Astra Connector를 설치하여 클러스터를 관리한다

Astra Control Service

NetApp
September 23, 2024

This PDF was generated from https://docs.netapp.com/ko-kr/astra-control-service/get-started/install-astra-connector-previous.html on September 23, 2024. Always check docs.netapp.com for the latest.

목차

Astra Connector를 설치하여 클러스터를 관리한다 1
이전 버전의 Astra Connector를 설치합니다 1
(기술 미리 보기) 선언적 Kubernetes Astra Connector를 설치합니다 4

Astra Connector를 설치하여 클러스터를 관리한다

Astra Connector는 관리형 클러스터에 상주하는 소프트웨어로, 관리형 클러스터와 Astra Control 간의 통신을 지원합니다. Astra Control Service를 사용하여 관리되는 클러스터의 경우 Astra Connector에 사용할 수 있는 2가지 버전이 있습니다.

- * 이전 버전의 Astra Connector *: "이전 버전의 Astra Connector를 설치합니다" Kubernetes 네이티브가 아닌 워크플로우를 통해 클러스터를 관리하려는 경우, 클러스터에 위치한다.
- [기술 미리보기] * 선언 Kubernetes Astra Connector *: "선언적 Kubernetes 워크플로로 관리되는 클러스터용 Astra Connector를 설치합니다" 선언적 Kubernetes 워크플로우를 사용하여 클러스터를 관리하려는 경우 클러스터에 두어야 합니다. 클러스터에 Astra Connector를 설치하면 클러스터가 Astra Control에 자동으로 추가됩니다.

> 선언적 Kubernetes Astra Connector는 Astra Control EAP(Early Adopter Program)의 일부로만 제공됩니다. 에 가입하는 방법에 대한 자세한 내용은 NetApp 영업 담당자에게 문의하십시오.

이전 버전의 Astra Connector를 설치합니다

Astra Control Service는 이전 버전의 Astra Connector를 사용하여 Kubernetes가 아닌 네이티브 워크플로우로 관리되는 프라이빗 클러스터 와 Astra Control Service 간의 통신을 지원합니다. Kubernetes가 아닌 네이티브 워크플로우로 관리하려는 프라이빗 클러스터에 Astra Connector를 설치해야 합니다.

이전 버전의 Astra Connector는 Kubernetes가 아닌 네이티브 워크플로로 관리되는 다음 유형의 프라이빗 클러스터를 지원합니다.

- Amazon Elastic Kubernetes Service(EKS)
- Azure Kubernetes 서비스(AKS)
- Google Kubernetes Engine(GKE)
- AWS 기반 Red Hat OpenShift Service(ROSA)
- AWS PrivateLink와 함께 Rosa
- Red Hat OpenShift Container Platform 온-프레미스

이 작업에 대해

- 이러한 단계를 수행할 때 Astra Control Service로 관리할 프라이빗 클러스터에 대해 다음 명령을 실행합니다.
- 방호 호스트를 사용하는 경우 방호 호스트의 명령줄에서 이러한 명령을 실행하십시오.

시작하기 전에

- Astra Control Service를 사용하여 관리할 프라이빗 클러스터에 액세스해야 합니다.
- 클러스터에 Astra Connector 연산자를 설치하려면 Kubernetes 관리자 권한이 필요합니다.

단계

1. Kubernetes가 아닌 네이티브 워크플로로 관리할 프라이빗 클러스터에 이전 Astra Connector 연산자를 설치합니다. 이 명령을 실행하면 네임스페이스가 생성됩니다 `astra-connector-operator` 이 만들어지고

구성이 네임스페이스에 적용됩니다.

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/23.07.0-
202310251519/astraconnector_operator.yaml
```

2. 작업자가 설치되어 있고 준비가 되었는지 확인합니다.

```
kubectl get all -n astra-connector-operator
```

3. Astra Control에서 API 토큰을 받습니다. 을 참조하십시오 "Astra 자동화 문서" 를 참조하십시오.
4. Astra-Connector 네임스페이스를 생성합니다.

```
kubectl create ns astra-connector
```

5. Astra Connector CR 파일을 생성하고 이름을 지정합니다 `astra-connector-cr.yaml`. 괄호 <> 의 값을 Astra Control 환경 및 클러스터 구성과 일치하도록 업데이트합니다.
   - * <ASTRA_CONTROL_SERVICE_URL> * : Astra Control Service의 웹 UI URL. 예를 들면 다음과 같습니다.

     ```
     https://astra.netapp.io
     ```

   - * <ASTRA_CONTROL_SERVICE_API_TOKEN> *: 이전 단계에서 얻은 Astra Control API 토큰.
   - * <PRIVATE_AKS_CLUSTER_NAME> *: (AKS 클러스터만 해당) - 전용 Azure Kubernetes Service 클러스터의 클러스터 이름입니다. 전용 AKS 클러스터를 추가하는 경우에만 주석을 해제하고 이 줄을 채웁니다.
   - * <ASTRA_CONTROL_ACCOUNT_ID> *: Astra Control 웹 UI에서 가져옵니다. 페이지 오른쪽 상단의 그림 아이콘을 선택하고 * API 액세스 * 를 선택합니다.

```yaml
apiVersion: netapp.astraconnector.com/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  natssync-client:
    cloud-bridge-url: <ASTRA_CONTROL_SERVICE_URL>
  imageRegistry:
    name: theotw
    secret: ""
  astra:
    token: <ASTRA_CONTROL_SERVICE_API_TOKEN>
    #clusterName: <PRIVATE_AKS_CLUSTER_NAME>
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    acceptEULA: yes
```

6. 를 채운 후 `astra-connector-cr.yaml` 올바른 값이 있는 파일에 CR을 적용합니다.

```
kubectl apply -f astra-connector-cr.yaml
```

7. Astra Connector가 완전히 구축되었는지 확인:

```
kubectl get all -n astra-connector
```

8. 클러스터가 Astra Control에 등록되었는지 확인:

```
kubectl get astraconnector -n astra-connector
```

다음과 유사한 출력이 표시됩니다.

```
NAME                REGISTERED   ASTRACONNECTORID
STATUS
astra-connector     true         be475ae5-1511-4eaa-9b9e-712f09b0d065
Registered with Astra
```

ASTRACONNECTORID를 기록해 둡니다. Astra Control에 클러스터를 추가할 때 필요합니다.

다음 단계

Astra Connector를 설치했으므로 이제 프라이빗 클러스터를 Astra Control Service에 추가할 준비가 되었습니다.

- "Astra Control Service에 프라이빗 공급자 관리 클러스터를 추가합니다": 다음 단계를 수행하여 프라이빗 IP 주소가 있고 클라우드 공급자가 관리하는 클러스터를 추가할 수 있습니다. 클라우드 공급자의 서비스 주 계정, 서비스 계정 또는 사용자 계정이 필요합니다.
- "Astra Control Service에 프라이빗 자체 관리 클러스터를 추가합니다": 다음 단계를 사용하여 개인 IP 주소가 있고 조직에서 관리하는 클러스터를 추가할 수 있습니다. 추가하려는 클러스터에 대해 kubecononfig 파일을 생성해야 합니다.

를 참조하십시오

- "클러스터를 추가합니다"

(기술 미리 보기) 선언적 **Kubernetes Astra Connector**를 설치합니다

선언적 Kubernetes 워크플로우를 사용하여 관리되는 클러스터는 Astra Connector를 사용하여 관리형 클러스터와 Astra Control 간의 통신을 지원합니다. 선언적 Kubernetes 워크플로로 관리할 모든 클러스터에 Astra Connector를 설치해야 합니다.

Kubernetes 명령 및 CR(Custom Resource) 파일을 사용하여 선언적 Kubernetes Astra Connector를 설치합니다.

이 작업에 대해

- 이러한 단계를 수행할 때 Astra Control으로 관리하려는 클러스터에서 다음 명령을 실행합니다.
- 방호 호스트를 사용하는 경우 방호 호스트의 명령줄에서 이러한 명령을 실행하십시오.

시작하기 전에

- Astra Control을 사용하여 관리할 클러스터에 액세스해야 합니다.
- 클러스터에 Astra Connector 연산자를 설치하려면 Kubernetes 관리자 권한이 필요합니다.

> Kubernetes 1.25 이상 클러스터의 기본값인 Pod 보안 승인 적용을 사용하여 클러스터를 구성한 경우 해당 네임스페이스에 PSA 제한을 활성화해야 합니다. 을 참조하십시오 "Astra Control을 사용하여 클러스터 관리를 위한 환경을 준비합니다" 를 참조하십시오.

단계

1. 선언적 Kubernetes 워크플로로 관리할 클러스터에 Astra Connector 연산자를 설치합니다. 이 명령을 실행하면 네임스페이스가 생성됩니다 `astra-connector-operator` 이 만들어지고 구성이 네임스페이스에 적용됩니다.

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/24.02.0-
202403151353/astraconnector_operator.yaml
```

2. 작업자가 설치되어 있고 준비가 되었는지 확인합니다.

```
kubectl get all -n astra-connector-operator
```

3. Astra Control에서 API 토큰을 받습니다. 을 참조하십시오 "Astra 자동화 문서" 를 참조하십시오.
4. 토큰을 사용하여 암호를 생성합니다. <API_TOKEN>를 Astra Control에서 받은 토큰으로 대체:

```
kubectl create secret generic astra-token \
--from-literal=apiToken=<API_TOKEN> \
-n astra-connector
```

5. Astra Connector 이미지를 가져오는 데 사용할 Docker 암호를 생성합니다. 괄호 안의 값을 사용자 환경의 정보로 대체:

> <ASTRA_CONTROL_ACCOUNT_ID>는 Astra Control 웹 UI에서 찾을 수 있습니다. 웹 UI에서 페이지 오른쪽 상단의 그림 아이콘을 선택하고 * API access * 를 선택합니다.

```
kubectl create secret docker-registry regcred \
--docker-username=<ASTRA_CONTROL_ACCOUNT_ID> \
--docker-password=<API_TOKEN> \
-n astra-connector \
--docker-server=cr.astra.netapp.io
```

6. Astra Connector CR 파일을 생성하고 이름을 지정합니다 `astra-connector-cr.yaml`. 괄호 <> 의 값을 Astra Control 환경 및 클러스터 구성과 일치하도록 업데이트합니다.
   - <ASTRA_CONTROL_ACCOUNT_ID>: 이전 단계에서 Astra Control 웹 UI에서 구함.
   - <CLUSTER_NAME>: Astra Control에서 이 클러스터를 할당해야 하는 이름입니다.
   - <ASTRA_CONTROL_URL>: Astra Control의 웹 UI URL입니다. 예를 들면 다음과 같습니다.

```
https://astra.control.url
```

```yaml
apiVersion: astra.netapp.io/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  astra:
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    clusterName: <CLUSTER_NAME>
    #Only set `skipTLSValidation` to `true` when using the default
self-signed
    #certificate in a proof-of-concept environment.
    skipTLSValidation: false #Should be set to false in production
environments
    tokenRef: astra-token
  natsSyncClient:
    cloudBridgeURL: <ASTRA_CONTROL_HOST_URL>
  imageRegistry:
    name: cr.astra.netapp.io
    secret: regcred
```

7. 를 채운 후 `astra-connector-cr.yaml` 올바른 값이 있는 파일에 CR을 적용합니다.

```
kubectl apply -n astra-connector -f astra-connector-cr.yaml
```

8. Astra Connector가 완전히 구축되었는지 확인:

```
kubectl get all -n astra-connector
```

9. 클러스터가 Astra Control에 등록되었는지 확인:

```
kubectl get astraconnectors.astra.netapp.io -A
```

다음과 유사한 출력이 표시됩니다.

```
NAMESPACE          NAME               REGISTERED   ASTRACONNECTORID
STATUS
astra-connector    astra-connector    true         00ac8-2cef-41ac-8777-
ed0583e   Registered with Astra
```

10. Astra Control 웹 UI의 * 클러스터 * 페이지에서 관리되는 클러스터 목록에 클러스터가 나타나는지 확인합니다.

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 라이센스와 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 라이센스도 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 라이센스를 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 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

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