



Astra를 사용하십시오

Astra Control Center

NetApp
November 21, 2023

목차

Astra를 사용하십시오.....	1
앱 관리	1
앱 보호	8
앱 및 클러스터 상태 보기	30
계정을 관리합니다	33
버킷을 관리합니다	38
스토리지 백엔드를 관리합니다	40
인프라 모니터링 및 보호	41
기존 라이센스를 업데이트합니다	48
앱 및 클러스터 관리를 취소합니다	49
Astra Control Center를 업그레이드합니다	50
Astra Control Center를 제거합니다	63

Astra를 사용하십시오

앱 관리

앱 관리를 시작합니다

먼저 해 ["Astra Control 관리에 클러스터를 추가합니다"](#), 클러스터(Astra Control 외부)에 앱을 설치한 다음, Astra Control의 앱 페이지로 이동하여 앱과 리소스 관리를 시작할 수 있습니다.

설명합니다

Astra Control에는 다음과 같은 앱 관리 요구 사항이 있습니다.

- * 라이선스 *: Astra Control Center를 사용하여 앱을 관리하려면 Astra Control Center 라이센스가 필요합니다.
- * Namespaces *: Astra Control은 앱이 단일 네임스페이스 이상의 범위를 포함하지 않도록 하지만 네임스페이스에는 여러 개의 앱이 포함될 수 있습니다.
- * StorageClass *: StorageClass가 명시적으로 설정된 앱을 설치하고 앱을 복제해야 하는 경우 클론 작업의 타겟 클러스터에 원래 지정된 StorageClass가 있어야 합니다. 명시적으로 StorageClass를 동일한 StorageClass가 없는 클러스터로 설정한 애플리케이션을 클론 복제하면 실패합니다.
- * Kubernetes 리소스 *: Astra Control에서 수집하지 않은 Kubernetes 리소스를 사용하는 앱에는 전체 앱 데이터 관리 기능이 없을 수 있습니다. Astra Control은 다음과 같은 Kubernetes 리소스를 수집합니다.
 - 클러스터 역할
 - ClusterRoleBinding 을 참조하십시오
 - ConfigMap을 클릭합니다
 - 사용자 지정 리소스 정의
 - CustomResource 를 선택합니다
 - DaemonSet
 - 구축
 - DeploymentConfig(배포 구성)
 - 침투
 - mutatingWebhook
 - PersistentVolumeClaim
 - 포드
 - ReplicaSet입니다
 - RoleBinding 을 클릭합니다
 - 역할
 - 루트
 - 비밀
 - 서비스

- 서비스 계정
- StatefulSet 을 선택합니다
- Webhook을 확인합니다

지원되는 앱 설치 방법

Astra Control은 다음과 같은 응용 프로그램 설치 방법을 지원합니다.

- * 매니페스트 파일 *: Astra Control은 kubectl을 사용하여 매니페스트 파일에서 설치된 앱을 지원합니다. 예를 들면 다음과 같습니다.

```
kubectl apply -f myapp.yaml
```

- * Helm 3 *: Helm을 사용하여 앱을 설치하는 경우 Astra Control에 Helm 버전 3이 필요합니다. Helm 3(또는 Helm 2에서 Helm 3으로 업그레이드)과 함께 설치된 앱의 관리 및 클론 생성이 완벽하게 지원됩니다. Helm 2가 설치된 앱 관리는 지원되지 않습니다.
- * 운영자 구축 앱 *: Astra Control은 네임스페이스 범위 연산자와 함께 설치된 앱을 지원합니다. 이러한 연산자는 일반적으로 "pass-by-reference" 아키텍처가 아니라 "pass-by-value"로 설계되었습니다. 다음은 이러한 패턴을 따르는 일부 운영자 앱에 대한 설명입니다.
 - "아파치 K8ssandra"
 - "젠킨스 CI"
 - "Percona XtraDB 클러스터"

Astra Control은 "pass-by-reference" 아키텍처(예: CockroachDB 운영자)로 설계된 운영자를 복제하지 못할 수 있습니다. 이러한 유형의 클론 복제 작업 중에 클론 복제 운영자는 클론 복제 프로세스의 일부로 고유한 새로운 암호가 있음에도 불구하고 소스 운영자의 Kubernetes 암호를 참조하려고 합니다. Astra Control이 소스 운영자의 Kubernetes 암호를 모르기 때문에 클론 작업이 실패할 수 있습니다.



운영자와 설치하는 앱은 동일한 네임스페이스를 사용해야 합니다. 운영자가 배포 .YAML 파일을 수정해야 할 수도 있습니다.

클러스터에 앱을 설치합니다

클러스터를 Astra Control에 추가했으므로 이제 클러스터에서 앱을 설치하거나 기존 앱을 관리할 수 있습니다. 네임스페이스로 범위가 지정된 모든 앱을 관리할 수 있습니다. Pod가 온라인 상태가 되면 Astra Control을 사용하여 앱을 관리할 수 있습니다.

제어 차트에서 검증된 애플리케이션을 구축하는 데 도움이 필요한 경우 다음을 참조하십시오.

- "제어 차트에서 MariaDB를 배포합니다"
- "제어 차트에서 MySQL을 배포합니다"
- "제어 차트에서 Postgres를 배포합니다"
- "제어 차트에서 Jenkins를 배포합니다"

앱 관리

Astra Control을 사용하면 네임스페이스 수준 또는 Kubernetes 레이블로 앱을 관리할 수 있습니다.



Helm 2와 함께 설치된 앱은 지원되지 않습니다.

다음 작업을 수행하여 앱을 관리할 수 있습니다.

- 앱 관리
 - 네임스페이스로 앱 관리
 - Kubernetes 레이블로 앱 관리
- 앱을 무시합니다
- 앱 관리 취소



Astra Control 자체는 표준 앱이 아니며 "시스템 앱"입니다. Astra Control 자체를 관리하려고 해서는 안 됩니다. 관리 시 Astra Control 자체는 기본적으로 표시되지 않습니다. 시스템 앱을 보려면 "Show system apps(시스템 앱 표시)" 필터를 사용합니다.

Astra Control API를 사용하여 앱을 관리하는 방법에 대한 지침은 [를 참조하십시오 "Astra 자동화 및 API 정보".](#)



데이터 보호 작업(클론, 백업, 복원)과 후속 영구 볼륨 크기 조정 후 UI에 새 볼륨 크기가 표시되기까지 최대 20분이 지연됩니다. 데이터 보호 작업이 몇 분 내에 성공적으로 완료되면 스토리지 백엔드에 관리 소프트웨어를 사용하여 볼륨 크기 변경을 확인할 수 있습니다.

네임스페이스로 앱 관리

앱 페이지의 * 검색됨 * 섹션에는 네임스페이스와 해당 네임스페이스에서 Helm이 설치한 앱 또는 사용자 지정 레이블이 지정된 앱이 표시됩니다. 각 앱을 개별적으로 또는 네임스페이스 수준에서 관리하도록 선택할 수 있습니다. 데이터 보호 작업에 필요한 세분화 수준으로 세분화됩니다.

예를 들어 주 단위 주기를 가진 "Maria"에 대한 백업 정책을 설정할 수 있지만, "MariaDB"(동일한 이름 공간에 있음)를 더 자주 백업해야 할 수 있습니다. 이러한 요구사항에 따라 단일 네임스페이스가 아닌 앱을 별도로 관리해야 합니다.

Astra Control을 사용하면 계층 구조의 수준(네임스페이스 및 해당 네임스페이스의 앱)을 모두 개별적으로 관리할 수 있지만, 가장 좋은 방법은 하나 또는 다른 수준을 선택하는 것입니다. 작업이 네임스페이스 및 앱 수준에서 동시에 발생하면 Astra Control에서 수행하는 작업이 실패할 수 있습니다.

단계

1. 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 검색된 * 를 선택합니다.

Apps						
Actions		+ Define	All Clusters	Search	Managed	Discovered 54 Ignored
	Name	Ready	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	default		se [REDACTED]	grp_default	2021/06/28 17:36 UTC	<button>Managed</button>
<input type="checkbox"/>	default1		se [REDACTED]	grp1_default	2021/06/28 17:36 UTC	<button>Unmanaged</button>
<input type="checkbox"/>	default2		se [REDACTED]	grp2_default	2021/06/28 17:36 UTC	<button>Unmanaged</button>
<input type="checkbox"/>	netapp-acc-operator		se [REDACTED]	netapp-acc-operator	2021/07/13 12:36 UTC	<button>Unmanaged</button>
<input type="checkbox"/>	pcloud		se [REDACTED]	pcloud	2021/07/13 12:37 UTC	<button>Unmanaged</button>

3. 검색된 네임스페이스 목록을 봅니다. 네임스페이스를 확장하여 앱 및 관련 리소스를 봅니다.

Astra Control은 네임스페이스에서 Helm 앱 및 사용자 지정 레이블 앱을 보여 줍니다. Helm 레이블을 사용할 수 있는 경우 태그 아이콘으로 지정됩니다.

4. 응용 프로그램이 실행 중인 이름 공간(폴더 아이콘으로 지정됨)을 확인하려면 * Group * 열을 확인합니다.
5. 각 앱을 개별적으로 관리할지 아니면 네임스페이스 수준에서 관리할지 결정합니다.
6. 계층 구조에서 원하는 수준에서 원하는 앱을 찾고 작업 메뉴에서 * 관리 *를 선택합니다.
7. 앱을 관리하지 않으려면 앱 옆의 작업 메뉴에서 * 무시 *를 선택합니다.

예를 들어 "Maria" 네임스페이스의 모든 앱을 함께 관리하여 동일한 스냅샷 및 백업 정책을 가지려면 네임스페이스를 관리하고 네임스페이스의 앱을 무시해야 합니다.

8. 관리되는 앱 목록을 보려면 디스플레이 필터로 * Managed *를 선택합니다.

Apps							
Actions		+ Define	All Clusters	Search	Managed	Discovered 54 Ignored	
	Name	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	app1			[REDACTED]	app-logging	2021/06/28 17:36 UTC	<button>Available</button> Snapshot Backup Clone Unmanage

방금 추가한 앱에는 Protected(보호) 열 아래에 백업 중이 아니며 아직 백업이 예약되지 않았음을 나타내는 경고 아이콘이 있습니다.

9. 특정 앱의 세부 정보를 보려면 앱 이름을 선택합니다.

결과

관리하기로 선택한 앱은 이제 * Managed * 탭에서 사용할 수 있습니다. 무시된 앱은 * ignored * 탭으로 이동합니다. 검색된 탭에 앱이 표시되지 않으므로 새 앱을 설치하면 찾아서 관리하기가 더 쉬워집니다.

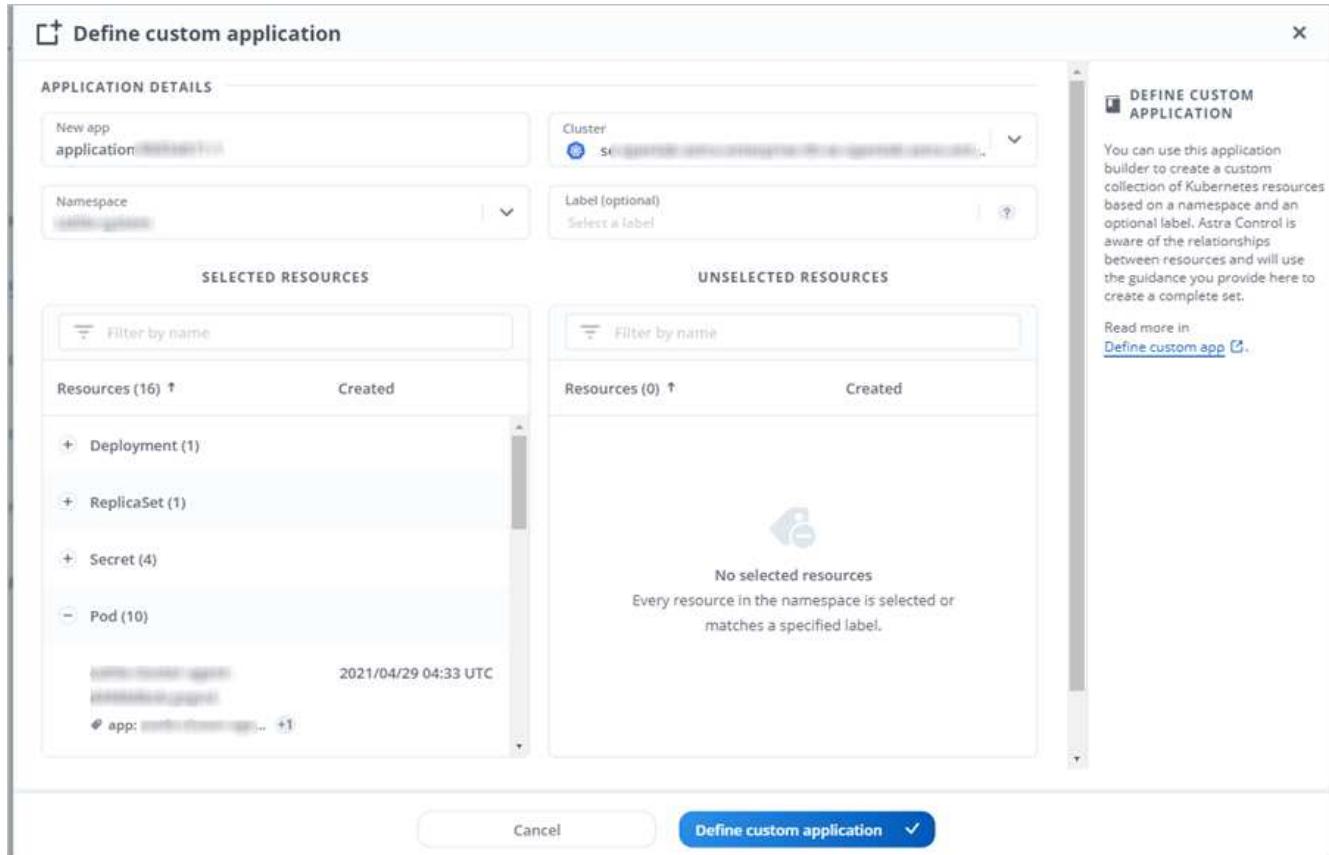
Kubernetes 레이블로 앱 관리

Astra Control에는 응용 프로그램 페이지 상단에 * 사용자 정의 앱 정의 *라는 작업이 포함되어 있습니다. 이 작업을

통해 Kubernetes 레이블로 식별된 앱을 관리할 수 있습니다. ["Kubernetes 레이블로 맞춤형 앱을 정의하는 방법에 대해 자세히 알아보십시오."](#).

단계

1. 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 정의 * 를 선택합니다.



3. 사용자 정의 응용 프로그램 정의 * 대화 상자에서 응용 프로그램을 관리하는 데 필요한 정보를 제공합니다.
 - a. * 새 앱 *: 앱의 표시 이름을 입력합니다.
 - b. * 클러스터 *: 앱이 있는 클러스터를 선택합니다.
 - c. * 네임스페이스*: 앱의 네임스페이스를 선택합니다.
 - d. * 레이블*: 레이블을 입력하거나 아래 리소스에서 레이블을 선택합니다.
 - e. * 선택한 리소스 *: 보호하려는 선택한 Kubernetes 리소스(Pod, 기밀, 영구 볼륨 등)를 보고 관리합니다.
 - 리소스를 확장하고 레이블 수를 선택하여 사용 가능한 레이블을 봅니다.
 - 레이블 중 하나를 선택합니다.레이블을 선택하면 * Label * (레이블 *) 필드에 표시됩니다. 또한 Astra Control은 선택한 레이블과 일치하지 않는 리소스를 표시하도록 * 선택되지 않은 리소스 * 섹션을 업데이트합니다.
 - f. 선택하지 않은 리소스 *: 보호하지 않을 앱 리소스를 확인합니다.
4. 사용자 정의 응용 프로그램 정의 * 를 선택합니다.

결과

Astra Control은 앱 관리를 지원합니다. 이제 * Managed * 탭에서 찾을 수 있습니다.

앱을 무시합니다

앱이 검색된 경우 검색된 목록에 표시됩니다. 이 경우 검색된 목록을 정리하여 새로 설치된 새 앱을 보다 쉽게 찾을 수 있습니다. 또는 관리하고 있는 앱이 있을 수 있으며 나중에 더 이상 앱을 관리하지 않기로 결정할 수 있습니다. 이러한 앱을 관리하지 않으려면 해당 앱을 무시해야 함을 나타낼 수 있습니다.

또한 하나의 네임스페이스(네임스페이스 관리)에서 앱을 관리할 수도 있습니다. 네임스페이스에서 제외할 앱을 무시할 수 있습니다.

단계

1. 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 검색됨 * 을 필터로 선택합니다.
3. 앱을 선택합니다.
4. 작업 메뉴에서 * 무시 * 를 선택합니다.
5. 무시 해제하려면 동작 메뉴에서 * 무시 해제 * 를 선택합니다.

앱 관리 취소

더 이상 앱을 백업, 스냅샷 또는 클론 복제하지 않으려는 경우 관리를 중지할 수 있습니다.



앱 관리를 해제하면 이전에 생성된 모든 백업 또는 스냅샷이 손실됩니다.

단계

1. 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 필터로 * Managed * 를 선택합니다.
3. 앱을 선택합니다.
4. Actions 메뉴에서 * Unmanage * 를 선택합니다.
5. 정보를 검토합니다.
6. "unmanage"를 입력하여 확인합니다.
7. 예, 응용 프로그램 관리 취소 * 를 선택합니다.

시스템 앱은 어떻습니까?

Astra Control은 Kubernetes 클러스터에서 실행 중인 시스템 앱을 검색합니다. 도구 모음의 클러스터 필터 아래에서 * 시스템 앱 표시 * 확인란을 선택하여 시스템 앱을 표시할 수 있습니다.

이러한 시스템 앱은 기본적으로 표시되지 않습니다. 백업해야 하는 경우는 드롭니다.



Astra Control 자체는 표준 앱이 아니며 "시스템 앱"입니다. Astra Control 자체를 관리하려고 해서는 안 됩니다. 관리 시 Astra Control 자체는 기본적으로 표시되지 않습니다. 시스템 앱을 보려면 "Show system apps(시스템 앱 표시)" 필터를 사용합니다.

자세한 내용을 확인하십시오

- ["Astra Control API를 사용합니다"](#)

사용자 지정 앱 예제를 정의합니다

사용자 지정 앱을 생성하면 Kubernetes 클러스터의 요소를 단일 앱으로 그룹화할 수 있습니다.

사용자 지정 앱을 사용하면 다음을 비롯하여 Astra Control 작업에 포함할 항목을 보다 세부적으로 제어할 수 있습니다.

- 복제
- 스냅샷
- 백업
- 보호 정책

대부분의 경우 전체 앱에서 Astra Control의 기능을 사용해야 합니다. 그러나 사용자 지정 앱을 만들어 네임스페이스에서 Kubernetes 개체에 할당하는 레이블을 통해 이러한 기능을 사용할 수도 있습니다.

사용자 지정 앱을 만들려면 앱 페이지로 이동하여 **+정의**를 선택합니다.

사용자 지정 앱 창에는 사용자 지정 앱에서 포함 또는 제외할 리소스가 표시됩니다. 이렇게 하면 사용자 지정 앱을 정의하는 올바른 기준을 선택할 수 있습니다.



맞춤형 앱은 단일 클러스터에서 지정된 네임스페이스 내에서만 생성할 수 있습니다. Astra Control은 사용자 지정 응용 프로그램이 여러 네임스페이스 또는 클러스터를 확장하는 기능을 지원하지 않습니다.

레이블은 식별을 위해 Kubernetes 객체에 할당할 수 있는 키/값 쌍입니다. 레이블을 사용하면 Kubernetes 오브젝트를 더 쉽게 정렬, 구성 및 찾을 수 있습니다. Kubernetes 레이블에 대해 자세히 알아보려면 ["Kubernetes 공식 문서를 참조하십시오"](#).



이름이 다른 동일한 리소스에 대해 정책을 중복하면 데이터 충돌이 발생할 수 있습니다. 리소스에 대한 사용자 지정 앱을 만드는 경우 다른 정책에 따라 복제되거나 백업되지 않도록 해야 합니다.

예: Canary 릴리즈에 대한 별도의 보호 정책

이 예제에서 DevOps 팀은 카나리아 릴리스 배포를 관리합니다. 그들의 클러스터에는 Nginx를 실행하는 3개의 포드가 있습니다. 포드 중 2개는 안정적인 릴리스 전용입니다. 세 번째 포드는 카나리 해제 시 사용합니다.

DevOps 팀의 Kubernetes 관리자가 안정적인 릴리즈 포드에 'duekment=stable'이라는 레이블을 추가합니다. 개발 팀은 카나리 릴리즈 포드에 'deement=canary' 레이블을 추가합니다.

이 팀의 안정적인 릴리즈에는 시간별 스냅샷 및 일일 백업에 대한 요구 사항이 포함됩니다. 카나리아 릴리스는 수명이 길기 때문에 '배포 = 카나리'라고 표시된 모든 것에 대해 공격적이고 단기적인 보호 정책을 만들고자 합니다.

데이터 충돌을 방지하기 위해 관리자는 Canary 릴리스용 응용 프로그램과 안정적 릴리스용 응용 프로그램의 두 가지 사용자 지정 응용 프로그램을 만듭니다. 이렇게 하면 두 Kubernetes 객체 그룹에 대해 백업, 스냅샷 및 클론 작업이 분리됩니다.

단계

- 팀이 Astra Control에 클러스터를 추가한 후 다음 단계는 사용자 지정 앱을 정의하는 것입니다. 이를 위해 팀은 앱 페이지에서 +정의 버튼을 선택합니다.
- 팝업 창이 나타나면 팀은 앱 이름으로 devops-canary-deployment를 설정합니다. 팀은 클러스터 드롭다운에서 클러스터를 선택한 다음 네임스페이스 드롭다운에서 해당 앱의 네임스페이스를 선택합니다.
- 팀은 **Labels** 필드에 duement=canary를 입력하거나 아래 나열된 리소스에서 해당 레이블을 선택할 수 있습니다.
- 카나리아 배포를 위한 사용자 지정 앱을 정의한 후 팀은 안정적인 배포를 위해 프로세스를 반복합니다.

팀이 두 개의 사용자 지정 앱을 모두 만든 후에는 이러한 리소스를 다른 Astra Control 응용 프로그램으로 처리할 수 있습니다. Kubernetes 레이블을 기반으로 각 리소스 그룹에 대해 클론을 생성하고, 백업과 스냅샷을 생성하고, 사용자 지정 보호 정책을 생성할 수 있습니다.

앱 보호

보호 개요

Astra Control Center를 사용하여 앱에 대한 백업, 클론, 스냅샷 및 보호 정책을 생성할 수 있습니다. 앱을 백업하면 서비스 및 관련 데이터를 가능한 한 사용할 수 있습니다. 재해 시나리오 중에 백업에서 복원하면 애플리케이션 및 관련 데이터를 중단 없이 완벽하게 복구할 수 있습니다. 백업, 클론, 스냅샷을 사용하면 랜섬웨어, 우발적인 데이터 손실 및 환경 재해와 같은 일반적인 위협으로부터 보호할 수 있습니다. ["Astra Control Center에서 사용 가능한 데이터 보호 유형과 사용 시기에 대해 알아보십시오".](#)

애플리케이션 보호 워크플로우

다음 예제 워크플로를 사용하여 앱 보호를 시작할 수 있습니다.

[1개] 모든 앱을 백업합니다

앱을 즉시 보호하려면 ["모든 앱의 수동 백업을 생성합니다"](#).

【2개】 각 앱에 대한 보호 정책을 구성합니다

향후 백업 및 스냅샷 자동화 "각 앱에 대한 보호 정책을 구성합니다". 예를 들어 주별 백업과 일별 스냅샷으로 시작할 수 있으며 두 가지 모두에 대해 한 달 동안 보존할 수 있습니다. 수동 백업 및 스냅샷보다 보호 정책을 사용하여 백업 및 스냅샷을 자동화하는 것이 좋습니다.

【세 가지】 선택 사항: 보호 정책을 조정합니다

앱과 사용 패턴이 변경되면 최적의 보호 기능을 제공하기 위해 필요에 따라 보호 정책을 조정합니다.

【네】 재해가 발생할 경우 앱을 복원합니다

데이터 손실이 발생하면 를 통해 복구할 수 있습니다 "최신 백업을 복원하는 중입니다" 각 앱에 대해 먼저 그런 다음 최신 스냅샷을 복구할 수 있습니다(사용 가능한 경우).

스냅샷 및 백업으로 애플리케이션 보호

자동화된 보호 정책을 사용하거나 필요에 따라 스냅샷과 백업을 생성하여 앱을 보호합니다. Astra UI 또는 를 사용할 수 있습니다 "Astra Control API" 앱을 보호합니다.



Helm을 사용하여 앱을 배포하는 경우 Astra Control Center에 Helm 버전 3이 필요합니다. Helm 3으로 배포된 애플리케이션 관리 및 복제(또는 Helm 2에서 Helm 3으로 업그레이드)가 완벽하게 지원됩니다. Helm 2와 함께 배포된 앱은 지원되지 않습니다.



OpenShift 클러스터에서 앱을 호스팅하기 위한 프로젝트(또는 Kubernetes 네임스페이스)에 SecurityContext UID가 할당됩니다. Astra Control Center에서 앱을 보호하고 OpenShift의 다른 클러스터 또는 프로젝트로 앱을 이동하려면 해당 앱을 UID로 실행할 수 있는 정책을 추가해야 합니다. 예를 들어 다음 OpenShift CLI 명령은 WordPress 앱에 적절한 정책을 부여합니다.

```
OC new-project WordPress OC adm policy add-SCC-to-group anyuid  
system:serviceaccounts:WordPress의 OC adm policy add-SCC-to-user privileged-z default-n  
WordPress
```

보호 정책을 구성합니다

보호 정책은 정의된 일정에 따라 스냅샷, 백업 또는 둘 다를 생성하여 앱을 보호합니다. 시간별, 일별, 주별 및 월별 스냅샷과 백업을 생성하도록 선택할 수 있으며, 보존할 복제본 수를 지정할 수 있습니다. 예를 들어 보호 정책은 주별 백업과 일별 스냅샷을 생성하고 백업 및 스냅샷을 한 달 동안 보존할 수 있습니다. 스냅샷 및 백업을 생성하는 빈도와 보관 기간은 조직의 요구 사항에 따라 다릅니다.

단계

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 데이터 보호 * 를 선택합니다.
3. 보호 정책 구성 * 을 선택합니다.
4. 시간별, 일별, 주별 및 월별로 유지할 스냅샷 및 백업 수를 선택하여 보호 스케줄을 정의합니다.

시간별, 일별, 주별 및 월별 스케줄을 동시에 정의할 수 있습니다. 보존 레벨을 설정하기 전에는 스케줄이 활성화되지 않습니다.

다음 예에서는 스냅샷 및 백업의 경우 매시간, 일별, 주별 및 월별로 4개의 보호 스케줄을 설정합니다.

PROTECTION SCHEDULE

- Hourly**: Every hour on the 0th minute, keep the last 4 snapshots.
- Daily**: Daily at 02:00 (UTC), keep the last 15 snapshots.
- Weekly**: Weekly on Mondays at 02:00 (UTC), keep the last 26 snapshots.
- Monthly**: Every 1st of the month at 02:00 (UTC), keep the last 12 backups.

SELECTED: Weekly

BACKUP DESTINATION

Bucket: ntp-nautilus-bucket-10 - ntp-nautils-bucket-10 (Default)

OVERVIEW

Schedule and retention

Define a policy to continuously protect your application on a schedule and configure a retention count to get started.

For select stateful applications, expect I/O to pause for a short time during a backup or snapshot operation.

Read more in [Protection policies](#).

5. Review * 를 선택합니다.
6. 보호 정책 설정 * 을 선택합니다

결과

Astra Control Center는 사용자가 정의한 스케줄 및 보존 정책을 사용하여 스냅샷 및 백업을 생성하고 유지함으로써 데이터 보호 정책을 구현합니다.

스냅샷을 생성합니다

언제든지 주문형 스냅샷을 생성할 수 있습니다.

단계

1. 응용 프로그램 * 을 선택합니다.
2. 원하는 앱의 * Actions * 열에서 드롭다운 목록을 선택합니다.
3. 스냅샷 * 을 선택합니다.
4. 스냅샷의 이름을 사용자 지정한 다음 * Review * 를 선택합니다.
5. 스냅샷 요약을 검토하고 * Snapshot * 을 선택합니다.

결과

스냅샷 프로세스가 시작됩니다. 데이터 보호 * > * 스냅샷 * 페이지의 * 작업 * 열에서 * 사용 가능 * 상태가 되면 스냅샷이 성공적으로 생성됩니다.

백업을 생성합니다

언제든지 앱을 백업할 수도 있습니다.



Astra Control Center의 S3 버킷은 가용 용량을 보고하지 않습니다. Astra Control Center에서 관리하는 앱을 백업 또는 클론 생성하기 전에 ONTAP 또는 StorageGRID 관리 시스템에서 버킷 정보를 확인하십시오.

단계

1. 응용 프로그램 * 을 선택합니다.
2. 원하는 앱의 * Actions * 열에서 드롭다운 목록을 선택합니다.
3. 백업 * 을 선택합니다.
4. 백업 이름을 사용자 지정합니다.
5. 기존 스냅샷에서 앱을 백업할지 여부를 선택합니다. 이 옵션을 선택하면 기존 스냅샷 목록에서 선택할 수 있습니다.
6. 스토리지 버킷 목록에서 선택하여 백업 대상을 선택합니다.
7. Review * 를 선택합니다.
8. 백업 요약을 검토하고 * Backup * 을 선택합니다.

결과

Astra Control Center는 앱 백업을 생성합니다.



네트워크에 정전이 발생했거나 비정상적으로 느린 경우 백업 작업이 시간 초과될 수 있습니다. 이로 인해 백업이 실패합니다.



실행 중인 백업을 중지할 방법은 없습니다. 백업을 삭제해야 하는 경우 백업이 완료될 때까지 기다린 다음의 지침을 따르십시오 [백업을 삭제합니다](#). 실패한 백업을 삭제하려면 "Astra Control API를 사용합니다".



데이터 보호 작업(클론, 백업, 복원)과 후속 영구 볼륨 크기 조정 후 UI에 새 볼륨 크기가 표시되기까지 최대 20분이 지연됩니다. 데이터 보호 작업이 몇 분 내에 성공적으로 완료되며 스토리지 백엔드에 관리 소프트웨어를 사용하여 볼륨 크기 변경을 확인할 수 있습니다.

스냅샷 및 백업을 봅니다

Data Protection 탭에서 앱의 스냅샷 및 백업을 볼 수 있습니다.

단계

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 데이터 보호 * 를 선택합니다.

스냅샷은 기본적으로 표시됩니다.

3. 백업 목록을 보려면 * backups * 를 선택합니다.

스냅샷을 삭제합니다

더 이상 필요하지 않은 예약된 스냅샷 또는 주문형 스냅샷을 삭제합니다.

단계

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 데이터 보호 * 를 선택합니다.
3. 원하는 스냅샷의 * Actions * 열에서 드롭다운 목록을 선택합니다.
4. 스냅샷 삭제 * 를 선택합니다.
5. 삭제를 확인하려면 "delete"라는 단어를 입력하고 * Yes, Delete snapshot * 을 선택합니다.

결과

Astra Control Center가 스냅샷을 삭제합니다.

백업을 삭제합니다

더 이상 필요하지 않은 예약된 백업 또는 필요 시 백업을 삭제합니다.



실행 중인 백업을 중지할 방법은 없습니다. 백업을 삭제해야 하는 경우 백업이 완료될 때까지 기다린 후 다음 지침을 따르십시오. 실패한 백업을 삭제하려면 "[Astra Control API를 사용합니다](#)".

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 데이터 보호 * 를 선택합니다.
3. Backups * 를 선택합니다.
4. 원하는 백업의 * Actions * 열에서 드롭다운 목록을 선택합니다.
5. 백업 삭제 * 를 선택합니다.
6. 삭제를 확인하려면 "delete"라는 단어를 입력하고 * Yes, Delete backup * 을 선택합니다.

결과

Astra Control Center가 백업을 삭제합니다.

앱 복원

Astra Control은 스냅샷 또는 백업에서 애플리케이션을 복원할 수 있습니다. 애플리케이션을 동일한 클러스터로 복구할 경우 기존 스냅샷에서 복구하는 속도가 빨라집니다. Astra Control UI 또는 를 사용할 수 있습니다 "[Astra Control API](#)" 앱을 복원합니다.



Helm을 사용하여 앱을 배포하는 경우 Astra Control Center에 Helm 버전 3이 필요합니다. Helm 3으로 배포된 애플리케이션 관리 및 복제(또는 Helm 2에서 Helm 3으로 업그레이드)가 완벽하게 지원됩니다. Helm 2와 함께 배포된 앱은 지원되지 않습니다.



다른 클러스터로 복원하는 경우 클러스터에서 동일한 영구 볼륨 액세스 모드(예: ReadWriteMany)를 사용하고 있는지 확인합니다. 대상 영구 볼륨 액세스 모드가 다르면 복원 작업이 실패합니다.



OpenShift 클러스터에서 앱을 호스팅하기 위한 프로젝트를 만들면 프로젝트(또는 Kubernetes 네임스페이스)에 SecurityContext UID가 할당됩니다. Astra Control Center에서 앱을 보호하고 OpenShift의 다른 클러스터 또는 프로젝트로 앱을 이동하려면 해당 앱을 UID로 실행할 수 있는 정책을 추가해야 합니다. 예를 들어 다음 OpenShift CLI 명령은 WordPress 앱에 적절한 정책을 부여합니다.

```
OC new-project WordPress OC adm policy add-SCC-to-group anyuid  
system:serviceaccounts:WordPress의 OC adm policy add-SCC-to-user privileged-z default-n  
WordPress
```

단계

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 데이터 보호 * 를 선택합니다.
3. 스냅샷에서 복구하려면 * 스냅샷 * 아이콘을 선택한 상태로 유지합니다. 그렇지 않으면 * Backups * 아이콘을 선택하여 백업에서 복원합니다.
4. 복구할 스냅샷 또는 백업의 * Actions * 열에서 드롭다운 목록을 선택합니다.
5. Restore application * 을 선택합니다.
6. * Restore details *: 복원된 앱에 대한 세부 정보를 지정합니다. 기본적으로 현재 클러스터와 네임스페이스가 표시됩니다. 앱을 원래 상태로 복원하려면 이 값을 그대로 두십시오. 이렇게 하면 앱이 이전 버전으로 되돌아갑니다. 다른 클러스터 또는 네임스페이스로 복원하려는 경우 이 값을 변경합니다.
 - 앱의 이름과 네임스페이스를 입력합니다.
 - 앱의 대상 클러스터를 선택합니다.
 - Review * 를 선택합니다.
7. * 복원 요약 *: 복원 작업에 대한 세부 정보를 검토하고 "복원"을 입력한 다음 * 복원 * 을 선택합니다.

결과

Astra Control Center는 사용자가 제공한 정보를 기반으로 앱을 복원합니다. 앱을 제자리에 복원한 경우 기존 영구 볼륨의 콘텐츠가 복원된 앱의 영구 볼륨 내용으로 바뀝니다.



데이터 보호 작업(클론, 백업, 복원)과 후속 영구 볼륨 크기 조정 후 UI에 새 볼륨 크기가 표시되기까지 최대 20분이 지연됩니다. 데이터 보호 작업이 몇 분 내에 성공적으로 완료되며 스토리지 백엔드에 관리 소프트웨어를 사용하여 볼륨 크기 변경을 확인할 수 있습니다.

애플리케이션 클론 복제 및 마이그레이션

기존 앱을 클론 복제하여 동일한 Kubernetes 클러스터 또는 다른 클러스터에 중복 앱을 생성합니다. Kubernetes 클러스터 간에 애플리케이션 및 스토리지를 이동해야 하는 경우 클로닝에 도움이 될 수 있습니다. 예를 들어, CI/CD 파이프라인과 Kubernetes 네임스페이스 전체에서 워크로드를 이동할 수 있습니다. Astra UI 또는 를 사용할 수 있습니다 "[Astra Control API](#)" 앱을 클론 복제 및 마이그레이션합니다.



StorageClass가 명시적으로 설정된 앱을 배포하고 앱을 복제해야 하는 경우 타겟 클러스터에 원래 지정된 StorageClass가 있어야 합니다. 명시적으로 StorageClass를 동일한 StorageClass가 없는 클러스터로 설정한 애플리케이션을 클론 복제하면 실패합니다.



Jenkins CI의 운영자 배포 인스턴스를 복제하는 경우 영구 데이터를 수동으로 복원해야 합니다. 이는 앱 배포 모델의 제한 사항입니다.



클러스터 간에 앱을 복제하는 경우 소스 클러스터와 대상 클러스터는 OpenShift의 배포 환경과 동일해야 합니다. 예를 들어 OpenShift 4.7 클러스터에서 앱을 클론하는 경우 OpenShift 4.7인 대상 클러스터를 사용합니다.

Astra Control Center에서 앱을 클론하면 애플리케이션 구성 및 영구 스토리지의 클론이 생성됩니다.



Astra Control Center의 S3 버킷은 가용 용량을 보고하지 않습니다. Astra Control Center에서 관리하는 앱을 백업 또는 클론 생성하기 전에 ONTAP 또는 StorageGRID 관리 시스템에서 버킷 정보를 확인하십시오.



OpenShift 클러스터에서 앱을 호스팅하기 위한 프로젝트를 만들면 프로젝트(또는 Kubernetes 네임스페이스)에 SecurityContext UID가 할당됩니다. Astra Control Center에서 앱을 보호하고 OpenShift의 다른 클러스터 또는 프로젝트로 앱을 이동하려면 해당 앱을 UID로 실행할 수 있는 정책을 추가해야 합니다. 예를 들어 다음 OpenShift CLI 명령은 WordPress 앱에 적절한 정책을 부여합니다.

```
OC new-project WordPress OC adm policy add-SCC-to-group anyuid  
system:serviceaccounts:WordPress의 OC adm policy add-SCC-to-user privileged-z default-n  
WordPress
```

필요한 것

앱을 다른 클러스터로 클론 복제하려면 기본 버킷이 필요합니다. 첫 번째 버킷을 추가하면 기본 버킷을 사용할 수 있습니다.

단계

1. 응용 프로그램 * 을 선택합니다.
2. 다음 중 하나를 수행합니다.
 - 원하는 앱의 * Actions * 열에서 드롭다운 목록을 선택합니다.
 - 원하는 앱의 이름을 선택하고 페이지 오른쪽 상단의 상태 드롭다운 목록을 선택합니다.
3. 클론 * 을 선택합니다.
4. * 클론 세부 정보 *: 클론에 대한 세부 정보 지정:
 - 이름을 입력합니다.
 - 클론의 네임스페이스를 입력합니다.
 - 클론의 대상 클러스터를 선택합니다.
 - 기존 스냅샷이나 백업에서 클론을 생성할지 여부를 선택합니다. 이 옵션을 선택하지 않으면 Astra Control Center는 앱의 현재 상태에서 클론을 생성합니다.
5. * 소스 *: 기존 스냅샷 또는 백업에서 복제하도록 선택한 경우 사용할 스냅샷 또는 백업을 선택합니다.
6. Review * 를 선택합니다.
7. * 클론 요약 *: 클론에 대한 세부 정보를 검토하고 * 클론 * 을 선택합니다.

결과

Astra Control Center는 사용자가 제공한 정보를 기반으로 해당 앱을 복제합니다. 새 앱 클론이 * 응용 프로그램 * 페이지의 "사용 가능" 상태가 되면 클론 작업이 성공적으로 수행됩니다.



데이터 보호 작업(클론, 백업, 복원)과 후속 영구 볼륨 크기 조정 후 UI에 새 볼륨 크기가 표시되기까지 최대 20분이 지연됩니다. 데이터 보호 작업이 몇 분 내에 성공적으로 완료되면 스토리지 백엔드에 관리 소프트웨어를 사용하여 볼륨 크기 변경을 확인할 수 있습니다.

앱 실행 후크 관리

실행 후크는 관리되는 앱의 스냅샷 전후에 실행할 수 있는 사용자 지정 스크립트입니다. 예를 들어 데이터베이스 앱이 있는 경우 실행 후크를 사용하여 스냅샷 전에 모든 데이터베이스 트랜잭션을 일시 중지하고 스냅샷이 완료된 후 트랜잭션을 다시 시작할 수 있습니다. 따라서 애플리케이션 정합성이 보장되는 스냅샷이 보장됩니다.

기본 실행 후크 및 정규식

일부 애플리케이션의 경우 Astra Control은 NetApp에서 제공하는 기본 실행 후크와 함께 제공되며, 스냅샷 전후에 고정 및 고정 작업을 처리합니다. Astra Control은 정규식을 사용하여 앱의 컨테이너 이미지를 다음과 같은 앱에 일치시킵니다.

- MariaDB
 - 일치하는 정규식:\bmariadb\b
- MySQL
 - 일치 정규식:\bmysql\b
- PostgreSQL
 - 일치하는 정규식:\bpostgres\b

일치하는 항목이 있으면 해당 앱에 대한 NetApp 제공 기본 실행 후크가 앱의 활성 실행 후크 목록에 나타나고, 해당 앱의 스냅샷을 생성하면 해당 후크가 자동으로 실행됩니다. 사용자 지정 앱 중 하나에 정규식과 일치하는 유사한 이미지 이름이 있는 경우(기본 실행 후크를 사용하지 않으려는 경우) 이미지 이름을 변경할 수 있습니다. 또는 해당 앱에 대한 기본 실행 후크를 비활성화하고 대신 사용자 지정 후크를 사용합니다.

기본 실행 후크는 삭제하거나 수정할 수 없습니다.

사용자 정의 실행 후크에 대한 중요 참고 사항

앱에 대한 실행 후크를 계획할 때 다음 사항을 고려하십시오.

- Astra Control을 사용하려면 실행 가능한 셀 스크립트 형식으로 실행 후크를 작성해야 합니다.
- 스크립트 크기는 128KB로 제한됩니다.
- Astra Control은 실행 후크 설정 및 모든 일치 기준을 사용하여 스냅샷에 적용할 후크를 결정합니다.
- 모든 실행 후크 오류는 소프트 장애이며, 후크에 장애가 발생해도 다른 후크와 스냅샷이 시도됩니다. 그러나 후크가 실패하면 * Activity * 페이지 이벤트 로그에 경고 이벤트가 기록됩니다.
- 실행 후크를 생성, 편집 또는 삭제하려면 소유자, 관리자 또는 구성원 권한이 있는 사용자여야 합니다.
- 실행 후크를 실행하는 데 25분 이상 걸리는 경우 후크에 장애가 발생하고 반환 코드가 "N/A"인 이벤트 로그 항목이 생성됩니다. 영향을 받는 모든 스냅샷은 시간 초과되어 실패로 표시되며, 그 결과 이벤트 로그 항목이 시간 초과를

나타냅니다.



실행 후크는 실행 중인 응용 프로그램의 기능을 줄이거나 완전히 비활성화하기 때문에 사용자 지정 실행 후크가 실행되는 시간을 최소화해야 합니다.

스냅샷이 실행되면 실행 후크 이벤트가 다음 순서로 발생합니다.

1. NetApp에서 제공하는 기본 사전 스냅샷 실행 후크는 해당 컨테이너에서 실행됩니다.
2. 해당되는 모든 사용자 지정 사전 스냅샷 실행 후크는 해당 컨테이너에서 실행됩니다. 필요에 따라 사용자 지정 사전 스냅샷 후크를 생성하고 실행할 수 있지만 스냅샷이 보장되거나 구성 가능해지기 전에 이러한 후크의 실행 순서가 보장되지 않습니다.
3. 스냅샷이 수행됩니다.
4. 해당되는 모든 사용자 지정 사후 스냅샷 실행 후크는 해당 컨테이너에서 실행됩니다. 필요에 따라 사용자 지정 사후 스냅샷 후크를 생성하고 실행할 수 있지만 스냅샷 후에 이러한 후크를 실행하는 순서는 보장되거나 구성할 수 없습니다.
5. NetApp에서 제공하는 모든 기본 사후 스냅샷 실행 후크는 해당 컨테이너에서 실행됩니다.



운영 환경에서 실행 후크 스크립트를 사용하려면 항상 해당 스크립트를 테스트해야 합니다. 'kubbeck exec' 명령을 사용하여 스크립트를 편리하게 테스트할 수 있습니다. 운영 환경에서 실행 후크를 활성화한 후 결과 스냅샷을 테스트하여 정합성이 보장되는지 확인합니다. 앱을 임시 네임스페이스에 클론 복제하고 스냅샷을 복구한 다음 앱을 테스트하여 이 작업을 수행할 수 있습니다.

기존 실행 후크를 봅니다

앱에 대한 기존 맞춤형 또는 NetApp에서 제공한 기본 실행 후크를 볼 수 있습니다.

단계

1. 응용 프로그램 * 으로 이동한 다음 관리되는 응용 프로그램의 이름을 선택합니다.
2. Execution hook * 탭을 선택합니다.

결과 목록에서 사용 가능하거나 비활성화된 실행 후크를 모두 볼 수 있습니다. 후크의 상태, 소스 및 실행 시간(사전 또는 사후 스냅샷)을 확인할 수 있습니다. 실행 후크를 둘러싼 이벤트 로그를 보려면 왼쪽 탐색 영역의 * Activity * 페이지로 이동합니다.

사용자 지정 실행 후크를 만듭니다

앱의 사용자 정의 실행 후크를 만들 수 있습니다. 을 참조하십시오 ["실행 후크 예"](#) 후크 예 실행 후크를 만들려면 소유자, 관리자 또는 구성원 권한이 있어야 합니다.



실행 후크로 사용할 사용자 정의 쉘 스크립트를 작성할 때는 Linux 명령을 실행하거나 실행 파일에 대한 전체 경로를 제공하지 않는 한 파일 시작 부분에 적절한 셀을 지정해야 합니다.

단계

1. 응용 프로그램 * 을 선택한 다음 관리되는 응용 프로그램의 이름을 선택합니다.
2. Execution hook * 탭을 선택합니다.
3. 새 후크 추가 * 를 선택합니다.

4. 후크 세부 정보 * 영역에서 후크를 실행해야 하는 시기에 따라 * 사전 스냅샷 * 또는 * 사후 스냅샷 * 을 선택합니다.
 5. 후크의 고유한 이름을 입력합니다.
 6. (선택 사항) 실행 중에 후크에 전달할 인수를 입력하고 각 인수 뒤에 Enter 키를 눌러 각 인수를 기록합니다.
 7. Container Images * (컨테이너 이미지 *) 영역에서 응용 프로그램에 포함된 모든 컨테이너 이미지에 대해 후크를 실행해야 하는 경우 * Apply to all container images * (모든 컨테이너 이미지에 적용) 확인란을 활성화합니다. 대신 후크가 하나 이상의 지정된 컨테이너 이미지에만 동작해야 하는 경우 일치시킬 * 컨테이너 이미지 이름 필드에 컨테이너 이미지 이름을 입력합니다.
 8. Script * 영역에서 다음 중 하나를 수행합니다.
 - 사용자 지정 스크립트를 업로드합니다.
 - i. 파일 업로드 * 옵션을 선택합니다.
 - ii. 파일을 찾아 업로드합니다.
 - iii. 스크립트에 고유한 이름을 지정합니다.
 - iv. (선택 사항) 다른 관리자가 스크립트에 대해 알아야 하는 참고 사항을 입력합니다.
 - 클립보드에서 사용자 정의 스크립트를 붙여 넣습니다.
 - i. 클립보드에서 붙여넣기 * 옵션을 선택합니다.
 - ii. 텍스트 필드를 선택하고 필드에 스크립트 텍스트를 붙여 넣습니다.
 - iii. 스크립트에 고유한 이름을 지정합니다.
 - iv. (선택 사항) 다른 관리자가 스크립트에 대해 알아야 하는 참고 사항을 입력합니다.
9. 후크 추가 * 를 선택합니다.

실행 후크를 비활성화합니다

앱 스냅샷 전후에 실행 후크가 실행되지 않도록 임시로 설정하려면 실행 후크를 사용하지 않도록 설정할 수 있습니다. 실행 후크를 비활성화하려면 소유자, 관리자 또는 구성원 권한이 있어야 합니다.

단계

1. 응용 프로그램 * 을 선택한 다음 관리되는 응용 프로그램의 이름을 선택합니다.
2. Execution hook * 탭을 선택합니다.
3. 비활성화하려는 후크에 대해 * 작업 * 드롭다운을 선택합니다.
4. 비활성화 * 를 선택합니다.

실행 후크를 삭제합니다

더 이상 필요 없는 경우 실행 후크를 완전히 제거할 수 있습니다. 실행 후크를 삭제하려면 소유자, 관리자 또는 구성원 권한이 있어야 합니다.

단계

1. 응용 프로그램 * 을 선택한 다음 관리되는 응용 프로그램의 이름을 선택합니다.
2. Execution hook * 탭을 선택합니다.
3. 삭제할 후크에 대한 * 작업 * 드롭다운을 선택합니다.

4. 삭제 * 를 선택합니다.

실행 후크 예

다음 예제를 사용하여 실행 후크를 구조화하는 방법에 대해 알아보십시오. 이러한 후크를 템플릿 또는 테스트 스크립트로 사용할 수 있습니다.

간단한 성공 사례

다음은 표준 출력 및 표준 오류에 성공하여 메시지를 기록하는 간단한 후크의 예입니다.

```
#!/bin/sh

# success_sample.sh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
```

```

        msg "ERROR: $*" 1>&2
    }

# 
# main
# 

# log something to stdout
info "running success_sample.sh"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

단순한 성공 사례(**bash** 버전)

다음은 **bash**용으로 작성된 표준 출력 및 표준 오류에 성공하여 메시지를 쓰는 간단한 후크의 예입니다.

```

#!/bin/bash

# success_sample.bash
#
# A simple noop success hook script for testing purposes.
#
# args: None

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

```

```

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.bash"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

간단한 성공 사례(zsh 버전)

다음은 Z 셸에 대해 작성된 표준 출력 및 표준 오류에 성공하여 메시지를 기록하는 간단한 후크의 예입니다.

```

#!/bin/zsh

# success_sample.zsh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

```

```

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#
# log something to stdout
info "running success_sample.zsh"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

인수 성공 예제

다음 예제에서는 후크에 args를 사용하는 방법을 보여 줍니다.

```

#!/bin/sh

# success_sample_args.sh
#
# A simple success hook script with args for testing purposes.
#
# args: Up to two optional args that are echoed to stdout

#
# Writes the given message to standard output
#
# $* - The message to write
#

```

```

msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#
# log something to stdout
info "running success_sample_args.sh"

# collect args
arg1=$1
arg2=$2

# output args and arg count to stdout
info "number of args: $#"
info "arg1 ${arg1}"
info "arg2 ${arg2}"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

사전 스냅샷/사후 스냅샷 후크의 예

다음 예제에서는 사전 스냅샷 및 사후 스냅샷 후크에 대해 동일한 스크립트를 사용하는 방법을 보여 줍니다.

```
#!/bin/sh

# success_sample_pre_post.sh
#
# A simple success hook script example with an arg for testing purposes
# to demonstrate how the same script can be used for both a prehook and
posthook
#
# args: [pre|post]

# unique error codes for every error case
ebase=100
eusage=$((ebase+1))
ebadstage=$((ebase+2))
epre=$((ebase+3))
epost=$((ebase+4))

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
```

```

error() {
    msg "ERROR: $*" 1>&2
}

#
# Would run prehook steps here
#
prehook() {
    info "Running noop prehook"
    return 0
}

#
# Would run posthook steps here
#
posthook() {
    info "Running noop posthook"
    return 0
}

#
# main
#

# check arg
stage=$1
if [ -z "${stage}" ]; then
    echo "Usage: $0 <pre|post>"
    exit ${eusage}
fi

if [ "${stage}" != "pre" ] && [ "${stage}" != "post" ]; then
    echo "Invalid arg: ${stage}"
    exit ${ebadstage}
fi

# log something to stdout
info "running success_sample_pre_post.sh"

if [ "${stage}" = "pre" ]; then
    prehook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during prehook"

```

```

    fi
fi

if [ "${stage}" = "post" ]; then
    posthook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during posthook"
    fi
fi

exit ${rc}

```

실패 예

다음 예제에서는 후크의 장애를 처리하는 방법을 보여 줍니다.

```

#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

```

```

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}

```

자세한 정보 표시 실패 예

다음 예제에서는 더 자세한 정보 로깅을 사용하여 후크의 오류를 처리하는 방법을 보여 줍니다.

```

#!/bin/sh

# failure_sample_verbose.sh
#
# A simple failure hook script with args for testing purposes.
#
# args: [The number of lines to output to stdout]

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

```

```

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#
# log something to stdout
info "running failure_sample_verbose.sh"

# output arg value to stdout
linecount=$1
info "line count ${linecount}"

# write out a line to stdout based on line count arg
i=1
while [ "$i" -le ${linecount} ]; do
    info "This is line ${i} from failure_sample_verbose.sh"
    i=$(( i + 1 ))
done

error "exiting with error code 8"
exit 8

```

종료 코드 예제에 오류가 발생했습니다

다음 예제에서는 종료 코드와 함께 후크 실패를 보여 줍니다.

```

#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#
# log something to stdout
info "running failure_sample_arg_exit_code.sh"

```

```

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}

```

실패 후 성공 예

다음 예제에서는 후크가 처음 실행될 때 후크가 실패하지만 두 번째 실행 후에 후크가 발생하는 방법을 보여 줍니다.

```

#!/bin/sh

# failure_then_success_sample.sh
#
# A hook script that fails on initial run but succeeds on second run for
testing purposes.
#
# Helpful for testing retry logic for post hooks.
#
# args: None
#
#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#

```

```

# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#
# log something to stdout
info "running failure_success sample.sh"

if [ -e /tmp/hook-test.junk ] ; then
    info "File does exist. Removing /tmp/hook-test.junk"
    rm /tmp/hook-test.junk
    info "Second run so returning exit code 0"
    exit 0
else
    info "File does not exist. Creating /tmp/hook-test.junk"
    echo "test" > /tmp/hook-test.junk
    error "Failed first run, returning exit code 5"
    exit 5
fi

```

앱 및 클러스터 상태 보기

앱 및 클러스터 상태 요약 보기

대시보드 * 를 선택하면 앱, 클러스터, 스토리지 백엔드 및 상태를 한눈에 파악할 수 있습니다.

The screenshot shows the Astra Control Center dashboard with the following details:

- Resource summary:**
 - Apps:** 4 Managed, All healthy, 0 Not fully protected, 48 Discovered.
 - Clusters:** 2 Managed, All healthy.
 - Storage backends:** 2 Managed, All healthy, 0 Discovered.
- Left sidebar:**
 - Dashboard (selected)
 - Apps
 - Clusters
 - Backends
 - Buckets
 - Account
 - Activity
 - Support
- Top right:** Welcome To Astra, !
- License notice:** You are currently using an Astra Control Center Evaluation license, which is scheduled to expire on September 15, 2021. Select Upgrade license for information about how to upgrade to the full version of Astra Control Center.
- Upgrade license button:** Upgrade license →

이것들은 단지 정적 숫자나 상태만이 아니라, 이것들 각각에서 드릴다운할 수 있습니다. 예를 들어 앱이 완전히 보호되지 않은 경우 아이콘 위로 마우스를 가져가면 완전히 보호되지 않은 앱을 확인할 수 있습니다. 여기에는 이유가 포함됩니다.

응용 프로그램 타일

응용 프로그램* 타일은 다음 사항을 식별하는 데 도움이 됩니다.

- 현재 관리 중인 애플리케이션 수는 Astra입니다.
- 관리된 앱이 정상 상태인지 여부
- 애플리케이션이 완전히 보호되는지 여부(최근 백업을 사용할 수 있는 경우 보호됨)
- 검색되었지만 아직 관리되지 않은 앱의 수입니다.

앱을 검색한 후 관리하거나 무시하면 되므로 이 숫자는 0이 되는 것이 좋습니다. 그런 다음 대시보드에서 검색된 앱의 수를 모니터링하여 개발자가 클러스터에 새 앱을 추가하는 시기를 파악할 수 있습니다.

클러스터 타일

클러스터 * 타일은 Astra Control Center를 사용하여 관리하고 있는 클러스터의 상태에 대한 유사한 세부 정보를 제공하며, 앱을 사용하는 것처럼 드릴다운하여 더 자세한 정보를 얻을 수 있습니다.

저장소 백엔드 타일

저장소 백엔드 * 타일은 다음을 포함하여 저장소 백엔드의 상태를 식별하는 데 도움이 되는 정보를 제공합니다.

- 관리되는 스토리지 백엔드 수
- 이러한 관리되는 백엔드가 정상 상태인지 여부
- 백엔드가 완전히 보호되는지 여부
- 검색되었지만 아직 관리되지 않은 백엔드 수입니다.

클러스터의 상태 및 세부 정보를 봅니다

Astra Control Center에서 관리할 클러스터를 추가한 후에는 클러스터의 위치, 작업자 노드, 영구 볼륨 및 스토리지 클래스 등의 클러스터에 대한 세부 정보를 볼 수 있습니다.

단계

1. Astra Control Center UI에서 * Clusters * 를 선택합니다.
2. 클러스터 * 페이지에서 세부 정보를 확인할 클러스터를 선택합니다.
3. Overview *, * Storage * 및 * Activity * 탭에서 원하는 정보를 확인할 수 있습니다.
 - * 개요 *: 해당 상태를 포함한 작업자 노드에 대한 세부 정보.
 - * 스토리지 *: 스토리지 클래스 및 상태를 비롯하여 컴퓨팅과 연관된 영구 볼륨입니다.
 - * Activity *: 클러스터와 관련된 활동을 표시합니다.



Astra Control Center * 대시보드 * 부터 클러스터 정보를 볼 수도 있습니다. 리소스 요약 * 의 * 클러스터 * 탭에서 * 클러스터 * 페이지로 이동하는 관리 클러스터를 선택할 수 있습니다. 클러스터 * 페이지로 이동한 후 위에 설명된 단계를 따릅니다.

앱의 상태 및 세부 정보를 봅니다

앱 관리를 시작한 후 Astra는 앱의 상태(정상 여부), 보호 상태(장애 시 완전히 보호되는지 여부), Pod, 영구 스토리지 등을 식별할 수 있는 앱에 대한 세부 정보를 제공합니다.

단계

1. Astra Control Center UI에서 * 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 원하는 정보를 찾습니다.

앱 상태

Kubernetes의 앱 상태를 반영하는 상태를 제공합니다. 예를 들어, Pod와 영구 볼륨을 온라인으로 전환합니까? 앱이 정상 상태가 아닌 경우 Kubernetes 로그를 확인하여 클러스터에서 문제를 해결해야 합니다. Astra는 고장 난 앱을 수정하는 데 도움이 되는 정보를 제공하지 않습니다.

앱 보호 상태

앱이 얼마나 잘 보호되는지 상태를 제공합니다.

- * 완전 보호 *: 이 앱에는 활성 백업 스케줄과 1주일 미만의 성공적인 백업이 있습니다
- * 부분 보호됨 *: 응용 프로그램에 활성 백업 일정, 활성 스냅샷 일정 또는 백업 또는 스냅샷이 있습니다
- * 보호되지 않음 *: 완전히 보호되거나 부분적으로 보호되지 않는 앱

최근 백업 _ 이(가) 있을 때까지 완전히 보호할 수 없습니다. 백업은 영구 볼륨으로부터 멀리 떨어진 개체 저장소에 저장되기 때문에 이 작업이 중요합니다. 장애 또는 사고로 인해 클러스터가 삭제되며 영구적 저장소인 경우 복구할 백업이 필요합니다. 스냅샷을 사용하면 복구할 수 없습니다.

개요

앱과 연결된 Pod의 상태에 대한 정보입니다.

데이터 보호

데이터 보호 정책을 구성하고 기존 스냅샷 및 백업을 볼 수 있습니다.

스토리지

에는 애플리케이션 레벨의 영구 볼륨이 나와 있습니다. 영구 볼륨의 상태는 Kubernetes 클러스터의 관점에서 나옵니다.

리소스

백업 및 관리되는 리소스를 확인할 수 있습니다.

활동입니다

앱 관련 활동을 보여줍니다.



Astra Control Center * Dashboard * 부터 앱 정보를 볼 수도 있습니다. 리소스 요약 * 의 * 응용 프로그램 * 탭에서 * 응용 프로그램 * 페이지로 이동하는 관리되는 앱을 선택할 수 있습니다. 응용 프로그램 * 페이지로 이동한 후 위에 설명된 단계를 따릅니다.

계정을 관리합니다

사용자 관리

Astra Control Center UI를 사용하여 Astra Control Center 설치 사용자를 추가, 제거 및 편집할 수 있습니다. Astra UI 또는 를 사용할 수 있습니다 "Astra Control API" 를 눌러 사용자를 관리합니다.

사용자 추가

계정 소유자와 관리자는 Astra Control Center 설치에 사용자를 더 추가할 수 있습니다.

단계

1. 계정 관리 * 탐색 영역에서 * 계정 * 을 선택합니다.
2. 사용자 * 탭을 선택합니다.
3. 사용자 추가 * 를 선택합니다.
4. 사용자 이름, 이메일 주소 및 임시 암호를 입력합니다.

사용자는 처음 로그인할 때 암호를 변경해야 합니다.

5. 적절한 시스템 권한이 있는 사용자 역할을 선택합니다.

각 역할은 다음과 같은 권한을 제공합니다.

- Viewer * 는 리소스를 볼 수 있습니다.
- 구성원 * 은 뷰어 역할 권한을 가지고 있으며 앱과 클러스터를 관리할 수 있지만 앱이나 클러스터를 관리하거나 스냅샷 또는 백업을 삭제할 수는 없습니다.
- Admin * 은 구성원 역할 권한을 가지며 소유자를 제외한 다른 사용자를 추가 및 제거할 수 있습니다.
- 소유자 * 는 관리자 역할 권한을 가지며 모든 사용자 계정을 추가 및 제거할 수 있습니다.

6. 추가 * 를 선택합니다.

암호 관리

Astra Control Center에서 사용자 계정의 암호를 관리할 수 있습니다.

암호를 변경합니다

언제든지 사용자 계정의 암호를 변경할 수 있습니다.

단계

1. 화면 오른쪽 상단에서 사용자 아이콘을 선택합니다.
2. 프로파일 * 을 선택합니다.
3. 작업 * 드롭다운 목록을 선택하고 * 암호 변경 * 을 선택합니다.
4. 암호 요구 사항에 맞는 암호를 입력합니다.
5. 암호를 다시 입력하여 확인합니다.

6. 암호 변경 * 을 선택합니다.

다른 사용자의 암호를 재설정합니다

계정에 관리자 또는 소유자 역할 권한이 있는 경우 다른 사용자 계정과 사용자의 암호를 재설정할 수 있습니다. 암호를 재설정할 때 사용자가 로그인할 때 변경해야 하는 임시 암호를 할당합니다.

단계

1. 계정 관리 * 탐색 영역에서 * 계정 * 을 선택합니다.
2. 사용자 * 탭의 * 상태 * 열에서 사용자의 드롭다운 목록을 선택합니다.
3. 암호 재설정 * 을 선택합니다.
4. 암호 요구 사항에 맞는 임시 암호를 입력합니다.
5. 암호를 다시 입력하여 확인합니다.



다음에 사용자가 로그인할 때 암호를 변경하라는 메시지가 표시됩니다.

6. 비밀번호 재설정 * 을 선택합니다.

사용자의 역할을 변경합니다

소유자 역할을 가진 사용자는 모든 사용자의 역할을 변경할 수 있지만 관리자 역할을 가진 사용자는 관리자, 구성원 또는 뷰어 역할을 가진 사용자의 역할을 변경할 수 있습니다.

단계

1. 계정 관리 * 탐색 영역에서 * 계정 * 을 선택합니다.
2. 사용자 * 탭의 * 역할 * 열에 있는 드롭다운 목록을 선택합니다.
3. 새 역할을 선택한 다음 메시지가 표시되면 * 역할 변경 * 을 선택합니다.

결과

Astra Control Center는 선택한 새 역할에 따라 사용자의 권한을 업데이트합니다.

사용자를 제거합니다

소유자 또는 관리자 역할을 가진 사용자는 언제든지 계정에서 다른 사용자를 제거할 수 있습니다.

단계

1. 계정 관리 * 탐색 영역에서 * 계정 * 을 선택합니다.
2. 사용자 * 탭에서 제거할 각 사용자 행의 확인란을 선택합니다.
3. Actions * 를 선택하고 * Remove user/s * 를 선택합니다.
4. 메시지가 표시되면 "remove(제거)"라는 단어를 입력한 다음 * Yes, Remove User(예, 사용자 제거) * 를 선택하여 삭제를 확인합니다.

결과

Astra Control Center는 계정에서 사용자를 제거합니다.

알림을 보고 관리합니다

Astra는 작업이 완료되거나 실패했을 때 알려줍니다. 예를 들어, 앱 백업이 성공적으로 완료되면 알림이 표시됩니다.

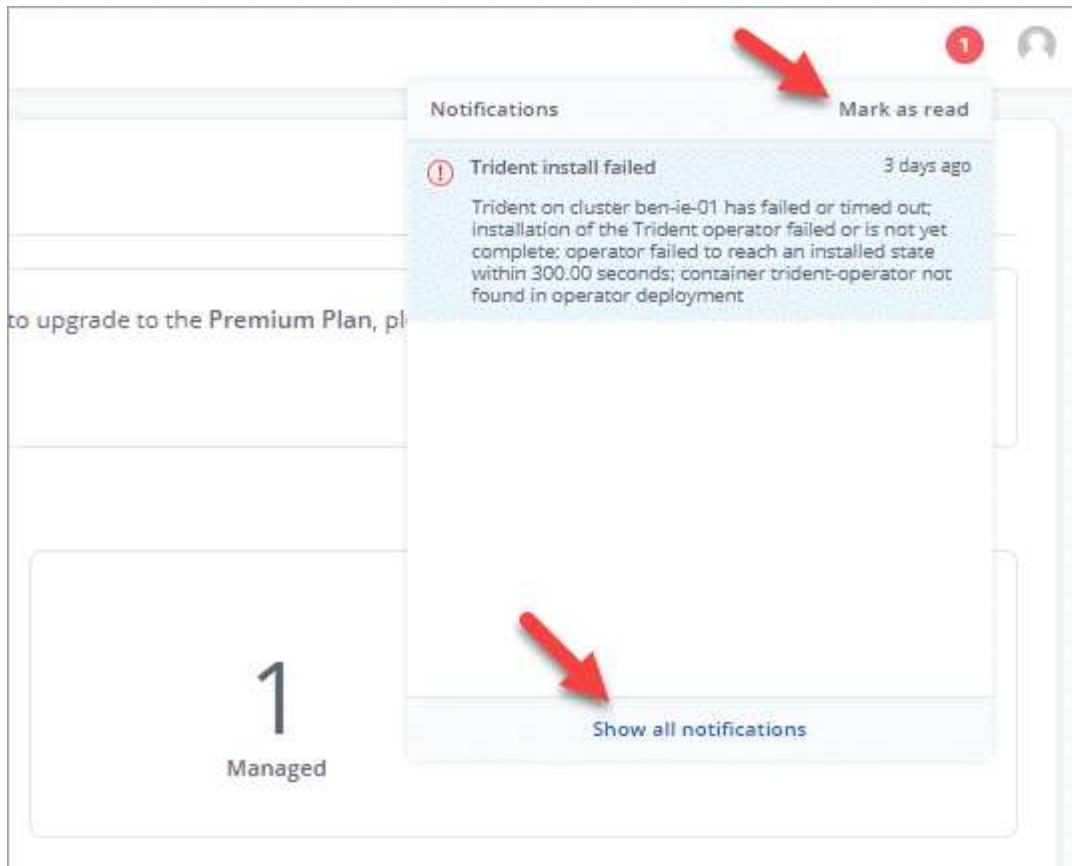
인터페이스의 오른쪽 상단에서 읽지 않은 알림 수를 사용할 수 있습니다.

The screenshot shows the LongBoat IE BC dashboard. In the top right corner, there is a red arrow pointing to a small red circle containing the number '1', indicating an unread alert. The main interface includes a 'Welcome to LongBoat IE BC' message, a 'Manage Plans' button, and a 'Resources summary' section. This section displays two boxes: one for 'Apps' (2/10 Managed) and one for 'Compute' (1 Managed). Below these boxes is a 'Getting started' section with four items: 'Manage Kubernetes compute', 'Add your applications', 'Manage applications', and 'Invite users'. At the bottom left, there is a 'NetApp' logo.

이러한 알림을 보고 읽은 상태로 표시할 수 있습니다. 이렇게 하면 읽지 않은 알림을 지우는 것이 편리합니다.

단계

1. 오른쪽 상단에서 읽지 않은 알림 수를 선택합니다.



2. 알림을 검토한 후 * 읽은 상태로 표시 * 또는 * 모든 알림 표시 * 를 선택합니다.

모든 알림 표시 * 를 선택한 경우 알림 페이지가 로드됩니다.

3. 알림 * 페이지에서 알림을 보고 읽음으로 표시할 알림을 선택하고 * 작업 * 을 선택한 다음 * 읽음으로 표시 * 를 선택합니다.

자격 증명을 추가 및 제거합니다

ONTAP S3, OpenShift로 관리되는 Kubernetes 클러스터, 또는 관리되지 않는 Kubernetes 클러스터와 같은 로컬 프라이빗 클라우드 공급자의 자격 증명을 언제든지 계정에서 추가 및 제거할 수 있습니다. Astra Control Center는 이러한 자격 증명을 사용하여 Kubernetes 클러스터 및 클러스터의 앱을 검색하고 대신 리소스를 프로비저닝합니다.

Astra Control Center의 모든 사용자는 동일한 자격 증명 세트를 공유합니다.

자격 증명을 추가합니다

클러스터를 관리할 때 Astra Control Center에 자격 증명을 추가할 수 있습니다. 새 클러스터를 추가하여 자격 증명을 추가하려면 을 참조하십시오 ["Kubernetes 클러스터 추가"](#).



고유한 "kubeconfig" 파일을 만들 경우 해당 파일에 * 하나의 * 컨텍스트 요소만 정의해야 합니다. 을 참조하십시오 ["Kubernetes 문서"](#) kubeconfig 파일을 만드는 방법에 대한 자세한 내용은

자격 증명을 제거합니다

언제든지 계정에서 자격 증명을 제거합니다. 자격 증명은 이후에 제거해야 합니다 ["연결된 모든 클러스터의 관리를"](#)

취소합니다".



Astra Control Center에 추가하는 첫 번째 자격 증명 세트는 항상 사용 중입니다. Astra Control Center는 자격 증명을 사용하여 백업 버킷에 인증하기 때문입니다. 이러한 자격 증명을 제거하지 않는 것이 좋습니다.

단계

1. 계정 > 자격 증명 * 을 선택합니다.
2. 제거할 자격 증명에 대한 * 상태 * 열의 드롭다운 목록을 선택합니다.
3. 제거 * 를 선택합니다.

The screenshot shows the 'Credentials' tab of the Astra Control Center's Account section. It lists one credential entry:

Name	Type	Provider	Created	Status
scale-sa@astradummy-01	service-account	Google Cloud Platform	2021/01/22 15:37 UTC	Active

On the right side of the row, there is a 'Remove' button with a trash can icon and a cursor pointing at it.

4. 삭제를 확인하려면 "remove(제거)"라는 단어를 입력한 다음 * Yes(예), Remove Credential(자격 증명 제거) * 을 선택합니다.

결과

Astra Control Center는 계정에서 자격 증명을 제거합니다.

계정 활동을 모니터링합니다

Astra Control 계정의 활동에 대한 세부 정보를 볼 수 있습니다. 예를 들어, 새 사용자를 초대하거나, 클러스터를 추가하거나, 스냅샷을 생성할 때 사용할 수 있습니다. 계정 활동을 CSV 파일로 내보낼 수도 있습니다.

Astra Control에서 모든 계정 활동을 봅니다

1. Activity * 를 선택합니다.
2. 필터를 사용하여 활동 목록의 범위를 좁히거나 검색 상자를 사용하여 원하는 항목을 정확하게 찾을 수 있습니다.
3. CSV로 내보내기 * 를 선택하여 계정 활동을 CSV 파일로 다운로드합니다.

특정 앱의 계정 활동을 봅니다

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. Activity * 를 선택합니다.

클러스터의 계정 활동을 봅니다

1. 클러스터 * 를 선택한 다음 클러스터 이름을 선택합니다.

2. Activity * 를 선택합니다.

주의가 필요한 이벤트를 해결하기 위한 조치를 취하십시오

1. Activity * 를 선택합니다.
2. 주의가 필요한 이벤트를 선택합니다.
3. 실행 * 드롭다운 옵션을 선택합니다.

이 목록에서 수행할 수 있는 수정 조치를 확인하고, 문제와 관련된 문서를 보고, 문제 해결을 위한 지원을 받을 수 있습니다.

기존 라이센스를 업데이트합니다

평가판 라이센스를 전체 라이센스로 변환하거나 기존 평가판 또는 전체 라이센스를 새 라이센스로 업데이트할 수 있습니다. 전체 라이센스가 없는 경우 NetApp 세일즈 담당자와 협력하여 전체 라이센스 및 일련 번호를 받으십시오. Astra UI 또는 를 사용할 수 있습니다 ["Astra Control API"](#) 기존 라이센스를 업데이트합니다.

단계

1. 예 로그인합니다 ["NetApp Support 사이트"](#).
2. Astra Control Center 다운로드 페이지에 액세스하여 일련 번호를 입력한 다음 전체 NetApp 라이센스 파일 (NLF)을 다운로드하십시오.
3. Astra Control Center UI에 로그인합니다.
4. 왼쪽 탐색 창에서 * 계정 * > * 라이센스 * 를 선택합니다.
5. 계정 * > * 라이선스 * 페이지에서 기존 라이선스의 상태 드롭다운 메뉴를 선택하고 * 교체 * 를 선택합니다.
6. 다운로드한 라이센스 파일을 찾습니다.
7. 추가 * 를 선택합니다.

Account * > * Licenses * 페이지에는 라이센스 정보, 만료 날짜, 라이센스 일련 번호, 계정 ID 및 사용된 CPU 단위가 표시됩니다.

버킷을 관리합니다

애플리케이션 및 영구 스토리지를 백업하려는 경우나 클러스터 간에 애플리케이션을 클론 복제하려는 경우에는 오브젝트 저장소 버킷 공급자가 필수적입니다. Astra Control Center를 사용하여 객체 저장소 공급자를 오프라인 클러스터, 앱의 백업 대상으로 추가합니다.

애플리케이션 구성과 영구 스토리지를 동일한 클러스터에 클론 복제할 경우 버킷이 필요하지 않습니다.

다음 버킷 공급자 중 하나를 사용하십시오.

- NetApp ONTAP S3
- NetApp StorageGRID S3
- 일반 S3



Astra Control Center는 Amazon S3를 일반 S3 버킷 공급자로 지원하지만, Astra Control Center는 Amazon의 S3 지원을 주장하는 모든 오브젝트 저장소 공급업체를 지원하지 않을 수 있습니다.

버킷은 삭제할 수 없지만 편집할 수는 있습니다.

버킷은 다음 상태 중 하나일 수 있습니다.

- 보류 중: 버킷이 검색되도록 예약되었습니다.
- 사용 가능: 버킷을 사용할 수 있습니다.
- 제거: 현재 버킷에 접근할 수 없습니다.

Astra Control API를 사용하여 버킷을 관리하는 방법에 대한 자침은 [을 참조하십시오 "Astra 자동화 및 API 정보"](#).

버킷 관리와 관련된 다음 작업을 수행할 수 있습니다.

- ["버킷을 추가합니다"](#)
- [버킷을 편집합니다](#)

 Astra Control Center의 S3 버킷은 가용 용량을 보고하지 않습니다. Astra Control Center에서 관리하는 앱을 백업 또는 클론 생성하기 전에 ONTAP 또는 StorageGRID 관리 시스템에서 버킷 정보를 확인하십시오.

자격 증명을 제거합니다

Astra Control API를 사용하여 언제든지 계정에서 S3 자격 증명을 제거합니다.

자세한 내용은 [을 참조하십시오 "Astra Control API를 사용합니다"](#).

 Astra Control에 추가하는 첫 번째 자격 증명 세트는 항상 사용 중입니다. Astra Control은 자격 증명을 사용하여 백업 버킷을 인증하기 때문입니다. 이러한 자격 증명을 제거하지 않는 것이 좋습니다.

버킷을 편집합니다

버킷의 액세스 자격 증명 정보를 변경하고 선택한 버킷이 기본 버킷인지 여부를 변경할 수 있습니다.

 버킷을 추가할 때 올바른 버킷 공급자를 선택하고 해당 공급자에 적합한 자격 증명을 제공합니다. 예를 들어, UI에서 NetApp ONTAP S3를 유형으로 받아들이고 StorageGRID 자격 증명을 받아들이지만, 이 버킷을 사용한 이후의 모든 애플리케이션 백업 및 복원이 실패합니다. [를 참조하십시오 "릴리즈 노트"](#).

단계

1. 왼쪽 탐색에서 * Bucket * 을 선택합니다.
2. 작업 메뉴에서 * 편집 * 을 선택합니다.
3. 버킷 유형 이외의 모든 정보를 변경합니다.



버킷 유형을 수정할 수 없습니다.

4. Update * 를 선택합니다.

자세한 내용을 확인하십시오

- ["Astra Control API를 사용합니다"](#)

스토리지 백엔드를 관리합니다

Astra Control에서 스토리지 클러스터를 스토리지 백엔드로 관리하면 PVS(영구적 볼륨)와 스토리지 백엔드 간의 연결 및 추가 스토리지 메트릭을 얻을 수 있습니다. Astra Control Center가 Cloud Insights에 연결된 경우, 스토리지 용량과 상태 세부 정보를 모니터링할 수 있습니다.

Astra Control API를 사용하여 스토리지 백엔드를 관리하는 방법에 대한 지침은 [를 참조하십시오 "Astra 자동화 및 API 정보"](#).

스토리지 백엔드 관리와 관련된 다음 작업을 완료할 수 있습니다.

- ["스토리지 백엔드를 추가합니다"](#)
- [스토리지 백엔드 세부 정보를 봅니다](#)
- [스토리지 백엔드의 관리를 취소합니다](#)

스토리지 백엔드 세부 정보를 봅니다

Dashboard 또는 Backend 옵션에서 스토리지 백엔드 정보를 볼 수 있습니다.

대시보드에서 스토리지 백엔드 세부 정보를 봅니다

단계

1. 왼쪽 탐색 모음에서 * 대시보드 *를 선택합니다.
2. 상태를 보여 주는 스토리지 백엔드 섹션을 검토합니다.
 - * 비정상 *: 스토리지가 최적 상태가 아닙니다. 이는 지연 시간 문제 또는 컨테이너 문제로 인해 앱 성능이 저하되었기 때문일 수 있습니다.
 - * 모두 정상 *: 스토리지가 관리되었으며 최적의 상태입니다.
 - * 검색됨 *: 스토리지를 검색했지만 Astra Control에서 관리하지 않았습니다.

백엔드 옵션에서 스토리지 백엔드 세부 정보를 봅니다

백엔드 상태, 용량 및 성능(IOPS 처리량 및/또는 지연 시간)에 대한 정보를 봅니다.

Cloud Insights에 연결하면 Kubernetes 애플리케이션에서 사용 중인 볼륨을 볼 수 있습니다. 볼륨은 선택한 스토리지 백엔드에 저장됩니다.

단계

1. 왼쪽 탐색 영역에서 * backends *를 선택합니다.
2. 스토리지 백엔드를 선택합니다.



NetApp Cloud Insights에 연결한 경우 Cloud Insights에서 발췌한 데이터가 백엔드 페이지에 나타납니다.

BASIC INFORMATION

- Type: ONTAP 9.7.0
- Cloud: private
- Credentials: Updated 2021/07/28 21:44 UTC

Persistent volumes

Name	Persistent volume	Capacity	App/s	Cluster/s	Cloud
trident_pvc_...	pvc-...	0.04/46.57 GiB: 0.1%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc-...	0.34/23.28 GiB: 1.44%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc-...	0.02/0.93 GiB: 2.33%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc-...	3.02/50.00 GiB: 6.04%	netapp-acc polaris-mongodb-mongodb	openshift-cluster010	private
trident_pvc_...	pvc-...	0.19/8.00 GiB: 2.39%	apps-mysql mysql-mysql	openshift-cluster010	private
trident_pvc_...	pvc-...	0.41/50.00 GiB: 0.81%	netapp-acc polaris-influxdb2-polaris-influxdb2	openshift-cluster010	private
trident_pvc_...	pvc-...	2.93/50.00 GiB: 5.87%	netapp-acc polaris-mongodb-mongodb	openshift-cluster010	private
trident_pvc_...	pvc-...	0.03/10.00 GiB: 0.36%	netapp-acc polaris-consul-consul	openshift-cluster010	private

3. Cloud Insights로 바로 이동하려면 메트릭 이미지 옆에 있는 * Cloud Insights * 아이콘을 선택합니다.

스토리지 백엔드의 관리를 취소합니다

백엔드의 관리를 해제할 수 있습니다.

단계

- 원쪽 탐색에서 * backends * 를 선택합니다.
- 스토리지 백엔드를 선택합니다.
- Actions 메뉴에서 * Unmanage * 를 선택합니다.
- "unmanage"를 입력하여 제거를 확인합니다.
- Yes, remove storage backend * 를 선택합니다.

자세한 내용을 확인하십시오

- "Astra Control API를 사용합니다"

인프라 모니터링 및 보호

Astra Control Center 환경을 향상시키기 위해 몇 가지 선택적 설정을 구성할 수 있습니다. Astra Control Center를 실행 중인 네트워크에 인터넷에 연결하기 위한 프록시가 필요한 경우(지원 번들을 NetApp 지원 사이트에 업로드하거나

Cloud Insights에 연결하려면 Astra Control Center에서 프록시 서버를 구성해야 합니다. 전체 인프라를 모니터링하고 통찰력을 확보하기 위해 NetApp Cloud Insights에 연결합니다. Astra Control Center에서 모니터링하는 시스템에서 Kubernetes 이벤트를 수집하려면 Fluentd 연결을 추가합니다.

프록시 서버를 추가합니다

Astra Control Center를 실행 중인 네트워크에 인터넷에 연결하기 위한 프록시가 필요한 경우(지원 번들을 NetApp 지원 사이트에 업로드하거나 Cloud Insights에 연결하려면) Astra Control Center에서 프록시 서버를 구성해야 합니다.



Astra Control Center는 프록시 서버에 대해 입력한 세부 정보를 확인하지 않습니다. 올바른 값을 입력했는지 확인하십시오.

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.
2. 계정 * > * 연결 * 을 선택합니다.
3. 드롭다운 목록에서 * 연결 * 을 선택하여 프록시 서버를 추가합니다.



HTTP PROXY

Disconnected

Connect

Configure Astra Control to send traffic through a proxy server.

4. 프록시 서버 이름 또는 IP 주소와 프록시 포트 번호를 입력합니다.
5. 프록시 서버에 인증이 필요한 경우 확인란을 선택하고 사용자 이름과 암호를 입력합니다.
6. Connect * 를 선택합니다.

결과

입력한 프록시 정보가 저장된 경우 * 계정 * > * 연결 * 페이지의 * HTTP 프록시 * 섹션에서 해당 정보가 연결되었음을 나타내고 서버 이름을 표시합니다.



Connected

HTTP PROXY ?

Server: proxy.example.com:8888

Authentication: Enabled

프록시 서버 설정을 편집합니다

프록시 서버 설정을 편집할 수 있습니다.

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.

2. 계정 * > * 연결 * 을 선택합니다.
3. 드롭다운 목록에서 * 편집 * 을 선택하여 연결을 편집합니다.
4. 서버 세부 정보 및 인증 정보를 편집합니다.
5. 저장 * 을 선택합니다.

프록시 서버 연결을 비활성화합니다

프록시 서버 연결을 비활성화할 수 있습니다. 다른 연결이 중단될 수 있다는 것을 비활성화하기 전에 경고가 표시됩니다.

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.
2. 계정 * > * 연결 * 을 선택합니다.
3. 드롭다운 목록에서 * 연결 끊기 * 를 선택하여 연결을 비활성화합니다.
4. 대화 상자가 열리면 작업을 확인합니다.

Cloud Insights에 연결합니다

전체 인프라를 모니터링하고 통찰력을 확보하기 위해 NetApp Cloud Insights를 Astra Control Center 인스턴스와 연결합니다. Cloud Insights는 Astra Control Center 라이선스에 포함되어 있습니다.

Cloud Insights는 Astra Control Center가 사용하는 네트워크나 프록시 서버를 통해 간접적으로 액세스할 수 있어야 합니다.

Astra Control Center가 Cloud Insights에 연결되면 획득 장치 포드가 생성됩니다. 이 Pod는 Astra Control Center에서 관리하는 스토리지 백엔드에서 데이터를 수집하여 Cloud Insights로 푸시합니다. 이 POD에는 8GB RAM과 2개의 CPU 코어가 필요합니다.



Cloud Insights 연결을 설정한 후에는 * backends * 페이지에서 처리량 정보를 확인하고 스토리지 백엔드를 선택한 후 여기에서 Cloud Insights에 연결할 수 있습니다. 또한 클러스터 섹션의 * 대시보드 *에서 정보를 찾고 Cloud Insights에 연결할 수도 있습니다.

필요한 것

- Astra Control Center 계정에는 * admin * / * owner * 권한이 있습니다.
- 유효한 Astra Control Center 라이센스가 있습니다.
- Astra Control Center를 실행 중인 네트워크에 인터넷에 연결하기 위한 프록시가 필요한 경우 프록시 서버



Cloud Insights를 처음 사용하는 경우 기능과 특징을 잘 익히십시오. 을 참조하십시오 ["Cloud Insights 설명서"](#).

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.
2. 계정 * > * 연결 * 을 선택합니다.
3. 연결을 추가하려면 드롭다운 목록에서 * 연결 끊김 * 이 표시되는 * 연결 * 을 선택합니다.



Disconnected

Connect

CLOUD INSIGHTS

Monitor and analyze your applications and the infrastructure they run on.

4. Cloud Insights API 토큰 및 테넌트 URL을 입력합니다. 테넌트 URL의 형식은 다음과 같습니다.

```
https://<environment-name>.c01.cloudinsights.netapp.com/
```

Cloud Insights 라이센스가 있으면 테넌트 URL을 가져옵니다. 테넌트 URL이 없는 경우 을 참조하십시오 ["Cloud Insights 설명서"](#).

- 를 다운로드하십시오 ["API 토큰"](#)에서 Cloud Insights 테넌트 URL에 로그인합니다.
- Cloud Insights에서 * 관리자 * > * API 액세스 * 를 클릭하여 * 읽기/쓰기 * 와 * 읽기 전용 * API 액세스 토큰을 모두 생성합니다.

The screenshot shows the Cloud Insights interface with the title "Cloud Insights (Trial)" at the top. The left sidebar has a "MONITOR & OPTIMIZE" section with links for HOME, DASHBOARDS, QUERIES, ALERTS, REPORTS, MANAGE, ADMIN (which is selected), CLOUD SECURE, and HELP. The main content area is titled "API Access Tokens (4)". It shows a table with columns: Name, Description, Token, API Type, and Permission. The tokens listed are:

Name	Description	Token	API Type	Permission
astr...		...zBskB1	All Categories	Read/Write
astr...		...xKOeL...	All Categories	Read/Write
astr...		...2_A6HP	All Categories	Read Only
astr...		...8BTKYY	All Categories	Read/Write

- 읽기 전용 * 키를 복사합니다. Cloud Insights 연결을 활성화하려면 Astra Control Center 창에 붙여 넣어야 합니다. Read API Access Token 키 권한에 대해 Assets, Alerts, Acquisition Unit 및 Data Collection을 선택합니다.
- 읽기/쓰기 * 키를 복사합니다. Astra Control Center * Connect Cloud Insights * 창에 붙여 넣어야 합니다. 읽기/쓰기 API 액세스 토큰 키 권한에 대해 자산, 데이터 수집, 로그 수집, 획득 단위, 및 데이터 수집 을 참조하십시오.



읽기 전용 * 키와 * 읽기/쓰기 * 키를 생성하고 두 가지 용도로 동일한 키를 사용하지 않는 것이 좋습니다. 기본적으로 토큰 만료 기간은 1년으로 설정됩니다. 토큰이 만료되기 전에 토큰을 최대 지속 시간으로 지정할 수 있도록 기본 선택을 유지하는 것이 좋습니다. 토큰이 만료되면 원격 측정이 중지됩니다.

- Cloud Insights에서 복사한 키를 Astra Control Center에 붙여 넣습니다.

5. Connect * 를 선택합니다.



연결을 선택하면 * 연결 상태가 * 계정 * > * 연결 * 페이지의 * Cloud Insights * 섹션에서 * 보류 * 로 변경됩니다. 연결이 활성화되고 상태가 * 연결됨 * 으로 변경되는 데 몇 분 정도 걸릴 수 있습니다.



Astra Control Center와 Cloud Insights UI 사이를 쉽게 오갈 수 있도록 두 가지 모두에 로그인했는지 확인하십시오.

Cloud Insights에서 데이터를 봅니다

연결에 성공하면 * 계정 * > * 연결 * 페이지의 * Cloud Insights * 섹션에 연결된 것으로 표시되고 테넌트 URL이 표시됩니다. Cloud Insights를 방문하여 성공적으로 수신 및 표시된 데이터를 볼 수 있습니다.

The screenshot shows the 'Connections' tab in the Astra Control Center. Under the 'EXTERNAL' section, there are two entries: 'HTTP PROXY' and 'CLOUD INSIGHTS'. Both entries show a 'Connected' status with a dropdown arrow. The 'HTTP PROXY' entry has a server URL 'proxy.example.com:8888' and authentication status 'Enabled'. The 'CLOUD INSIGHTS' entry has a tenant name 'Cloud Insights'.

어떤 이유로 연결에 실패한 경우 상태가 * 실패 * 로 표시됩니다. UI 오른쪽 상단의 * 알림 * 에서 실패 원인을 찾을 수 있습니다.

The screenshot shows the 'Notifications' page with a red circular badge indicating 33 unread notifications. The first notification is highlighted with a blue background and white text, showing an error message: 'Unable to connect to Cloud Insights' with a timestamp 'an hour ago'. The message continues: 'The Cloud Insights API token is invalid. Create a new API token in Cloud Insights and update Astra Control connection settings with the new token.'

계정 * > * 알림 * 에서 동일한 정보를 찾을 수도 있습니다.

Astra Control Center에서 * backend * 페이지의 처리량 정보를 볼 수 있을 뿐 아니라 스토리지 백엔드를 선택한 후 여기에서 Cloud Insights에 연결할 수도 있습니다.

Cloud Insights로 바로 이동하려면 메트릭 이미지 옆에 있는 * Cloud Insights * 아이콘을 선택합니다.

또한 * 대시보드 *에서 정보를 찾을 수 있습니다.



Cloud Insights 연결을 활성화한 후 Astra 제어 센터에서 추가한 백엔드를 제거하면 백엔드에서 Cloud Insights에 대한 보고를 중지합니다.

Cloud Insights 연결을 편집합니다

Cloud Insights 연결을 편집할 수 있습니다.



API 키만 편집할 수 있습니다. Cloud Insights 테넌트 URL을 변경하려면 Cloud Insights 연결을 끊고 새 URL에 연결하는 것이 좋습니다.

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.
2. 계정 * > * 연결 * 을 선택합니다.
3. 드롭다운 목록에서 * 편집 * 을 선택하여 연결을 편집합니다.
4. Cloud Insights 연결 설정을 편집합니다.
5. 저장 * 을 선택합니다.

Cloud Insights 연결을 비활성화합니다

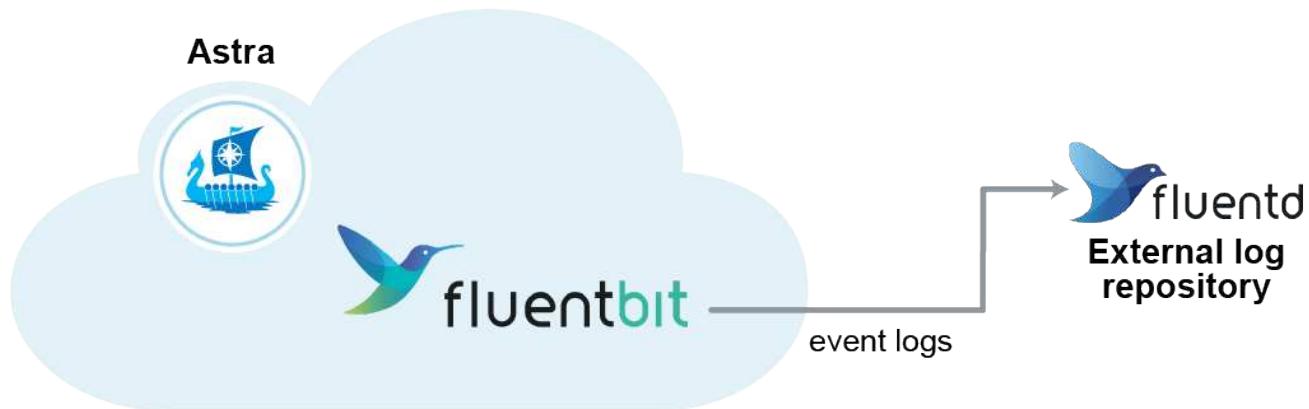
Astra Control Center에서 관리하는 Kubernetes 클러스터에 대한 Cloud Insights 연결을 해제할 수 있습니다. Cloud Insights 연결을 비활성화해도 이미 Cloud Insights에 업로드된 원격 측정 데이터는 삭제되지 않습니다.

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.
2. 계정 * > * 연결 * 을 선택합니다.
3. 드롭다운 목록에서 * 연결 끊기 * 를 선택하여 연결을 비활성화합니다.
4. 대화 상자가 열리면 작업을 확인합니다. 작업을 확인한 후 * 계정 * > * 연결 * 페이지에서 Cloud Insights 상태가 * 보류 * 로 변경됩니다. 상태가 * 연결 끊김 * 으로 변경되는 데 몇 분 정도 걸립니다.

Fluentd에 연결합니다

Astra Control Center에서 Fluentd 엔드포인트로 로그(Kubernetes 이벤트)를 보낼 수 있습니다. Fluentd 연결은 기본적으로 비활성화되어 있습니다.



관리되는 클러스터의 이벤트 로그만 Fluentd로 전달됩니다.

필요한 것

- Astra Control Center 계정에는 * admin * / * owner * 권한이 있습니다.
- Kubernetes 클러스터에 설치 및 실행 중인 Astra Control Center



Astra Control Center는 Fluentd 서버에 대해 입력한 세부 정보를 확인하지 않습니다. 올바른 값을 입력했는지 확인하십시오.

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.
2. 계정 * > * 연결 * 을 선택합니다.
3. 연결을 추가하려면 * 연결 끊김 * 이 표시된 드롭다운 목록에서 * 연결 * 을 선택합니다.



Disconnected ▾

FLUENTD

Connect Astra Control logs to Fluentd for use by your log analysis software.

4. Fluentd 서버의 호스트 IP 주소, 포트 번호 및 공유 키를 입력합니다.

5. Connect * 를 선택합니다.

결과

Fluentd 서버에 대해 입력한 세부 정보가 저장된 경우 * 계정 * > * 연결 * 페이지의 * Fluentd * 섹션에서 해당 정보가 연결되었음을 나타냅니다. 이제 연결한 Fluentd 서버를 방문하여 이벤트 로그를 볼 수 있습니다.

어떤 이유로 연결에 실패한 경우 상태가 * 실패 * 로 표시됩니다. UI 오른쪽 상단의 * 알림 *에서 실패 원인을 찾을 수 있습니다.

계정 * > * 알림 *에서 동일한 정보를 찾을 수도 있습니다.



로그 수집에 문제가 있는 경우 작업자 노드에 로그인하여 로그를 '/var/log/containers/'에서 사용할 수 있는지 확인해야 합니다.

Fluentd 연결을 편집합니다

Fluentd 연결을 Astra Control Center 인스턴스에 편집할 수 있습니다.

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.
2. 계정 * > * 연결 * 을 선택합니다.
3. 드롭다운 목록에서 * 편집 * 을 선택하여 연결을 편집합니다.
4. Fluentd 끝점 설정을 변경합니다.
5. 저장 * 을 선택합니다.

Fluentd 연결을 비활성화합니다

Astra Control Center 인스턴스에 대한 Fluentd 연결을 비활성화할 수 있습니다.

단계

1. admin * / * owner * 권한이 있는 계정을 사용하여 Astra Control Center에 로그인합니다.
2. 계정 * > * 연결 * 을 선택합니다.
3. 드롭다운 목록에서 * 연결 끊기 * 를 선택하여 연결을 비활성화합니다.
4. 대화 상자가 열리면 작업을 확인합니다.

기존 라이센스를 업데이트합니다

평가판 라이센스를 전체 라이센스로 변환하거나 기존 평가판 또는 전체 라이센스를 새 라이센스로 업데이트할 수 있습니다. 전체 라이센스가 없는 경우 NetApp 세일즈 담당자와 협력하여 전체 라이센스 및 일련 번호를 받으십시오. Astra UI 또는 를 사용할 수 있습니다 "[Astra Control API](#)" 기존 라이센스를 업데이트합니다.

단계

1. 에 로그인합니다 ["NetApp Support 사이트"](#).
2. Astra Control Center 다운로드 페이지에 액세스하여 일련 번호를 입력한 다음 전체 NetApp 라이센스 파일

(NLF)을 다운로드하십시오.

3. Astra Control Center UI에 로그인합니다.
4. 왼쪽 탐색 창에서 * 계정 * > * 라이센스 * 를 선택합니다.
5. 계정 * > * 라이선스 * 페이지에서 기존 라이선스의 상태 드롭다운 메뉴를 선택하고 * 교체 * 를 선택합니다.
6. 다운로드한 라이센스 파일을 찾습니다.
7. 추가 * 를 선택합니다.

Account * > * Licenses * 페이지에는 라이센스 정보, 만료 날짜, 라이센스 일련 번호, 계정 ID 및 사용된 CPU 단위가 표시됩니다.

앱 및 클러스터 관리를 취소합니다

Astra Control Center에서 더 이상 관리하지 않으려는 응용 프로그램 또는 클러스터를 제거합니다.

앱 관리를 취소합니다

Astra Control Center에서 더 이상 백업, 스냅샷 또는 클론 복제하지 않을 애플리케이션 관리를 중지합니다.

- 기존 백업 및 스냅샷이 삭제됩니다.
- 애플리케이션과 데이터는 사용 가능한 상태로 유지됩니다.

단계

1. 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 더 이상 관리하지 않을 앱의 확인란을 선택합니다.
3. Action * 메뉴에서 * Unmanage * 를 선택합니다.
4. "unmanage"를 입력하여 확인합니다.
5. 앱 관리를 해제할지 확인한 다음 * 예, 애플리케이션 관리 취소 * 를 선택합니다.

결과

Astra Control Center가 앱 관리를 중지합니다.

클러스터 관리를 취소합니다

Astra Control Center에서 더 이상 관리하지 않으려는 클러스터를 관리 해제합니다.

- 이 작업을 수행하면 Astra Control Center에서 클러스터를 관리할 수 없습니다. 클러스터 구성을 변경하지 않고 클러스터를 삭제하지 않습니다.
- Trident가 클러스터에서 제거되지 않습니다. "[Trident를 제거하는 방법을 알아보십시오](#)".



클러스터를 관리하기 전에 클러스터와 연결된 앱의 관리를 해제해야 합니다.

단계

1. 왼쪽 탐색 모음에서 * 클러스터 * 를 선택합니다.

2. Astra Control Center에서 더 이상 관리하지 않으려는 클러스터의 확인란을 선택합니다.
3. Actions * 메뉴에서 * Unmanage * 를 선택합니다.
4. 클러스터 관리를 해제할지 확인한 다음 * 예, 클러스터 관리 취소 * 를 선택합니다.

결과

클러스터의 상태가 * Removing * 으로 변경되고 그 후에는 * Clusters * 페이지에서 클러스터가 제거되고 Astra Control Center에서 더 이상 관리되지 않습니다.



* Astra Control Center와 Cloud Insights가 연결되지 않은 경우 * 클러스터를 관리하지 않으면 원격 측정 데이터를 전송하기 위해 설치된 모든 리소스가 제거됩니다. * Astra Control Center와 Cloud Insights가 연결되어 있는 경우 * 클러스터를 관리하지 않으면 'fluentbit'와 'event-lavientes' Pod만 삭제됩니다.

Astra Control Center를 업그레이드합니다

Astra Control Center를 업그레이드하려면 NetApp Support 사이트에서 설치 번들을 다운로드하고 해당 지침에 따라 Astra Control Center 구성 요소를 업그레이드하십시오. 이 절차를 사용하여 인터넷에 연결되거나 공기가 연결된 환경에서 Astra Control Center를 업그레이드할 수 있습니다.

필요한 것

- "업그레이드를 시작하기 전에 운영 환경이 Astra Control Center 배포에 대한 최소 요구 사항을 충족하는지 확인하십시오".
- 모든 클러스터 운영자가 양호한 상태이며 사용 가능한지 확인합니다.

OpenShift 예:

```
oc get clusteroperators
```

- 모든 API 서비스가 정상 상태이며 사용 가능한지 확인합니다.

OpenShift 예:

```
oc get apiservices
```

- Astra Control Center에서 로그아웃합니다.

이 작업에 대해

Astra Control Center 업그레이드 프로세스는 다음과 같은 고급 단계를 안내합니다.

- Astra Control Center 번들을 다운로드합니다
- 번들의 포장을 풀고 디렉토리를 변경합니다
- 이미지를 로컬 레지스트리에 추가합니다
- 업데이트된 Astra Control Center 운영자를 설치합니다

- Astra Control Center를 업그레이드합니다
- 타사 서비스를 업그레이드합니다
- 시스템 상태를 확인합니다



Astra Control Center POD를 모두 삭제하지 않도록 업그레이드 프로세스 내내 다음 명령을 실행하지 마십시오: "kubectl delete -f Astra_control_center_operator_deploy.yaml"



스케줄, 백업 및 스냅샷이 실행되고 있지 않은 경우 유지보수 창에서 업그레이드를 수행합니다.



Docker Engine 대신 Red Hat의 Podman 명령을 사용하는 경우 Docker 명령 대신 Podman 명령을 사용할 수 있습니다.

Astra Control Center 번들을 다운로드합니다

1. 에서 Astra Control Center 업그레이드 번들("Astra-control-center-[version].tar.gz")을 다운로드합니다 ["NetApp Support 사이트"](#).
2. (선택 사항) 다음 명령을 사용하여 번들의 서명을 확인합니다.

```
openssl dgst -sha256 -verify astra-control-center[version].pub
-signature <astra-control-center[version].sig astra-control-
center[version].tar.gz
```

번들의 포장을 풀고 디렉토리를 변경합니다

1. 이미지 추출:

```
tar -vxzf astra-control-center-[version].tar.gz
```

2. Astra 디렉토리로 변경합니다.

```
cd astra-control-center-[version]
```

이미지를 로컬 레지스트리에 추가합니다

1. Astra Control Center 이미지 디렉토리의 파일을 로컬 레지스트리에 추가합니다.



아래 이미지의 자동 로드에 대한 샘플 스크립트를 참조하십시오.

- a. Docker 레지스트리에 로그인합니다.

```
docker login [your_registry_path]
```

- b. Docker에 이미지를 로드합니다.
- c. 이미지에 태그를 지정합니다.
- d. 이미지를 로컬 레지스트리에 푸시합니다.

```
export REGISTRY=[your_registry_path]
for astraImageFile in $(ls images/*.tar)
    # Load to local cache. And store the name of the loaded image
    trimming the 'Loaded images: '
    do astraImage=$(docker load --input ${astrImageFile} | sed
's/Loaded image: //')
        astraImage=$(echo ${astrImage} | sed 's!localhost/!!')
        # Tag with local image repo.
        docker tag ${astrImage} ${REGISTRY}/${astrImage}
        # Push to the local repo.
        docker push ${REGISTRY}/${astrImage}
done
```

업데이트된 Astra Control Center 운영자를 설치합니다

1. Astra Control Center 운영자 배포 YAML('Astra_control_center_operator_deploy.YAML')을 편집하여 현지 등록부와 비밀을 참조하십시오.

```
vim astra_control_center_operator_deploy.yaml
```

- a. 인증이 필요한 레지스트리를 사용하는 경우 'imagePullSecrets:[]'의 기본 줄을 다음과 같이 바꿉니다.

```
imagePullSecrets:
- name: <name_of_secret_with_creds_to_local_registry>
```

- b. kuby-RBAC-proxy 이미지의 [your_registry_path]를 이미지를 에서 푸시한 레지스트리 경로로 변경합니다 [이전 단계](#).
- c. "acc-operator-controller-manager" 이미지의 [your_registry_path]를 이미지를 에서 푸시한 레지스트리 경로로 변경합니다 [이전 단계](#).

```

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
        - args:
            - --secure-listen-address=0.0.0.0:8443
            - --upstream=http://127.0.0.1:8080/
            - --logtostderr=true
            - --v=10
          image: [your_registry_path]/kube-rbac-proxy:v4.8.0
          name: kube-rbac-proxy
          ports:
            - containerPort: 8443
              name: https
        - args:
            - --health-probe-bind-address=:8081
            - --metrics-bind-address=127.0.0.1:8080
            - --leader-elect
          command:
            - /manager
          env:
            - name: ACCOP_LOG_LEVEL
              value: "2"
          image: [your_registry_path]/acc-operator:[version x.y.z]
          imagePullPolicy: IfNotPresent
          imagePullSecrets: []

```

2. 업데이트된 Astra Control Center 운영자를 설치합니다.

```
kubectl apply -f astra_control_center_operator_deploy.yaml
```

샘플 반응:

```
namespace/netapp-acc-operator unchanged
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.astra.
netapp.io configured
role.rbac.authorization.k8s.io/acc-operator-leader-election-role
unchanged
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role
configured
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
unchanged
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role unchanged
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding unchanged
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding configured
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding unchanged
configmap/acc-operator-manager-config unchanged
service/acc-operator-controller-manager-metrics-service unchanged
deployment.apps/acc-operator-controller-manager configured
```

Astra Control Center를 업그레이드합니다

1. Astra Control Center 사용자 정의 자원(CR)을 편집하여 Astra version('epec' 내부의 astraVersion) 번호를 최신 버전으로 변경합니다.

```
kubectl edit acc -n [netapp-acc or custom namespace]
```



Astra Control Center 업그레이드를 위해서는 Astra 버전을 변경해야 합니다. 레지스트리 경로는
에서 이미지를 푸시한 레지스트리 경로와 일치해야 합니다 [이전 단계](#).

2. 포드가 종료되고 다시 사용 가능한지 확인합니다.

```
watch kubectl get pods -n [netapp-acc or custom namespace]
```

3. 모든 시스템 구성 요소가 성공적으로 업그레이드되었는지 확인합니다.

```
kubectl get pods -n [netapp-acc or custom namespace]
```

각 포드는 최근 런닝과 나이의 상태를 가져야 합니다. 시스템 포드를 구축하는 데 몇 분 정도 걸릴 수 있습니다.

샘플 반응:

NAME	READY	STATUS	RESTARTS
AGE			
acc-helm-repo-5f75c5f564-bzqmt	1/1	Running	0
11m			
activity-6b8f7cccb9-mlrn4	1/1	Running	0
9m2s			
api-token-authentication-6hznt	1/1	Running	0
8m50s			
api-token-authentication-qpfgb	1/1	Running	0
8m50s			
api-token-authentication-sqnb7	1/1	Running	0
8m50s			
asup-5578bbdd57-dxkbp	1/1	Running	0
9m3s			
authentication-56bff4f95d-mspmq	1/1	Running	0
7m31s			
bucketservice-6f7968b95d-9rrrl	1/1	Running	0
8m36s			
cert-manager-5f6cf4bc4b-82khn	1/1	Running	0
6m19s			
cert-manager-cainjector-76cf976458-sdrbc	1/1	Running	0
6m19s			
cert-manager-webhook-5b7896bfd8-2n45j	1/1	Running	0
6m19s			
cloud-extension-749d9f684c-8bdhq	1/1	Running	0
9m6s			
cloud-insights-service-7d58687d9-h5tzw	1/1	Running	2
8m56s			
composite-compute-968c79cb5-nv714	1/1	Running	0
9m11s			
composite-volume-7687569985-jg9gg	1/1	Running	0
8m33s			
credentials-5c9b75f4d6-nx9cz	1/1	Running	0
8m42s			
entitlement-6c96fd8b78-zt7f8	1/1	Running	0
8m28s			
features-5f7bfc9f68-gsjnl	1/1	Running	0
8m57s			
fluent-bit-ds-h88p7	1/1	Running	0
7m22s			
fluent-bit-ds-krhnj	1/1	Running	0
7m23s			
fluent-bit-ds-l5bjj	1/1	Running	0
7m22s			

fluent-bit-ds-lrclb		1/1	Running	0
7m23s				
fluent-bit-ds-s5t4n		1/1	Running	0
7m23s				
fluent-bit-ds-zpr6v		1/1	Running	0
7m22s				
graphql-server-5f5976f4bd-vbb4z		1/1	Running	0
7m13s				
identity-56f78b8f9f-8h9p9		1/1	Running	0
8m29s				
influxdb2-0		1/1	Running	0
11m				
krakend-6f8d995b4d-5khkl		1/1	Running	0
7m7s				
license-5b5db87c97-jmxzc		1/1	Running	0
9m				
login-ui-57b57c74b8-6xtv7		1/1	Running	0
7m10s				
loki-0		1/1	Running	0
11m				
monitoring-operator-9dbc9c76d-8znck		2/2	Running	0
7m33s				
nats-0		1/1	Running	0
11m				
nats-1		1/1	Running	0
10m				
nats-2		1/1	Running	0
10m				
nautilus-6b9d88bc86-h8kfb		1/1	Running	0
8m6s				
nautilus-6b9d88bc86-vn68r		1/1	Running	0
8m35s				
openapi-b87d77dd8-5dz9h		1/1	Running	0
9m7s				
polaris-consul-consul-5ljfb		1/1	Running	0
11m				
polaris-consul-consul-s5d5z		1/1	Running	0
11m				
polaris-consul-consul-server-0		1/1	Running	0
11m				
polaris-consul-consul-server-1		1/1	Running	0
11m				
polaris-consul-consul-server-2		1/1	Running	0
11m				
polaris-consul-consul-twmpq		1/1	Running	0
11m				

polaris-mongodb-0	2/2	Running	0
11m			
polaris-mongodb-1	2/2	Running	0
10m			
polaris-mongodb-2	2/2	Running	0
10m			
polaris-ui-84dc87847f-zrg8w	1/1	Running	0
7m12s			
polaris-vault-0	1/1	Running	0
11m			
polaris-vault-1	1/1	Running	0
11m			
polaris-vault-2	1/1	Running	0
11m			
public-metrics-657698b66f-67pgt	1/1	Running	0
8m47s			
storage-backend-metrics-6848b9fd87-w7x8r	1/1	Running	0
8m39s			
storage-provider-5ff5868cd5-r9hj7	1/1	Running	0
8m45s			
telegraf-ds-dw4hg	1/1	Running	0
7m23s			
telegraf-ds-k92gn	1/1	Running	0
7m23s			
telegraf-ds-mmxjl	1/1	Running	0
7m23s			
telegraf-ds-nhs8s	1/1	Running	0
7m23s			
telegraf-ds-rj7lw	1/1	Running	0
7m23s			
telegraf-ds-tqrkb	1/1	Running	0
7m23s			
telegraf-rs-9mwgj	1/1	Running	0
7m23s			
telemetry-service-56c49d689b-ffrzx	1/1	Running	0
8m42s			
tenancy-767c77fb9d-g9ctv	1/1	Running	0
8m52s			
traefik-5857d87f85-7pmx8	1/1	Running	0
6m49s			
traefik-5857d87f85-cpxgv	1/1	Running	0
5m34s			
traefik-5857d87f85-lvmlb	1/1	Running	0
4m33s			
traefik-5857d87f85-t2x1k	1/1	Running	0
4m33s			

traefik-5857d87f85-v9wpf	1/1	Running	0
7m3s			
trident-svc-595f84dd78-zb816	1/1	Running	0
8m54s			
vault-controller-86c94fbf4f-krttq	1/1	Running	0
9m24s			

4. Astra 상태 조건에 업그레이드가 완료되었으며 준비가 되었음을 나타내는지 확인합니다.

```
kubectl get -o yaml -n [netapp-acc or custom namespace]
astraccontrolcenters.astra.netapp.io astra
```

응답:

```
conditions:
- lastTransitionTime: "2021-10-25T18:49:26Z"
  message: Astra is deployed
  reason: Complete
  status: "True"
  type: Ready
- lastTransitionTime: "2021-10-25T18:49:26Z"
  message: Upgrading succeeded.
  reason: Complete
  status: "False"
  type: Upgrading
```

타사 서비스를 업그레이드합니다

타사 서비스 Traefik 및 Cert-manager는 이전 업그레이드 단계 중에 업그레이드되지 않습니다. 여기에 설명된 절차를 사용하여 필요에 따라 업그레이드하거나 시스템에 필요한 경우 기존 서비스 버전을 유지할 수 있습니다. 권장되는 Traefik 및 Certs-manager 업그레이드 순서는 다음과 같습니다.

1. [Traefik 및 Cert-manager를 업그레이드하기 위해 acc-helm-repo를 설정합니다](#)
2. [acc-helm-repo를 사용하여 Traefik 서비스를 업데이트합니다](#)
3. [인증서 관리자 서비스를 업데이트합니다](#)

Traefik 및 Cert-manager를 업그레이드하기 위해 acc-helm-repo를 설정합니다

1. 로컬 Docker 캐시에 로드된 'enterprise-helm-repo'를 찾습니다.

```
docker images enterprise-helm-repo
```

응답:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
enterprise-helm-repo	21.10.218	7a182d6b30f3	20 hours ago	464MB

2. 이전 단계의 태그를 사용하여 컨테이너를 시작합니다.

```
docker run -dp 8082:8080 enterprise-helm-repo:21.10.218
```

응답:

```
940436e67fa86d2c4559ac4987b96bb35588313c2c9ddc9cec195651963f08d8
```

3. 로컬 호스트 리포지토리에 Helm repo를 추가합니다.

```
helm repo add acc-helm-repo http://localhost:8082/
```

응답:

```
"acc-helm-repo" has been added to your repositories
```

4. 다음 Python 스크립트를 파일로 저장합니다(예: 'et_previous_values.py').



이 Python 스크립트는 Helm 값을 유지하기 위해 이후 업그레이드 단계에서 사용되는 두 개의 파일을 만듭니다.

```

#!/usr/bin/env python3
import json
import os

NAMESPACE = "netapp-acc"

os.system(f"helm get values traefik -n {NAMESPACE} -o json >
traefik_values.json")
os.system(f"helm get values cert-manager -n {NAMESPACE} -o json >
cert_manager_values.json")

# reformat traefik values
f = open("traefik_values.json", "r")
traefik_values = {'traefik': json.load(f)}
f.close()

with open('traefik_values.json', 'w') as output_file:
    json.dump(traefik_values, output_file)

# reformat cert-manager values
f = open("cert_manager_values.json", "r")
cm_values = {'cert-manager': json.load(f)}
f.close()

cm_values['global'] = cm_values['cert-manager']['global']
del cm_values['cert-manager']['global']

with open('cert_manager_values.json', 'w') as output_file:
    json.dump(cm_values, output_file)

print('Done')

```

5. 스크립트를 실행합니다.

```
python3.7 ./set_previous_values.py
```

acc-helm-repo를 사용하여 Traefik 서비스를 업데이트합니다



이미 이(가) 있어야 합니다 [acc-helm-repo 설정](#) 다음 절차를 완료하기 전에

1. GNU wget과 같은 안전한 파일 전송 도구를 사용하여 Traefik 번들을 다운로드합니다.

```
wget http://localhost:8082/traefik-0.2.0.tgz
```

2. 이미지 추출:

```
tar -vxzf traefik-0.2.0.tgz
```

3. Traefik CRD 적용:

```
kubectl apply -f ./traefik/charts/traefik/crds/
```

4. 업그레이드된 Traefik에 사용할 Helm 차트 버전을 찾습니다.

```
helm search repo acc-helm-repo/traefik
```

응답:

NAME	CHART VERSION	APP VERSION	
DESCRIPTION			
acc-helm-repo/traefik	0.2.0	2.5.3	Helm
chart for Traefik Ingress controller			
acc-helm-repo/traefik-ingressroutes	0.2.0	2.5.3	A Helm
chart for Kubernetes			

5. 업그레이드용 traefik_values.json 파일 검증:

- traefik_values.json 파일을 엽니다.
- imagePullSecret 필드에 값이 있는지 확인합니다. 비어 있는 경우 파일에서 다음 텍스트를 제거합니다.

```
"imagePullSecrets": [ {"name": ""} ],
```

- traefik 영상이 올바른 위치로 이동되고 올바른 이름이 지정되었는지 확인합니다.

```
image: [your_registry_path]/traefik
```

6. Traefik 구성 업그레이드:

```
helm upgrade --version 0.2.0 --namespace netapp-acc -f  
traefik_values.json traefik acc-helm-repo/traefik
```

응답:

```
Release "traefik" has been upgraded. Happy Helming!
NAME: traefik
LAST DEPLOYED: Mon Oct 25 22:53:19 2021
NAMESPACE: netapp-acc
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

인증서 관리자 서비스를 업데이트합니다



을(를) 이미 완료해야 합니다 [Traefik 업데이트](#) 및 [Helm에 acc-helm-repo 추가](#) 다음 절차를 완료하기 전에

1. 업그레이드된 인증서 관리자에 사용할 Helm Chart 버전을 찾습니다.

```
helm search repo acc-helm-repo/cert-manager
```

응답:

```
NAME CHART VERSION APP VERSION DESCRIPTION
acc-helm-repo/cert-manager 0.3.0 v1.5.4 A Helm chart for cert-manager
acc-helm-repo/cert-manager-certificates 0.1.0 1.16.0 A Helm chart for
Kubernetes
```

2. 업그레이드할 cert_manager_values.json 파일을 확인합니다.

- a. cert_manager_values.json 파일을 엽니다.
- b. imagePullSecret 필드에 값이 있는지 확인합니다. 비어 있는 경우 파일에서 다음 텍스트를 제거합니다.

```
"imagePullSecrets": [ {"name": ""} ],
```

- c. 인증서 관리자 이미지 3개가 올바른 위치로 리디렉션되고 올바른 이름이 있는지 확인합니다.

3. Cert-Manager 구성 업그레이드:

```
helm upgrade --version 0.3.0 --namespace netapp-acc -f
cert_manager_values.json cert-manager acc-helm-repo/cert-manager
```

응답:

```
Release "cert-manager" has been upgraded. Happy Helming!
NAME: cert-manager
LAST DEPLOYED: Tue Nov 23 11:20:05 2021
NAMESPACE: netapp-acc
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

시스템 상태를 확인합니다

1. Astra Control Center에 로그인합니다.
2. 모든 관리되는 클러스터와 앱이 여전히 존재하고 보호되고 있는지 확인합니다.

Astra Control Center를 제거합니다

평가판을 정식 버전으로 업그레이드하는 경우 Astra Control Center 구성 요소를 제거해야 할 수 있습니다. Astra Control Center 및 Astra Control Center 운영자를 제거하려면 이 절차에 설명된 명령을 순서대로 실행하십시오.

필요한 것

- Astra Control Center UI를 사용하여 모두 관리합니다 ["클러스터"](#).

단계

1. Astra Control Center를 삭제합니다. 다음 샘플 명령은 기본 설치를 기반으로 합니다. 사용자 정의 설정을 만든 경우 명령을 수정합니다.

```
kubectl delete -f astra_control_center_min.yaml -n netapp-acc
```

결과:

```
astracontrolcenter.astra.netapp.io "astra" deleted
```

2. 다음 명령을 사용하여 'NetApp-acc' 네임스페이스를 삭제합니다.

```
kubectl delete ns netapp-acc
```

결과:

```
namespace "netapp-acc" deleted
```

3. 다음 명령을 사용하여 Astra Control Center 운영자 시스템 구성 요소를 삭제합니다.

```
kubectl delete -f astra_control_center_operator_deploy.yaml
```

결과:

```
namespace "netapp-acc-operator" deleted
customresourcedefinition.apiextensions.k8s.io
"astracontrolcenters.astra.netapp.io" deleted
role.rbac.authorization.k8s.io "acc-operator-leader-election-role"
deleted
clusterrole.rbac.authorization.k8s.io "acc-operator-manager-role"
deleted
clusterrole.rbac.authorization.k8s.io "acc-operator-metrics-reader"
deleted
clusterrole.rbac.authorization.k8s.io "acc-operator-proxy-role" deleted
rolebinding.rbac.authorization.k8s.io "acc-operator-leader-election-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "acc-operator-manager-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "acc-operator-proxy-
rolebinding" deleted
configmap "acc-operator-manager-config" deleted
service "acc-operator-controller-manager-metrics-service" deleted
deployment.apps "acc-operator-controller-manager" deleted
```

자세한 내용을 확인하십시오

- ["제거 관련 알려진 문제입니다"](#)

저작권 정보

Copyright © 2023 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그레픽, 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 있으며 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/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.