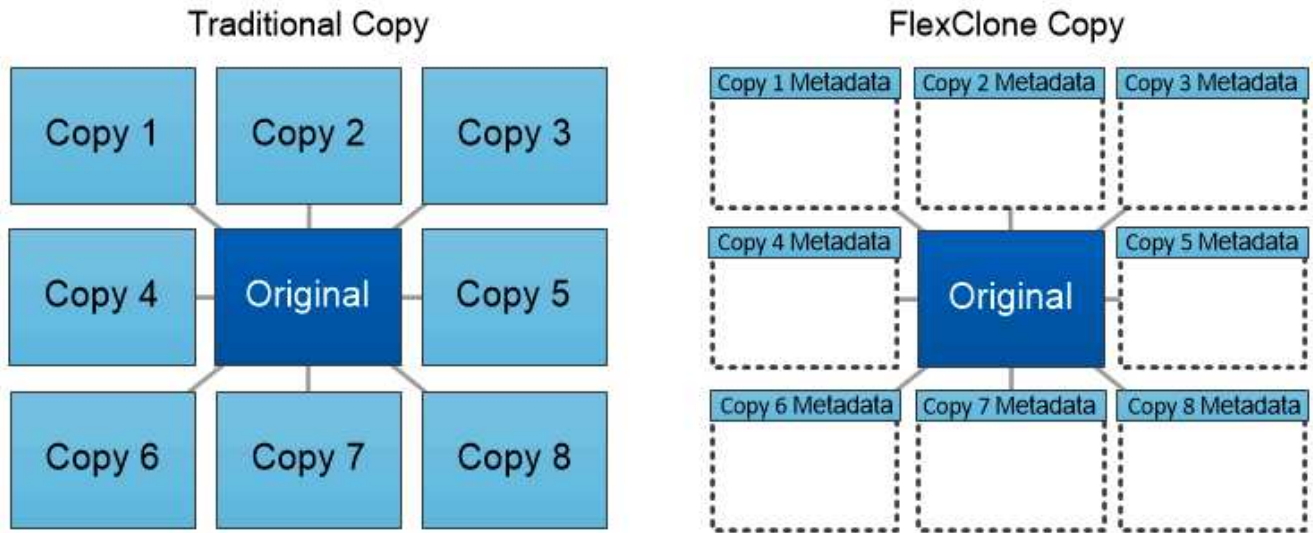
스냅샷 복사본을 사용하여 개별 파일 또는 LUN을 복구하거나 볼륨의 전체 콘텐츠를 복원할 수 있습니다. ONTAP은 스냅샷 복사본의 포인터 정보를 디스크의 데이터와 비교하여 다운타임 또는 상당한 성능 비용 없이 누락 또는 손상된 개체를 재구성합니다.



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

NetApp FlexClone 기술

NetApp FlexClone 기술은 Snapshot 메타데이터를 참조하여 볼륨의 쓰기 가능한 특정 시점 복사본을 생성합니다. 복사본은 다음 그림과 같이 복사본에 변경 사항이 기록될 때까지 메타데이터에 필요한 사항을 제외하고 데이터 블록을 부모와 공유하고 스토리지를 사용하지 않습니다. FlexClone 소프트웨어를 사용하면 기존 복사본을 생성하는 데 몇 분 또는 몇 시간이 걸릴 수 있으며 최대 규모의 데이터 세트도 거의 즉시 복사할 수 있습니다. 따라서 동일한 데이터 세트의 여러 복사본(예: 개발 작업 공간)이 필요하거나 데이터 세트의 임시 복사본(운영 데이터 세트에 대해 애플리케이션 테스트)이 필요한 경우에 적합합니다.

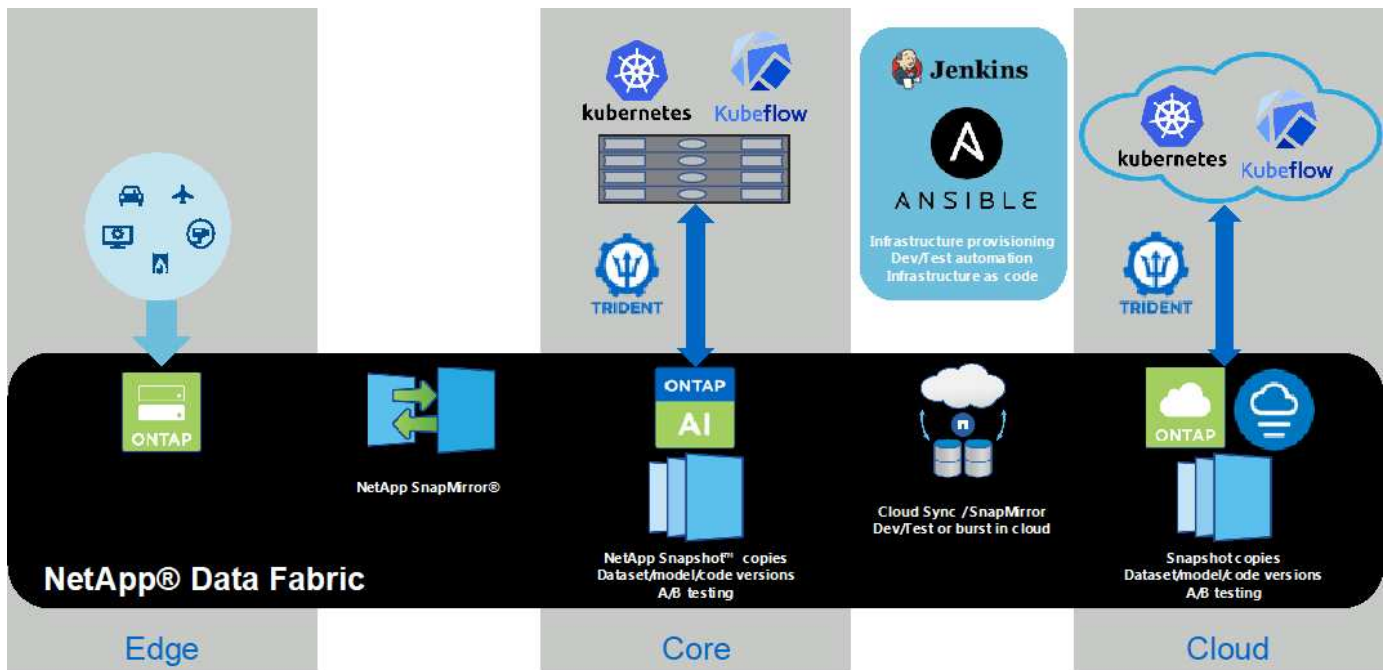


FlexClone copies share data blocks with their parents, consuming no storage except what is required for metadata.

NetApp SnapMirror 데이터 복제 기술

NetApp SnapMirror 소프트웨어는 Data Fabric에서 사용하기 쉬운 비용 효율적인 통합 복제 솔루션입니다. LAN 또는 WAN을 통해 데이터를 고속으로 복제합니다. 가상 환경과 기존 환경 모두에서 비즈니스 크리티컬 애플리케이션을 포함한 모든 유형의 애플리케이션에 대해 높은 데이터 가용성과 빠른 데이터 복제를 제공합니다. 하나 이상의 NetApp 스토리지 시스템에 데이터를 복제하고 2차 데이터를 지속적으로 업데이트함으로써 데이터가 최신 상태로 유지되고 필요할 때마다 사용할 수 있으며 외부 복제 서버가 필요하지 않습니다. 다음 그림은 SnapMirror 기술을 활용하는 아키텍처의 예입니다.

SnapMirror 소프트웨어는 변경된 블록만 네트워크를 통해 전송함으로써 NetApp ONTAP 스토리지 효율성을 활용합니다. SnapMirror 소프트웨어는 또한 내장된 네트워크 압축 기능을 사용하여 데이터 전송을 더 신속하게 수행하고 네트워크 대역폭 활용률을 70%까지 줄입니다. SnapMirror 기술을 사용하면 하나의 씬 복제 데이터 스트림을 활용하여 활성 미러와 이전 시점의 복사본을 둘 다 유지 관리하는 단일 저장소를 만들 수 있으므로 네트워크 트래픽이 최대 50% 감소합니다.



NetApp BlueXP 복사 및 동기화

BlueXP 복사 및 동기화는 빠르고 안전한 데이터 동기화를 제공하는 NetApp 서비스입니다. 온프레미스 NFS 또는 SMB 파일 공유 간에 파일을 전송해야 하는 경우, NetApp StorageGRID, NetApp ONTAP S3, NetApp Cloud Volumes Service, Azure NetApp Files, AWS S3, AWS EFS, Azure Blob, Google Cloud Storage, 즉 IBM Cloud Object Storage, BlueXP Copy and Sync는 필요한 파일을 빠르고 안전하게 이동합니다.

데이터가 전송되면 소스와 타겟 모두에서 사용할 수 있습니다. BlueXP 복사 및 동기화는 업데이트가 트리거되거나 미리 정의된 일정에 따라 데이터가 지속적으로 동기화되는 경우 필요 시 데이터를 동기화할 수 있습니다. BlueXP Copy 및 Sync는 변경된 부분만 이동하므로 데이터 복제에 소비되는 시간과 비용이 최소화됩니다.

BlueXP Copy and Sync는 매우 간단하게 설정하고 사용할 수 있는 서비스형 소프트웨어(SaaS) 툴입니다. BlueXP Copy 및 Sync에 의해 트리거되는 데이터 전송은 데이터 브로커에 의해 수행됩니다. BlueXP Copy 및 Sync 데이터 브로커는 AWS, Azure, Google Cloud Platform 또는 사내에 구축할 수 있습니다.

NetApp XCP

NetApp XCP는 모든 NetApp 및 NetApp 간 데이터 마이그레이션 및 파일 시스템 통찰력을 위한 클라이언트 기반 소프트웨어입니다. xCP는 사용 가능한 모든 시스템 리소스를 활용하여 대용량 데이터 세트 및 고성능 마이그레이션을 처리함으로써 최대한의 성능을 발휘하도록 설계되었습니다. xCP를 사용하면 보고서를 생성하는 옵션을 통해 파일 시스템에 대한 완벽한 가시성을 확보할 수 있습니다.

NetApp XCP는 NFS 및 SMB 프로토콜을 지원하는 단일 패키지로 제공됩니다. xCP에는 NFS 데이터 세트용 Linux 바이너리와 SMB 데이터 세트용 Windows 실행 파일이 포함되어 있습니다.

NetApp XCP File Analytics는 파일 공유를 감지하고 파일 시스템에서 스캔을 실행하며 파일 분석을 위한 대시보드를 제공하는 호스트 기반 소프트웨어입니다. XCP File Analytics는 NetApp 및 타사 시스템과 모두 호환되며 Linux 또는 Windows 호스트에서 실행되어 NFS 및 SMB에서 내보낸 파일 시스템에 대한 분석 기능을 제공합니다.

NetApp ONTAP FlexGroup 볼륨

교육 데이터 세트는 잠재적으로 수십억 개의 파일로 구성됩니다. 파일에는 텍스트, 오디오, 비디오 및 기타 형식의 비정형 데이터가 포함될 수 있으며, 이 데이터를 병렬로 읽고 저장해야 합니다. 스토리지 시스템은 수많은 작은 파일을

저장해야 하며 순차적 I/O 및 랜덤 I/O를 위해 병렬로 이들 파일을 읽어야 합니다

FlexGroup 볼륨은 다음 그림과 같이 여러 개의 구성 멤버 볼륨으로 구성된 단일 네임스페이스입니다. 스토리지 관리자 관점에서 FlexGroup 볼륨은 NetApp FlexVol 볼륨과 마찬가지로 관리되고 작동합니다. FlexGroup 볼륨의 파일은 개별 구성원 볼륨에 할당되며 볼륨 또는 노드에 스트라이핑되지 않습니다. 다음과 같은 기능을 지원합니다.

- FlexGroup 볼륨은 메타데이터가 많은 워크로드에 수 페타바이트에 달하는 용량과 예측 가능한 짧은 지연 시간을 제공합니다.
- 동일한 네임스페이스에서 최대 4천억 개의 파일을 지원합니다.
- CPU, 노드, 애그리게이트, 구성 FlexVol 볼륨에서 NAS 워크로드에 병렬 작업을 지원합니다.



하드웨어 및 소프트웨어 요구 사항

NetApp AI Control Plane 솔루션은 이 특정 하드웨어에 종속되지 않습니다. 이 솔루션은 Trident에서 지원하는 모든 NetApp 물리적 스토리지 어플라이언스, 소프트웨어 정의 인스턴스 또는 클라우드 서비스와 호환됩니다. 예를 들어 NetApp AFF 스토리지 시스템, Azure NetApp Files, NetApp Cloud Volumes Service, NetApp ONTAP Select 소프트웨어 정의 스토리지 인스턴스 또는 NetApp Cloud Volumes ONTAP 인스턴스가 있습니다. 또한, 사용된 Kubernetes 버전이 Kubeflow 및 NetApp Trident에서 지원하는 경우 모든 Kubernetes 클러스터에서 구현할 수 있습니다. Kubeflow에서 지원하는 Kubernetes 버전 목록은 [참조하십시오 "Kubeflow 공식 문서"](#). Trident에서 지원하는 Kubernetes 버전 목록은 [참조하십시오 "Trident 문서"](#). 솔루션을 검증하는 데 사용된 환경에 대한 자세한 내용은 다음 표를 참조하십시오.

인프라 구성 요소	수량	세부 정보	운영 체제
배포 점프 호스트	1	VM	Ubuntu 20.04.2 LTS
Kubernetes 마스터 노드	1	VM	Ubuntu 20.04.2 LTS
Kubernetes 작업자 노드	2	VM	Ubuntu 20.04.2 LTS

인프라 구성 요소	수량	세부 정보	운영 체제
Kubernetes GPU 작업자 노드	2	NVIDIA DGX-1(베어 메탈)	NVIDIA DGX OS 4.0.5(Ubuntu 18.04.2 LTS 기준)
스토리지	1개의 HA 쌍	NetApp AFF A220을 참조하십시오	NetApp ONTAP 9.7 P6

소프트웨어 구성 요소	버전
아파치 기류	2.0.1
Apache Airflow Helm Chart(Apache Airflow 제어 차트)	8.0.8
Docker 를 참조하십시오	2012년 3월 19일
Kubeflow	1.2
쿠버네티스	1.18.9
NetApp 트라이던트	21.01.2
NVIDIA DeepOps	커밋 시 마스터 분기의 Trident 배포 기능 "61898cdfda"버전 21.03의 다른 모든 기능

지원

NetApp은 Apache Airflow, Docker, Kubeflow, Kubernetes 또는 NVIDIA DeepOps에 대한 엔터프라이즈 지원을 제공하지 않습니다. NetApp AI Control Plane 솔루션과 유사한 기능을 갖춘 완벽한 지원 솔루션을 원하는 경우, "[NetApp에 문의하십시오](#)" NetApp이 파트너와 공동으로 제공하는 완전 지원되는 AI/ML 솔루션 정보

Kubernetes 구축

이 섹션에서는 NetApp AI Control Plane 솔루션을 구현하기 위해 Kubernetes 클러스터를 구축하는 데 필요한 작업에 대해 설명합니다. Kubernetes 클러스터가 이미 있는 경우, Kubeflow 및 NetApp Trident에서 지원하는 Kubernetes 버전을 실행 중인 경우 이 섹션을 건너뛸 수 있습니다. Kubeflow에서 지원하는 Kubernetes 버전 목록은 를 참조하십시오 "[Kubeflow 공식 문서](#)". Trident에서 지원하는 Kubernetes 버전 목록은 를 참조하십시오 "[Trident 문서](#)".

NVIDIA GPU를 탑재한 베어 메탈 노드를 포함하는 온프레미스 Kubernetes 배포의 경우 NVIDIA의 DeepOps Kubernetes 배포 도구를 사용하는 것이 좋습니다. 이 섹션에서는 DeepOps를 사용하여 Kubernetes 클러스터 구축에 대해 간략하게 설명합니다.

필수 구성 요소

이 섹션에 요약된 배포 연습을 수행하기 전에 이미 다음 작업을 수행했다고 가정합니다.

- 표준 구성 지침에 따라 베어 메탈 Kubernetes 노드(예: ONTAP AI 포드의 일부인 NVIDIA DGX 시스템)를 이미 구성했습니다.
- 모든 Kubernetes 마스터 및 작업자 노드와 배포 점프 호스트에 지원되는 운영 체제를 설치했습니다. DeepOps에서 지원하는 운영 체제 목록은 를 참조하십시오 "[DeepOps GitHub 사이트](#)".

NVIDIA DeepOps를 사용하여 Kubernetes 설치 및 구성

NVIDIA DeepOps를 사용하여 Kubernetes 클러스터를 구축하고 구성하려면 배포 점프 호스트에서 다음 작업을 수행하십시오.

1. 의 지침에 따라 NVIDIA DeepOps를 다운로드합니다 "[시작 페이지](#)" NVIDIA DeepOps GitHub 사이트에서 다운로드할 수 있습니다.
2. 의 지침에 따라 클러스터에 Kubernetes를 배포합니다 "[Kubernetes 배포 가이드 페이지](#)" NVIDIA DeepOps GitHub 사이트에서 다운로드할 수 있습니다.

NetApp Trident 구축 및 구성

NetApp Trident 구축 및 구성

이 섹션에서는 Kubernetes 클러스터에 NetApp Trident를 설치 및 구성하기 위해 완료해야 하는 작업에 대해 설명합니다.

필수 구성 요소

이 섹션에 요약된 배포 연습을 수행하기 전에 이미 다음 작업을 수행했다고 가정합니다.

1. Kubernetes 작업 클러스터가 이미 있으며, Trident에서 지원하는 Kubernetes 버전을 실행 중입니다. 지원되는 버전 목록은 를 참조하십시오 "[Trident 문서](#)".
2. Trident에서 지원하는 작업 중인 NetApp 스토리지 어플라이언스, 소프트웨어 정의 인스턴스 또는 클라우드 스토리지 서비스가 이미 있습니다.

Trident를 설치합니다

Kubernetes 클러스터에 NetApp Trident를 설치 및 구성하려면 배포 점프 호스트에서 다음 작업을 수행하십시오.

1. 다음 방법 중 하나를 사용하여 Trident를 배포합니다.
 - NVIDIA DeepOps를 사용하여 Kubernetes 클러스터를 구축한 경우, NVIDIA DeepOps를 사용하여 Kubernetes 클러스터에 Trident를 구축할 수도 있습니다. DeepOps를 사용하여 Trident를 배포하려면 을 따릅니다 "[Trident 배포 지침](#)" NVIDIA DeepOps GitHub 사이트에서 다운로드할 수 있습니다.
 - NVIDIA DeepOps를 사용하여 Kubernetes 클러스터를 배포하지 않았거나 Trident를 수동으로 배포하려는 경우 에 따라 Trident를 배포할 수 있습니다 "[배포 지침](#)" Trident 문서 구성 방법에 대한 자세한 내용은 Trident 백엔드와 하나 이상의 Kubernetes StorageClass를 생성해야 합니다 "[백엔드](#)" 및 "[StorageClaes](#)를 참조하십시오" NetApp Docs에서 연결된 하위 섹션을 참조하십시오.



ONTAP AI Pod에 NetApp AI Control Plane 솔루션을 구축하는 경우 를 참조하십시오 "[ONTAP AI 배포에 대한 Trident 백엔드 예](#)" 및 를 생성할 수 있는 다른 Trident 백엔드의 몇 가지 예를 확인하십시오 "[ONTAP AI 배포를 위한 Kubernetes Storagecles의 예](#)" 생성할 수 있는 여러 Kubernetes StorageClasses의 몇 가지 예를 확인하십시오.

ONTAP AI 배포에 대한 Trident 백엔드 예

Trident를 사용하여 Kubernetes 클러스터 내에서 스토리지 리소스를 동적으로 프로비저닝하려면 먼저 하나 이상의 Trident 백엔드를 생성해야 합니다. 다음 예는 ONTAP AI

포드에 NetApp AI Control Plane 솔루션을 구축할 경우 생성할 수 있는 다양한 유형의 백엔드를 보여줍니다. 백엔드에 대한 자세한 내용은 을 참조하십시오 ["Trident 문서"](#).

1. NetApp AFF 시스템에서 사용할 각 데이터 LIF(데이터 액세스를 제공하는 논리적 네트워크 인터페이스)에 대해 FlexGroup 지원 Trident 백엔드를 생성하는 것이 좋습니다. LIF 간 볼륨 마운트의 균형을 조정할 수 있습니다

다음 명령 예는 동일한 ONTAP 스토리지 가상 시스템(SVM)과 연관된 2개의 서로 다른 데이터 LIF에 대해 2개의 FlexGroup 지원 Trident 백엔드를 생성하는 것을 보여줍니다. 이러한 백엔드는 'ONTAP-NAS-Flexgroup' 스토리지 드라이버를 사용합니다. ONTAP는 FlexVol과 FlexGroup의 두 가지 기본 데이터 볼륨 유형을 지원합니다. FlexVol 볼륨의 크기는 제한되어 있습니다(이 쓰기 작업 시 최대 크기는 특정 구축에 따라 다름). 반면 FlexGroup 볼륨은 최대 20PB 및 4천억 개 파일까지 선형적으로 확장할 수 있으므로 데이터 관리를 크게 간소화하는 단일 네임스페이스를 제공합니다. 따라서 FlexGroup 볼륨은 대량의 데이터를 사용하는 AI 및 ML 워크로드에 최적화되어 있습니다.

소량의 데이터로 작업하고 FlexGroup 볼륨 대신 FlexVol 볼륨을 사용하려는 경우, ONTAP-NAS-Flexgroup 스토리지 드라이버 대신 'ONTAP-NAS' 스토리지 드라이버를 사용하는 Trident 백엔드를 생성할 수 있습니다.

```
$ cat << EOF > ./trident-backend-ontap-ai-flexgroups-iface1.json
{
  "version": 1,
  "storageDriverName": "ontap-nas-flexgroup",
  "backendName": "ontap-ai-flexgroups-iface1",
  "managementLIF": "10.61.218.100",
  "dataLIF": "192.168.11.11",
  "svm": "ontapai_nfs",
  "username": "admin",
  "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexgroups-
iface1.json -n trident
+-----+-----+
+-----+-----+-----+
|          NAME          | STORAGE DRIVER |
| UUID                   | STATE  | VOLUMES |
+-----+-----+-----+
+-----+-----+-----+
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online |      0 |
+-----+-----+-----+
+-----+-----+-----+
$ cat << EOF > ./trident-backend-ontap-ai-flexgroups-iface2.json
{
  "version": 1,
  "storageDriverName": "ontap-nas-flexgroup",
  "backendName": "ontap-ai-flexgroups-iface2",
  "managementLIF": "10.61.218.100",
  "dataLIF": "192.168.12.12",
```

```

    "svm": "ontapai_nfs",
    "username": "admin",
    "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexgroups-
iface2.json -n trident
+-----+-----+
+-----+-----+-----+
|          NAME          | STORAGE DRIVER |
UUID                  | STATE  | VOLUMES |
+-----+-----+-----+
+-----+-----+-----+
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online |      0 |
+-----+-----+-----+
+-----+-----+-----+
$ tridentctl get backend -n trident
+-----+-----+
+-----+-----+-----+
|          NAME          | STORAGE DRIVER |
UUID                  | STATE  | VOLUMES |
+-----+-----+-----+
+-----+-----+-----+
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online |      0 |
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online |      0 |
+-----+-----+-----+
+-----+-----+-----+

```

2. 또한 하나 이상의 FlexVol 지원 Trident 백엔드를 생성하는 것이 좋습니다. 데이터 세트 스토리지를 훈련하는 데 FlexGroup 볼륨을 사용하는 경우 FlexVol 볼륨을 사용하여 결과, 출력, 디버그 정보 등을 저장할 수 있습니다. FlexVol 볼륨을 사용하려면 하나 이상의 FlexVol 지원 Trident 백엔드를 생성해야 합니다. 다음 명령의 예는 단일 데이터 LIF를 사용하는 단일 FlexVol 지원 Trident 백엔드를 생성하는 것입니다.

```
$ cat << EOF > ./trident-backend-ontap-ai-flexvols.json
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-ai-flexvols",
  "managementLIF": "10.61.218.100",
  "dataLIF": "192.168.11.11",
  "svm": "ontapai_nfs",
  "username": "admin",
  "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexvols.json -n
trident
+-----+-----+-----+
+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE | VOLUMES | |
+-----+-----+-----+
+-----+-----+-----+
| ontap-ai-flexvols      | ontap-nas      | 52bdb3b1-13a5-4513-
a9c1-52a69657fabe | online | 0 |
+-----+-----+-----+
+-----+-----+-----+
$ tridentctl get backend -n trident
+-----+-----+-----+
+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE | VOLUMES | |
+-----+-----+-----+
+-----+-----+-----+
| ontap-ai-flexvols      | ontap-nas      | 52bdb3b1-13a5-4513-
a9c1-52a69657fabe | online | 0 |
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online | 0 |
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online | 0 |
+-----+-----+-----+
+-----+-----+-----+
```

ONTAP AI 구축을 위한 Kubernetes StorageClasses의 예

Trident를 사용하여 Kubernetes 클러스터 내에서 스토리지 리소스를 동적으로 프로비저닝하려면 먼저 하나 이상의 Kubernetes StorageClasses를 생성해야 합니다. 다음 예제는 ONTAP AI POD에 NetApp AI Control Plane 솔루션을 구축할 경우 생성할 수 있는

다양한 유형의 StorageClasses를 보여줍니다. StorageClasses에 대한 자세한 내용은 [참조하십시오 "Trident 문서"](#).

1. NetApp은 섹션에 생성한 각 FlexGroup 지원 Trident 백엔드에 대해 별도의 StorageClass를 생성할 것을 권장합니다 ["ONTAP AI 배포에 대한 Trident 백엔드 예"](#), 1단계. 이러한 세분화된 StorageClasses를 사용하면 특정 LIF(Trident 백엔드를 생성할 때 지정한 LIF)에 해당하는 NFS 마운트를 StorageClass 사양 파일에 지정된 특정 백엔드에서 추가할 수 있습니다. 다음 예제 명령은 섹션에 생성된 두 예제 백엔드에 해당하는 두 개의 StorageClasses를 생성하는 방법을 보여 줍니다 ["ONTAP AI 배포에 대한 Trident 백엔드 예"](#), 1단계. StorageClasses에 대한 자세한 내용은 [참조하십시오 "Trident 문서"](#).

영구 볼륨은 해당 PersistentVolumeClaim(PVC)이 삭제되어도 삭제되지 않도록 다음 예에서는 "Retain"의 "reclaimPolicy" 값을 사용합니다. '청구 정책' 필드에 대한 자세한 내용은 [공식 을 참조하십시오 "Kubernetes 문서"](#).

```
$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain-iface1.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain-iface1
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
  storagePools: "ontap-ai-flexgroups-iface1:.*"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain-
iface1.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain-iface1 created
$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain-iface2.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain-iface2
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
  storagePools: "ontap-ai-flexgroups-iface2:.*"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain-
iface2.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain-iface2 created
$ kubectl get storageclass
```

NAME	PROVISIONER	AGE
ontap-ai-flexgroups-retain-iface1	netapp.io/trident	0m
ontap-ai-flexgroups-retain-iface2	netapp.io/trident	0m

2. 또한 섹션에서 생성한 FlexVol 지원 Trident 백엔드에 해당하는 StorageClass를 생성하는 것이 좋습니다 ["ONTAP AI 배포에 대한 Trident 백엔드 예"](#), 2단계. 다음 명령 예에서는 FlexVol 볼륨에 대한 단일 StorageClass를 생성하는 것을 보여 줍니다.

다음 예에서는 FlexVol 지원 Trident 백엔드가 하나만 생성되었기 때문에 StorageClass 정의 파일에 특정 백엔드가 지정되지 않습니다. Kubernetes를 사용하여 이 StorageClass를 사용하는 볼륨을 관리할 경우 Trident는 'ONTAP-NAS' 드라이버를 사용하는 사용 가능한 백엔드를 사용하려고 합니다.

```
$ cat << EOF > ./storage-class-ontap-ai-flexvols-retain.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexvols-retain
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexvols-retain.yaml
storageclass.storage.k8s.io/ontap-ai-flexvols-retain created
$ kubectl get storageclass
NAME                                PROVISIONER          AGE
ontap-ai-flexgroups-retain-iface1   netapp.io/trident    1m
ontap-ai-flexgroups-retain-iface2   netapp.io/trident    1m
ontap-ai-flexvols-retain            netapp.io/trident    0m
```

3. 또한 FlexGroup 볼륨에 대한 일반 StorageClass를 생성하는 것이 좋습니다. 다음 예제 명령은 FlexGroup 볼륨에 대한 단일 일반 StorageClass를 생성하는 방법을 보여 줍니다.

StorageClass 정의 파일에 특정 백엔드가 지정되지 않았습니다. 따라서 Kubernetes를 사용하여 이 StorageClass를 사용하는 볼륨을 관리할 때 Trident는 'ONTAP-NAS-Flexgroup' 드라이버를 사용하는 사용 가능한 백엔드를 사용하려고 합니다.

```
$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain created
$ kubectl get storageclass
```

NAME	PROVISIONER	AGE
ontap-ai-flexgroups-retain	netapp.io/trident	0m
ontap-ai-flexgroups-retain-iface1	netapp.io/trident	2m
ontap-ai-flexgroups-retain-iface2	netapp.io/trident	2m
ontap-ai-flexvols-retain	netapp.io/trident	1m

Kubeflow 구축

이 섹션에서는 Kubernetes 클러스터에 Kubeflow를 구축하기 위해 완료해야 하는 작업에 대해 설명합니다.

필수 구성 요소

이 섹션에 요약된 배포 연습을 수행하기 전에 이미 다음 작업을 수행했다고 가정합니다.

1. Kubernetes 작업 클러스터가 이미 있으며, Kubeflow에서 지원하는 Kubernetes 버전을 실행하고 있습니다. 지원되는 버전 목록은 [참조하십시오 "Kubeflow 공식 문서"](#).
2. 에 설명된 대로 Kubernetes 클러스터에 NetApp Trident를 이미 설치 및 구성했습니다 ["Trident 구축 및 구성"](#).

기본 Kubernetes StorageClass를 설정합니다

Kubeflow를 구현하기 전에 Kubernetes 클러스터 내에서 기본 StorageClass를 지정해야 합니다. Kubeflow 배포 프로세스에서는 기본 StorageClass를 사용하여 새 영구 볼륨의 프로비저닝을 시도합니다. 기본 StorageClass로 지정된 StorageClass가 없으면 배포가 실패합니다. 클러스터 내에서 기본 StorageClass를 지정하려면 배포 점프 호스트에서 다음 작업을 수행합니다. 클러스터 내에서 기본 StorageClass를 이미 지정한 경우에는 이 단계를 건너뛸 수 있습니다.

1. 기존 StorageClasses 중 하나를 기본 StorageClass로 지정합니다. 다음 명령을 실행하면 기본 StorageClass로 ONTAP-ai-FlexVols-Retain이라는 StorageClass가 지정됩니다.



ONTAP-NAS-Flexgroup Trident 백엔드 유형은 PVC 크기가 매우 큼니다. 기본적으로 Kubeflow는 크기가 몇 GB인 PVC를 프로비저닝하려고 시도합니다. 따라서 Kubeflow 구축을 위해 "ONTAP-NAS-flexgroup" 백엔드 유형을 기본 StorageClass로 사용하는 StorageClass를 지정할 수 없습니다.


```
$ kubectl get sc
NAME                                PROVISIONER                        AGE
ontap-ai-flexgroups-retain         csi.trident.netapp.io            25h
ontap-ai-flexgroups-retain-iface1  csi.trident.netapp.io            25h
ontap-ai-flexgroups-retain-iface2  csi.trident.netapp.io            25h
ontap-ai-flexvols-retain           csi.trident.netapp.io            3s
$ kubectl patch storageclass ontap-ai-flexvols-retain -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
storageclass.storage.k8s.io/ontap-ai-flexvols-retain patched
$ kubectl get sc
NAME                                PROVISIONER                        AGE
ontap-ai-flexgroups-retain         csi.trident.netapp.io            25h
ontap-ai-flexgroups-retain-iface1  csi.trident.netapp.io            25h
ontap-ai-flexgroups-retain-iface2  csi.trident.netapp.io            25h
ontap-ai-flexvols-retain (default) csi.trident.netapp.io            54s
```

NVIDIA DeepOps를 사용하여 Kubeflow를 배포합니다

NVIDIA DeepOps에서 제공하는 Kubeflow 구현 툴을 사용할 것을 권장합니다. DeepOps 구축 툴을 사용하여 Kubernetes 클러스터에 Kubeflow를 배포하려면 배포 점프 호스트에서 다음 작업을 수행합니다.



또는 에 따라 Kubeflow를 수동으로 배포할 수도 있습니다 ["설치 지침"](#) 공식 Kubeflow 문서에서 제공됩니다

1. 에 따라 클러스터에 Kubeflow를 구현합니다 ["Kubeflow 구축 지침"](#) NVIDIA DeepOps GitHub 사이트에서 다운로드할 수 있습니다.
2. DeepOps Kubeflow 구현 도구에서 출력하는 Kubeflow 대시보드 URL을 기록합니다.

```
$ ./scripts/k8s/deploy_kubeflow.sh -x
...
INFO[0007] Applied the configuration Successfully!
filename="cmd/apply.go:72"
Kubeflow app installed to: /home/ai/kubeflow
It may take several minutes for all services to start. Run 'kubectl get
pods -n kubeflow' to verify
To remove (excluding CRDs, istio, auth, and cert-manager), run:
./scripts/k8s_deploy_kubeflow.sh -d
To perform a full uninstall : ./scripts/k8s_deploy_kubeflow.sh -D
Kubeflow Dashboard (HTTP NodePort): http://10.61.188.111:31380
```

3. Kubeflow 네임스페이스 내에 배포된 모든 Pod에 'Running'이라는 'Status'가 표시되는지 확인하고 네임스페이스 내에 배포된 구성 요소가 오류 상태에 있지 않은지 확인합니다. 모든 Pod를 시작하는 데 몇 분 정도 걸릴 수 있습니다.

```
$ kubectl get all -n kubeflow
```

NAME			READY
STATUS	RESTARTS	AGE	
pod/admission-webhook-bootstrap-stateful-set-0			1/1
Running	0	95s	
pod/admission-webhook-deployment-6b89c84c98-vrtbh			1/1
Running	0	91s	
pod/application-controller-stateful-set-0			1/1
Running	0	98s	
pod/argo-ui-5dcf5d8b4f-m2wn4			1/1
Running	0	97s	
pod/centraldashboard-cf4874ddc-7hcr8			1/1
Running	0	97s	
pod/jupyter-web-app-deployment-685b455447-gjhh7			1/1
Running	0	96s	
pod/katib-controller-88c97d85c-kgq66			1/1
Running	1	95s	
pod/katib-db-8598468fd8-5jw2c			1/1
Running	0	95s	
pod/katib-manager-574c8c67f9-wtrf5			1/1
Running	1	95s	
pod/katib-manager-rest-778857c989-fjbzn			1/1
Running	0	95s	
pod/katib-suggestion-bayesianoptimization-65df4d7455-qthmw			1/1
Running	0	94s	
pod/katib-suggestion-grid-56bf69f597-98vwn			1/1
Running	0	94s	
pod/katib-suggestion-hyperband-7777b76cb9-9v6dq			1/1
Running	0	93s	
pod/katib-suggestion-nasrl-77f6f9458c-2qzxq			1/1
Running	0	93s	
pod/katib-suggestion-random-77b88b5c79-l64j9			1/1
Running	0	93s	
pod/katib-ui-7587c5b967-nd629			1/1
Running	0	95s	
pod/metacontroller-0			1/1
Running	0	96s	
pod/metadata-db-5dd459cc-swzkm			1/1
Running	0	94s	
pod/metadata-deployment-6cf77db994-69fk7			1/1
Running	3	93s	
pod/metadata-deployment-6cf77db994-mpbjt			1/1
Running	3	93s	
pod/metadata-deployment-6cf77db994-xg7tz			1/1
Running	3	94s	
pod/metadata-ui-78f5b59b56-qb6kr			1/1

```

Running    0          94s
pod/minio-758b769d67-1lvdr                      1/1
Running    0          91s
pod/ml-pipeline-5875b9db95-g8t2k                1/1
Running    0          91s
pod/ml-pipeline-persistenceagent-9b69ddd46-bt9r9 1/1
Running    0          90s
pod/ml-pipeline-scheduledworkflow-7b8d756c76-7x56s 1/1
Running    0          90s
pod/ml-pipeline-ui-79ffd9c76-fcwpd              1/1
Running    0          90s
pod/ml-pipeline-viewer-controller-deployment-5fdc87f58-b2t9r 1/1
Running    0          90s
pod/mysql-657f87857d-15k9z                      1/1
Running    0          91s
pod/notebook-controller-deployment-56b4f59bbf-8bvnr 1/1
Running    0          92s
pod/profiles-deployment-6bc745947-mrdkh          2/2
Running    0          90s
pod/pytorch-operator-77c97f4879-hmlrv           1/1
Running    0          92s
pod/seldon-operator-controller-manager-0         1/1
Running    1          91s
pod/spartakus-volunteer-5fdfddb779-17qkm        1/1
Running    0          92s
pod/tensorboard-6544748d94-nh8b2                1/1
Running    0          92s
pod/tf-job-dashboard-56f79c59dd-6w59t           1/1
Running    0          92s
pod/tf-job-operator-79cbfd6dbc-rb58c            1/1
Running    0          91s
pod/workflow-controller-db644d554-cwrnb          1/1
Running    0          97s

```

NAME	TYPE
CLUSTER-IP	AGE
service/admission-webhook-service	ClusterIP
10.233.51.169 <none> 443/TCP	97s
service/application-controller-service	ClusterIP
10.233.4.54 <none> 443/TCP	98s
service/argo-ui	NodePort
10.233.47.191 <none> 80:31799/TCP	97s
service/centraldashboard	ClusterIP
10.233.8.36 <none> 80/TCP	97s
service/jupyter-web-app-service	ClusterIP
10.233.1.42 <none> 80/TCP	97s
service/katib-controller	ClusterIP

10.233.25.226	<none>	443/TCP	96s
service/katib-db			ClusterIP
10.233.33.151	<none>	3306/TCP	97s
service/katib-manager			ClusterIP
10.233.46.239	<none>	6789/TCP	96s
service/katib-manager-rest			ClusterIP
10.233.55.32	<none>	80/TCP	96s
service/katib-suggestion-bayesianoptimization			ClusterIP
10.233.49.191	<none>	6789/TCP	95s
service/katib-suggestion-grid			ClusterIP
10.233.9.105	<none>	6789/TCP	95s
service/katib-suggestion-hyperband			ClusterIP
10.233.22.2	<none>	6789/TCP	95s
service/katib-suggestion-nasrl			ClusterIP
10.233.63.73	<none>	6789/TCP	95s
service/katib-suggestion-random			ClusterIP
10.233.57.210	<none>	6789/TCP	95s
service/katib-ui			ClusterIP
10.233.6.116	<none>	80/TCP	96s
service/metadata-db			ClusterIP
10.233.31.2	<none>	3306/TCP	96s
service/metadata-service			ClusterIP
10.233.27.104	<none>	8080/TCP	96s
service/metadata-ui			ClusterIP
10.233.57.177	<none>	80/TCP	96s
service/minio-service			ClusterIP
10.233.44.90	<none>	9000/TCP	94s
service/ml-pipeline			ClusterIP
10.233.41.201	<none>	8888/TCP, 8887/TCP	94s
service/ml-pipeline-tensorboard-ui			ClusterIP
10.233.36.207	<none>	80/TCP	93s
service/ml-pipeline-ui			ClusterIP
10.233.61.150	<none>	80/TCP	93s
service/mysql			ClusterIP
10.233.55.117	<none>	3306/TCP	94s
service/notebook-controller-service			ClusterIP
10.233.10.166	<none>	443/TCP	95s
service/profiles-kfam			ClusterIP
10.233.33.79	<none>	8081/TCP	92s
service/pytorch-operator			ClusterIP
10.233.37.112	<none>	8443/TCP	95s
service/seldon-operator-controller-manager-service			ClusterIP
10.233.30.178	<none>	443/TCP	92s
service/tensorboard			ClusterIP
10.233.58.151	<none>	9000/TCP	94s
service/tf-job-dashboard			ClusterIP

```

10.233.4.17      <none>          80/TCP          94s
service/tf-job-operator          ClusterIP
10.233.60.32     <none>          8443/TCP        94s
service/webhook-server-service   ClusterIP
10.233.32.167    <none>          443/TCP         87s
NAME                                                    READY    UP-
TO-DATE    AVAILABLE    AGE
deployment.apps/admission-webhook-deployment          1/1      1
1           97s
deployment.apps/argo-ui                               1/1      1
1           97s
deployment.apps/centraldashboard                     1/1      1
1           97s
deployment.apps/jupyter-web-app-deployment            1/1      1
1           97s
deployment.apps/katib-controller                     1/1      1
1           96s
deployment.apps/katib-db                             1/1      1
1           97s
deployment.apps/katib-manager                       1/1      1
1           96s
deployment.apps/katib-manager-rest                   1/1      1
1           96s
deployment.apps/katib-suggestion-bayesianoptimization 1/1      1
1           95s
deployment.apps/katib-suggestion-grid                1/1      1
1           95s
deployment.apps/katib-suggestion-hyperband           1/1      1
1           95s
deployment.apps/katib-suggestion-nasrl               1/1      1
1           95s
deployment.apps/katib-suggestion-random              1/1      1
1           95s
deployment.apps/katib-ui                             1/1      1
1           96s
deployment.apps/metadata-db                          1/1      1
1           96s
deployment.apps/metadata-deployment                  3/3      3
3           96s
deployment.apps/metadata-ui                          1/1      1
1           96s
deployment.apps/minio                                1/1      1
1           94s
deployment.apps/ml-pipeline                          1/1      1
1           94s
deployment.apps/ml-pipeline-persistenceagent         1/1      1

```

1	93s			
deployment.apps/ml-pipeline-scheduledworkflow		1/1		1
1	93s			
deployment.apps/ml-pipeline-ui		1/1		1
1	93s			
deployment.apps/ml-pipeline-viewer-controller-deployment		1/1		1
1	93s			
deployment.apps/mysql		1/1		1
1	94s			
deployment.apps/notebook-controller-deployment		1/1		1
1	95s			
deployment.apps/profiles-deployment		1/1		1
1	92s			
deployment.apps/pytorch-operator		1/1		1
1	95s			
deployment.apps/spartakus-volunteer		1/1		1
1	94s			
deployment.apps/tensorboard		1/1		1
1	94s			
deployment.apps/tf-job-dashboard		1/1		1
1	94s			
deployment.apps/tf-job-operator		1/1		1
1	94s			
deployment.apps/workflow-controller		1/1		1
1	97s			
NAME				
DESIRED	CURRENT	READY	AGE	
replicaset.apps/admission-webhook-deployment-6b89c84c98				1
1	1	97s		
replicaset.apps/argo-ui-5dcf5d8b4f				1
1	1	97s		
replicaset.apps/centraldashboard-cf4874ddc				1
1	1	97s		
replicaset.apps/jupyter-web-app-deployment-685b455447				1
1	1	97s		
replicaset.apps/katib-controller-88c97d85c				1
1	1	96s		
replicaset.apps/katib-db-8598468fd8				1
1	1	97s		
replicaset.apps/katib-manager-574c8c67f9				1
1	1	96s		
replicaset.apps/katib-manager-rest-778857c989				1
1	1	96s		
replicaset.apps/katib-suggestion-bayesianoptimization-65df4d7455				1
1	1	95s		
replicaset.apps/katib-suggestion-grid-56bf69f597				1

1	1	95s		
replicaset.apps/katib-suggestion-hyperband-7777b76cb9			1	
1	1	95s		
replicaset.apps/katib-suggestion-nasrl-77f6f9458c			1	
1	1	95s		
replicaset.apps/katib-suggestion-random-77b88b5c79			1	
1	1	95s		
replicaset.apps/katib-ui-7587c5b967			1	
1	1	96s		
replicaset.apps/metadata-db-5dd459cc			1	
1	1	96s		
replicaset.apps/metadata-deployment-6cf77db994			3	
3	3	96s		
replicaset.apps/metadata-ui-78f5b59b56			1	
1	1	96s		
replicaset.apps/minio-758b769d67			1	
1	1	93s		
replicaset.apps/ml-pipeline-5875b9db95			1	
1	1	93s		
replicaset.apps/ml-pipeline-persistenceagent-9b69ddd46			1	
1	1	92s		
replicaset.apps/ml-pipeline-scheduledworkflow-7b8d756c76			1	
1	1	91s		
replicaset.apps/ml-pipeline-ui-79ffd9c76			1	
1	1	91s		
replicaset.apps/ml-pipeline-viewer-controller-deployment-5fdc87f58			1	
1	1	91s		
replicaset.apps/mysql-657f87857d			1	
1	1	92s		
replicaset.apps/notebook-controller-deployment-56b4f59bbf			1	
1	1	94s		
replicaset.apps/profiles-deployment-6bc745947			1	
1	1	91s		
replicaset.apps/pytorch-operator-77c97f4879			1	
1	1	94s		
replicaset.apps/spartakus-volunteer-5fdfd9b779			1	
1	1	94s		
replicaset.apps/tensorboard-6544748d94			1	
1	1	93s		
replicaset.apps/tf-job-dashboard-56f79c59dd			1	
1	1	93s		
replicaset.apps/tf-job-operator-79cbfd6dbc			1	
1	1	93s		
replicaset.apps/workflow-controller-db644d554			1	
1	1	97s		
NAME			READY	AGE

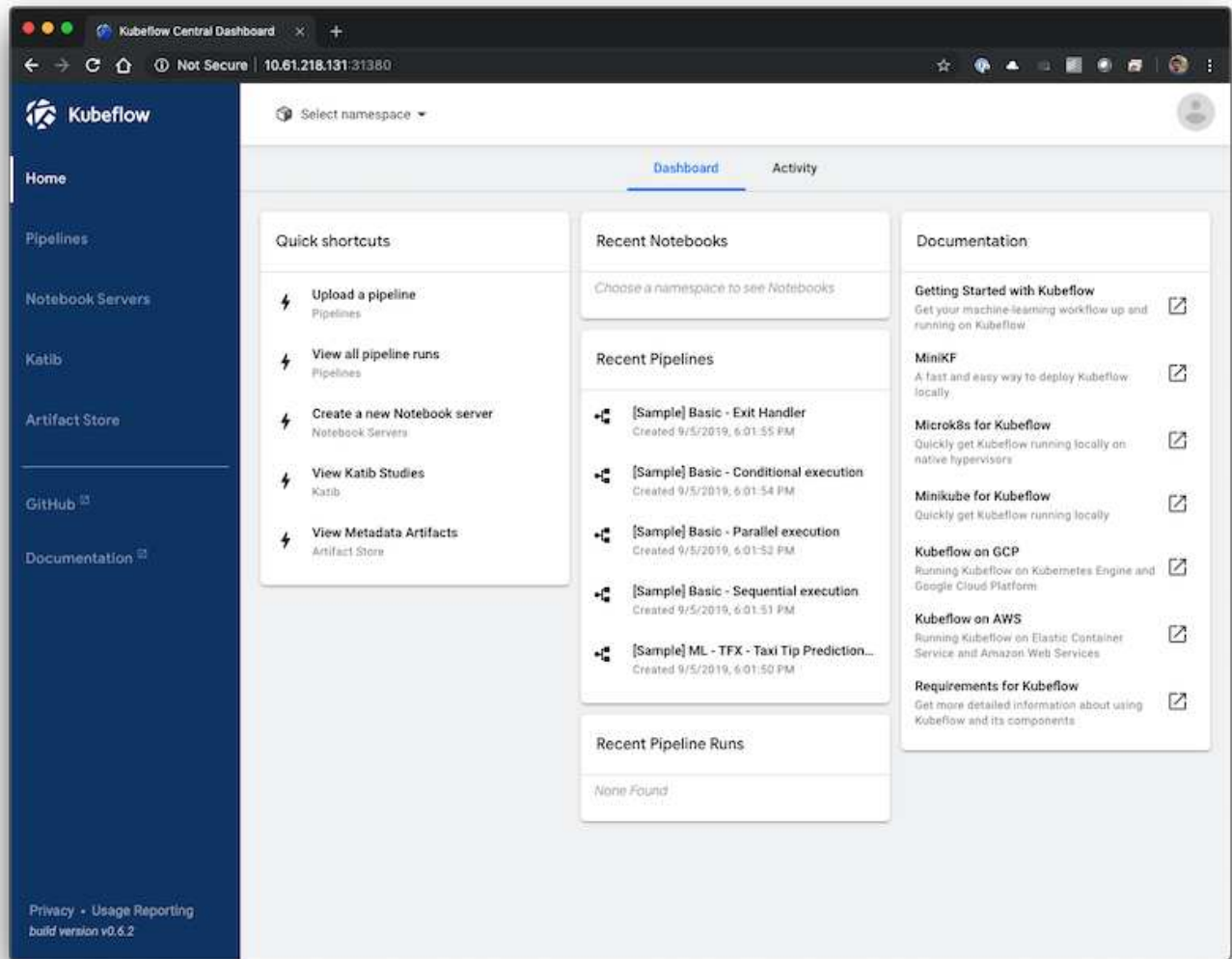
```

statefulset.apps/admission-webhook-bootstrap-stateful-set 1/1 97s
statefulset.apps/application-controller-stateful-set 1/1 98s
statefulset.apps/metacontroller 1/1 98s
statefulset.apps/seldon-operator-controller-manager 1/1 92s
$ kubectl get pvc -n kubeflow
NAME                                STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS          AGE
katib-mysql                         Bound   pvc-b07f293e-d028-11e9-9b9d-00505681a82d  10Gi      RWO            ontap-ai-flexvols-retain 27m
metadata-mysql                     Bound   pvc-b0f3f032-d028-11e9-9b9d-00505681a82d  10Gi      RWO            ontap-ai-flexvols-retain 27m
minio-pv-claim                     Bound   pvc-b22727ee-d028-11e9-9b9d-00505681a82d  20Gi      RWO            ontap-ai-flexvols-retain 27m
mysql-pv-claim                     Bound   pvc-b2429afd-d028-11e9-9b9d-00505681a82d  20Gi      RWO            ontap-ai-flexvols-retain 27m

```

4. 웹 브라우저에서 2단계에서 기록해 둔 URL로 이동하여 Kubeflow 중앙 대시보드에 액세스합니다.

기본 사용자 이름은 admin@kubeflow.org, 기본 암호는 12341234입니다. 추가 사용자를 생성하려면 [의 지침을](#) 따르십시오 "[Kubeflow 공식 문서](#)".



Kubeflow 작업 및 작업 예

이 섹션에서는 Kubeflow를 사용하여 수행할 수 있는 다양한 작업 및 작업의 예를 제공합니다.

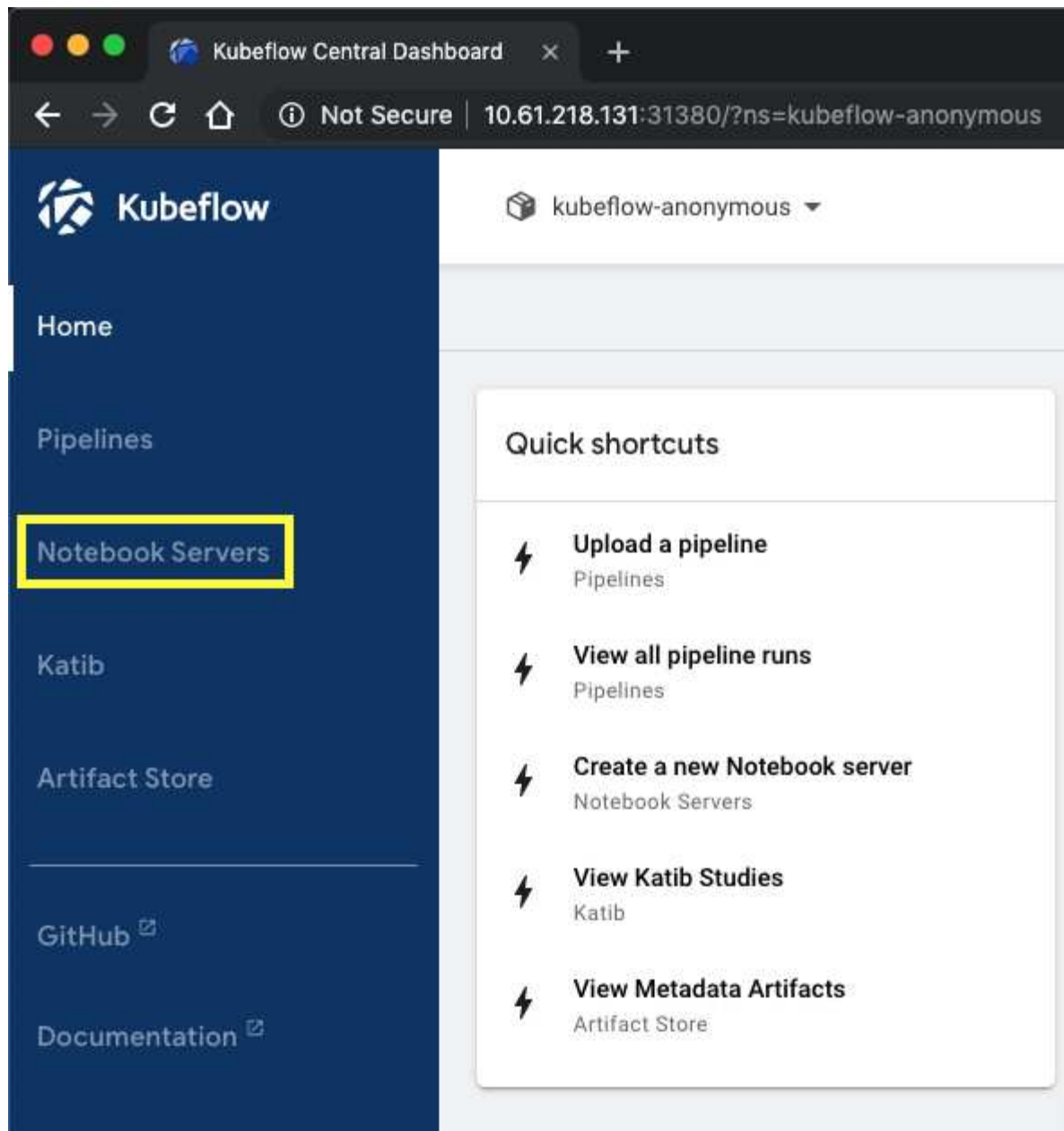
Kubeflow 작업 및 작업 예

이 섹션에서는 Kubeflow를 사용하여 수행할 수 있는 다양한 작업 및 작업의 예를 제공합니다.

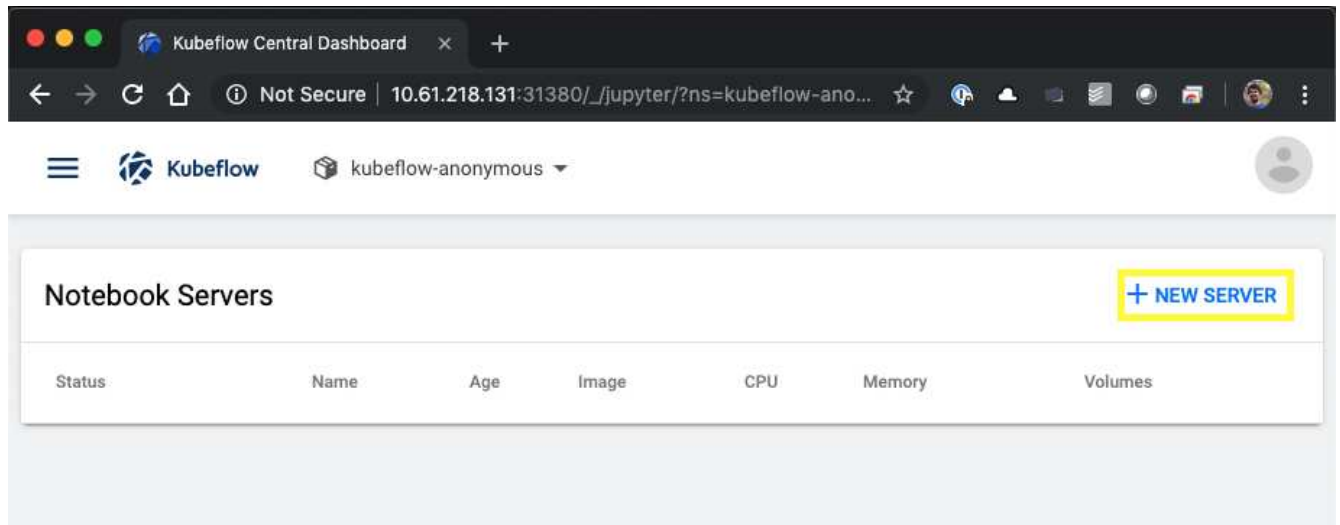
데이터 과학자 또는 개발자 사용을 위한 **Jupyter Notebook Workspace**를 제공합니다

Kubeflow는 새로운 Jupyter Notebook 서버를 신속하게 프로비저닝하여 데이터 과학자 작업 공간 역할을 할 수 있습니다. Kubeflow와 함께 새로운 Jupyter Notebook 서버를 프로비저닝하려면 다음 작업을 수행합니다. Kubeflow 컨텍스트 내의 Jupyter Notebooks에 대한 자세한 내용은 ["Kubeflow 공식 문서"](#)를 참조하십시오.

1. Kubeflow 중앙 대시보드에서 기본 메뉴의 Notebook Servers를 클릭하여 Jupyter Notebook 서버 관리 페이지로 이동합니다.



2. New Server를 클릭하여 새 Jupyter Notebook 서버를 프로비저닝합니다.



3. 새 서버의 이름을 지정하고, 서버를 기반으로 할 Docker 이미지를 선택한 다음, 서버에서 예약할 CPU와 RAM의 양을 지정합니다. 네임스페이스 필드가 비어 있는 경우 페이지 머릿글의 네임스페이스 선택 메뉴를 사용하여 네임스페이스를 선택합니다. 그러면 Namespace 필드가 선택한 네임스페이스로 자동 채워집니다.

다음 예에서는 kubeflow-anonymous 네임스페이스가 선택됩니다. 또한 Docker 이미지, CPU 및 RAM에 대한 기본값을 사용할 수 있습니다.

Name

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

Name: Namespace:

Image

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

☐ Custom Image

Image:

CPU / RAM

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

CPU: Memory:

- 작업 공간 볼륨 세부 정보를 지정합니다. 새 볼륨을 생성하기로 선택한 경우 기본 StorageClass를 사용하여 해당 볼륨 또는 PVC가 프로비저닝됩니다. Trident를 사용하는 StorageClass가 섹션에서 기본 StorageClass로 지정되었기 때문입니다. "Kubeflow 구축", 볼륨 또는 PVC는 Trident를 사용하여 프로비저닝됩니다. 이 볼륨은 Jupyter Notebook Server 컨테이너 내의 기본 작업 공간으로 자동으로 마운트됩니다. 사용자가 서버에서 별도의 데이터 볼륨에 저장되지 않은 전자 필기장은 이 작업 영역 볼륨에 자동으로 저장됩니다. 따라서 재부팅 시에도 노트북이 유지됩니다.

Workspace Volume

Configure the Volume to be mounted as your personal Workspace.

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

Type: Name: Size: Mode: Mount Point:

- 데이터 볼륨을 추가합니다. 다음 예에서는 'PB-FG-ALL'이라는 기존 PVC를 지정하고 기본 마운트 지점을 적용합니다.

Data Volumes

Configure the Volumes to be mounted as your Datasets.

[+ ADD VOLUME](#)

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

6. * 선택 사항: * 원하는 수의 GPU를 노트북 서버에 할당하도록 요청합니다. 다음 예에서는 GPU 1개가 요청됩니다.

Configurations

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

Configurations

Extra Resources

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

☒ Enable Shared Memory

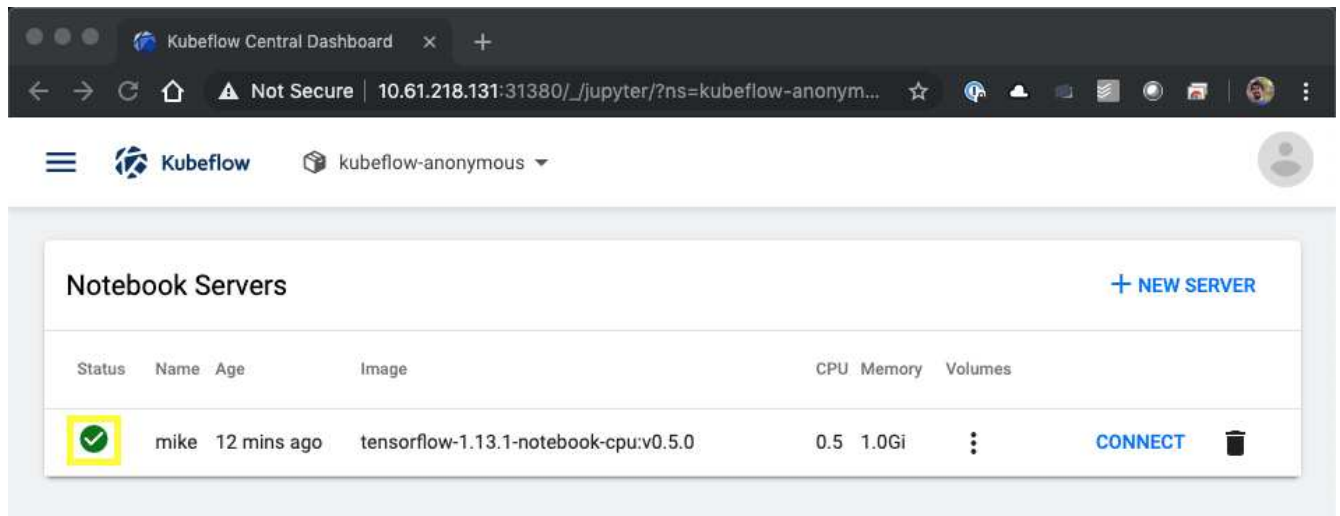
Extra Resources *

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

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

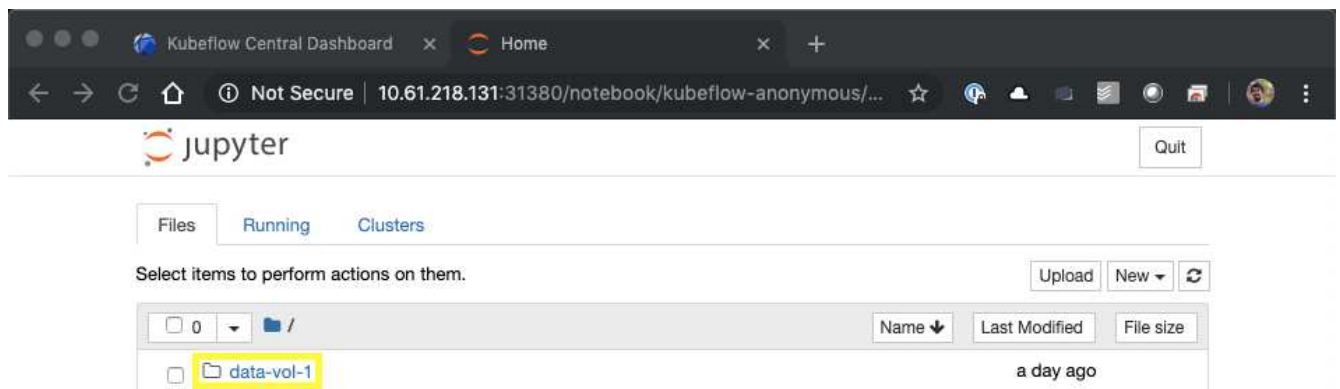
[LAUNCH](#) [CANCEL](#)

- 시작 을 클릭하여 새 노트북 서버를 프로비저닝합니다.
- 노트북 서버가 완전히 준비될 때까지 기다립니다. 지정한 Docker 이미지를 사용하여 서버를 프로비저닝하지 않은 경우 이미지를 다운로드해야 하므로 몇 분 정도 걸릴 수 있습니다. 서버가 완전히 프로비저닝되면 Jupyter Notebook 서버 관리 페이지의 상태 열에 녹색 확인 표시가 나타납니다.



9. 연결을 클릭하여 새 서버 웹 인터페이스에 연결합니다.

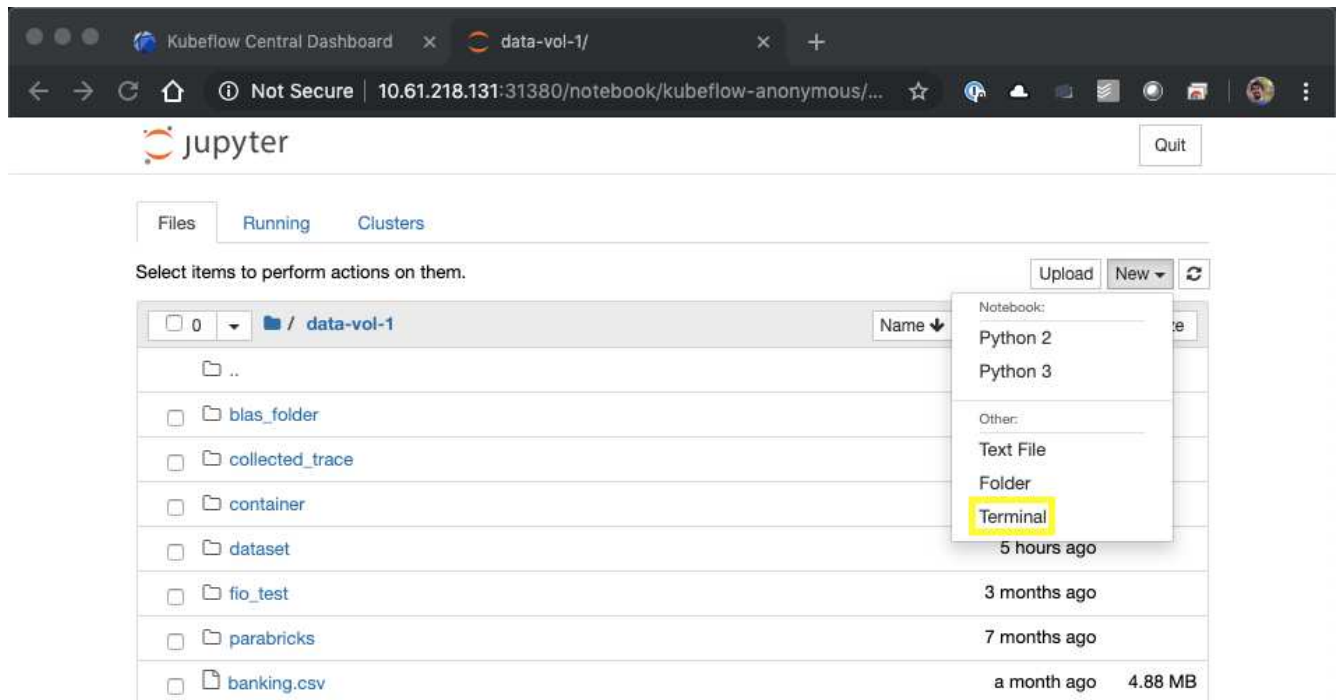
10. 6단계에서 지정한 데이터 세트 볼륨이 서버에 마운트되었는지 확인합니다. 이 볼륨은 기본적으로 기본 작업 공간 내에 마운트됩니다. 사용자의 관점에서 이것은 작업 영역 내의 다른 폴더일 뿐입니다. 인프라 전문가가 아닌 데이터 과학자인 사용자는 이 볼륨을 사용하기 위해 스토리지 전문 지식을 보유할 필요가 없습니다.





11. 터미널을 열고 5단계에서 새 볼륨이 요청되었다고 가정하고 `df -h`를 실행하여 새로운 Trident 제공 영구 볼륨이 기본 작업 공간으로 마운트되었는지 확인합니다.

기본 작업 공간 디렉터리는 서버의 웹 인터페이스에 처음 액세스할 때 표시되는 기본 디렉터리입니다. 따라서 웹 인터페이스를 사용하여 생성한 아티팩트는 Trident가 제공하는 영구 볼륨에 저장됩니다.



```

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

```

12. 터미널을 사용하여 NVIDIA-SMI를 실행하여 올바른 개수의 GPU가 노트북 서버에 할당되었는지 확인합니다. 다음 예에서는 7단계에서 요청했던 대로 하나의 GPU가 노트북 서버에 할당되었습니다.

```

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

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

```


를 클릭합니다 "[Kubernetes용 NetApp 데이터 과학 툴킷](#)" Kubeflow와 함께 사용할 수 있습니다. Kubeflow와 함께 NetApp Data Science Toolkit을 사용하면 다음과 같은 이점이 있습니다.

- 데이터 과학자는 Jupyter Notebook 내에서 직접 고급 NetApp 데이터 관리 작업을 수행할 수 있습니다.
- 고급 NetApp 데이터 관리 작업은 Kubeflow 파이프라인 프레임워크를 사용하여 자동화된 워크플로우에 통합할 수 있습니다.

을 참조하십시오 "[Kubeflow 예](#)" Kubeflow 기반 툴킷 사용에 대한 자세한 내용은 NetApp Data Science Toolkit GitHub 리포지토리 를 참조하십시오.

Apache Airflow Deployment

Kubernetes에서 Apache Airflow를 실행하는 것이 좋습니다. 이 섹션에서는 Kubernetes 클러스터에 공기 흐름을 구축하기 위해 완료해야 하는 작업에 대해 설명합니다.



Kubernetes 이외의 플랫폼에 공기 흐름을 배포할 수 있습니다. Kubernetes가 아닌 다른 플랫폼에 공기 흐름을 배포하는 것은 이 솔루션의 범위를 벗어납니다.

필수 구성 요소

이 섹션에 요약된 배포 연습을 수행하기 전에 이미 다음 작업을 수행했다고 가정합니다.

1. Kubernetes 클러스터 작업이 이미 진행 중입니다.
2. "NetApp Trident 배포 및 구성" 섹션에 설명된 대로 Kubernetes 클러스터에 NetApp Trident를 이미 설치 및 구성했습니다.

제어 장치를 설치합니다

Kubernetes의 유명 패키지 매니저인 Helm을 사용하여 공기 흐름을 구축합니다. 공기 흐름을 배치하기 전에 배포 점프 호스트에 Helm을 설치해야 합니다. 배포 점프 호스트에 Helm을 설치하려면 에 따르십시오 "[설치 지침](#)" 공식 Helm 문서.

기본 Kubernetes StorageClass를 설정합니다

공기 흐름을 구축하기 전에 Kubernetes 클러스터 내에서 기본 StorageClass를 지정해야 합니다. Airflow 배포 프로세스는 기본 StorageClass를 사용하여 새 영구 볼륨의 프로비저닝을 시도합니다. 기본 StorageClass로 지정된 StorageClass가 없으면 배포가 실패합니다. 클러스터 내에서 기본 StorageClass를 지정하려면 섹션에 설명된 지침을 따르십시오 "[Kubeflow 구축](#)". 클러스터 내에서 기본 StorageClass를 이미 지정한 경우에는 이 단계를 건너뛸 수 있습니다.

Helm을 사용하여 공기 흐름을 전개하십시오

Helm을 사용하여 Kubernetes 클러스터에 공기 흐름을 배포하려면 배포 점프 호스트에서 다음 작업을 수행하십시오.

1. 에 따라 헬름으로 공기 흐름을 전개하십시오 "[배포 지침](#)" Artifact Hub의 공식 공기 흐름 도표입니다. 다음 명령 예는 Helm을 사용한 공기 흐름의 배치를 보여줍니다. 환경과 원하는 구성에 따라 필요에 따라 'custom-values.yaml' 파일에서 값을 수정, 추가 및/또는 제거합니다.

```

$ cat << EOF > custom-values.yaml
#####
# Airflow - Common Configs
#####
airflow:
  ## the airflow executor type to use
  ##
  executor: "CeleryExecutor"
  ## environment variables for the web/scheduler/worker Pods (for
airflow configs)
  ##
  #
#####
# Airflow - WebUI Configs
#####
web:
  ## configs for the Service of the web Pods
  ##
  service:
    type: NodePort
#####
# Airflow - Logs Configs
#####
logs:
  persistence:
    enabled: true
#####
# Airflow - DAGs Configs
#####
dags:
  ## configs for the DAG git repository & sync container
  ##
  gitSync:
    enabled: true
    ## url of the git repository
    ##
    repo: "git@github.com:mboglesby/airflow-dev.git"
    ## the branch/tag/sha1 which we clone
    ##
    branch: master
    revision: HEAD
    ## the name of a pre-created secret containing files for ~/.ssh/
    ##
    ## NOTE:
    ## - this is ONLY RELEVANT for SSH git repos
    ## - the secret commonly includes files: id_rsa, id_rsa.pub,

```

```

known_hosts
## - known_hosts is NOT NEEDED if `git.sshKeyscan` is true
##
sshSecret: "airflow-ssh-git-secret"
## the name of the private key file in your `git.secret`
##
## NOTE:
## - this is ONLY RELEVANT for PRIVATE SSH git repos
##
sshSecretKey: id_rsa
## the git sync interval in seconds
##
syncWait: 60
EOF
$ helm install airflow airflow-stable/airflow -n airflow --version 8.0.8
--values ./custom-values.yaml
...
Congratulations. You have just deployed Apache Airflow!
1. Get the Airflow Service URL by running these commands:
    export NODE_PORT=$(kubectl get --namespace airflow -o
jsonpath="{.spec.ports[0].nodePort}" services airflow-web)
    export NODE_IP=$(kubectl get nodes --namespace airflow -o
jsonpath="{.items[0].status.addresses[0].address}")
    echo http://$NODE_IP:$NODE_PORT/
2. Open Airflow in your web browser

```

2. 모든 공기 흐름 포드가 실행 중인지 확인합니다. 모든 Pod를 시작하는 데 몇 분 정도 걸릴 수 있습니다.

```

$ kubectl -n airflow get pod

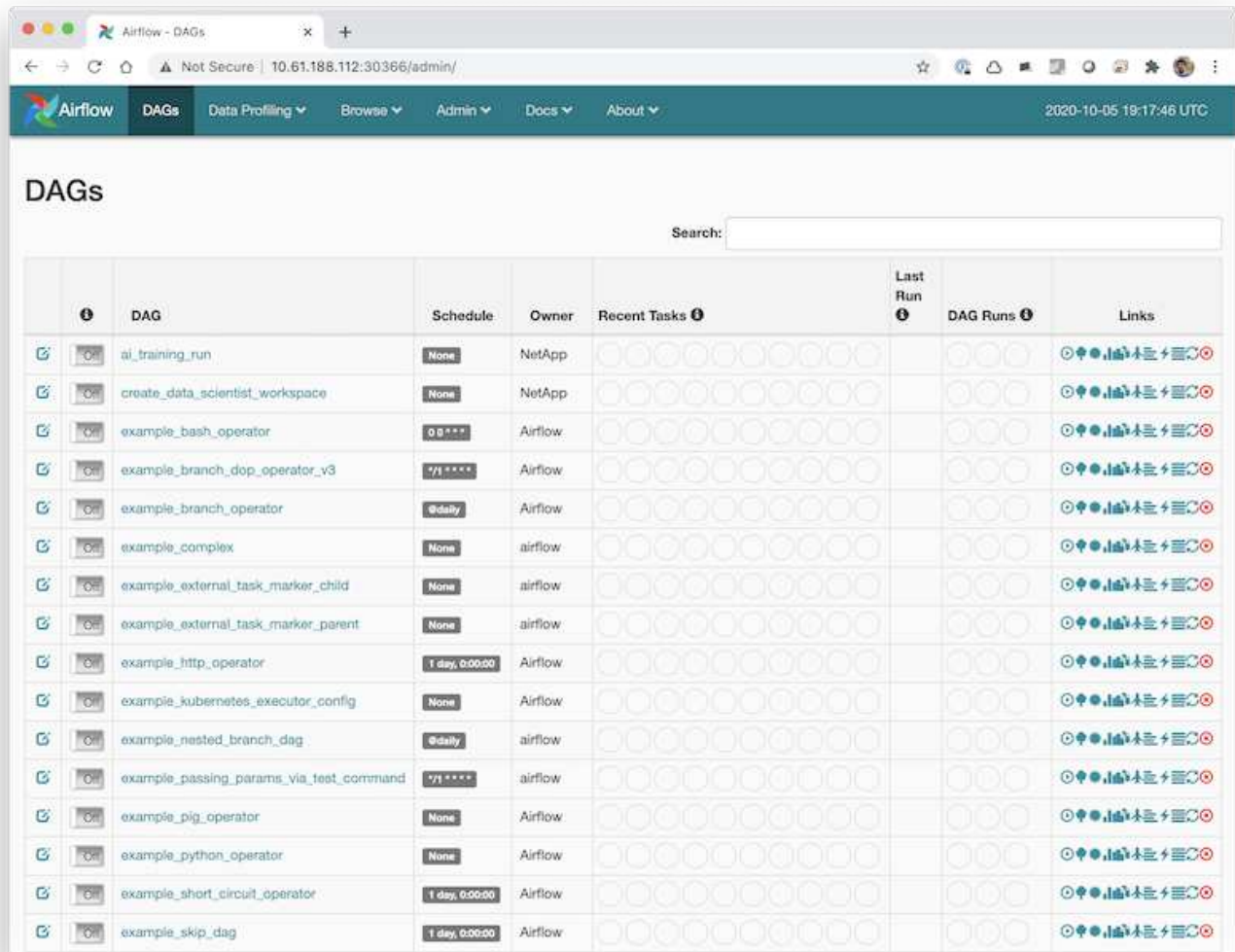
```

NAME	READY	STATUS	RESTARTS	AGE
airflow-flower-b5656d44f-h8qjk	1/1	Running	0	2h
airflow-postgresql-0	1/1	Running	0	2h
airflow-redis-master-0	1/1	Running	0	2h
airflow-scheduler-9d95fcd9-clf4b	2/2	Running	2	2h
airflow-web-59c94db9c5-z7rg4	1/1	Running	0	2h
airflow-worker-0	2/2	Running	2	2h

3. 1단계에서 제어 장치를 사용하여 공기 흐름을 배포할 때 콘솔에 인쇄된 지침에 따라 공기 흐름 웹 서비스 URL을 얻습니다.

```
$ export NODE_PORT=$(kubectl get --namespace airflow -o
jsonpath="{.spec.ports[0].nodePort}" services airflow-web)
$ export NODE_IP=$(kubectl get nodes --namespace airflow -o
jsonpath="{.items[0].status.addresses[0].address}")
$ echo http://$NODE_IP:$NODE_PORT/
```

4. Airflow 웹 서비스에 액세스할 수 있는지 확인합니다.



Apache Airflow 워크플로의 예

를 클릭합니다 **"Kubernetes용 NetApp 데이터 과학 툴킷"** 공기 흐름과 함께 사용할 수 있습니다. 공기 흐름이 원활한 NetApp 데이터 과학 툴킷을 사용하면 NetApp 데이터 관리 작업을 공기 흐름으로 조율되는 자동화된 워크플로우에 통합할 수 있습니다.

을 참조하십시오 **"공기 흐름의 예"** NetApp Data Science Toolkit GitHub 리포지토리 내의 섹션에서 공기 흐름이 포함된 툴킷 사용에 대한 자세한 내용을 확인하십시오.

Trident 작업의 예

이 섹션에는 Trident를 사용하여 수행할 수 있는 다양한 작업의 예가 포함되어 있습니다.

기존 볼륨을 가져옵니다

Kubernetes 클러스터 내의 컨테이너에 마운트할 NetApp 스토리지 시스템/플랫폼에 기존 볼륨이 있지만, 클러스터의 PVC와 연결되지 않은 경우 이러한 볼륨을 가져와야 합니다. Trident 볼륨 가져오기 기능을 사용하여 이러한 볼륨을 가져올 수 있습니다.

다음 예제 명령은 섹션의 예에서 생성된 각 Trident 백엔드에 대해 동일한 볼륨("PB_FG_ALL"이라는 이름)을 두 번 가져오는 것을 보여 줍니다 ["ONTAP AI 배포에 대한 Trident 백엔드 예"](#), 1단계. 같은 볼륨을 이러한 방식으로 두 번 가져오면 섹션에 설명된 대로 여러 LIF에서 볼륨(기존 FlexGroup 볼륨)을 여러 번 마운트할 수 있습니다 ["ONTAP AI 배포에 대한 Trident 백엔드 예"](#), 1단계. PVC에 대한 자세한 내용은 를 참조하십시오 ["Kubernetes 공식 문서"](#). 볼륨 가져오기 기능에 대한 자세한 내용은 를 참조하십시오 ["Trident 문서"](#).

예제 PVC 규격 파일에는 'ReadOnlyMany'의 'accessModes' 값이 지정되어 있습니다. 'accessMode' 필드에 대한 자세한 내용은 를 참조하십시오 ["Kubernetes 공식 문서"](#).



다음 가져오기 명령에 지정된 백엔드 이름은 섹션의 예제에서 생성한 백엔드에 해당합니다 ["ONTAP AI 배포에 대한 Trident 백엔드 예"](#), 1단계. 다음 예제 PVC 정의 파일에 지정된 StorageClass 이름은 섹션의 예제에서 만든 StorageClasses에 해당합니다 ["ONTAP AI 구축을 위한 Kubernetes StorageClasses의 예"](#), 1단계.

```
$ cat << EOF > ./pvc-import-pb_fg_all-iface1.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface1
  namespace: default
spec:
  accessModes:
    - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface1
EOF
$ tridentctl import volume ontap-ai-flexgroups-iface1 pb_fg_all -f ./pvc-
import-pb_fg_all-iface1.yaml -n trident
+-----+-----+
+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  |          STORAGE CLASS          |
| PROTOCOL |          BACKEND UUID          |  STATE  |
MANAGED |
+-----+-----+
+-----+-----+
+-----+-----+-----+-----+
| default-pb-fg-all-iface1-7d9f1 | 10 TiB | ontap-ai-flexgroups-retain-
iface1 | file      | b74cbddb-e0b8-40b7-b263-b6da6dec0bdd | online | true
```

```

|
+-----+-----+
+-----+-----+
+-----+-----+-----+
$ cat << EOF > ./pvc-import-pb_fg_all-iface2.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface2
  namespace: default
spec:
  accessModes:
    - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface2
EOF
$ tridentctl import volume ontap-ai-flexgroups-iface2 pb_fg_all -f ./pvc-
import-pb_fg_all-iface2.yaml -n trident
+-----+-----+
+-----+-----+
+-----+-----+-----+
|          NAME          |  SIZE  |          STORAGE CLASS
| PROTOCOL |          BACKEND UUID          | STATE |
MANAGED |
+-----+-----+
+-----+-----+
+-----+-----+-----+
| default-pb-fg-all-iface2-85aee | 10 TiB | ontap-ai-flexgroups-retain-
iface2 | file      | 61814d48-c770-436b-9cb4-cf7ee661274d | online | true
|
+-----+-----+
+-----+-----+
+-----+-----+-----+
$ tridentctl get volume -n trident
+-----+-----+
+-----+-----+
+-----+-----+-----+
|          NAME          |  SIZE  |          STORAGE CLASS
| PROTOCOL |          BACKEND UUID          | STATE | MANAGED |
+-----+-----+
+-----+-----+
+-----+-----+-----+
| default-pb-fg-all-iface1-7d9f1 | 10 TiB | ontap-ai-flexgroups-retain-
iface1 | file      | b74cbddb-e0b8-40b7-b263-b6da6dec0bdd | online | true
|
| default-pb-fg-all-iface2-85aee | 10 TiB | ontap-ai-flexgroups-retain-
iface2 | file      | 61814d48-c770-436b-9cb4-cf7ee661274d | online | true

```

```

|
+-----+-----+
+-----+-----+
+-----+-----+-----+
$ kubectl get pvc
NAME                                STATUS    VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS          AGE
pb-fg-all-iface1    Bound    default-pb-fg-all-iface1-7d9f1
10995116277760    ROX      ontap-ai-flexgroups-retain-iface1    25h
pb-fg-all-iface2    Bound    default-pb-fg-all-iface2-85aee
10995116277760    ROX      ontap-ai-flexgroups-retain-iface2    25h

```

새 볼륨을 프로비저닝합니다

Trident를 사용하여 NetApp 스토리지 시스템 또는 플랫폼에서 새 볼륨을 프로비저닝할 수 있습니다. 다음 명령 예에서는 새 FlexVol 볼륨의 프로비저닝을 보여 줍니다. 이 예제에서는 섹션의 예제에서 만든 StorageClass 를 사용하여 볼륨을 프로비저닝합니다 "[ONTAP AI 구축을 위한 Kubernetes StorageClasses의 예](#)", 2단계.

다음 PVC 정의 파일에는 ReadWriteMany 의 accessModes 값이 지정되어 있습니다. 'accessMode' 필드에 대한 자세한 내용은 를 참조하십시오 "[Kubernetes 공식 문서](#)".

```
$ cat << EOF > ./pvc-tensorflow-results.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: tensorflow-results
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-ai-flexvols-retain
EOF
$ kubectl create -f ./pvc-tensorflow-results.yaml
persistentvolumeclaim/tensorflow-results created
$ kubectl get pvc
NAME                                STATUS      VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS          AGE
pb-fg-all-iface1                    Bound      default-pb-fg-all-iface1-7d9f1          10995116277760    ROX           ontap-ai-flexgroups-retain-iface1    26h
pb-fg-all-iface2                    Bound      default-pb-fg-all-iface2-85aee          10995116277760    ROX           ontap-ai-flexgroups-retain-iface2    26h
tensorflow-results                   Bound      default-tensorflow-results-2fd60        1073741824        RWX           ontap-ai-flexvols-retain            25h
```

ONTAP AI 배포에 대한 고성능 작업 예

이 섹션에는 Kubernetes를 ONTAP AI Pod에 구축할 때 실행할 수 있는 다양한 고성능 작업의 예가 포함되어 있습니다.

ONTAP AI 배포에 대한 고성능 작업 예

이 섹션에는 Kubernetes를 ONTAP AI Pod에 구축할 때 실행할 수 있는 다양한 고성능 작업의 예가 포함되어 있습니다.

단일 노드 AI 워크로드 실행

Kubernetes 클러스터에서 단일 노드 AI 및 ML 작업을 실행하려면 배포 점프 호스트에서 다음 작업을 수행하십시오. Trident를 사용하면 페타바이트에 이를 수 있는 데이터 볼륨을 빠르고 쉽게 만들어 Kubernetes 워크로드에 액세스할 수 있습니다. Kubernetes Pod에서 데이터 볼륨에 액세스할 수 있도록 하려면 POD 정의에 PVC를 지정하기만 하면 됩니다. 이 단계는 Kubernetes 네이티브 운영이므로 NetApp의 전문성이 필요하지 않습니다.



이 섹션에서는 Kubernetes 클러스터에서 실행하려고 하는 특정 AI 및 ML 워크로드를 이미 컨테이너화(Docker 컨테이너 형식)했다고 가정합니다.

1. 다음 명령 예는 ImageNet 데이터 세트를 사용하는 TensorFlow 벤치마크 워크로드에 대한 Kubernetes 작업 생성을 보여줍니다. ImageNet 데이터 세트에 대한 자세한 내용은 [참조하십시오 "ImageNet 웹 사이트"](#).

이 예시 작업은 8개의 GPU를 요청하므로 8개 이상의 GPU를 갖춘 단일 GPU 작업자 노드에서 실행할 수 있습니다. 이 예시 작업은 8개 이상의 GPU를 갖춘 작업자 노드가 없거나 현재 다른 워크로드를 사용 중인 클러스터에 제출할 수 있습니다. 이 경우 해당 작업자 노드를 사용할 수 있을 때까지 작업은 보류 중 상태로 유지됩니다.

또한 스토리지 대역폭을 최대화하기 위해 필요한 교육 데이터가 들어 있는 볼륨이 이 작업에서 생성되는 POD 내에 두 번 마운트됩니다. 포드에도 다른 볼륨이 마운트됩니다. 이 두 번째 볼륨은 결과 및 메트릭을 저장하는 데 사용됩니다. 이러한 용적은 PVC 이름을 사용하여 작업 정의에서 참조됩니다. Kubernetes 작업에 대한 자세한 내용은 [참조하십시오 "Kubernetes 공식 문서"](#).

이 예시 작업이 생성하는 포드의 /dev/shm에 Memory의 midium 값을 가진 emptyDir 볼륨이 실장된다. Docker 컨테이너 런타임을 통해 자동으로 생성되는 '/dev/shm' 가상 볼륨의 기본 크기는 TensorFlow의 요구 사항에 비해 부족할 수 있습니다. 다음 예제와 같이 "emptyDir" 볼륨을 마운트하면 충분히 큰 "/dev/shm" 가상 볼륨이 제공됩니다. 'emptyDir' 볼륨에 대한 자세한 내용은 [참조하십시오 "Kubernetes 공식 문서"](#).

이 예제 작업 정의에 지정된 단일 컨테이너에는 'securityContext > privileged' 값이 'true'로 지정됩니다. 이 값은 컨테이너가 호스트에 대한 루트 액세스 권한을 효과적으로 가지고 있음을 의미합니다. 이 경우 실행되는 특정 워크로드에 루트 액세스가 필요하므로 이 주석이 사용됩니다. 특히, 워크로드가 수행하는 명확한 캐시 작업에서는 루트 액세스가 필요합니다. 이 "특권" 주석이 필요한지 여부는 실행 중인 특정 워크로드의 요구 사항에 따라 달라집니다.

```
$ cat << EOF > ./netapp-tensorflow-single-imagenet.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-tensorflow-single-imagenet
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: dshm
        emptyDir:
          medium: Memory
      - name: testdata-iface1
        persistentVolumeClaim:
          claimName: pb-fg-all-iface1
      - name: testdata-iface2
        persistentVolumeClaim:
          claimName: pb-fg-all-iface2
      - name: results
        persistentVolumeClaim:
          claimName: tensorflow-results
    containers:
```

```

- name: netapp-tensorflow-py2
  image: netapp/tensorflow-py2:19.03.0
  command: ["python", "/netapp/scripts/run.py", "--
dataset_dir=/mnt/mount_0/dataset/imagenet", "--dgx_version=dgx1", "--
num_devices=8"]
  resources:
    limits:
      nvidia.com/gpu: 8
  volumeMounts:
  - mountPath: /dev/shm
    name: dshm
  - mountPath: /mnt/mount_0
    name: testdata-iface1
  - mountPath: /mnt/mount_1
    name: testdata-iface2
  - mountPath: /tmp
    name: results
  securityContext:
    privileged: true
  restartPolicy: Never
EOF
$ kubectl create -f ./netapp-tensorflow-single-imagenet.yaml
job.batch/netapp-tensorflow-single-imagenet created
$ kubectl get jobs
NAME                                COMPLETIONS   DURATION   AGE
netapp-tensorflow-single-imagenet   0/1            24s        24s

```

2. 1단계에서 만든 작업이 올바르게 실행 중인지 확인합니다. 다음 명령 예에서는 작업 정의에 지정된 대로 작업에 대해 단일 POD가 생성되었으며 이 POD가 현재 GPU 작업자 노드 중 하나에서 실행되고 있음을 확인합니다.

```

$ kubectl get pods -o wide
NAME                                READY   STATUS
RESTARTS   AGE
IP          NODE          NOMINATED NODE
netapp-tensorflow-single-imagenet-m7x92   1/1     Running   0
3m         10.233.68.61   10.61.218.154   <none>

```

3. 1단계에서 생성한 작업이 성공적으로 완료되었는지 확인합니다. 다음 명령 예에서는 작업이 성공적으로 완료되었음을 확인합니다.

```

$ kubectl get jobs
NAME                                     COMPLETIONS   DURATION
AGE
netapp-tensorflow-single-imagenet      1/1            5m42s
10m
$ kubectl get pods
NAME                                     READY   STATUS
RESTARTS   AGE
netapp-tensorflow-single-imagenet-m7x92 0/1     Completed
0         11m
$ kubectl logs netapp-tensorflow-single-imagenet-m7x92
[netapp-tensorflow-single-imagenet-m7x92:00008] PMIX ERROR: NO-
PERMISSIONS in file gds_dstore.c at line 702
[netapp-tensorflow-single-imagenet-m7x92:00008] PMIX ERROR: NO-
PERMISSIONS in file gds_dstore.c at line 711
Total images/sec = 6530.59125
===== Clean Cache !!! =====
mpirun -allow-run-as-root -np 1 -H localhost:1 bash -c 'sync; echo 1 >
/proc/sys/vm/drop_caches'
=====
mpirun -allow-run-as-root -np 8 -H localhost:8 -bind-to none -map-by
slot -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH python
/netapp/tensorflow/benchmarks_190205/scripts/tf_cnn_benchmarks/tf_cnn_be
nchmarks.py --model=resnet50 --batch_size=256 --device=gpu
--force_gpu_compatible=True --num_intra_threads=1 --num_inter_threads=48
--variable_update=horovod --batch_group_size=20 --num_batches=500
--nodistortions --num_gpus=1 --data_format=NCHW --use_fp16=True
--use_tf_layers=False --data_name=imagenet --use_datasets=True
--data_dir=/mnt/mount_0/dataset/imagenet
--datasets_parallel_interleave_cycle_length=10
--datasets_sloppy_parallel_interleave=False --num_mounts=2
--mount_prefix=/mnt/mount_%d --datasets_prefetch_buffer_size=2000
--datasets_use_prefetch=True --datasets_num_private_threads=4
--horovod_device=gpu >
/tmp/20190814_105450_tensorflow_horovod_rdma_resnet50_gpu_8_256_b500_ima
genet_nodistort_fp16_r10_m2_nockpt.txt 2>&1

```

4. * 선택 사항: * 작업 아티팩트 정리. 다음 예제 명령은 1단계에서 만든 작업 오브젝트의 삭제를 보여 줍니다.

작업 개체를 삭제하면 Kubernetes에서 연결된 포드를 자동으로 삭제합니다.

```

$ kubectl get jobs
NAME                                     COMPLETIONS   DURATION
AGE
netapp-tensorflow-single-imagenet      1/1            5m42s
10m
$ kubectl get pods
NAME                                     READY   STATUS
RESTARTS   AGE
netapp-tensorflow-single-imagenet-m7x92 0/1     Completed
0         11m
$ kubectl delete job netapp-tensorflow-single-imagenet
job.batch "netapp-tensorflow-single-imagenet" deleted
$ kubectl get jobs
No resources found.
$ kubectl get pods
No resources found.

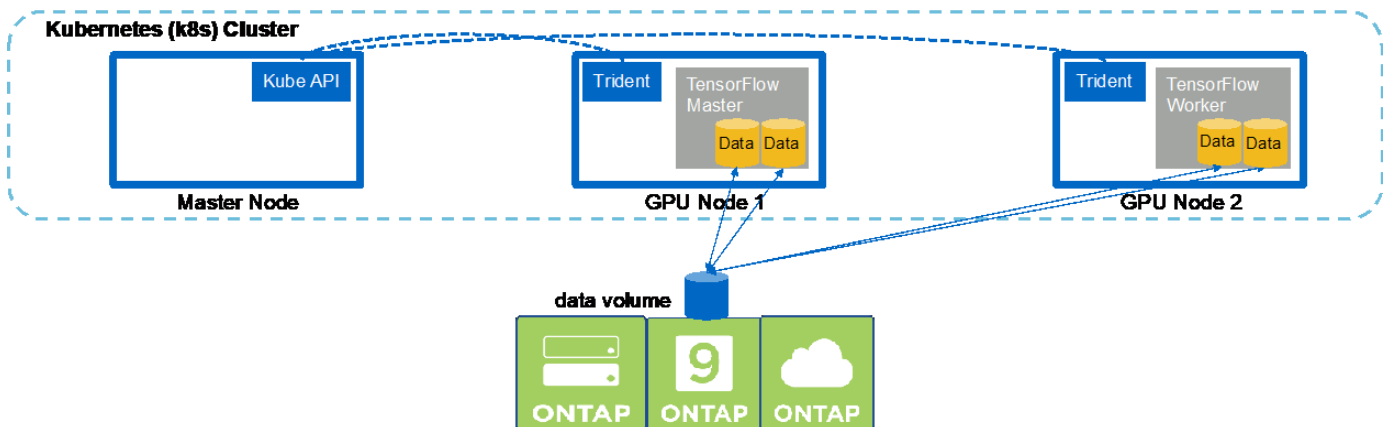
```

동기식 분산 AI 워크로드 실행

Kubernetes 클러스터에서 동기식 다중 노드 AI 및 ML 작업을 실행하려면 배포 점프 호스트에서 다음 작업을 수행하십시오. 이 프로세스를 통해 NetApp 볼륨에 저장된 데이터를 활용하고 단일 작업자 노드가 제공할 수 있는 것보다 더 많은 GPU를 사용할 수 있습니다. 동기식 분산 AI 작업을 설명하는 방법은 다음 그림을 참조하십시오.



동기 분산 작업은 비동기 분산 작업에 비해 성능 및 교육 정확도를 높일 수 있습니다. 동기 작업과 비동기 작업의 장단점을 논하는 것은 이 문서의 범위를 벗어납니다.



1. 다음 명령 예는 섹션의 예에서 단일 노드에서 실행된 동일한 TensorFlow 벤치마크 작업의 동기식 분산 실행에 참여하는 작업자 1명의 생성을 보여 줍니다. "[단일 노드 AI 워크로드 실행](#)". 이 특정 예제에서는 작업이 두 작업자 노드에 걸쳐 실행되므로 한 명의 작업자만 배포됩니다.

이 작업자 배포는 8개의 GPU를 요청하므로 8개 이상의 GPU를 갖춘 단일 GPU 작업자 노드에서 실행할 수 있습니다. GPU 작업자 노드에서 8개 이상의 GPU를 사용하여 성능을 극대화한 경우, 이 숫자를 작업자 노드가 갖춘 GPU 수와 같게 늘리고 싶을 수 있습니다. Kubernetes 구축에 대한 자세한 내용은 [Kubernetes](#)를 참조하십시오.

공식 문서".

이 예에서는 특정 컨테이너형 작업자가 자체적으로 완료되지 않기 때문에 Kubernetes 구축이 생성됩니다. 따라서 Kubernetes 작업 구성을 사용하여 구축하는 것은 타당하지 않습니다. 작업자가 혼자서 완료되도록 설계되거나 작성된 경우 작업 구성을 사용하여 작업자를 배포하는 것이 합리일 수 있습니다.

이 예제 배포 사양에 지정된 POD에 "true"의 "hostNetwork" 값이 제공됩니다. 이 값은 이 Pod가 일반적으로 Kubernetes에서 각 Pod에 생성하는 가상 네트워킹 스택 대신 호스트 작업자 노드의 네트워킹 스택을 사용한다는 것을 의미합니다. 이 경우 특정 워크로드는 개방형 MPI, NCCL 및 Horovod를 통해 동기식 분산 방식으로 워크로드를 실행하기 때문에 이 주석이 사용됩니다. 따라서 호스트 네트워킹 스택에 액세스해야 합니다. 공개 MPI, NCCL 및 Horovod에 대한 논의는 이 문서의 범위를 벗어납니다. 이 "hostNetwork: true" 주석이 필요한지 여부는 실행 중인 특정 워크로드의 요구 사항에 따라 달라집니다. hostNetwork 필드에 대한 자세한 내용은 [참조하십시오 "Kubernetes 공식 문서"](#).

```
$ cat << EOF > ./netapp-tensorflow-multi-imagenet-worker.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: netapp-tensorflow-multi-imagenet-worker
spec:
  replicas: 1
  selector:
    matchLabels:
      app: netapp-tensorflow-multi-imagenet-worker
  template:
    metadata:
      labels:
        app: netapp-tensorflow-multi-imagenet-worker
    spec:
      hostNetwork: true
      volumes:
        - name: dshm
          emptyDir:
            medium: Memory
        - name: testdata-iface1
          persistentVolumeClaim:
            claimName: pb-fg-all-iface1
        - name: testdata-iface2
          persistentVolumeClaim:
            claimName: pb-fg-all-iface2
        - name: results
          persistentVolumeClaim:
            claimName: tensorflow-results
      containers:
        - name: netapp-tensorflow-py2
          image: netapp/tensorflow-py2:19.03.0
          command: ["bash", "/netapp/scripts/start-slave-multi.sh",
"22122"]
```

```

resources:
  limits:
    nvidia.com/gpu: 8
  volumeMounts:
  - mountPath: /dev/shm
    name: dshm
  - mountPath: /mnt/mount_0
    name: testdata-iface1
  - mountPath: /mnt/mount_1
    name: testdata-iface2
  - mountPath: /tmp
    name: results
  securityContext:
    privileged: true
EOF
$ kubectl create -f ./netapp-tensorflow-multi-imagenet-worker.yaml
deployment.apps/netapp-tensorflow-multi-imagenet-worker created
$ kubectl get deployments
NAME                                                    DESIRED   CURRENT   UP-TO-DATE
AVAILABLE   AGE
netapp-tensorflow-multi-imagenet-worker  1         1         1
1         4s

```

2. 1단계에서 만든 작업자 배포가 성공적으로 시작되었는지 확인합니다. 다음 예제 명령은 배포 정의에 나와 있는 것처럼 단일 작업자 POD가 배포용으로 생성되었으며 이 POD가 현재 GPU 작업자 노드 중 하나에서 실행되고 있음을 확인합니다.

```

$ kubectl get pods -o wide
NAME                                                    READY
STATUS    RESTARTS   AGE
IP          NODE          NOMINATED NODE
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725  1/1
Running    0          60s  10.61.218.154  10.61.218.154  <none>
$ kubectl logs netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725
22122

```

3. 동기 다중 노드 작업의 실행을 종료, 참여 및 추적하는 마스터에 대한 Kubernetes 작업을 생성합니다. 다음 명령 예에서는 이 섹션의 예제에서 단일 노드에서 실행된 것과 동일한 TensorFlow 벤치마크 작업의 동기식 분산 실행을 시작, 참여 및 추적하는 하나의 마스터를 생성합니다 ["단일 노드 AI 워크로드 실행"](#).

이 마스터 작업에서는 8개의 GPU를 요청하므로 8개 이상의 GPU를 갖춘 단일 GPU 작업자 노드에서 실행할 수 있습니다. GPU 작업자 노드에서 8개 이상의 GPU를 사용하여 성능을 극대화한 경우, 이 숫자를 작업자 노드가 갖춘 GPU 수와 같게 늘리고 싶을 수 있습니다.

이 예에서 지정한 마스터 포드는 1단계에서 작업자 포드가 `hostNetwork` 값이 `true`인 것처럼 `true`의 `hostNetwork` 값이 지정됩니다. 이 값이 필요한 이유에 대한 자세한 내용은 1단계를 참조하십시오.

```

$ cat << EOF > ./netapp-tensorflow-multi-imagenet-master.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-tensorflow-multi-imagenet-master
spec:
  backoffLimit: 5
  template:
    spec:
      hostNetwork: true
      volumes:
      - name: dshm
        emptyDir:
          medium: Memory
      - name: testdata-iface1
        persistentVolumeClaim:
          claimName: pb-fg-all-iface1
      - name: testdata-iface2
        persistentVolumeClaim:
          claimName: pb-fg-all-iface2
      - name: results
        persistentVolumeClaim:
          claimName: tensorflow-results
    containers:
    - name: netapp-tensorflow-py2
      image: netapp/tensorflow-py2:19.03.0
      command: ["python", "/netapp/scripts/run.py", "--
dataset_dir=/mnt/mount_0/dataset/imagenet", "--port=22122", "--
num_devices=16", "--dgx_version=dgx1", "--
nodes=10.61.218.152,10.61.218.154"]
      resources:
        limits:
          nvidia.com/gpu: 8
      volumeMounts:
      - mountPath: /dev/shm
        name: dshm
      - mountPath: /mnt/mount_0
        name: testdata-iface1
      - mountPath: /mnt/mount_1
        name: testdata-iface2
      - mountPath: /tmp
        name: results
      securityContext:
        privileged: true
      restartPolicy: Never

```

```
EOF
$ kubectl create -f ./netapp-tensorflow-multi-imagenet-master.yaml
job.batch/netapp-tensorflow-multi-imagenet-master created
$ kubectl get jobs
```

NAME	COMPLETIONS	DURATION	AGE
netapp-tensorflow-multi-imagenet-master	0/1	25s	25s

4. 3단계에서 만든 마스터 작업이 올바르게 실행되고 있는지 확인합니다. 다음 예제 명령은 작업 정의에 나와 있는 것처럼 작업에 대해 단일 마스터 포드가 생성되었으며 이 포드가 현재 GPU 작업자 노드 중 하나에서 실행되고 있음을 확인합니다. 또한 1단계에서 처음 보았던 작업자 포드가 여전히 실행 중이고 마스터 포드와 작업자 포드가 다른 노드에서 실행되고 있음을 확인해야 합니다.

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

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

5. 3단계에서 만든 마스터 작업이 성공적으로 완료되었는지 확인합니다. 다음 명령 예에서는 작업이 성공적으로 완료되었음을 확인합니다.

```
$ kubectl get jobs
```

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

```
$ kubectl get pods
```

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

```
$ kubectl logs netapp-tensorflow-multi-imagenet-master-ppwwj
[10.61.218.152:00008] WARNING: local probe returned unhandled
shell:unknown assuming bash
rm: cannot remove '/lib': Is a directory
[10.61.218.154:00033] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 702
[10.61.218.154:00033] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 711
[10.61.218.152:00008] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 702
[10.61.218.152:00008] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
```



```

line 711
Total images/sec = 12881.33875
===== Clean Cache !!! =====
mpirun -allow-run-as-root -np 2 -H 10.61.218.152:1,10.61.218.154:1 -mca
pml obl -mca btl ^openib -mca btl_tcp_if_include enpls0f0 -mca
plm_rsh_agent ssh -mca plm_rsh_args "-p 22122" bash -c 'sync; echo 1 >
/proc/sys/vm/drop_caches'
=====
mpirun -allow-run-as-root -np 16 -H 10.61.218.152:8,10.61.218.154:8
-bind-to none -map-by slot -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH
-mca pml obl -mca btl ^openib -mca btl_tcp_if_include enpls0f0 -x
NCCL_IB_HCA=mlx5 -x NCCL_NET_GDR_READ=1 -x NCCL_IB_SL=3 -x
NCCL_IB_GID_INDEX=3 -x
NCCL_SOCKET_IFNAME=enp5s0.3091,enp12s0.3092,enp132s0.3093,enp139s0.3094
-x NCCL_IB_CUDA_SUPPORT=1 -mca orte_base_help_aggregate 0 -mca
plm_rsh_agent ssh -mca plm_rsh_args "-p 22122" python
/netapp/tensorflow/benchmarks_190205/scripts/tf_cnn_benchmarks/tf_cnn_be
nchmarks.py --model=resnet50 --batch_size=256 --device=gpu
--force_gpu_compatible=True --num_intra_threads=1 --num_inter_threads=48
--variable_update=horovod --batch_group_size=20 --num_batches=500
--nodistortions --num_gpus=1 --data_format=NCHW --use_fp16=True
--use_tf_layers=False --data_name=imagenet --use_datasets=True
--data_dir=/mnt/mount_0/dataset/imagenet
--datasets_parallel_interleave_cycle_length=10
--datasets_sloppy_parallel_interleave=False --num_mounts=2
--mount_prefix=/mnt/mount_%d --datasets_prefetch_buffer_size=2000 --
datasets_use_prefetch=True --datasets_num_private_threads=4
--horovod_device=gpu >
/tmp/20190814_161609_tensorflow_horovod_rdma_resnet50_gpu_16_256_b500_im
agenet_nodistort_fp16_r10_m2_nockpt.txt 2>&1

```

6. 작업자 배포가 더 이상 필요하지 않으면 삭제합니다. 다음 예제 명령은 1단계에서 만든 작업자 배포 개체를 삭제하는 방법을 보여 줍니다.

작업자 배포 개체를 삭제하면 Kubernetes에서 연결된 작업자 포드를 자동으로 삭제합니다.

```

$ kubectl get deployments
NAME                                                    DESIRED   CURRENT   UP-TO-DATE
AVAILABLE      AGE
netapp-tensorflow-multi-imagenet-worker      1         1         1
1         43m
$ kubectl get pods
NAME                                                    READY
STATUS      RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj      0/1
Completed    0         17m
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725  1/1
Running      0         43m
$ kubectl delete deployment netapp-tensorflow-multi-imagenet-worker
deployment.extensions "netapp-tensorflow-multi-imagenet-worker" deleted
$ kubectl get deployments
No resources found.
$ kubectl get pods
NAME                                                    READY   STATUS
RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj      0/1     Completed    0
18m

```

7. * 선택 사항: * 마스터 작업 아티팩트를 정리하십시오. 다음 예제 명령은 3단계에서 만든 마스터 작업 오브젝트의 삭제를 보여 줍니다.

마스터 작업 개체를 삭제하면 연결된 마스터 포드가 자동으로 삭제됩니다.

```

$ kubectl get jobs
NAME                                                    COMPLETIONS   DURATION   AGE
netapp-tensorflow-multi-imagenet-master      1/1           5m50s     19m
$ kubectl get pods
NAME                                                    READY   STATUS
RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj      0/1     Completed    0
19m
$ kubectl delete job netapp-tensorflow-multi-imagenet-master
job.batch "netapp-tensorflow-multi-imagenet-master" deleted
$ kubectl get jobs
No resources found.
$ kubectl get pods
No resources found.

```

성능 테스트

우리는 이 솔루션 작성의 일부로 간단한 성능 비교를 수행했습니다. Kubernetes를 사용하여 몇 가지 표준 NetApp AI 벤치마킹 작업을 실행했으며, 단순한 Docker 실행 명령을 사용하여 수행한 실행과 벤치마크 결과를 비교했습니다. 뚜렷한 성능 차이는 없었습니다. 따라서, 컨테이너화된 AI 교육 작업을 오케스트레이션하기 위해 Kubernetes를 사용할 경우 성능에 부정적인 영향을 미치지 않는다는 결론을 내렸습니다. 성능 비교 결과는 다음 표를 참조하십시오.

벤치마크	데이터 세트	Docker 실행(이미지/초)	Kubernetes(이미지/초)
단일 노드 TensorFlow	합성 데이터	6,667.2475	6,661.93125
단일 노드 TensorFlow	ImageNet	6,570.2025	6,530.59125
동기식 분산 2노드 TensorFlow	합성 데이터	13,213.70625	13,218.288125
동기식 분산 2노드 TensorFlow	ImageNet	12,941.69125	12,881.33875

결론

모든 규모와 업종에 상관없이 모든 기업과 조직은 실제 문제를 해결하고 혁신적인 제품과 서비스를 제공하며 경쟁이 갈수록 치열해지는 시장에서 경쟁 우위를 확보하기 위해 인공지능(AI), 머신 러닝(ML), 딥 러닝(DL)으로 눈을 돌리고 있습니다. AI, ML 및 DL의 사용이 증가함에 따라 워크로드 확장성 및 데이터 가용성을 비롯한 많은 과제에 직면하게 됩니다. 이러한 문제는 NetApp AI Control Plane 솔루션을 사용하여 해결할 수 있습니다.

이 솔루션을 사용하면 데이터 네임스페이스를 빠르게 클론 복제할 수 있습니다. 또한 추적 가능성 및 버전 관리를 위한 데이터 및 모델 기준선의 거의 즉각적인 생성을 통합하는 AI, ML 및 DL 교육 워크플로우를 정의하고 구현할 수 있습니다. 이 솔루션을 사용하면 모든 단일 모델 교육을 모델이 훈련 및/또는 검증을 거친 정확한 데이터 세트로 추적할 수 있습니다. 끝으로, 이 솔루션을 사용하면 방대한 데이터 세트에 액세스할 수 있는 Jupyter Notebook 작업 공간을 신속하게 프로비저닝할 수 있습니다.

이 솔루션은 데이터 과학자 및 데이터 엔지니어들을 대상으로 하기 때문에 최소한의 NetApp 또는 NetApp ONTAP 전문 지식이 필요합니다. 이 솔루션에서는 단순하고 친숙한 톨과 인터페이스를 사용하여 데이터 관리 기능을 실행할 수 있습니다. 또한 이 솔루션은 완전한 오픈 소스 및 무료 구성 요소를 활용합니다. 따라서 환경에 NetApp 스토리지가 이미 있는 경우 지금 이 솔루션을 구현할 수 있습니다. 이 솔루션을 시험하고 싶지만 NetApp 스토리지가 없는 경우를 방문하십시오 ["cloud.netapp.com"](https://cloud.netapp.com) 또한 클라우드 기반 NetApp 스토리지 솔루션을 사용하여 시스템을 신속하게 가동할 수 있습니다.

Iguazio를 사용한 MLRun 파이프라인

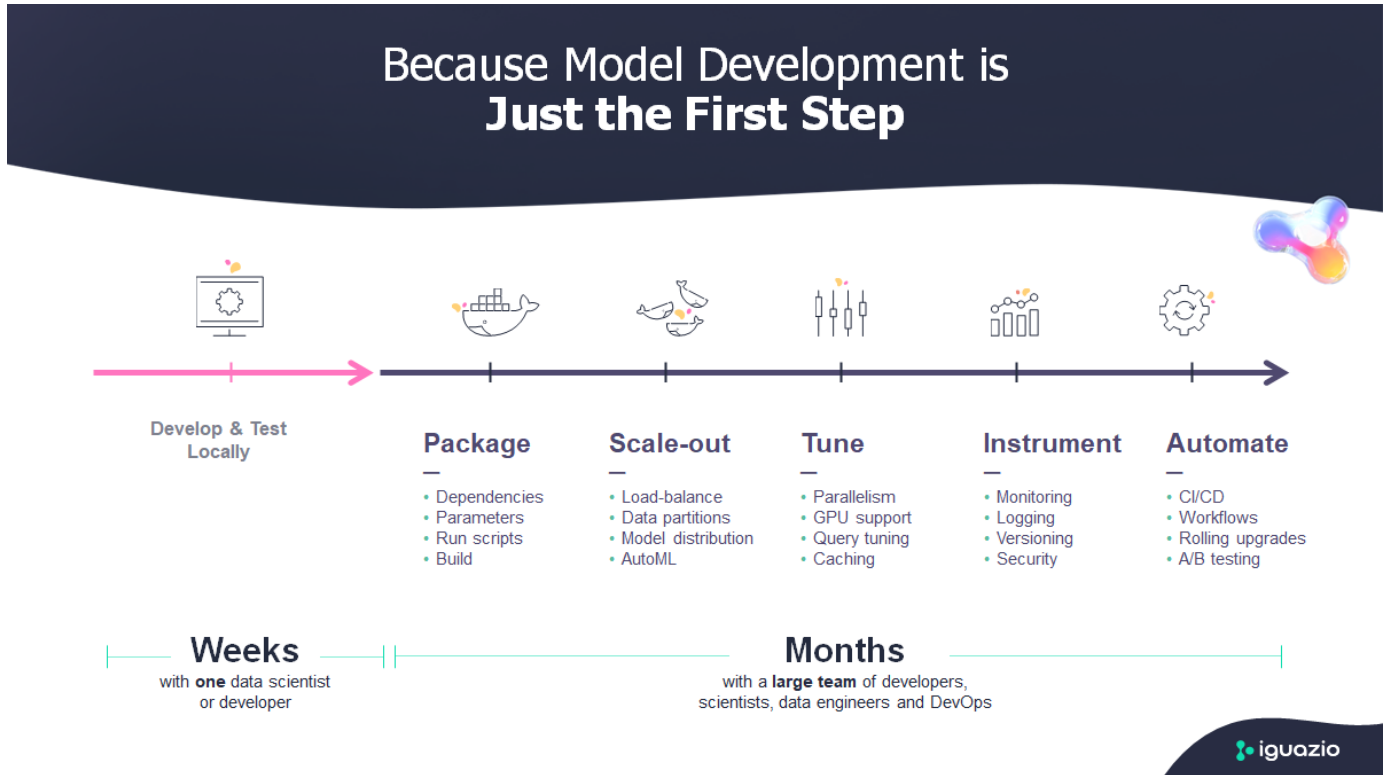
TR-4834: MLRun 파이프라인에 대한 NetApp과 Iguazio

Rick Huang, David Arnette, NetApp Marcelo Litovsky, Iguazio

이 문서에서는 NetApp ONTAP AI, NetApp AI Control Plane, NetApp Cloud Volumes 소프트웨어 및 Iguazio 데이터 과학 플랫폼을 사용하는 MLRun 파이프라인의 세부 정보를 다룹니다. Nuclio serverless 기능, Kubernetes Persistent Volumes, NetApp Cloud Volumes,

NetApp Snapshot 복사본, Grafana 대시보드를 사용했습니다. 네트워크 장애 감지 시뮬레이션을 위한 종단 간 데이터 파이프라인을 구축하기 위한 Iguazio 플랫폼의 기타 서비스. Iguazio 및 NetApp 기술을 통합하여 사내와 클라우드에서 모델 구축, 데이터 복제 및 운영 모니터링 기능을 빠르게 구현합니다.

데이터 과학자의 작업은 머신 러닝(ML) 및 인공지능(AI) 모델의 훈련 및 튜닝에 중점을 두어야 합니다. 그러나 Google의 조사에 따르면 데이터 과학자는 다음 이미지에서와 같이 모델을 엔터프라이즈 애플리케이션과 연동하고 대규모로 실행하는 방법을 찾는 데 80% 정도 시간을 소비합니다. AI/ML 워크플로우에서 모델 개발을 묘사한 것으로 나타났습니다.



엔드 투 엔드 AI/ML 프로젝트를 관리하려면 엔터프라이즈 구성 요소를 더 잘 이해해야 합니다. DevOps가 이러한 유형의 구성 요소를 정의, 통합 및 배포했지만 머신 러닝 작업은 AI/ML 프로젝트를 포함하는 비슷한 흐름을 목표로 합니다. 엔터프라이즈에서 엔드 투 엔드 AI/ML 파이프라인이 어떤 영향을 받는지 알아보려면 다음 필수 구성요소 목록을 참조하십시오.

- 스토리지
- 네트워킹
- 데이터베이스를 지원합니다
- 파일 시스템
- 컨테이너
- CI/CD(Continuous Integration and Continuous Deployment) 파이프라인
- 개발 IDE(통합 개발 환경)
- 보안
- 데이터 액세스 정책

- 하드웨어
- 클라우드
- 포함되었습니다
- 데이터 과학 도구 세트 및 라이브러리

이 문서에서는 NetApp과 Iguazio 간의 파트너십을 통해 엔드 투 엔드 AI/ML 파이프라인 개발을 획기적으로 단순화하는 방법을 보여줍니다. 이러한 단순화 덕분에 모든 AI/ML 애플리케이션의 출시 시기를 앞당길 수 있습니다.

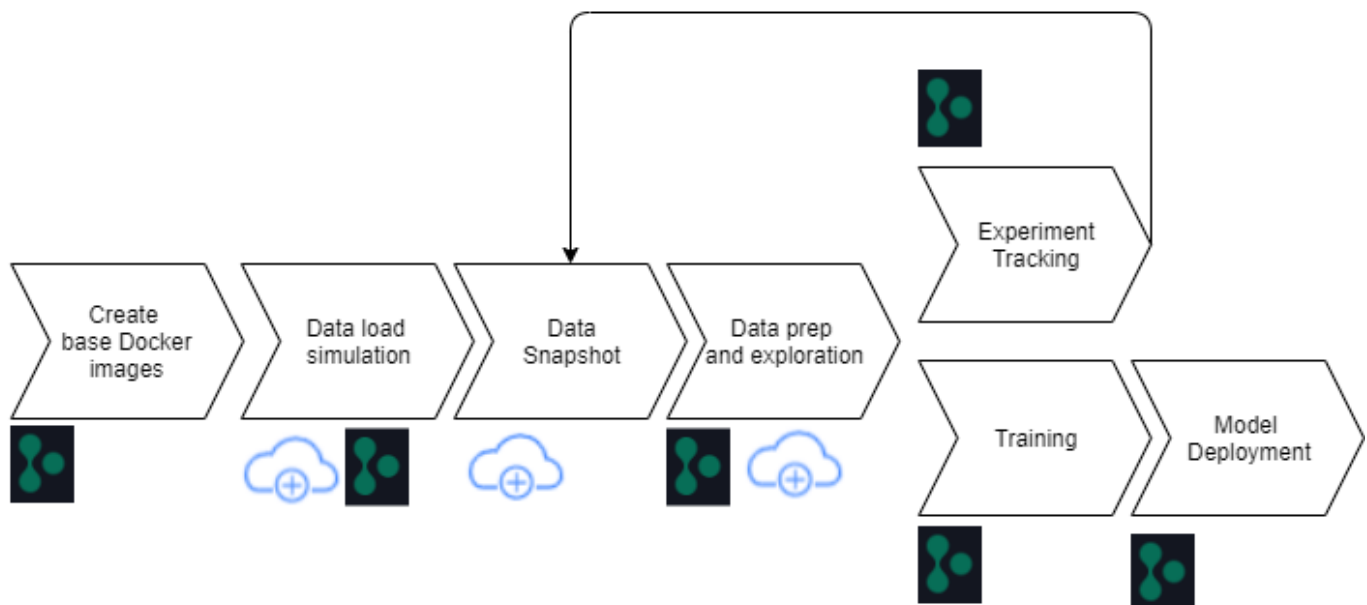
대상

데이터 과학의 세계는 정보 기술 및 비즈니스의 여러 분야에 영향을 줍니다.

- 데이터 과학자는 자신이 선택한 도구와 라이브러리를 사용할 수 있는 유연성이 필요합니다.
- 데이터 엔지니어는 데이터 흐름과 데이터 위치를 알아야 합니다.
- DevOps 엔지니어는 새로운 AI/ML 애플리케이션을 CI/CD 파이프라인에 통합하는 툴을 필요로 합니다.
- 비즈니스 사용자는 AI/ML 애플리케이션에 액세스할 수 있기를 원합니다. NetApp과 Iguazio가 각 역할을 통해 당사의 플랫폼을 통해 비즈니스에서 가치를 창출하는 방법을 설명합니다.

솔루션 개요

이 솔루션은 AI/ML 애플리케이션의 라이프사이클 뒤에 있습니다. 먼저 데이터 과학자의 작업을 통해 데이터를 준비하고 모델을 훈련 및 구축하는 데 필요한 다양한 단계를 정의합니다. 또한 아티팩트를 추적하고, 실행을 실험하고, Kubeflow에 배포할 수 있는 능력을 갖춘 전체 파이프라인을 생성하는 데 필요한 작업을 수행합니다. 전체 주기를 완료하기 위해 NetApp Cloud Volumes와 파이프라인을 통합하여 다음 이미지와 같이 데이터 버전 관리를 지원합니다.



기술 개요

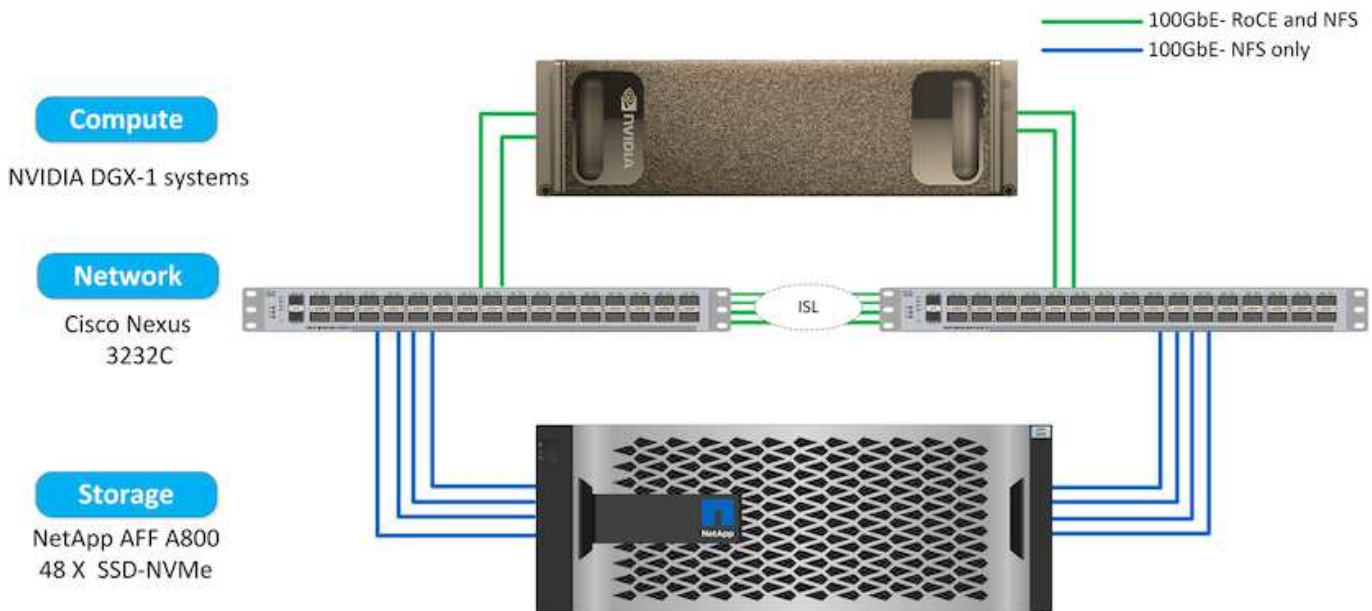
NetApp 개요

NetApp은 하이브리드 클라우드 환경에서 데이터 관련 최고의 권위자입니다. NetApp은 클라우드와 온프레미스 환경에서 애플리케이션 및 데이터의 관리를 간소화하여 디지털 혁신을 앞당기는 다양한 하이브리드 클라우드 데이터 서비스를 제공합니다. NetApp은 파트너와 함께 글로벌 조직이 데이터의 잠재력을 극대화하여 고객과의 접점을 확대하고 혁신을 촉진하며 운영을 최적화할 수 있도록 돕고 있습니다.

NetApp ONTAP AI를 참조하십시오

NVIDIA DGX 시스템과 NetApp 클라우드 연결형 All-Flash 스토리지를 기반으로 하는 NetApp ONTAP AI를 사용하면 데이터 흐름을 안정적으로 간소화하고 에지에서 코어 및 클라우드에 이르는 Data Fabric을 사용하여 분석, 훈련, 추론의 속도를 높일 수 있습니다. 이 아키텍처는 IT 조직에 다음과 같은 이점을 제공하는 아키텍처를 제공합니다.


- 설계 복잡성 제거
- 컴퓨팅과 스토리지를 독립적으로 확장할 수 있습니다
- 고객이 작은 규모로 시작한 후 원활하게 확장할 수 있도록 지원
- 다양한 성능 및 비용 포인트를 위한 다양한 스토리지 옵션을 제공합니다. NetApp ONTAP AI는 NVIDIA DGX-1, 페타플롭 확장 AI 시스템, NVIDIA Mellanox 고성능 이더넷 스위치를 통합한 통합 인프라 스택을 제공하여 AI 워크로드를 통합하고 구축을 간소화하고 ROI를 가속합니다. NetApp은 이 기술 보고서를 위해 하나의 DGX-1 및 NetApp AFF A800 스토리지 시스템과 함께 ONTAP AI를 활용했습니다. 다음 이미지는 이 검증에 사용된 DGX-1 시스템과 ONTAP AI의 토폴로지를 보여줍니다.




NetApp AI Control Plane


NetApp AI Control Plane을 사용하면 최고의 확장성, 간소화된 구축, 무중단 데이터 가용성을 제공하는 솔루션을 통해 AI 및 ML을 활용할 수 있습니다. AI Control Plane 솔루션은 Kubernetes 및 Kubeflow를 NetApp 구현 Data Fabric과 통합합니다. 클라우드 네이티브 구현을 위한 업계 표준 컨테이너 오케스트레이션 플랫폼인 Kubernetes는 워크로드 확장성 및 이동성을 지원합니다. Kubeflow는 관리 및 구현을 간소화하여 개발자가 더 많은 데이터 과학을 더 빠르게 수행할 수 있도록 지원하는 오픈 소스 머신 러닝 플랫폼입니다. NetApp 구현 Data Fabric은 최고의 데이터 가용성 및 이동성을 제공하여 에지, 코어, 클라우드에 이르는 파이프라인 전반에서 데이터에 액세스할 수 있도록 합니다. 이 기술 보고서에서는 MLRun 파이프라인에서 NetApp AI Control Plane을 사용합니다. 다음 이미지는 각 클러스터에 대해


서로 다른 엔드포인트를 가질 수 있는 Kubernetes 클러스터 관리 페이지를 보여줍니다. NFS 영구 볼륨을 Kubernetes 클러스터에 연결했고 다음 이미지에는 클러스터에 연결된 영구 볼륨이 표시됩니다. 여기서 "NetApp 트라이던트" 영구 스토리지 지원 및 데이터 관리 기능을 제공합니다.


 Kubernetes Clusters Discover Cluster


4 Kubernetes Clusters


 kubernet


 <https://3.20.111.39:6443>
Cluster Endpoint


 v1.15.5
Cluster Version


 19.07.1
Trident Version


 0
Working Environments

 kubernet

 <https://172.31.14.31:6443>
Cluster Endpoint

 v1.15.5
Cluster Version

 19.07.1
Trident Version

 1
Working Environments

Persistent Volumes for Kubernetes

Connected with Kubernetes Cluster

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

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

Kubernetes Cluster

Select Kubernetes Cluster

kubernetes

Custom Export Policy (Optional)

Custom Export Policy

172.31.0.0/16

☒ Set as default storage class

☒ NFS ☐ iSCSI

Connect

Cancel

Volumes

4 Volumes | 300 GB Allocated | 1.43 GB Total Used


kubernetes_trident_pvc_551720fa_3758_461...
ONLINE

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

Iguazio 개요

Iguazio Data Science Platform은 완전히 통합되고 안전한 PaaS(Data Science Platform as a Service)로서 개발을 간소화하고, 성능을 가속화하고, 협업을 촉진하며, 운영 문제를 해결합니다. 이 플랫폼에는 다음과 같은 구성 요소가 포함되어 있으며 이과지오 데이터 과학 플랫폼이 다음 이미지에 나와 있습니다.

- Jupyter Notebooks, 통합 분석 엔진 및 Python 패키지를 포함하는 데이터 과학 워크벤치입니다
- 실험 추적 및 자동화된 파이프라인 기능을 사용하여 관리 모델링
- 확장 가능한 Kubernetes 클러스터를 통해 데이터 및 ML 서비스를 관리했습니다
- Nuclio는 실시간 서버리스 기능 프레임워크입니다
- SQL, NoSQL, 시계열 데이터베이스, 파일(단순한 개체), 스트리밍을 지원하는 매우 빠르고 안전한 데이터 계층입니다
- NetApp, Amazon S3, HDFS, SQL 데이터베이스, 스트리밍 또는 메시징 프로토콜 등의 타사 데이터 소스와 통합
- Grafana 기반의 실시간 대시보드



소프트웨어 및 하드웨어 요구 사항

네트워크 구성

다음은 클라우드에서 설정하기 위한 네트워크 구성 요구 사항입니다.

- Iguazio 클러스터와 NetApp Cloud Volumes는 동일한 가상 프라이빗 클라우드에 있어야 합니다.
- 클라우드 관리자는 Iguazio 앱 노드의 포트 6443에 액세스할 수 있어야 합니다.
- 이 기술 보고서에서는 Amazon Web Services를 사용했습니다. 그러나 사용자는 모든 클라우드 공급자에 솔루션을 배포할 수 있습니다. NVIDIA DGX-1을 사용하는 ONTAP AI에서의 온프레미스 테스트를 위해 이과지오에서 호스팅하는 DNS 서비스를 편리하게 사용했습니다.

클라이언트는 동적으로 생성된 DNS 도메인에 액세스할 수 있어야 합니다. 고객은 원하는 경우 자체 DNS를 사용할 수 있습니다.

하드웨어 요구 사항

Iguazio는 자체 클러스터에 설치할 수 있습니다. NetApp은 NVIDIA DGX-1 시스템을 통해 NetApp ONTAP AI의 솔루션을 검증했습니다. 다음 표에는 이 솔루션을 테스트하는 데 사용되는 하드웨어가 정리되어 있습니다.

하드웨어	수량
DGX-1 시스템	1
NetApp AFF A800 시스템	1개의 고가용성(HA) 2노드에 컨트롤러 2개와 NVMe SSD 48개(3.8TB 이상) 포함
Cisco Nexus 3232C 네트워크 스위치	2

다음 표에는 사내 테스트에 필요한 소프트웨어 구성 요소가 나열되어 있습니다.

소프트웨어	버전 또는 기타 정보
NetApp ONTAP 데이터 관리 소프트웨어	9.7
Cisco NX-OS 스위치 펌웨어	7.0(3) I6(1)
NVIDIA DGX OS	4.4-Ubuntu 18.04 LTS
Docker 컨테이너 플랫폼	19.03.5
컨테이너 버전입니다	20.01-tf1-py2
머신 러닝 프레임워크	TensorFlow 1.15.0
이과시오	버전 2.8 이상
ESX Server를 선택합니다	6.5

이 솔루션은 Iguazio 버전 2.5 및 NetApp Cloud Volumes ONTAP for AWS로 완전히 테스트되었습니다. Iguazio 클러스터와 NetApp 소프트웨어가 모두 AWS에서 실행되고 있습니다.

소프트웨어	버전 또는 유형
이과시오	버전 2.8 이상
앱 노드	M5.4xLarge
데이터 노드	I3.4xLarge

네트워크 장치 오류 예측 사용 사례 요약

이 사용 사례는 아시아 통신 분야의 이과시오 고객을 기반으로 합니다. 매년 100K 기업 고객 및 125k 네트워크 중단 이벤트가 발생하면서 네트워크 장애가 고객에게 영향을 미치지 않도록 사전에 조치를 취하고 예측해야 했습니다. 이 솔루션은 다음과 같은 이점을 제공합니다.

- 네트워크 장애에 대한 예측 분석
- 발권 시스템과 통합
- 네트워크 장애를 방지하기 위한 사전 예방 조치 이과시오(Iguazio)의 구현 결과, 60%의 장애를 사전에 예방했습니다.

설정 개요

Iguazio는 온프레미스 또는 클라우드 공급자에 설치할 수 있습니다.

Iguazio 설치

프로비저닝은 서비스로 수행하고 Iguazio 또는 고객이 관리할 수 있습니다. 두 경우 모두 Iguazio는 클러스터를 배포 및 관리하기 위한 배포 애플리케이션(Provazio)을 제공합니다.

온-프레미스 설치의 경우 을 참조하십시오 ["NVA-1121"](#) 컴퓨팅, 네트워크 및 스토리지 설정을 위해 사용할 수 있습니다. Iguazio는 고객의 추가 비용 없이 구내 배치 서비스를 제공합니다. 을 참조하십시오 ["이 페이지"](#) DNS 및 SMTP 서버 구성의 경우 Provazio 설치 페이지는 다음과 같다.

×

New System (dev)

Installation Scenario

General

Clusters

Cloud

Bare metal / virtual machines

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

AWS

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

Azure

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

AWS (pre-provisioned)

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

Azure (pre-provisioned)

Installs the system on Microsoft Azure instances, manually provisioned beforehand

Advanced

Show advanced options in the next steps

BACK

NEXT

Kubernetes 클러스터 구성 중






이 섹션은 클라우드 및 온프레미스 구축을 위한 두 부분으로 나누어져 있습니다.





클라우드 구축 Kubernetes 구성

NetApp Cloud Manager를 통해 Iguazio Kubernetes 클러스터 연결을 정의할 수 있습니다. Trident를 사용하려면 클러스터의 여러 리소스에 액세스하여 볼륨을 사용할 수 있어야 합니다.

1. 액세스를 설정하려면 Iguazio 노드 중 하나에서 Kubernetes 구성 파일을 가져옵니다. 이 파일은 ``/home/Iguazio/.kube/config.'` 아래에 있습니다 이 파일을 바탕 화면에 다운로드합니다.
2. 구성할 클러스터 검색 으로 이동합니다.

4 Kubernetes Clusters

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

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

3. Kubernetes 구성 파일을 업로드합니다. 다음 이미지를 참조하십시오.

Upload Kubernetes Configuration File

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

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

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

Upload File

4. Trident를 구축하고 클러스터와 볼륨을 연결합니다. Iguazio 클러스터에 영구 볼륨 정의 및 할당에 대한 다음 이미지를 참조하십시오. 이 프로세스는 Iguazio의 Kubernetes 클러스터에 영구 볼륨(PV)을 만듭니다. 이를 사용하려면 먼저 영구 볼륨 클레임(PVC)을 정의해야 합니다.

Persistent Volumes for Kubernetes

Connected with Kubernetes Cluster

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

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

Kubernetes Cluster

Custom Export Policy (Optional) ⓘ

Select Kubernetes Cluster

kubernetes

Custom Export Policy

172.31.0.0/16

☒ Set as default storage class

☒ NFS ☐ iSCSI

Connect

Cancel

온프레미스 구축 **Kubernetes** 구성

NetApp Trident의 사내 설치에 대한 자세한 내용은 [을 참조하십시오](#) "TR-4798" 를 참조하십시오. Kubernetes 클러스터를 구성하고 NetApp Trident를 설치한 후 Trident를 Iguazio 클러스터에 연결하여 데이터 및 모델의 Snapshot 복사본 생성 등의 NetApp 데이터 관리 기능을 사용할 수 있습니다.

영구 볼륨 클레임을 정의합니다

1. 다음 YAML을 파일에 저장하여 기본 유형의 PVC를 생성합니다.

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

2. Iguazio Kubernetes 클러스터에 YAML 파일을 적용하십시오.

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

Jupyter Notebook에 NetApp Volume을 연결합니다

Iguazio는 데이터 과학자에게 AI/ML 애플리케이션의 개발 및 배포를 위한 완벽한 종단 간 스택을 제공하기 위해 여러 가지 관리 서비스를 제공합니다. 에서 이러한 구성 요소에 대해 자세히 알아볼 수 있습니다 "[Iguazio 응용 프로그램 서비스 및 도구 개요](#)".

관리 서비스 중 하나는 Jupyter Notebook입니다. 각 개발자는 개발에 필요한 리소스와 함께 노트북 컨테이너를 직접 배포할 수 있습니다. NetApp Cloud Volume에 대한 액세스 권한을 부여하려면 볼륨을 해당 컨테이너에 할당하고 리소스 할당, 실행 중인 사용자 및 영구 볼륨 청구에 대한 환경 변수 설정을 다음 이미지에 표시할 수 있습니다.

온-프레미스 구성의 경우 를 참조할 수 있습니다 "[TR-4798](#)" 버전 관리를 위한 데이터 또는 모델의 스냅샷 복사본 생성 등 NetApp ONTAP 데이터 관리 기능을 지원하는 Trident 설정에서 Trident 백엔드 구성 파일에 다음 줄을 추가하여 스냅샷 디렉토리를 표시합니다.

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

JSON 형식으로 Trident 백 엔드 구성 파일을 생성한 후 다음을 실행해야 합니다 "[Trident 명령](#)" 이를 참조하려면:

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

The screenshot shows the Jupyter Notebook configuration interface. At the top, there is a toggle for "Enabled" and a slider for "Inactivity window" with values 5m, 10m, 1h, 2h, and 4h. Below this is the "Resources" section, which includes a link to "Kubernetes documentation" and a note: "The memory and CPU configurations are applied to each replica." There are two rows of configuration for "Memory" and "CPU", each with "Request" and "Limit" fields. The "Running User" section at the bottom has a text input field with "admin" and a "Username" label.

The screenshot shows the Jupyter Notebook configuration interface. At the top, there is a "Flavor" dropdown menu with "Full stack without GPU" selected. Below this is a "Spark" dropdown menu with "spark" selected and a "Create new..." link. The "Environment Variables" section has a "Create a new environment variable" button. The "Persistent Volume Claims (PVCs)" section has a table with columns "Name" and "Mount Path". The table has one row with "basic" in the "Name" column and "/netapp" in the "Mount Path" column. There is an "Add PVC" button at the bottom.

응용 프로그램 배포

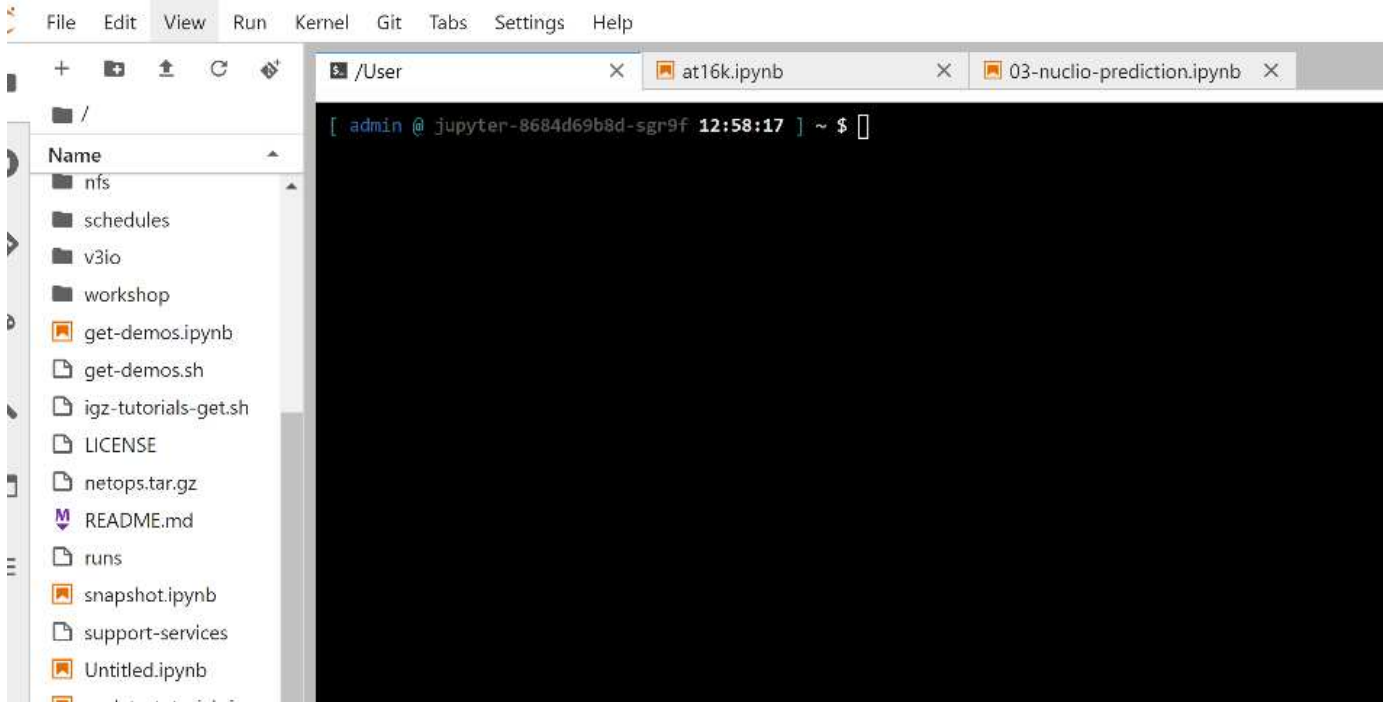
다음 섹션에서는 응용 프로그램을 설치하고 배포하는 방법에 대해 설명합니다.

GitHub에서 코드를 가져옵니다

이제 Iguazio 클러스터와 개발자 환경에서 NetApp Cloud Volume 또는 NetApp Trident 볼륨을 사용할 수 있으므로 애플리케이션 검토를 시작할 수 있습니다.

사용자는 고유한 작업 공간(디렉터리)을 가지고 있습니다. 모든 노트북에서 사용자 디렉토리 경로는 "/User"입니다. Iguazio 플랫폼은 디렉토리를 관리합니다. 위의 지침을 따르면 NetApp Cloud 볼륨은 '/NetApp' 디렉토리에 있습니다.

Jupyter 터미널을 사용하여 GitHub에서 코드를 가져옵니다.



Jupyter 터미널 프롬프트에서 프로젝트를 복제합니다.

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

이제 Jupyter 작업 공간의 파일 트리에 'netops-netapp' 폴더가 표시됩니다.

작업 환경을 구성합니다

'노트북'의 et_env-example.ipynb를 'et_env.ipynb'로 복사합니다. 'et_env.ipynb'를 열고 편집합니다. 이 노트북은 자격 증명, 파일 위치 및 실행 드라이버에 대한 변수를 설정합니다.

위의 지침을 따르면 다음 단계만 변경할 수 있습니다.

1. 이과지오 서비스 대시보드에서 이 값을 'docker_registry'로 얻습니다

예: dddocker-registry.default-tenant.app.clusterq.iguaziodev.com:80`

2. 'admin'을 Iguazio 사용자 이름으로 변경합니다.

```
"IGZ_container_path='/users/admin'"
```

다음은 ONTAP 시스템 연결 세부 정보입니다. Trident를 설치할 때 생성한 볼륨 이름을 포함합니다. 다음은 온-프레미스 ONTAP 클러스터에 대한 설정입니다.

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

Cloud Volumes ONTAP에 대한 설정은 다음과 같습니다.

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

기본 **Docker** 이미지를 생성합니다

ML 파이프라인을 구축하는 데 필요한 모든 것이 Iguazio 플랫폼에 포함되어 있습니다. 개발자는 파이프라인을 실행하고 Jupyter Notebook에서 이미지 생성을 실행하는 데 필요한 Docker 이미지의 사양을 정의할 수 있습니다. 노트북 'create-images.ipynb'를 열고 모든 셀을 실행합니다.

이 노트북은 파이프라인에서 사용하는 두 개의 이미지를 만듭니다.

- "iguazio/NetApp. ML 작업을 처리하는 데 사용됩니다.

Create image for training pipeline

```
[4]: fn.build_config(image=docker_registry+'/iguazio/netapp', commands=['pip install \
v3io_frames fsspec>=0.3.3 PyYAML==5.1.2 pyarrow==0.15.1 pandas==0.25.3 matplotlib seaborn yellowb
fn.deploy()
```

- 'NetApp/파이프라인' NetApp Snapshot 복사본을 처리하는 유틸리티를 포함합니다.

Create image for Ontap utilites

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

개별 **Jupyter Notebooks**를 검토합니다

다음 표에는 이 작업을 만드는 데 사용한 라이브러리와 프레임워크가 나와 있습니다. 이러한 모든 구성 요소는 Iguazio의 역할 기반 액세스 및 보안 제어와 완벽하게 통합되어 있습니다.

라이브러리/프레임워크	설명
MLRun	머신 러닝/AI 파이프라인을 조립, 실행 및 모니터링할 수 있도록 Iguazio에서 관리합니다.
뉴클레오	서버를 사용하지 않는 함수 프레임워크가 Iguazio와 통합되었습니다. 이과지오(Iguazio)에서 관리하는 오픈 소스 프로젝트로도 사용할 수 있습니다.
Kubeflow	파이프라인을 구축하는 Kubernetes 기반 프레임워크 이과지오가 기여하는 오픈 소스 프로젝트이기도 합니다. 이과지오와 통합되어 나머지 인프라와의 보안 및 통합을 강화합니다.
Docker 를 참조하십시오	Docker 레지스트리는 Iguazio 플랫폼에서 서비스로 실행됩니다. 이 설정을 변경하여 레지스트리에 연결할 수도 있습니다.
NetApp Cloud Volumes를 참조하십시오	AWS에서 실행되는 Cloud Volumes를 통해 대량의 데이터에 액세스하고 Snapshot 복사본을 교육에 사용되는 데이터 세트의 버전으로 만들 수 있습니다.
트라이던트	Trident는 NetApp에서 관리하는 오픈 소스 프로젝트입니다. Kubernetes에서 스토리지 및 컴퓨팅 리소스와 통합할 수 있도록 지원합니다.

우리는 여러 노트북을 사용하여 ML 파이프라인을 구성하였습니다. 각 노트북은 파이프라인에 함께 들어가기 전에 개별적으로 테스트할 수 있습니다. 이 데모 애플리케이션의 배포 흐름에 따라 각 노트북을 개별적으로 다룹니다.

원하는 결과는 데이터의 스냅샷 복사본을 기반으로 모델을 교육하고 추론을 위해 모델을 구축하는 파이프라인입니다. 완료된 MLRun 파이프라인에 대한 블록 다이어그램이 다음 이미지에 나와 있습니다.



데이터 생성 기능 구축

이 섹션에서는 Nuclio 서버리스 기능을 사용하여 네트워크 장치 데이터를 생성하는 방법에 대해 설명합니다. 이 활용 사례는 파이프라인을 구축한 이과지오 클라이언트에서 수정되었으며 이과지오 서비스를 사용하여 네트워크 디바이스 장애를 모니터링하고 예측합니다.

네트워크 장치에서 오는 데이터를 시뮬레이션했습니다. Jupyter Notebook `data-generator.ipynb`를 실행하면 10분마다 실행되는 서버리스 기능이 생성되고 새 데이터가 있는 Parquet 파일이 생성됩니다. 함수를 배포하려면 이 전자 필기장의 모든 셀을 실행합니다. 를 참조하십시오 ["Nuclio 웹 사이트"](#) 이 노트북에서 잘 모르는 구성 요소를 검토합니다.

함수를 생성할 때 다음 주석이 있는 셀은 무시됩니다. 노트북의 모든 셀은 기능의 일부로 간주됩니다. `'%nuclio magic'` 을(를) 활성화하려면 Nuclio 모듈을 가져옵니다.

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

기능에 대한 사양에서는 함수가 실행되는 환경, 함수가 트리거되는 방식 및 사용되는 리소스를 정의했습니다.

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

INIT_CONTEXT 함수는 함수 초기화 시 Nuclio Framework에 의해 호출됩니다.

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

함수에 없는 코드는 함수가 초기화될 때 호출됩니다. 이 함수를 호출하면 처리기 함수가 실행됩니다. 처리기의 이름을 변경하고 함수 사양에서 지정할 수 있습니다.

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

배포 전에 노트북에서 기능을 테스트할 수 있습니다.

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

이 기능은 노트북에서 배포하거나 CI/CD 파이프라인에서 배포할 수 있습니다(이 코드 조정).

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

파이프라인 노트북

이 노트북은 이 설정을 위해 개별적으로 실행할 수 없습니다. 이 내용은 각 전자 필기장에 대한 검토일 뿐입니다. 파이프라인을 구성하는 요소로 호출한 것입니다. 개별적으로 실행하려면 MLRun 설명서를 검토하여 Kubernetes 작업으로 실행합니다.

SNAP_CV.iynb

이 노트북은 파이프라인의 시작 부분에 있는 Cloud Volume Snapshot 복사본을 처리합니다. 볼륨의 이름을 파이프라인 컨텍스트로 전달합니다. 이 노트북은 스냅샷 복사본을 처리하기 위해 셀 스크립트를 호출합니다. 파이프라인에서 실행되는 동안 실행 컨텍스트에는 실행에 필요한 모든 파일을 찾는 데 도움이 되는 변수가 포함되어 있습니다. 이 코드를

작성하는 동안 개발자는 이 코드를 실행하는 컨테이너의 파일 위치에 대해 걱정할 필요가 없습니다. 나중에 설명했듯이 이 응용 프로그램은 모든 종속성을 포함하여 배포되며 실행 컨텍스트를 제공하는 파이프라인 매개 변수의 정의입니다.

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

생성된 스냅샷 복사본 위치는 파이프라인의 단계에서 사용할 MLRun 컨텍스트에 배치됩니다.

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

다음 세 개의 노트북은 병렬로 실행됩니다.

데이터 준비 .ipynb

원시 메트릭을 기능으로 전환하여 모델 교육을 활성화해야 합니다. 이 노트북은 Snapshot 디렉토리에서 원시 메트릭을 읽고 모델 훈련을 위한 기능을 NetApp 볼륨에 씁니다.

파이프라인 컨텍스트에서 실행되는 경우 입력 DATA_DIR에 스냅샷 복사 위치가 포함됩니다.

```
metrics_table = os.path.join(str(mlruncontext.get_input('DATA_DIR',
os.getenv('DATA_DIR', '/netpp'))),
                             mlruncontext.get_param('metrics_table',
os.getenv('metrics_table', 'netops_metrics_parquet')))
```

ipynb 설명

수신 메트릭을 시각화하기 위해 Kubeflow 및 MLRun UI를 통해 사용할 수 있는 플롯 및 그래프를 제공하는 파이프라인 단계를 배포합니다. 각 실행에는 이 시각화 도구의 고유 버전이 있습니다.

```
ax.set_title("features correlation")
plt.savefig(os.path.join(base_path, "plots/corr.png"))
context.log_artifact(PlotArtifact("correlation", body=plt.gcf()),
local_path="plots/corr.html")
```

Deploy-feature-function.ipynb

NetApp은 이상 징후를 찾기 위한 메트릭을 지속적으로 모니터링합니다. 이 노트북은 들어오는 메트릭에 대한 예측을 실행하는 데 필요한 기능을 생성하는 서버리스 기능을 생성합니다. 이 노트북은 함수 생성을 호출합니다. 기능 코드는 노트북 data-prep.ipynb에 있다. 이러한 목적을 위해 파이프라인에서 한 단계씩 동일한 전자 필기장을 사용합니다.

훈련.iynb

피처를 작성한 후 모델 교육을 시작합니다. 이 단계의 출력은 추론을 위해 사용할 모델입니다. 또한 각 실행(실험)을 추적하기 위해 통계를 수집합니다.

예를 들어 다음 명령은 해당 실험의 컨텍스트에 정확도 점수를 입력합니다. 이 값은 Kubeflow 및 MLRun에서 볼 수

있습니다.

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

deploy-추론-function.ipynb입니다

파이프라인의 마지막 단계는 모델을 서버리스 기능으로 구축하여 연속 추론을 수행하는 것입니다. 이 노트북은 'nuclio-추론-function.ipynb'에 정의된 서버리스 기능의 생성을 호출합니다.

파이프라인 검토 및 구축

파이프라인에서 모든 노트북을 함께 실행할 경우 실험을 지속적으로 실행하여 새로운 측정 지표를 기준으로 모델의 정확성을 재평가할 수 있습니다. 먼저 파이프라인 ipynb 노트북을 엽니다. NetApp과 Iguazio가 이 ML 파이프라인 구축을 단순화하는 방법을 자세히 설명 드리겠습니다.

MLRun을 사용하여 컨텍스트를 제공하고 파이프라인의 각 단계에 대한 리소스 할당을 처리합니다. MLRun API 서비스는 Iguazio 플랫폼에서 실행되며 Kubernetes 리소스와 상호 작용하는 지점입니다. 각 개발자는 리소스를 직접 요청할 수 없습니다. API는 요청을 처리하고 액세스 제어를 활성화합니다.

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

파이프라인은 NetApp Cloud Volumes 및 온프레미스 볼륨과 함께 사용할 수 있습니다. Cloud Volumes를 사용하기 위해 이 데모를 구축했지만 코드에서 온프레미스 실행 옵션을 확인할 수 있습니다.

```
# Initialize the NetApp snap function once for all functions in a notebook
if [ NETAPP_CLOUD_VOLUME ]:
    snapfn =
code_to_function('snap',project='NetApp',kind='job',filename="snap_cv.ipyn
b").apply(mount_v3io())
    snap_params = {
        "metrics_table" : metrics_table,
        "NETAPP_MOUNT_PATH" : NETAPP_MOUNT_PATH,
        'MANAGER' : MANAGER,
        'svm' : svm,
        'email': email,
        'password': password ,
        'weid': weid,
        'volume': volume,
        "APP_DIR" : APP_DIR
    }
else:
    snapfn =
code_to_function('snap',project='NetApp',kind='job',filename="snapshot.ipyn
b").apply(mount_v3io())
...
snapfn.spec.image = docker_registry + '/netapp/pipeline:latest'
snapfn.spec.volume_mounts =
[snapfn.spec.volume_mounts[0],netapp_volume_mounts]
    snapfn.spec.volumes = [ snapfn.spec.volumes[0],netapp_volumes]
```

Jupyter 노트북을 Kubeflow 단계로 전환하는 데 필요한 첫 번째 작업은 코드를 함수로 전환하는 것입니다. 기능에는 해당 노트북을 실행하는 데 필요한 모든 사양이 있습니다. 전자 필기장을 아래로 스크롤하면 파이프라인의 모든 단계에 대한 기능을 정의하는 것을 볼 수 있습니다.

노트북의 일부입니다	설명
code_to_function> (MLRun 모듈의 일부)	함수 이름: 프로젝트 이름. 모든 프로젝트 아티팩트를 구성하는 데 사용됩니다. 이것은 MLRun UI에서 볼 수 있습니다. 있습니다. 이 경우에는 Kubernetes 작업입니다. 이는 Dask, MPI, 스파크k8s 등이 될 수 있습니다. 자세한 내용은 MLRun 설명서를 참조하십시오. 파일. 전자 필기장의 이름입니다. Git(HTTP)의 위치일 수도 있습니다.
이미지	이 단계에서 사용 중인 Docker 이미지의 이름입니다. 앞에서 create-image.ipynb 전자 필기장으로 이 기능을 만들었습니다.
volume_mounts 및 volume	런타임에 NetApp Cloud Volume을 마운트하기 위한 세부 정보

단계에 대한 매개 변수도 정의합니다.

```

params={
    "FEATURES_TABLE":FEATURES_TABLE,
    "SAVE_TO" : SAVE_TO,
    "metrics_table" : metrics_table,
    'FROM_TSDB': 0,
    'PREDICTIONS_TABLE': PREDICTIONS_TABLE,
    'TRAIN_ON_LAST': '1d',
    'TRAIN_SIZE':0.7,
    'NUMBER_OF_SHARDS' : 4,
    'MODEL_FILENAME' : 'netops.v3.model.pickle',
    'APP_DIR' : APP_DIR,
    'FUNCTION_NAME' : 'netops-inference',
    'PROJECT_NAME' : 'netops',
    'NETAPP_SIM' : NETAPP_SIM,
    'NETAPP_MOUNT_PATH': NETAPP_MOUNT_PATH,
    'NETAPP_PVC_CLAIM' : NETAPP_PVC_CLAIM,
    'IGZ_CONTAINER_PATH' : IGZ_CONTAINER_PATH,
    'IGZ_MOUNT_PATH' : IGZ_MOUNT_PATH
}

```

모든 단계에 대한 함수 정의가 있으면 파이프라인을 구성할 수 있습니다. 우리는 이 정의를 만들기 위해 'kfp' 모듈을 사용합니다. MLRun을 사용하는 것과 자체적으로 구축하는 것의 차이점은 코딩의 단순화 및 단축입니다.

정의한 기능은 MLRun의 AS_STEP 기능을 이용하여 STEP 부품으로 변환한다.

스냅샷 단계 정의

스냅샷 기능을 시작하고 v3io를 소스로 출력 및 마운트합니다.

```

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

```

매개 변수	세부 정보
새 작업	NewTask 는 함수 실행의 정의입니다.
(MLRun 모듈)	핸들러. 호출할 Python 함수의 이름입니다. 전자 필기장에서 이름 처리기를 사용했지만 필수 사항은 아닙니다. 매개 변수 실행에 전달된 매개 변수. 코드 안에서 context.get_param('parameter')을 사용하여 값을 가져옵니다.
AS_STEP	이름. Kubeflow 파이프라인 단계의 이름입니다. 출력. 이 값은 완료 시 단계에서 사전에 추가하는 값입니다. SNAP_CV.iynb 노트북을 살펴보세요. mount_v3io(). 이를 통해 파이프라인을 실행하는 사용자에게 대해 /User를 마운트하는 단계를 구성합니다.

```

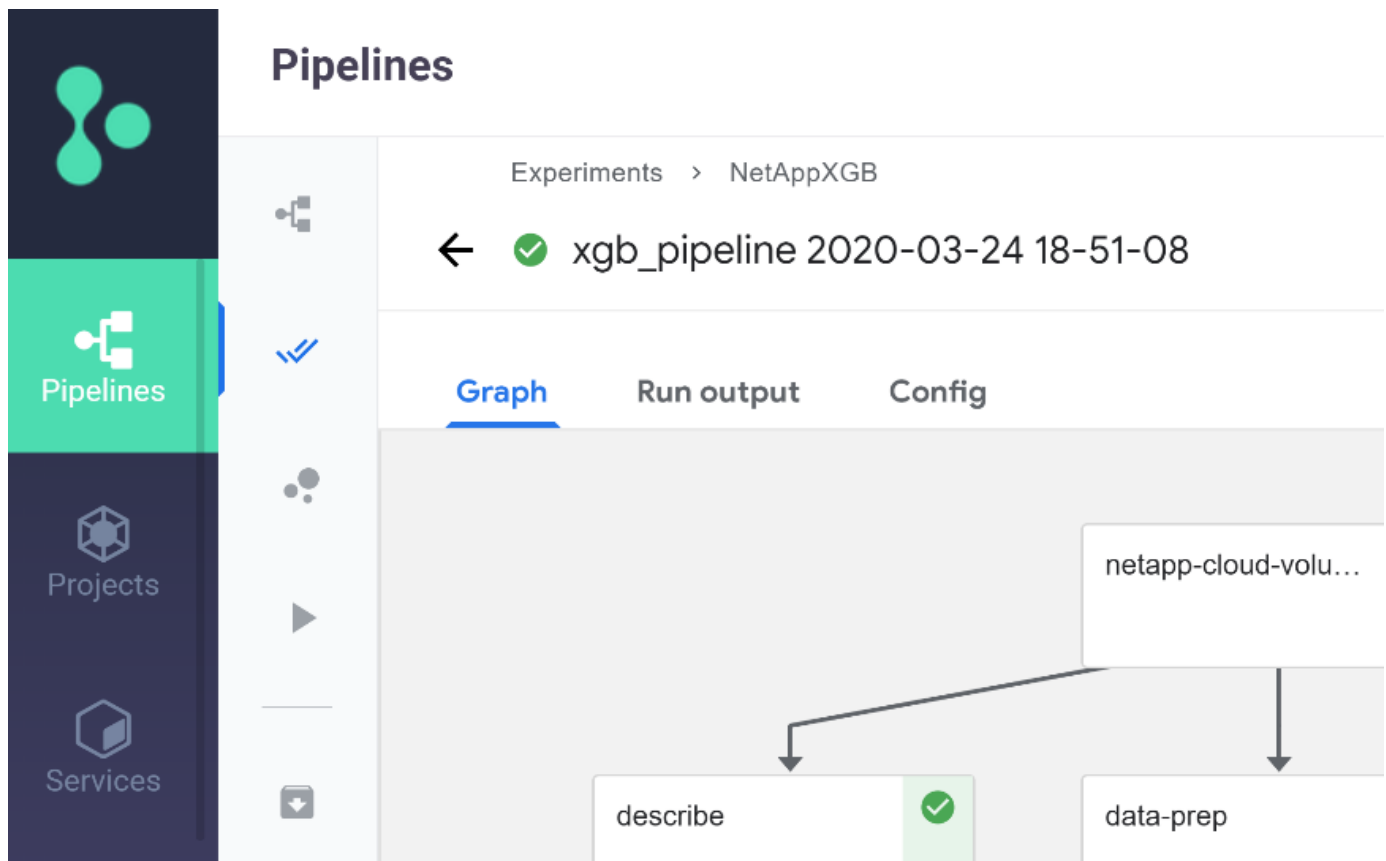
prep = data_prep.as_step(name='data-prep',
handler='handler',params=params,
                        inputs = {'DATA_DIR':
snap.outputs['snapVolumeDetails']} ,

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

```

매개 변수	세부 정보
입력	이전 단계의 출력을 단계별로 전달할 수 있습니다. 이 경우 snap.outputs ['sapVolumeDetails']는 스냅 단계에서 생성한 스냅샷 복사본의 이름입니다.
아웃_경로	MLRun 모듈 log_artifacts를 사용하여 생성하는 아티팩트를 배치할 위치입니다.

pipeline.ipynb는 위에서 아래로 실행할 수 있다. 그런 다음 Iguazio 대시보드에서 Pipelines 탭으로 이동하여 Iguazio 대시보드 파이프라인 탭에 표시된 진행 상황을 모니터링할 수 있습니다.



모든 러닝에서 훈련 단계의 정확성을 기록했기 때문에 훈련 정확도 기록에서도 볼 수 있듯이 각 실험마다 정확한 기록을 가지고 있습니다.

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

스냅샷 단계를 선택하면 이 실험을 실행하는 데 사용된 스냅샷 복사본의 이름을 볼 수 있습니다.

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

Artifacts Input/Output Volumes Manifest Logs

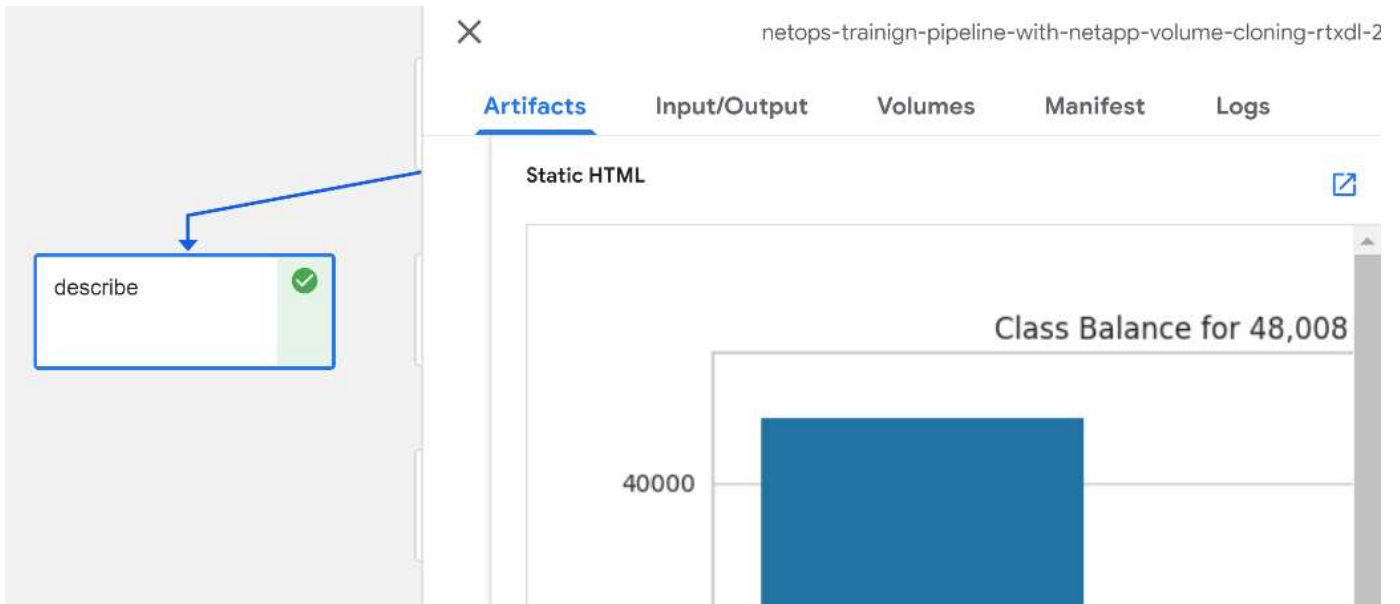
input artifacts

Output parameters

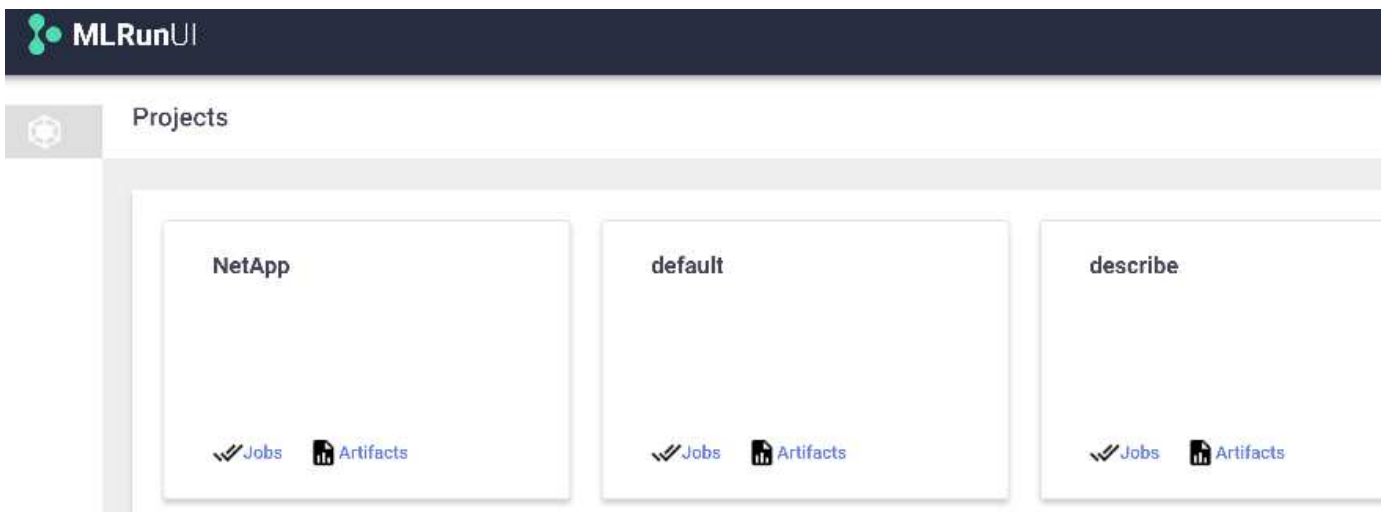
netapp-cloud-volume-snapshot-snapVolumeDetails	/netapp/.snapshot/kfp_20200324_185122
netapp-cloud-volume-snapshot-training_parquet_file	/netapp/.snapshot/kfp_20200324_18512...

Output artifacts

설명된 단계에는 우리가 사용한 지표를 탐색할 수 있는 시각적 인공물이 있습니다. 다음 이미지와 같이 전체 플롯을 보기 위해 확장할 수 있습니다.



또한 MLRun API 데이터베이스는 프로젝트별로 구성된 각 실행의 입력, 출력 및 아티팩트를 추적합니다. 각 시리즈의 입력, 출력 및 아티팩트의 예는 다음 영상에서 확인할 수 있습니다.



각 직무마다 추가 세부 정보를 저장합니다.

Name

deploy-model

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

xgb_train

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

data-prep

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

describe

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

deploy-features-function

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

NetApp_Cloud_Volume_Sna

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

describe

24 Mar, 14:52:45

Info

Inputs

Artifacts

Results

Logs

UID

66ef22187efb4ad89e8da8433c2a460e

Start time

24 Mar, 14:52:45

Parameters

Completed

Results

class_label...

key: summary

label_colu...

MLRun에 대한 자세한 내용은 이 문서에서 다룰 수 있는 것보다 많습니다. 단계와 함수의 정의를 비롯한 AI 아티팩트는 API 데이터베이스에 저장하고 버전을 지정한 후 개별 또는 전체 프로젝트로 호출할 수 있습니다. 프로젝트를 저장하고 나중에 사용할 수 있도록 Git에 푸시할 수도 있습니다. 자세한 내용은 에서 확인하시기 바랍니다 "[MLRun GitHub 사이트](#)".








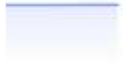











Grafana 대시보드 배포

모든 것이 배포되면 새 데이터에 대한 추론을 실행합니다. 이 모델은 네트워크 장치 장비의 고장을 예측합니다. 예측 결과는 Iguazio 시간 계열 테이블에 저장됩니다. Iguazio의 보안 및 데이터 액세스 정책과 통합된 플랫폼을 통해 Grafana로 결과를 시각화할 수 있습니다.

제공된 JSON 파일을 클러스터의 Grafana 인터페이스로 가져와 대시보드를 구축할 수 있습니다.

1. Grafana 서비스가 실행 중인지 확인하려면 서비스 아래를 살펴보세요.

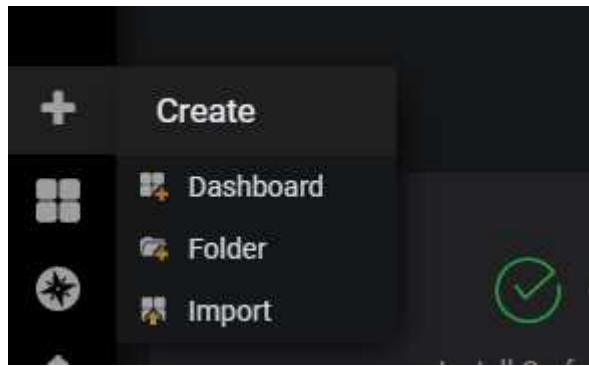
Services

<input type="checkbox"/>	Name ↑	Running User	Version ↕	CPU (cores)	Memory	AF
<input type="checkbox"/>	 docker-registry Type: Docker Regi		2.7.1	96μ 	1.67 GB 	H
<input type="checkbox"/>	 framesd Type: V3ID Frame		0.6.10	369μ 	795.19 MB 	H
<input type="checkbox"/>	 grafana Type: Grafana		6.6.0	1m 	38.39 MB 	
<input type="checkbox"/>	 jupyter Type: Jupyter Note	admin	1.0.2	81m 	3.27 GB 	
<input type="checkbox"/>	 log-forwarder Type: Log forward		6.7.2	0 	0 bytes 	

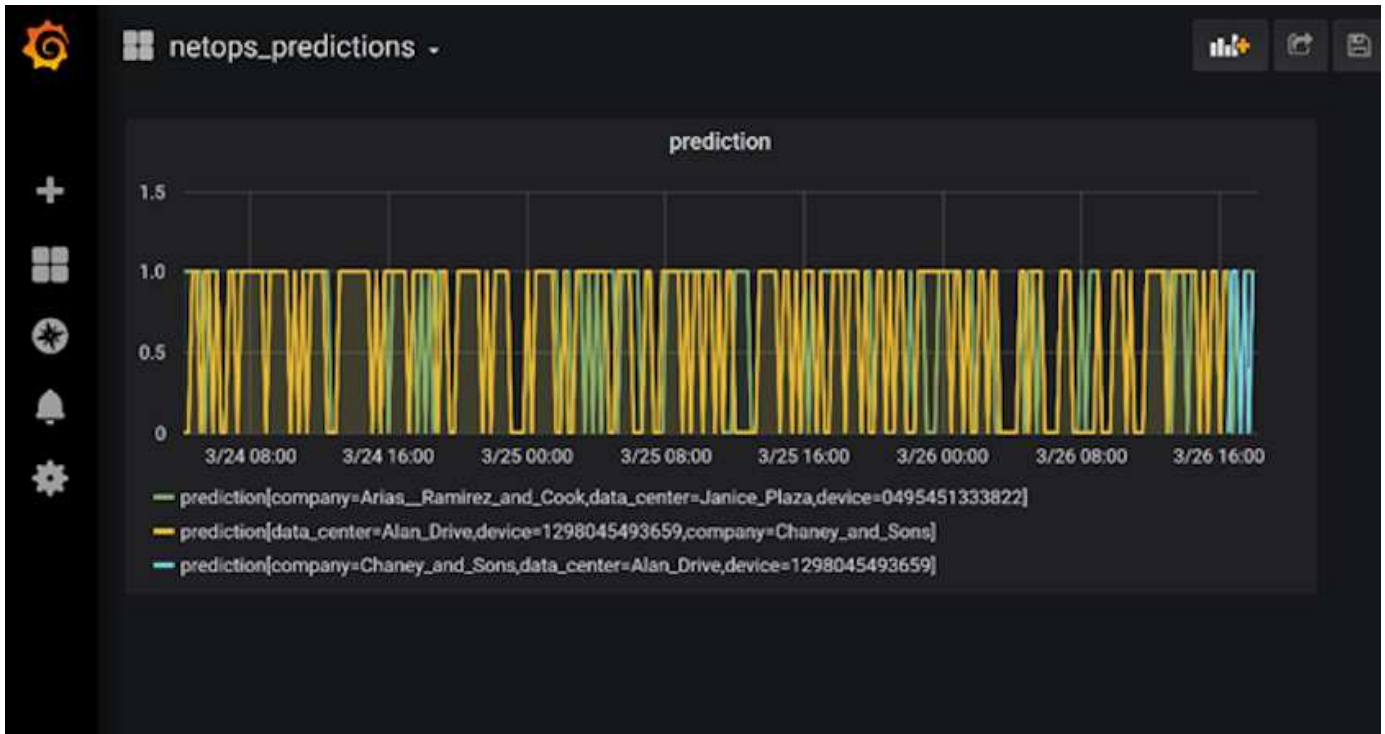
2. 이 인스턴스가 없으면 서비스 섹션에서 인스턴스를 배포합니다.
 - a. 새 서비스를 클릭합니다.
 - b. 목록에서 Grafana를 선택합니다.
 - c. 기본값을 그대로 사용합니다.
 - d. 다음 단계를 클릭합니다.
 - e. 사용자 ID를 입력합니다.
 - f. 서비스 저장 을 클릭합니다.
 - g. 맨 위에 있는 변경 내용 적용을 클릭합니다.
3. 대시보드를 배포하려면 Jupyter 인터페이스를 통해 NetopsPredictions-Dashboard.json 파일을 다운로드합니다.



4. 서비스 섹션에서 Grafana를 열고 대시보드를 가져옵니다.



5. Upload(*.json) File을 클릭하고 앞서 다운로드한 파일('NetopsPredictions - Dashboard.json')을 선택한다. 업로드가 완료되면 대시보드가 표시됩니다.



정리 기능을 배포합니다

많은 데이터를 생성할 때 정리 및 정리하는 것이 중요합니다. 이를 위해 정리 기능을 정리 iynb 공책에 배치한다.

이점

NetApp과 Iguazio는 Docker, Kubernetes와 같은 오케스트레이션 툴과 함께 Kubeflow, Apache Spark 및 TensorFlow 등의 필수 프레임워크를 구축하여 AI 및 ML 애플리케이션의 구축 속도를 높이고 단순화합니다. NetApp과 Iguazio는 엔드 투 엔드 데이터 파이프라인을 통합함으로써 수많은 고급 컴퓨팅 워크로드에서 발생하는 지연 시간과 복잡성을 줄이고 개발 및 운영 간의 격차를 효과적으로 좁혀줍니다. 데이터 과학자는 대규모 데이터 세트에서 쿼리를 실행하고 교육 단계 동안 권한 있는 사용자와 데이터 및 알고리즘 모델을 안전하게 공유할 수 있습니다. 컨테이너식 모델이 운영 환경에 준비되면 개발 환경에서 운영 환경으로 쉽게 이동할 수 있습니다.

결론

자체 AI/ML 파이프라인을 구축할 때는 아키텍처 구성 요소의 통합, 관리, 보안 및 접근성을 구성하는 것이 매우 어렵습니다. 개발자가 자신의 환경에 액세스하고 제어하도록 하는 것은 또 다른 도전 과제입니다.

NetApp과 Iguazio는 이러한 기술을 관리형 서비스로 결합하여 기술 채택을 가속하고 새로운 AI/ML 애플리케이션의 출시 시기를 앞지였습니다.

TR-4915: AI 및 분석 워크플로우를 위해 E-Series 및 BeeGFS로 데이터 이동

Cody Harryman과 NetApp의 Ryan Rodine이 함께 합니다

TR-4915에서는 NetApp E-Series SAN 스토리지를 통해 데이터 저장소에서 BeeGFS 파일 시스템으로 데이터를 이동하는 방법에 대해 설명합니다. 고객은 인공지능(AI) 및 머신 러닝(ML)

애플리케이션의 경우 모델 개발을 위해 페타바이트급 데이터가 넘는 대규모 데이터 세트를 BeeGFS 클러스터로 이동해야 할 수 있습니다. 이 문서에서는 NetApp XCP 및 NetApp BlueXP 복사 및 동기화 툴을 사용하여 이 작업을 수행하는 방법에 대해 알아봅니다.

"TR-4915: AI 및 분석 워크플로우를 위해 E-Series 및 BeeGFS로 데이터 이동"

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.