



요구 사항

Kubernetes clusters

NetApp
June 02, 2022

목차

요구 사항	1
AWS의 Kubernetes 클러스터 요구사항	1

요구 사항

AWS의 Kubernetes 클러스터 요구사항

AWS에서 관리되는 Amazon EKS(Elastic Kubernetes Service) 클러스터 또는 자체 관리되는 Kubernetes 클러스터를 Cloud Manager에 추가할 수 있습니다. 클러스터를 Cloud Manager에 추가하려면 먼저 다음 요구사항을 충족해야 합니다.



이 항목에서는 EKS 및 자체 관리 Kubernetes 클러스터에 대한 구성이 동일한 _Kubernetes 클러스터를 사용합니다. 클러스터 유형은 구성이 다른 곳에서 지정됩니다.

요구 사항

아스트라 트리덴트

Astra Trident의 최신 버전 4개 중 하나가 필요합니다. Cloud Manager에서 Astra Trident를 직접 설치할 수 있습니다. 당신은 해야 한다 ["사전 요구 사항을 검토합니다"](#) Astra Trident를 설치하기 전

Astra Trident를 업그레이드하려면 ["운영자와 함께 업그레이드하십시오"](#).

Cloud Volumes ONTAP

Cloud Volumes ONTAP for AWS는 클러스터를 위한 백엔드 스토리지로 설정해야 합니다. ["구성 단계를 보려면 Astra Trident 문서로 이동합니다"](#).

Cloud Manager 커넥터

필요한 권한이 있는 Connector가 AWS에서 실행되고 있어야 합니다. [아래에서 자세히 알아보십시오.](#)

네트워크 연결

Kubernetes 클러스터와 Connector 간, Kubernetes 클러스터와 Cloud Volumes ONTAP 사이에 네트워크 연결이 필요합니다. [아래에서 자세히 알아보십시오.](#)

RBAC 인증

Cloud Manager Connector 역할은 각 Kubernetes 클러스터에서 승인되어야 합니다. [아래에서 자세히 알아보십시오.](#)

커넥터를 준비합니다

Kubernetes 클러스터를 검색하고 관리하려면 AWS에 Cloud Manager Connector가 필요합니다. 새 Connector를 만들거나 필요한 권한이 있는 기존 Connector를 사용해야 합니다.

새 커넥터를 작성합니다

아래 링크 중 하나에 있는 단계를 따르십시오.

- ["Cloud Manager에서 커넥터를 생성합니다"](#) (권장)
- ["AWS Marketplace에서 Connector를 생성합니다"](#)
- ["AWS의 기존 Linux 호스트에 커넥터를 설치합니다"](#)

기존 **Connector**에 필요한 권한을 추가합니다

3.9.13 릴리스부터 `_NEWED_DEPLOY` 커넥터에는 Kubernetes 클러스터의 검색 및 관리를 지원하는 세 가지 새로운 AWS 권한이 포함되어 있습니다. 이 릴리스 전에 Connector를 생성한 경우, Connector의 IAM 역할에 대한 기존 정책을 수정하여 권한을 제공해야 합니다.

단계

1. AWS 콘솔로 이동하여 EC2 서비스를 엽니다.
2. Connector 인스턴스를 선택하고 * Security * 를 클릭한 다음 IAM 역할의 이름을 클릭하여 IAM 서비스의 역할을 확인합니다.



3. 사용 권한 * 탭에서 정책을 확장하고 * 정책 편집 * 을 클릭합니다.



4. JSON * 을 클릭하고 첫 번째 작업 세트에서 다음 권한을 추가합니다.

```
"eks:ListClusters",
"eks:DescribeCluster",
"iam:GetInstanceProfile"
```

"정책의 전체 JSON 형식을 봅니다".

5. 정책 검토 * 를 클릭한 다음 * 변경 사항 저장 * 을 클릭합니다.

네트워킹 요구 사항을 검토합니다

Kubernetes 클러스터와 Connector 간, Kubernetes 클러스터와 클러스터에 백엔드 스토리지를 제공하는 Cloud Volumes ONTAP 시스템 간에 네트워크 연결을 제공해야 합니다.

- 각 Kubernetes 클러스터에는 Connector로부터 인바운드 연결이 있어야 합니다
- Connector는 포트 443을 통해 각 Kubernetes 클러스터에 대한 아웃바운드 연결을 가지고 있어야 합니다

이 연결을 제공하는 가장 간단한 방법은 Kubernetes 클러스터와 같은 VPC에 Connector와 Cloud Volumes ONTAP를 구축하는 것입니다. 그렇지 않으면 다른 VPC 간에 VPC 피어링 연결을 설정해야 합니다.

다음은 동일한 VPC의 각 구성 요소를 보여 주는 예입니다.



이 또 다른 예는 다른 VPC에서 실행되는 EKS 클러스터를 보여 줍니다. 이 예에서 VPC 피어링은 EKS 클러스터용 VPC와 커넥터 및 Cloud Volumes ONTAP용 VPC 간에 연결을 제공합니다.



RBAC 승인을 설정합니다

Connector가 클러스터를 검색 및 관리할 수 있도록 각 Kubernetes 클러스터에서 Connector 역할을 승인해야 합니다.

다른 기능을 사용하려면 다른 권한이 필요합니다.

백업 및 복원

백업 및 복원에는 기본 인증만 필요합니다.

스토리지 클래스를 추가합니다

Cloud Manager를 사용하여 스토리지 클래스를 추가하려면 확장된 인증이 필요합니다.

Astra 트리덴트 설치

Astra Trident를 설치하려면 Cloud Manager에 대한 전체 인증을 제공해야 합니다.



Astra Trident를 설치할 때 Cloud Manager는 Astra Trident와 스토리지 클러스터와 통신하는 데 필요한 자격 증명이 포함된 Astra Trident 백엔드 및 Kubernetes 암호를 설치합니다.

단계

1. 클러스터 역할 및 역할 바인딩을 생성합니다.
 - a. 귀하의 승인 요구 사항에 따라 다음 텍스트가 포함된 YAML 파일을 생성합니다.

백업/복원

Kubernetes 클러스터의 백업 및 복원을 위한 기본 인증을 추가하십시오.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: cloudmanager-access-clusterrole
rules:
  - apiGroups:
      - ''
    resources:
      - namespaces
    verbs:
      - list
  - apiGroups:
      - ''
    resources:
      - persistentvolumes
    verbs:
      - list
  - apiGroups:
      - ''
    resources:
      - pods
      - pods/exec
    verbs:
      - get
      - list
  - apiGroups:
      - ''
    resources:
      - persistentvolumeclaims
    verbs:
      - list
      - create
  - apiGroups:
      - storage.k8s.io
    resources:
      - storageclasses
    verbs:
      - list
  - apiGroups:
      - trident.netapp.io
    resources:
      - tridentbackends
```



```

    verbs:
      - list
  - apiGroups:
      - trident.netapp.io
    resources:
      - tridentorchestrators
    verbs:
      - get
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: k8s-access-binding
subjects:
  - kind: Group
    name: cloudmanager-access-group
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: cloudmanager-access-clusterrole
  apiGroup: rbac.authorization.k8s.io

```

스토리지 클래스

Cloud Manager를 사용하여 스토리지 클래스를 추가하려면 확장 인증을 추가합니다.

```

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: cloudmanager-access-clusterrole
rules:
  - apiGroups:
      - ''
    resources:
      - secrets
      - namespaces
      - persistentvolumeclaims
      - persistentvolumes
      - pods
      - pods/exec
    verbs:
      - get
      - list
      - create
      - delete
  - apiGroups:

```

```

      - storage.k8s.io
resources:
  - storageclasses
verbs:
  - get
  - create
  - list
  - delete
  - patch
- apiGroups:
  - trident.netapp.io
resources:
  - tridentbackends
  - tridentorchestrators
  - tridentbackendconfigs
verbs:
  - get
  - list
  - create
  - delete

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: k8s-access-binding
subjects:
  - kind: Group
    name: cloudmanager-access-group
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: cloudmanager-access-clusterrole
  apiGroup: rbac.authorization.k8s.io

```

Trident를 설치합니다

명령줄을 사용하여 전체 인증을 제공하고 Cloud Manager에서 Astra Trident를 설치할 수 있도록 합니다.

```

eksctl create iamidentitymapping --cluster < > --region < > --arn
< > --group "system:masters" --username
system:node:{{EC2PrivateDNSName}}

```

b. 클러스터에 구성을 적용합니다.

```
kubectl apply -f <file-name>
```

2. 권한 그룹에 대한 ID 매핑을 만듭니다.

eksctl을 사용합니다

eksctl을 사용하여 클러스터와 Cloud Manager Connector의 IAM 역할 사이에 IAM ID 매핑을 생성합니다.

"전체 지침은 [eksctl 설명서를 참조하십시오](#)".

아래에 예가 나와 있습니다.

```
eksctl create iamidentitymapping --cluster <eksCluster> --region  
<us-east-2> --arn <ARN of the Connector IAM role> --group  
cloudmanager-access-group --username  
system:node:{{EC2PrivateDNSName}}
```

AWS-auth를 편집합니다

AWS-auth ConfigMap을 직접 편집하여 RBAC 액세스를 Cloud Manager Connector의 IAM 역할에 추가합니다.

"전체 지침은 [AWS EKS 설명서를 참조하십시오](#)".

아래에 예가 나와 있습니다.

```
apiVersion: v1  
data:  
  mapRoles: |  
    - groups:  
      - cloudmanager-access-group  
      rolearn: <ARN of the Connector IAM role>  
      username: system:node:{{EC2PrivateDNSName}}  
kind: ConfigMap  
metadata:  
  creationTimestamp: "2021-09-30T21:09:18Z"  
  name: aws-auth  
  namespace: kube-system  
  resourceVersion: "1021"  
  selfLink: /api/v1/namespaces/kube-system/configmaps/aws-auth  
  uid: dcc31de5-3838-11e8-af26-02e00430057c
```

저작권 정보

Copyright © 2022 NetApp, Inc. All rights reserved. 미국에서 인쇄된 본 문서의 어떤 부분도 저작권 소유자의 사전 서면 승인 없이 어떠한 형식이나 그래픽, 전자적 또는 기계적 수단(사진 복사, 레코딩 등)으로도 저작권 소유자의 사전 서면 승인 없이 전자 검색 시스템에 저장 또는 저장.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지 사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 "있는 그대로" 제공되며 상품성 및 특정 목적에 대한 적합성에 대한 명시적 또는 묵시적 보증을 포함하여 이에 제한되지 않고, 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 또는 파생적 손해(소계 물품 또는 서비스의 조달, 사용 손실, 데이터 또는 수익 손실, 계약, 엄격한 책임 또는 불법 행위(과실 또는 그렇지 않은 경우)에 관계없이 어떠한 책임도 지지 않으며, 이는 이러한 손해의 가능성을 사전에 알고 있던 경우에도 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 구입의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

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

권리 제한 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.277-7103(1988년 10월) 및 FAR 52-227-19(1987년 6월)의 기술 데이터 및 컴퓨터 소프트웨어의 권리(Rights in Technical Data and Computer Software) 조항의 하위 조항 (c)(1)(ii)에 설명된 제한사항이 적용됩니다.

상표 정보

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