설치 개요

Astra Data Store

NetApp
February 16, 2022

This PDF was generated from https://docs.netapp.com/ko-kr/astra-data-store/get-started/install-ads.html on February 16, 2022. Always check docs.netapp.com for the latest.

목차

Astra Data Store 설치 개요

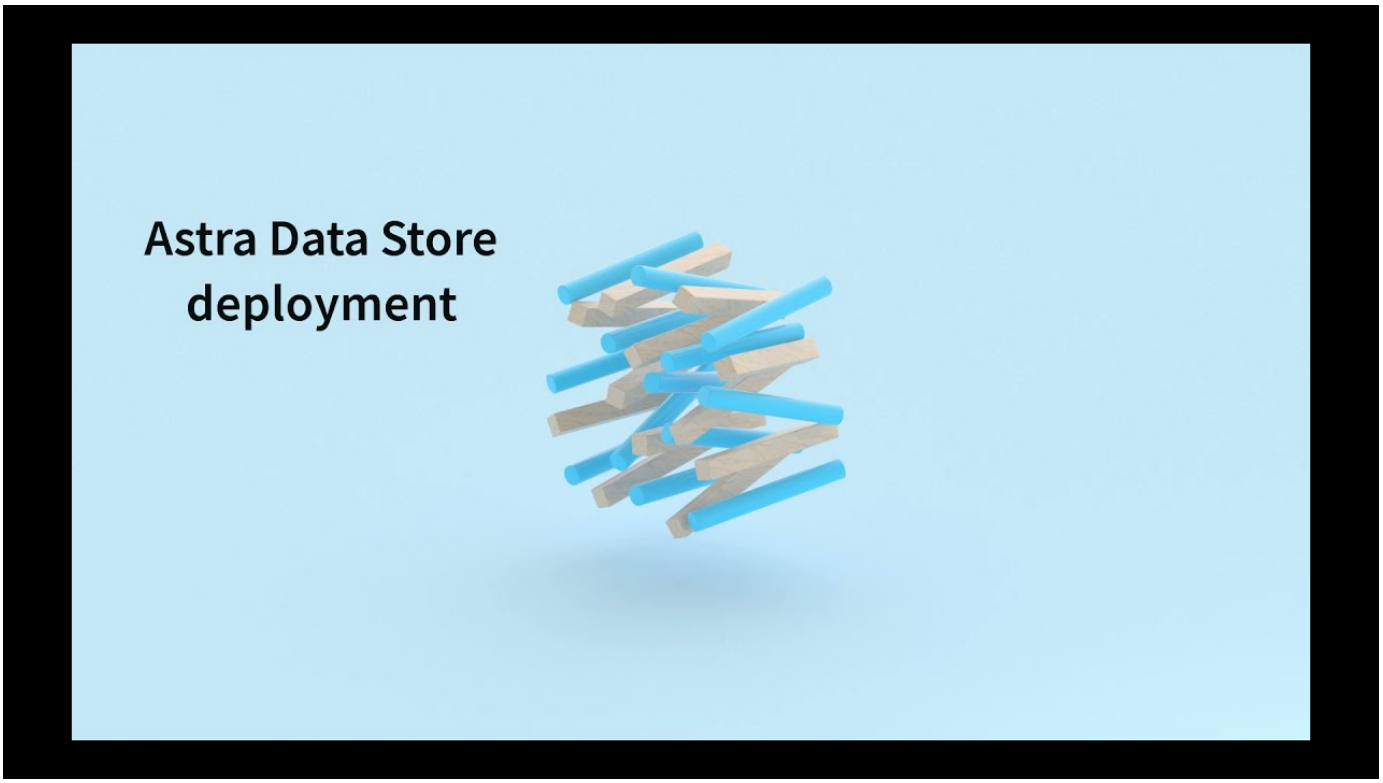
다음 Astra Data Store 설치 절차 중 하나를 선택하여 완료합니다.

- "표준 프로세스를 사용하여 Astra Data Store를 설치합니다".
- "Red Hat OpenShift를 사용하는 경우 OpenShift를 사용하여 Astra Data Store를 설치합니다".

Astra Data Store 미리 보기를 설치합니다

Astra Data Store 미리 보기를 설치하려면 에서 설치 번들을 다운로드하십시오 "NetApp Support 사이트" 및 이 절차에 설명된 설치 단계를 완료합니다.

또는 을(를) 사용할 수 있습니다 "Red Hat OpenShift Container Platform(OCP)에 Astra Data Store Preview 설치".



무엇을 '필요로 할거야

- "설치를 시작하기 전에 Astra Data Store 미리 보기 구축을 위한 환경을 준비합니다".
- 에 액세스합니다 "NetApp Support 사이트". 전체 액세스 권한이 없는 NetApp Support 사이트 계정이 없는 경우 미리 보기를 위해
- A "NetApp 라이센스 파일(NLF)" Astra Data Store 미리 보기입니다. 라이센스 다운로드 지침이 사용자에게 전송됩니다 "가입하세요".
- 활성 컨텍스트에 대한 클러스터 관리자 권한이 있는 활성 kubecononfig.
- 에 대한 이해 "역할 및 권한" Astra Data Store 미리 보기에서 사용합니다.
- 인터넷 연결. Astra Data Store 미리 보기는 대기 환경을 지원하지 않습니다. 직접 또는 프록시를 통해 support.netapp.com 에 접속하려면 인터넷 연결이 필요합니다.

Astra Data Store 미리 보기 설치 프로세스는 다음과 같은 고급 단계를 안내합니다.

- [Download the Astra Data Store preview bundle and extract the images]
- [Copy the binary and push images to your local registry]
- [Install the Astra Data Store preview operator]
- [Deploy the Astra Data Store preview version YAML]
- [Apply the Astra Data Store preview license]
- [Install the Astra Data Store preview cluster]
- [Understand deployment-related events]
- [Configure Astra Data Store preview monitoring]

Astra Data Store 미리 보기를 사용하여 비밀이 있는 이미지 등록부와 함께 작업하려면 을 참조하십시오 "이 KB입니다 ".

Astra Data Store 미리보기 번들을 다운로드하고 이미지를 추출합니다

1. 에 로그인합니다 "NetApp Support 사이트" 그리고 Astra Data Store 미리보기 번들('2021.12_astradatastore.tar')을 다운로드하십시오.
2. (선택 사항) 다음 명령을 사용하여 번들의 서명을 확인합니다.

```
openssl dgst -sha256 -verify 2021.12_astradatastore.pub -signature
2021.12_astradatastore.sig 2021.12_astradatastore.tar
```

3. 이미지 추출:

```
tar -xvf 2021.12_astradatastore.tar
```

이진 파일을 복사하고 이미지를 로컬 레지스트리에 푸시합니다

1. k8kubbeck 바이너리가 설치된 표준 경로(예: "/usr/bin/")로 이미지를 추출하기 위해 사용한 디렉토리에서 kubbctl-astrs 바이너리를 복사합니다. kubbctl-astrs는 Astra Data Store Preview 클러스터를 설치 및 관리하는 사용자 지정 kubtl 확장입니다.

```
cp -p ./bin/kubectl-astrads /usr/bin/.
```

2. Astra Data Store 미리보기 이미지 디렉토리의 파일을 로컬 레지스트리에 추가합니다.

   아래 이미지의 자동 로드에 대한 샘플 스크립트를 참조하십시오.

   a. 레지스트리에 로그인합니다.

```
docker login [your_registry_path]
```

b. 환경 변수를 레지스트리 경로로 설정하여 Astra Data Store 미리 보기 이미지를 푸시합니다(예: repo.company.com`).

```
export REGISTRY=repo.company.com/astrads
```

c. 스크립트를 실행하여 이미지를 Docker에 로드하고, 이미지에 태그를 지정하고, 를 눌러 이미지를 로컬 레지스트리에 로드합니다.

```
for astraImageFile in $(ls images/*.tar) ; do
   astraImage=$(docker load --input ${astraImageFile} | sed 's~Loaded
image: ~~')
   astraImageShort=`echo $astraImage | sed 's~.*/~~'`
   docker tag ${astraImage} ${REGISTRY}/${astraImageShort}
   docker push ${REGISTRY}/${astraImageShort}
done
sed -i 's~\[YOUR REGISTRY\]~'${REGISTRY}'~' ./manifests/*.yaml
```

Astra Data Store preview operator를 설치한다

1. Astra Data Store 미리 보기 목록을 나열합니다.

```
ls manifests/*yaml
```

응답:

```
manifests/astradscluster.yaml
manifests/astradsoperator.yaml
manifests/astradsversion.yaml
manifests/monitoring_operator.yaml
```

2. kubeck 적용 시 운용자 배치:

```
kubectl apply -f ./manifests/astradsoperator.yaml
```

응답:

```
namespace/astrads-system created
customresourcedefinition.apiextensions.k8s.io/astradsautosupports.astrad
```

```
s.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradscloudsnapshots.astr
ads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsclusters.astrads.ne
tapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsdeployments.astrads
.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsexportpolicies.astr
ads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsfaileddrives.astrad
s.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradslicenses.astrads.ne
tapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsnfsoptions.astrads.
netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsnodeinfoes.astrads.
netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsqospolicies.astrads
.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumefiles.astrads
.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumes.astrads.net
app.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumesnapshots.ast
rads.netapp.io created
role.rbac.authorization.k8s.io/astrads-leader-election-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscloudsnapshot-
editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscloudsnapshot-
viewer-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscluster-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradscluster-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradslicense-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradslicense-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolume-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolume-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-autosupport-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-autosupport-viewer-role
created
```

```
clusterrole.rbac.authorization.k8s.io/astrads-manager-role created
clusterrole.rbac.authorization.k8s.io/astrads-metrics-reader created
clusterrole.rbac.authorization.k8s.io/astrads-netappexportpolicy-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappexportpolicy-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsdeployment-
editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsdeployment-
viewer-role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnfsoption-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnfsoption-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnodeinfo-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnodeinfo-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-proxy-role created
rolebinding.rbac.authorization.k8s.io/astrads-leader-election-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/astrads-manager-rolebinding
created
clusterrolebinding.rbac.authorization.k8s.io/astrads-proxy-rolebinding
created
configmap/astrads-autosupport-cm created
configmap/astrads-firetap-cm created
configmap/astrads-fluent-bit-cm created
configmap/astrads-kevents-asup created
configmap/astrads-metrics-cm created
service/astrads-operator-metrics-service created
deployment.apps/astrads-operator created
```

3. Astra Data Store 운영자 POD가 시작되고 실행 중인지 확인합니다.

```
kubectl get pods -n astrads-system
```

응답:

```
NAME                                READY   STATUS    RESTARTS   AGE
astrads-operator-5ffb94fbf-7ln4h    1/1     Running   0          17m
```

Astra Data Store Preview 버전 YAML을 배포하십시오

1. kubeck을 사용하여 구축 적용:

```
kubectl apply -f ./manifests/astradsversion.yaml
```

2. Pod가 실행 중인지 확인합니다.

```
kubectl get pods -n astrads-system
```

응답:

```
NAME                                          READY   STATUS    RESTARTS
AGE
astrads-cluster-controller-7f6f884645-xxf2n   1/1     Running   0
117s
astrads-ds-nodeinfo-astradsversion-2jqnk      1/1     Running   0
2m7s
astrads-ds-nodeinfo-astradsversion-dbk7v      1/1     Running   0
2m7s
astrads-ds-nodeinfo-astradsversion-rn9tt      1/1     Running   0
2m7s
astrads-ds-nodeinfo-astradsversion-vsmhv      1/1     Running   0
2m7s
astrads-license-controller-fb8fd56bc-bxq7j    1/1     Running   0
2m2s
astrads-operator-5ffb94fbf-7ln4h              1/1     Running   0
2m10s
```

Astra Data Store Preview 라이센스를 적용합니다

1. 미리 보기에 등록할 때 얻은 NetApp 라이센스 파일(NLF)을 적용합니다. 명령을 실행하기 전에 현재 클러스터 이름('<Astra-Data-Store-cluster-name>')을 입력합니다 배포로 이동합니다 또는 이미 배포되어 있고 사용권 파일('<file_path/file.txt>')에 대한 경로가 있습니다.

```
kubectl astrads license add --license-file-path <file_path/file.txt>
--ads-cluster-name <Astra-Data-Store-cluster-name> -n astrads-system
```

2. 라이센스가 추가되었는지 확인합니다.

```
kubectl astrads license list
```

응답:

```
NAME         ADSCLUSTER                  VALID   PRODUCT
EVALUATION   ENDDATE     VALIDATED
p100000006   astrads-example-cluster     true    Astra Data Store Preview
true         2022-01-23  2021-11-04T14:38:54Z
```

Astra Data Store Preview 클러스터를 설치합니다

1. YAML 파일을 엽니다.

```
vim ./manifests/astradscluster.yaml
```

2. YAML 파일에서 다음 값을 편집합니다.

   YAML 파일의 간단한 예는 다음과 같습니다.

   a. (필수) * 메타데이터 *: metadata에서 이름 문자열을 클러스터 이름으로 변경합니다. 이 이름은 사용 시 사용한 클러스터 이름과 같아야 합니다 라이센스를 적용합니다.

   b. (필수) * Spec *: 'sepec'에서 다음 필수 값을 변경합니다.

      - 클러스터의 작업자 노드에서 라우팅할 수 있는 부동 관리 IP의 IP 주소로 mVIP 문자열을 변경합니다.
      - adsDataNetworks에서 NetApp 볼륨을 마운트할 호스트에서 라우팅할 수 있는 쉼표로 구분된 부동 IP 주소 목록("주소")을 추가합니다. 노드당 하나의 부동 IP 주소를 사용합니다. Astra Data Store 미리 보기 노드만큼 데이터 네트워크 IP 주소가 적어도 몇 개 있어야 합니다. Astra Data Store 미리 보기의 경우 4개 이상의 주소가 있어야 하며, 나중에 5개 노드로 클러스터를 확장할 계획이면 5개 이상의 주소가 필요합니다.
      - adsDataNetworks에서 데이터 네트워크에서 사용하는 넷마스크를 지정한다.
      - adsNetworkInterfaces에서 '<mgmt_interface_name>' 및 '<cluster_and_storage_interface_name>' 값을 관리, 클러스터 및 스토리지에 사용할 네트워크 인터페이스 이름으로 바꿉니다. 이름을 지정하지 않으면 노드의 기본 인터페이스가 관리, 클러스터 및 스토리지 네트워킹에 사용됩니다.

        클러스터 및 스토리지 네트워크는 동일한 인터페이스에 있어야 합니다. Astra Data Store 미리 보기 관리 인터페이스는 Kubernetes 노드의 관리 인터페이스와 동일해야 합니다.

   c. (선택 사항) * monitoringConfig *: 를 구성하려는 경우 운전자 모니터링 (모니터링을 위해 Astra Control Center를 사용하지 않는 경우 선택 사항) 섹션에서 메모를 제거하고 에이전트 CR(모니터링 운영자 리소스)이 적용되는 네임스페이스(기본값은 NetApp 모니터링)를 추가한 다음 이전 단계에서 사용한 레지스트리('your_registry_path')의 경로를 추가합니다.

   d. (선택 사항) * autoSupportConfig *: 를 유지합니다 "AutoSupport" 프록시를 구성할 필요가 없는 경우 기본값:

      - proxyURL의 경우 AutoSupport 번들 전송에 사용할 포트를 사용하여 프록시 URL을 설정합니다.

      아래의 YAML 샘플에서 대부분의 의견이 제거되었습니다.

```
```

```
apiVersion: astrads.netapp.io/v1alpha1
kind: AstraDSCluster
metadata:
  name: astrads-cluster-name
  namespace: astrads-system
spec:
  adsNodeConfig:
    cpu: 9
    memory: 34
  adsNodeCount: 4
  mvip: ""
  adsDataNetworks:
    - addresses: ""
      netmask:
  # Specify the network interface names to use for management, cluster
and storage networks.
  # If none are specified, the node's primary interface will be used for
management, cluster and storage networking.
  # To move the cluster and storage networks to a different interface
than management, specify all three interfaces to use here.
  # NOTE: The cluster and storage networks need to be on the same
interface.
  adsNetworkInterfaces:
    managementInterface: "<mgmt_interface_name>"
    clusterInterface: "<cluster_and_storage_interface_name>"
    storageInterface: "<cluster_and_storage_interface_name>"
  # [Optional] Provide a k8s label key that defines which protection
domain a node belongs to.
    # adsProtectionDomainKey: ""
  # [Optional] Provide a monitoring config to be used to setup/configure
a monitoring agent.
 # monitoringConfig:
   # namespace: "netapp-monitoring"
   # repo: "[YOUR REGISTRY]"
  autoSupportConfig:
    autoUpload: true
    enabled: true
    coredumpUpload: false
    historyRetentionCount: 25
    destinationURL: "https://support.netapp.com/put/AsupPut"
    # ProxyURL defines the URL of the proxy with port to be used for
AutoSupport bundle transfer
    # proxyURL:
    periodic:
      - schedule: "0 0 * * *"
        periodicconfig:
```

```
        - component:
            name: storage
            event: dailyMonitoring
          userMessage: Daily Monitoring Storage AutoSupport bundle
          nodes: all
        - component:
            name: controlplane
            event: daily
          userMessage: Daily Control Plane AutoSupport bundle
```

3. "kubbtl apply"를 사용하여 클러스터를 구축합니다.

```
kubectl apply -f ./manifests/astradscluster.yaml
```

4. 클러스터 생성 작업이 완료될 때까지 몇 분 정도 기다린 후 Pod가 실행 중인지 확인합니다.

```
kubectl get pods -n astrads-system
```

샘플 반응:

```
NAME                        READY     STATUS    RESTARTS    AGE
astrads-cluster-controller-7c67cc7f7b-2jww2 1/1 Running 0 7h31m
astrads-deployment-support-788b859c65-2qjkn 3/3 Running 19 12d
astrads-ds-astrads-cluster-1ab0dbc-j9jzc 1/1 Running 0 5d2h
astrads-ds-astrads-cluster-1ab0dbc-k9wp8 1/1 Running 0 5d1h
astrads-ds-astrads-cluster-1ab0dbc-pwk42 1/1 Running 0 5d2h
astrads-ds-astrads-cluster-1ab0dbc-qhvc6 1/1 Running 0 8h
astrads-ds-nodeinfo-astradsversion-gcmj8 1/1 Running 1 12d
astrads-ds-nodeinfo-astradsversion-j826x 1/1 Running 3 12d
astrads-ds-nodeinfo-astradsversion-vdthh 1/1 Running 3 12d
astrads-ds-nodeinfo-astradsversion-xwgsf 1/1 Running 0 12d
astrads-ds-support-828vw 2/2 Running 2 5d2h
astrads-ds-support-cfzts 2/2 Running 0 8h
astrads-ds-support-nzkkr 2/2 Running 15 7h49m
astrads-ds-support-xxbnp 2/2 Running 1 5d2h
astrads-license-controller-86c69f76bb-s6fb7 1/1 Running 0 8h
astrads-operator-79ff8fbb6d-vpz9m 1/1 Running 0 8h
```

5. 클러스터 배포 진행 상태 확인:

```
kubectl get astradscluster -n astrads-system
```

샘플 반응:

```
NAME                          STATUS    VERSION    SERIAL NUMBER    MVIP
AGE

astrads-example-cluster   created   2021.10.0   p100000006
10.x.x.x   10m
```

배포 관련 이벤트를 이해합니다

클러스터 배치 중에는 작동 상태가 공란에서 진행 중 상태로 변경되어야 합니다. 클러스터 구축은 약 8~10분간 지속됩니다. 구축하는 동안 클러스터 이벤트를 모니터링하려면 다음 명령 중 하나를 실행합니다.

```
kubectl get events --field-selector involvedObject.kind=AstraDSCluster -n
astrads-system
```

```
kubectl describe astradscluster <cluster name> -n astrads-system
```

다음은 배포 중에 발생하는 주요 이벤트입니다.

이벤트 메시지입니다	의미
ADS 클러스터를 연결할 4개의 컨트롤 플레인 노드를 성공적으로 선택했습니다	Astra Data Store 미리보기 운영자는 CPU, 메모리, 스토리지 및 네트워킹으로 Astra Data Store 미리보기 클러스터를 생성할 수 있는 충분한 노드를 식별했습니다.
ADS 클러스터 생성 진행 중	Astra Data Store 미리보기 클러스터 컨트롤러가 클러스터 생성 작업을 시작했습니다.
ADS 클러스터가 생성되었습니다	클러스터가 생성되었습니다.

클러스터의 상태가 "In progress(진행 중)"로 변경되지 않는 경우 운영자 로그에서 노드 선택에 대한 자세한 내용을 확인하십시오.

```
kubectl logs -n astrads-system <astrads operator pod name>
```

클러스터의 상태가 "in progress(진행 중)"로 고착된 경우 클러스터 컨트롤러의 로그를 확인하십시오.

```
kubectl logs -n astrads-system <astrads cluster controller pod name>
```

Astra Data Store 미리보기 모니터링을 구성합니다

Astra Control Center 모니터링 또는 다른 원격 측정 서비스의 모니터링을 위해 Astra Data Store 미리보기를 구성할 수 있습니다.

Astra Control Center 미리 보기에 대한 모니터링을 구성합니다

Astra Control Center에서 Astra Data Store 미리 보기를 백엔드로 관리하는 경우에만 다음 단계를 수행하십시오.

1. Astra Control Center에서 모니터링하는 Astra Data Store 미리 보기를 구성합니다.

```
kubectl astrads monitoring -m netapp-monitoring -r [YOUR REGISTRY] setup
```

모니터링 운전자를 설치합니다

(선택 사항) Astra Data Store Preview를 Astra Control Center로 가져오지 않는 경우 모니터링 운용자를 권장합니다. Astra 데이터 저장소 미리 보기 인스턴스가 독립 실행형 배포이거나, Cloud Insights를 사용하여 원격 측정을 모니터링하거나, Elastic과 같은 타사 엔드포인트로 로그를 스트리밍하는 경우 모니터링 연산자를 설치할 수 있습니다.

1. 다음 설치 명령을 실행합니다.

```
kubectl apply -f ./manifests/monitoring_operator.yaml
```

2. 모니터링을 위해 Astra Data Store 미리 보기를 구성합니다.

```
kubectl astrads monitoring -m netapp-monitoring -r [YOUR REGISTRY] setup
```

다음 단계

를 수행하여 배포를 완료합니다 "설정 작업".

Red Hat OpenShift Container Platform에 Astra Data Store 미리보기를 설치합니다

Red Hat OpenShift Container Platform(OCP)에 Astra Data Store 미리보기를 설치하려면 에서 설치 번들을 다운로드하십시오 "NetApp Support 사이트" 및 이 절차에 설명된 설치 단계를 완료합니다.

무엇을 '필요로 할거야

- 설치를 시작하기 전에 "Astra Data Store 구축을 위한 환경을 준비합니다".
- 에 액세스합니다 "NetApp Support 사이트". 전체 액세스 권한이 없는 NetApp Support 사이트 계정이 없는 경우 미리 보기를 위해
- A "NetApp 라이센스 파일" (NLF) - Astra Data Store 미리 보기. 라이센스 다운로드 지침은 가입 후 발송됩니다.
- 활성 컨텍스트에 대한 클러스터 관리자 권한이 있는 활성 kubecononfig.
- 에 대한 이해 "역할 및 권한" Astra Data Store 미리 보기에서 사용합니다.
- 인터넷 연결. Astra Data Store Preview는 대기 환경을 지원하지 않습니다. 직접 또는 프록시를 통해 support.netapp.com 에 접속하려면 인터넷 연결이 필요합니다.

이 작업에 대해

Astra Data Store 미리 보기 설치 프로세스는 다음과 같은 고급 단계를 안내합니다.

- [Download the Astra Data Store preview bundle and extract the images]
- [Copy the binary and push images to your local registry]
- [Create a namespace to deploy Astra Data Store preview]
- [Create a custom SCC]
- [Create the roles and role bindings]
- [Install the Astra Data Store preview operator]
- [Deploy the Astra Data Store preview version YAML]
- [Apply the Astra Data Store preview license]
- [Install the Astra Data Store preview cluster]
- [Understand deployment-related events]
- [Configure Astra Data Store preview monitoring]
- [Install the monitoring operator]

Astra Data Store 미리 보기를 사용하여 비밀이 있는 이미지 등록부와 함께 작업하려면 을 참조하십시오 "이 KB입니다 ".

Astra Data Store 미리보기 번들을 다운로드하고 이미지를 추출합니다

1. 에 로그인합니다 "NetApp Support 사이트" 그리고 Astra Data Store 미리보기 번들('2021.12_astradatastore.tar')을 다운로드하십시오.
2. (선택 사항) 번들의 서명을 확인합니다.

```
openssl dgst -sha256 -verify 2021.12_astradatastore.pub -signature
2021.12_astradatastore.sig 2021.12_astradatastore.tar
```

3. 이미지 추출:

```
tar -xvf 2021.12_astradatastore.tar
```

이진 파일을 복사하고 이미지를 로컬 레지스트리에 푸시합니다

1. k8kubbeck 바이너리가 설치된 표준 경로(예: "/usr/bin/")로 이미지를 추출하기 위해 사용한 디렉토리에서 kubbctl-astrs 바이너리를 복사합니다. kubbctl-astrs는 Astra Data Store Preview 클러스터를 설치 및 관리하는 사용자 지정 kubtl 확장입니다.

```
cp -p ./bin/kubectl-astrads /usr/bin/.
```

2. Astra Data Store 미리보기 이미지 디렉토리의 파일을 로컬 레지스트리에 추가합니다.

   아래 이미지의 자동 로드에 대한 샘플 스크립트를 참조하십시오.

   a. 레지스트리에 로그인합니다.

   ```
   docker login [your_registry_path]
   ```

   b. 환경 변수를 레지스트리 경로로 설정하여 Astra Data Store 미리 보기 이미지를 푸시합니다(예: repo.company.com`).

   ```
   export REGISTRY=repo.company.com/astrads
   ```

   c. 스크립트를 실행하여 이미지를 Docker에 로드하고, 이미지에 태그를 지정하고, 를 눌러 이미지를 로컬 레지스트리에 로드합니다.

   ```
   for astraImageFile in $(ls images/*.tar) ; do
      astraImage=$(docker load --input ${astraImageFile} | sed 's~Loaded
   image: ~~')
      astraImageShort=`echo $astraImage | sed 's~.*/~~'`
      docker tag ${astraImage} ${REGISTRY}/${astraImageShort}
      docker push ${REGISTRY}/${astraImageShort}
   done
   sed -i 's~\[YOUR REGISTRY\]~'${REGISTRY}'~' ./manifests/*.yaml
   ```

Astra Data Store 미리 보기를 배포할 네임스페이스를 생성합니다

모든 Astra Data Store 미리 보기 구성 요소를 설치할 네임스페이스 'astrads-system'을 만듭니다.

1. 네임스페이스 만들기:

   ```
   kubectl create -f ads_namespace.yaml
   ```

   샘플: ads_namespace.YAML

   ```
   apiVersion: v1
   kind: Namespace
   metadata:
     labels:
       control-plane: operator
     name: astrads-system
   ```

사용자 지정 **SCC**를 생성합니다

OpenShift는 POD가 수행할 수 있는 작업을 제어하는 SCC(Security Context Constraints)를 사용합니다. 기본적으로 모든 컨테이너의 실행에는 제한된 SCC와 해당 SCC에 의해 정의된 기능만 부여됩니다.

제한된 SCC는 Astra Data Store 미리보기 클러스터 포드에 필요한 권한을 제공하지 않습니다. 이 절차를 사용하여 Astra Data Store 미리 보기에 필요한 권한(샘플에 나열되어 있음)을 제공합니다.

사용자 지정 SCC를 Astra Data Store 미리 보기 네임스페이스의 기본 서비스 계정에 할당합니다.

단계

1. 사용자 지정 SCC 생성:

```
kubectl create -f ads_privileged_scc.yaml
```

샘플: ads_privileged_csC.yAML

```
allowHostDirVolumePlugin: true
allowHostIPC: true
allowHostNetwork: true
allowHostPID: true
allowHostPorts: true
allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities:
- '*'
allowedUnsafeSysctls:
- '*'
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups: []
kind: SecurityContextConstraints
metadata:
  annotations:
    kubernetes.io/description: 'ADS privileged. Grant with caution.'
    release.openshift.io/create-only: "true"
  name: ads-privileged
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities: null
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
seccompProfiles:
- '*'
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:astrads-system:default
volumes:
- '*'
```

2. OC Get SCC 명령을 사용하여 새로 추가한 SCC를 출력한다.

```
# oc get scc/ads-privileged
NAME              PRIV    CAPS    SELINUX    RUNASUSER   FSGROUP
SUPGROUP   PRIORITY     READONLYROOTFS   VOLUMES
ads-privileged   true    ["*"]   RunAsAny   RunAsAny    RunAsAny
RunAsAny   <no value>   false            ["*"]
#
```

역할 및 역할 바인딩을 만듭니다

Astra Data Store 미리 보기의 기본 서비스 계정에서 사용할 필수 역할 및 역할 바인딩을 만듭니다.

다음 YAML 정의는 Astra Data Store 미리보기 리소스에 필요한 다양한 역할(rolebindings)을 'astrads.netapp.io` API 그룹'에서 할당합니다.

1. 정의된 역할 및 역할 바인딩을 생성합니다.

```
kubectl create -f oc_role_bindings.yaml
```

샘플: OC_ROLE_BINDINGS.YAML

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: privcrole
rules:
- apiGroups:
  - security.openshift.io
  resourceNames:
  - ads-privileged
  resources:
  - securitycontextconstraints
  verbs:
  - use
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: default-scc-rolebinding
  namespace: astrads-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privcrole
subjects:
```

```
- kind: ServiceAccount
  name: default
  namespace: astrads-system
---

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: ownerref
  namespace: astrads-system
rules:
- apiGroups:
  - astrads.netapp.io
  resources:
  - '*/finalizers'
  verbs:
  - update
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: or-rb
  namespace: astrads-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: ownerref
subjects:
- kind: ServiceAccount
  name: default
  namespace: astrads-system
```

작업자 노드를 준비합니다

Astra Data Store 클러스터 미리 보기 구축을 위해 작업자 노드를 준비합니다. Astra Data Store preview 클러스터에 사용되는 모든 작업자 노드에서 이 절차를 수행합니다.

OpenShift는 kubelet 구성 파일('/var/lib/kubelet/config.json')에 json 형식을 사용합니다. Astra Data Store Preview 클러스터는 kubelet config 파일의 YAML 형식을 찾습니다.

단계

1. 클러스터 설치를 시작하기 전에 각 작업자 노드에 '/var/lib/kubelet/config.YAML' 파일을 생성합니다.

```
sudo cp /var/lib/kubelet/config.json /var/lib/kubelet/config.yaml`
```

2. 클러스터 YAML이 적용되기 전에 모든 Kubernetes 노드에서 이 절차를 완료하십시오.

이렇게 하지 않으면 Astra Data Store Preview 클러스터 설치가 실패합니다.

Astra Data Store preview operator를 설치한다

1. Astra Data Store 미리 보기 목록을 나열합니다.

```
ls manifests/*yaml
```

응답:

```
manifests/astradscluster.yaml
manifests/astradsoperator.yaml
manifests/astradsversion.yaml
manifests/monitoring_operator.yaml
```

2. kubbeck apply 명령을 사용하여 운용자를 배치한다.

```
kubectl apply -f ./manifests/astradsoperator.yaml
```

응답:

```
namespace/astrads-system created
customresourcedefinition.apiextensions.k8s.io/astradsautosupports.astrad
s.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradscloudsnapshots.astr
ads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsclusters.astrads.ne
tapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsdeployments.astrads
.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsexportpolicies.astr
ads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsfaileddrives.astrad
s.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradslicenses.astrads.ne
tapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsnfsoptions.astrads.
netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsnodeinfoes.astrads.
netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsqospolicies.astrads
```

```
.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumefiles.astrads
.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumes.astrads.net
app.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumesnapshots.ast
rads.netapp.io created
role.rbac.authorization.k8s.io/astrads-leader-election-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscloudsnapshot-
editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscloudsnapshot-
viewer-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscluster-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradscluster-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradslicense-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradslicense-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolume-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolume-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-autosupport-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-autosupport-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-manager-role created
clusterrole.rbac.authorization.k8s.io/astrads-metrics-reader created
clusterrole.rbac.authorization.k8s.io/astrads-netappexportpolicy-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappexportpolicy-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsdeployment-
editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsdeployment-
viewer-role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnfsoption-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnfsoption-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnodeinfo-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnodeinfo-viewer-
role created
```

```
clusterrole.rbac.authorization.k8s.io/astrads-proxy-role created
rolebinding.rbac.authorization.k8s.io/astrads-leader-election-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/astrads-manager-rolebinding
created
clusterrolebinding.rbac.authorization.k8s.io/astrads-proxy-rolebinding
created
configmap/astrads-autosupport-cm created
configmap/astrads-firetap-cm created
configmap/astrads-fluent-bit-cm created
configmap/astrads-kevents-asup created
configmap/astrads-metrics-cm created
service/astrads-operator-metrics-service created
deployment.apps/astrads-operator created
```

3. Astra Data Store 운영자 POD가 시작되고 실행 중인지 확인합니다.

```
kubectl get pods -n astrads-system
```

응답:

```
NAME                                  READY   STATUS    RESTARTS   AGE
astrads-operator-5ffb94fbf-7ln4h      1/1     Running   0          17m
```

Astra Data Store Preview 버전 YAML을 배포하십시오

1. kubbeck apply 명령을 사용하여 구축:

```
kubectl apply -f ./manifests/astradsversion.yaml
```

2. Pod가 실행 중인지 확인합니다.

```
kubectl get pods -n astrads-system
```

응답:

```
NAME                                          READY   STATUS    RESTARTS
AGE
astrads-cluster-controller-7f6f884645-xxf2n   1/1     Running   0
117s
astrads-ds-nodeinfo-astradsversion-2jqnk      1/1     Running   0
2m7s
astrads-ds-nodeinfo-astradsversion-dbk7v      1/1     Running   0
2m7s
astrads-ds-nodeinfo-astradsversion-rn9tt      1/1     Running   0
2m7s
astrads-ds-nodeinfo-astradsversion-vsmhv      1/1     Running   0
2m7s
astrads-license-controller-fb8fd56bc-bxq7j    1/1     Running   0
2m2s
astrads-operator-5ffb94fbf-7ln4h              1/1     Running   0
2m10s
```

Astra Data Store Preview 라이센스를 적용합니다

1. 미리 보기에 등록할 때 얻은 NetApp 라이센스 파일(NLF)을 적용합니다. 명령을 실행하기 전에 현재 클러스터 이름('<Astra-Data-Store-cluster-name>')을 입력합니다 배포로 이동합니다 또는 이미 배포되어 있고 사용권 파일('<file_path/file.txt>')에 대한 경로가 있습니다.

```
kubectl astrads license add --license-file-path <file_path/file.txt>
--ads-cluster-name <Astra-Data-Store-cluster-name> -n astrads-system
```

2. 라이센스가 추가되었는지 확인합니다.

```
kubectl astrads license list
```

응답:

```
NAME        ADSCLUSTER                  VALID   PRODUCT
EVALUATION  ENDDATE     VALIDATED
p100000006  astrads-example-cluster    true    Astra Data Store Preview
true        2022-01-23  2021-11-04T14:38:54Z
```

Astra Data Store Preview 클러스터를 설치합니다

1. YAML 파일을 엽니다.

```
vim ./manifests/astradscluster.yaml
```

2. YAML 파일에서 다음 값을 편집합니다.

   YAML 파일의 간단한 예는 다음과 같습니다.

   a. (필수) * 메타데이터 *: metadata에서 이름 문자열을 클러스터 이름으로 변경합니다. 이 이름은 사용 시 사용한 클러스터 이름과 같아야 합니다 라이센스를 적용합니다.

   b. (필수) * Spec *: 'sepec'에서 다음 필수 값을 변경합니다.

      - 클러스터의 작업자 노드에서 라우팅할 수 있는 부동 관리 IP의 IP 주소로 mVIP 문자열을 변경합니다.
      - adsDataNetworks에서 NetApp 볼륨을 마운트할 호스트에서 라우팅할 수 있는 쉼표로 구분된 부동 IP 주소 목록("주소")을 추가합니다. 노드당 하나의 부동 IP 주소를 사용합니다. Astra Data Store 미리 보기 노드만큼 데이터 네트워크 IP 주소가 적어도 몇 개 있어야 합니다. Astra Data Store 미리 보기의 경우 4개 이상의 주소가 있어야 하며, 나중에 5개 노드로 클러스터를 확장할 계획이면 5개 이상의 주소가 필요합니다.
      - adsDataNetworks에서 데이터 네트워크에서 사용하는 넷마스크를 지정한다.
      - adsNetworkInterfaces에서 '<mgmt_interface_name>' 및 '<cluster_and_storage_interface_name>' 값을 관리, 클러스터 및 스토리지에 사용할 네트워크 인터페이스 이름으로 바꿉니다. 이름을 지정하지 않으면 노드의 기본 인터페이스가 관리, 클러스터 및 스토리지 네트워킹에 사용됩니다.

        클러스터 및 스토리지 네트워크는 동일한 인터페이스에 있어야 합니다. Astra Data Store 미리 보기 관리 인터페이스는 Kubernetes 노드의 관리 인터페이스와 동일해야 합니다.

   c. (선택 사항) * monitoringConfig *: 를 구성하려는 경우 운전자 모니터링 (모니터링을 위해 Astra Control Center를 사용하지 않는 경우 선택 사항) 섹션에서 메모를 제거하고 에이전트 CR(모니터링 운영자 리소스)이 적용되는 네임스페이스(기본값은 NetApp 모니터링)를 추가한 다음 이전 단계에서 사용한 레지스트리('your_registry_path')의 경로를 추가합니다.

   d. (선택 사항) * autoSupportConfig *: 를 유지합니다 "AutoSupport" 프록시를 구성할 필요가 없는 경우 기본값:

      - proxyURL의 경우 AutoSupport 번들 전송에 사용할 포트를 사용하여 프록시 URL을 설정합니다.

      아래의 YAML 샘플에서 대부분의 의견이 제거되었습니다.

```
apiVersion: astrads.netapp.io/v1alpha1
kind: AstraDSCluster
metadata:
  name: astrads-cluster-name
  namespace: astrads-system
spec:
  adsNodeConfig:
    cpu: 9
    memory: 34
  adsNodeCount: 4
  mvip: ""
  adsDataNetworks:
```

```
    - addresses: ""
      netmask:
  # Specify the network interface names to use for management, cluster
and storage networks.
  # If none are specified, the node's primary interface will be used for
management, cluster and storage networking.
  # To move the cluster and storage networks to a different interface
than management, specify all three interfaces to use here.
  # NOTE: The cluster and storage networks need to be on the same
interface.
  adsNetworkInterfaces:
    managementInterface: "<mgmt_interface_name>"
    clusterInterface: "<cluster_and_storage_interface_name>"
    storageInterface: "<cluster_and_storage_interface_name>"
  # [Optional] Provide a k8s label key that defines which protection
domain a node belongs to.
    # adsProtectionDomainKey: ""
  # [Optional] Provide a monitoring config to be used to setup/configure
a monitoring agent.
 # monitoringConfig:
   # namespace: "netapp-monitoring"
   # repo: "[YOUR REGISTRY]"
  autoSupportConfig:
    autoUpload: true
    enabled: true
    coredumpUpload: false
    historyRetentionCount: 25
    destinationURL: "https://support.netapp.com/put/AsupPut"
    # ProxyURL defines the URL of the proxy with port to be used for
AutoSupport bundle transfer
    # proxyURL:
    periodic:
      - schedule: "0 0 * * *"
        periodicconfig:
        - component:
            name: storage
            event: dailyMonitoring
          userMessage: Daily Monitoring Storage AutoSupport bundle
          nodes: all
        - component:
            name: controlplane
            event: daily
          userMessage: Daily Control Plane AutoSupport bundle
```

3. "kubbtl apply"를 사용하여 클러스터를 구축합니다.

```
kubectl apply -f ./manifests/astradscluster.yaml
```

4. SELinux가 설정된 경우 Astra Data Store preview 클러스터의 노드에 있는 다음 디렉토리에 대한 'strinux' 컨텍스트의 레이블을 다시 지정합니다.

```
sudo chcon -R -t container_file_t
/var/opt/netapp/firetap/rootfs/var/asup/notification/firetap/
```

```
sudo chcon -R -t container_file_t /var/netapp/firetap/firegen/persist/
```

> 이 단계는 셀린스가 이 디렉토리를 쓸 수 없도록 하여 지원 포드가 CrashLoopoff 상태로 들어가는 원인이 되기 때문에 필요합니다. 이 단계는 Astra Data Store 미리 보기 클러스터의 모든 노드에서 수행해야 합니다.

5. 클러스터 생성 작업이 완료될 때까지 몇 분 정도 기다린 후 Pod가 실행 중인지 확인합니다.

```
kubectl get pods -n astrads-system
```

샘플 반응:

```
NAME READY STATUS RESTARTS AGE
astrads-cluster-controller-7c67cc7f7b-2jww2 1/1 Running 0 7h31m
astrads-deployment-support-788b859c65-2qjkn 3/3 Running 19 12d
astrads-ds-astrads-cluster-1ab0dbc-j9jzc 1/1 Running 0 5d2h
astrads-ds-astrads-cluster-1ab0dbc-k9wp8 1/1 Running 0 5d1h
astrads-ds-astrads-cluster-1ab0dbc-pwk42 1/1 Running 0 5d2h
astrads-ds-astrads-cluster-1ab0dbc-qhvc6 1/1 Running 0 8h
astrads-ds-nodeinfo-astradsversion-gcmj8 1/1 Running 1 12d
astrads-ds-nodeinfo-astradsversion-j826x 1/1 Running 3 12d
astrads-ds-nodeinfo-astradsversion-vdthh 1/1 Running 3 12d
astrads-ds-nodeinfo-astradsversion-xwgsf 1/1 Running 0 12d
astrads-ds-support-828vw 2/2 Running 2 5d2h
astrads-ds-support-cfzts 2/2 Running 0 8h
astrads-ds-support-nzkkr 2/2 Running 15 7h49m
astrads-ds-support-xxbnp 2/2 Running 1 5d2h
astrads-license-controller-86c69f76bb-s6fb7 1/1 Running 0 8h
astrads-operator-79ff8fbb6d-vpz9m 1/1 Running 0 8h
```

6. 클러스터 배포 진행 상태 확인:

```
kubectl get astradscluster -n astrads-system
```

샘플 반응:

```
NAME                          STATUS    VERSION    SERIAL NUMBER    MVIP
AGE

astrads-example-cluster   created   2021.10.0   p100000006
10.x.x.x   10m
```

배포 관련 이벤트를 이해합니다

클러스터 배치 중에는 작동 상태가 공란에서 진행 중 상태로 변경되어야 합니다. 클러스터 구축은 약 8~10분간 지속됩니다. 구축하는 동안 클러스터 이벤트를 모니터링하려면 다음 명령 중 하나를 실행합니다.

```
kubectl get events --field-selector involvedObject.kind=AstraDSCluster -n
astrads-system
```

```
kubectl describe astradscluster <cluster name> -n astrads-system
```

다음은 배포 중에 발생하는 주요 이벤트입니다.

이벤트 메시지입니다	의미
ADS 클러스터를 연결할 4개의 컨트롤 플레인 노드를 성공적으로 선택했습니다	Astra Data Store 미리보기 운영자는 CPU, 메모리, 스토리지 및 네트워킹으로 Astra Data Store 미리보기 클러스터를 생성할 수 있는 충분한 노드를 식별했습니다.
ADS 클러스터 생성 진행 중	Astra Data Store 미리보기 클러스터 컨트롤러가 클러스터 생성 작업을 시작했습니다.
ADS 클러스터가 생성되었습니다	클러스터가 생성되었습니다.

클러스터의 상태가 "In progress(진행 중)"로 변경되지 않는 경우 운영자 로그에서 노드 선택에 대한 자세한 내용을 확인하십시오.

```
kubectl logs -n astrads-system <astrads operator pod name>
```

클러스터의 상태가 "in progress(진행 중)"로 고착된 경우 클러스터 컨트롤러의 로그를 확인하십시오.

```
kubectl logs -n astrads-system <astrads cluster controller pod name>
```

Astra Data Store 미리보기 모니터링을 구성합니다

Astra Control Center 모니터링 또는 다른 원격 측정 서비스의 모니터링을 위해 Astra Data Store 미리보기를 구성할 수 있습니다.

Astra Control Center 미리 보기에 대한 모니터링을 구성합니다

Astra Control Center에서 Astra Data Store 미리 보기를 백엔드로 관리하는 경우에만 다음 단계를 수행하십시오.

1. Astra Control Center에서 모니터링하는 Astra Data Store 미리 보기를 구성합니다.

```
kubectl astrads monitoring -m netapp-monitoring -r [YOUR REGISTRY] setup
```

모니터링 운전자를 설치합니다

(선택 사항) Astra Data Store Preview를 Astra Control Center로 가져오지 않는 경우 모니터링 운용자를 권장합니다. Astra 데이터 저장소 미리 보기 인스턴스가 독립 실행형 배포이거나, Cloud Insights를 사용하여 원격 측정을 모니터링하거나, Elastic과 같은 타사 엔드포인트로 로그를 스트리밍하는 경우 모니터링 연산자를 설치할 수 있습니다.

1. 다음 설치 명령을 실행합니다.

```
kubectl apply -f ./manifests/monitoring_operator.yaml
```

2. 모니터링을 위해 Astra Data Store 미리 보기를 구성합니다.

```
kubectl astrads monitoring -m netapp-monitoring -r [YOUR REGISTRY] setup
```

다음 단계

를 수행하여 배포를 완료합니다 "설정 작업".

Copyright Information

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.