Astra Control Center를 사용합니다

Astra Control Center

NetApp
November 27, 2023

목차

Astra Control Center를 사용합니다

앱 관리를 시작합니다

먼저 해 "Astra Control 관리에 클러스터를 추가합니다", 클러스터(Astra Control 외부)에 앱을 설치한 다음 Astra Control의 애플리케이션 페이지로 이동하여 앱과 리소스를 정의할 수 있습니다.

설명합니다

Astra Control에는 다음과 같은 애플리케이션 관리 요구 사항이 있습니다.

- * 라이선스 *: Astra Control Center를 사용하여 애플리케이션을 관리하려면 Astra Control Center 평가판 라이센스 또는 전체 라이센스가 필요합니다.
- * 네임스페이스 *: Astra Control을 사용하여 단일 클러스터에서 하나 이상의 지정된 네임스페이스 내에서 응용 프로그램을 정의할 수 있습니다. 앱은 동일한 클러스터 내에서 여러 네임스페이스에 걸쳐 있는 리소스를 포함할 수 있습니다. Astra Control은 여러 클러스터에서 앱을 정의하는 기능을 지원하지 않습니다.
- * 스토리지 클래스 *: 스토리지 클래스가 명시적으로 설정된 애플리케이션을 설치하고 앱을 복제해야 하는 경우 클론 작업의 타겟 클러스터에 원래 지정된 스토리지 클래스가 있어야 합니다. 명시적으로 설정된 스토리지 클래스를 가진 애플리케이션을 동일한 스토리지 클래스가 없는 클러스터로 클론 복제하면 실패합니다.
- * Kubernetes 리소스 *: Astra Control에서 수집하지 않은 Kubernetes 리소스를 사용하는 애플리케이션에는 전체 앱 데이터 관리 기능이 없을 수 있습니다. Astra Control은 다음과 같은 Kubernetes 리소스를 수집합니다.

ClusterRole	ClusterRoleBinding	ConfigMap
CronJob	CustomResourceDefinition	CustomResource
DaemonSet	DeploymentConfig	HorizontalPodAutoscaler
Ingress	MutatingWebhook	NetworkPolicy
PersistentVolumeClaim	Pod	PodDisruptionBudget
PodTemplate	ReplicaSet	Role
RoleBinding	Route	Secret
Service	ServiceAccount	StatefulSet
ValidatingWebhook		

지원되는 앱 설치 방법

Astra Control은 다음과 같은 응용 프로그램 설치 방법을 지원합니다.

- * 매니페스트 파일 *: Astra Control은 kubctl을 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.
- * 운용자 구축 앱 *: Astra Control은 네임스페이스 범위 연산자로 설치된 앱을 지원합니다. 일반적으로 "pass-by-reference" 아키텍처가 아니라 "pass-by-value"로 설계되었습니다. 운영자와 설치하는 앱은 동일한 네임스페이스를 사용해야 합니다. 운영자의 배포 YAML 파일을 수정해야 이 문제가 발생할 수 있습니다.

  다음은 이러한 패턴을 따르는 일부 운영자 앱에 대한 설명입니다.

  - "아파치 K8ssandra"

    > K8ssandra 의 경우 현재 위치 복원 작업이 지원됩니다. 새 네임스페이스 또는 클러스터에 대한 복원 작업을 수행하려면 응용 프로그램의 원래 인스턴스를 중단해야 합니다. 이는 이월된 피어 그룹 정보가 인스턴스 간 통신으로 이어지지 않도록 하기 위한 것입니다. 앱 복제는 지원되지 않습니다.

  - "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.

클러스터에 앱을 설치합니다

먼저 해 "클러스터가 추가되었습니다" Astra Control은 클러스터에서 앱을 설치하거나 기존 앱을 관리할 수 있습니다. 하나 이상의 네임스페이스로 범위가 지정된 모든 앱을 관리할 수 있습니다.

앱 정의

Astra Control이 클러스터에서 네임스페이스를 검색한 후 관리할 애플리케이션을 정의할 수 있습니다. 선택할 수 있습니다 하나 이상의 네임스페이스를 포괄하는 응용 프로그램을 관리합니다 또는 전체 네임스페이스를 단일 애플리케이션으로 관리합니다. 데이터 보호 작업에 필요한 세분화 수준으로 세분화됩니다.

Astra Control을 사용하면 계층 구조의 수준(네임스페이스 및 해당 네임스페이스 또는 스패닝 네임스페이스의 응용 프로그램)을 별도로 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.

> 예를 들어, "Maria"에 대해 주간 백업 주기를 갖는 백업 정책을 설정할 수 있지만 "MariaDB"(동일한 네임스페이스)를 더 자주 백업해야 할 수 있습니다. 이러한 요구사항에 따라 단일 네임스페이스 앱이 아니라 앱을 별도로 관리해야 합니다.

시작하기 전에

- Astra Control에 Kubernetes 클러스터가 추가되었습니다.
- 클러스터에 설치된 애플리케이션 하나 이상 지원되는 앱 설치 방법에 대해 자세히 알아보십시오.
- Astra Control에 추가한 Kubernetes 클러스터의 기존 네임스페이스
- (선택 사항) Any의 Kubernetes 레이블 "지원되는 Kubernetes 리소스".

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".

이 작업에 대해

- 시작하기 전에, 또한 이해해야 합니다 "표준 및 시스템 네임스페이스 관리".
- Astra Control에서 앱과 여러 네임스페이스를 사용하려면 "네임스페이스 제약 조건을 사용하여 사용자 역할을 수정합니다" 여러 네임스페이스 지원이 있는 Astra Control Center 버전으로 업그레이드한 후
- Astra Control API를 사용하여 앱을 관리하는 방법에 대한 지침은 를 참조하십시오 "Astra 자동화 및 API 정보".

애플리케이션 관리 옵션

- 앱으로 관리할 리소스를 정의합니다
- 앱으로 관리할 네임스페이스를 정의합니다

앱으로 관리할 리소스를 정의합니다

를 지정할 수 있습니다 "앱을 구성하는 Kubernetes 리소스" Astra Control을 통해 관리하고자 하는 것입니다. 앱을 정의하면 Kubernetes 클러스터의 요소를 단일 애플리케이션으로 그룹화할 수 있습니다. 이 Kubernetes 리소스 모음은 네임스페이스 및 레이블 선택기 기준에 따라 구성됩니다.

앱을 정의하면 클론, 스냅샷, 백업을 비롯한 Astra Control 작업에 포함할 항목을 보다 세부적으로 제어할 수 있습니다.

앱을 정의할 때 보호 정책이 있는 여러 앱에 Kubernetes 리소스를 포함하지 않아야 합니다. Kubernetes 리소스의 보호 정책이 중복되어 데이터 충돌이 발생할 수 있습니다. 예를 들어, 자세한 내용을 읽어보십시오.

앱 네임스페이스에 클러스터 범위 리소스를 추가하는 방법에 대한 자세한 내용은 을(를) 참조하십시오.

Namespace 리소스와 연결된 클러스터 리소스 및 자동으로 포함된 Astra Control을 가져올 수 있습니다. 특정 그룹, 종류, 버전 및 레이블(선택 사항)의 리소스를 포함할 규칙을 추가할 수 있습니다. Astra Control에 자동으로 포함되지 않는 리소스가 있는 경우 이 작업을 수행할 수 있습니다.

Astra Control에 의해 자동으로 포함되는 클러스터 범위 리소스는 제외할 수 없습니다.

다음을 추가할 수 있습니다 `apiVersions` (API 버전과 결합된 그룹):

자원 종류	**apiVersions**(그룹 + 버전)
`ClusterRole`	rbac.authorization.k8s.io/v1
`ClusterRoleBinding`	rbac.authorization.k8s.io/v1
`CustomResource`	apiextensions.k8s.io/v1, apiextensions.k8s.io/v1beta1
`CustomResourceDefinition`	apiextensions.k8s.io/v1, apiextensions.k8s.io/v1beta1
`MutatingWebhookConfiguration`	Admissions registration.k8s.io/v1
`ValidatingWebhookConfiguration`	Admissions registration.k8s.io/v1

단계

1. 응용 프로그램 페이지에서 * 정의 * 를 선택합니다.
2. 응용 프로그램 정의 * 창에서 응용 프로그램 이름을 입력합니다.
3. 응용 프로그램이 실행되는 클러스터를 * 클러스터 * 드롭다운 목록에서 선택합니다.
4. Namespace* 드롭다운 목록에서 응용 프로그램의 네임스페이스를 선택합니다.

   Astra Control을 사용하여 단일 클러스터에서 하나 이상의 지정된 네임스페이스 내에서 앱을 정의할 수 있습니다. 앱은 동일한 클러스터 내에서 여러 네임스페이스에 걸쳐 있는 리소스를 포함할 수 있습니다. Astra Control은 여러 클러스터에서 앱을 정의하는 기능을 지원하지 않습니다.

5. (선택 사항) 각 네임스페이스에서 Kubernetes 리소스에 대한 레이블을 입력합니다. 단일 레이블 또는 레이블 선택 조건(쿼리)을 지정할 수 있습니다.

   Kubernetes 레이블에 대해 자세히 알아보려면 "Kubernetes 공식 문서를 참조하십시오".

6. (선택 사항) * 네임스페이스 추가 * 를 선택하고 드롭다운 목록에서 네임스페이스를 선택하여 앱에 대한 네임스페이스를 추가합니다.
7. (선택 사항) 추가하는 모든 추가 네임스페이스에 대한 단일 레이블 또는 레이블 선택기 조건을 입력합니다.
8. (선택 사항) Astra Control에 자동으로 포함되는 리소스 외에 클러스터 범위 리소스를 포함하려면 * 추가 클러스터 범위 리소스 포함 * 을 선택하여 다음을 완료합니다.
   a. 포함 규칙 추가 * 를 선택합니다.
   b. * Group *: 드롭다운 목록에서 리소스의 API 그룹을 선택합니다.

c. * Kind *: 드롭다운 목록에서 개체 스키마의 이름을 선택합니다.

d. * 버전 *: API 버전을 입력합니다.

e. * 라벨 선택기 *: 규칙에 추가할 라벨을 선택적으로 포함합니다. 이 레이블은 이 레이블과 일치하는 리소스만 검색하는 데 사용됩니다. 레이블을 제공하지 않으면 Astra Control은 해당 클러스터에 대해 지정된 리소스 유형의 모든 인스턴스를 수집합니다.

f. 항목에 따라 만들어진 규칙을 검토합니다.

g. 추가 * 를 선택합니다.

클러스터 범위의 리소스 규칙을 원하는 만큼 만들 수 있습니다. 규칙은 애플리케이션 요약 정의에 나타납니다.

9. 정의 * 를 선택합니다.

10. 정의 * 를 선택한 후 필요에 따라 다른 앱에 대해 프로세스를 반복합니다.

앱 정의를 마치면 앱이 에 나타납니다 `Healthy` 응용 프로그램 페이지의 응용 프로그램 목록에서 상태를 지정합니다. 이제 클론을 생성하고 백업과 스냅샷을 생성할 수 있습니다.

방금 추가한 앱에는 Protected(보호) 열 아래에 백업이 없고 아직 백업이 예약되지 않았음을 나타내는 경고 아이콘이 있을 수 있습니다.

특정 앱의 세부 정보를 보려면 앱 이름을 선택합니다.

이 앱에 추가된 리소스를 보려면 * 리소스 * 탭을 선택하십시오. 리소스 열에서 리소스 이름 뒤의 숫자를 선택하거나 검색에 리소스 이름을 입력하여 추가 클러스터 범위 리소스가 포함되도록 합니다.

앱으로 관리할 네임스페이스를 정의합니다

네임스페이스의 리소스를 애플리케이션으로 정의하여 Astra Control 관리에 네임스페이스의 모든 Kubernetes 리소스를 추가할 수 있습니다. 이 방법은 특정 네임스페이스의 모든 리소스를 비슷한 방식으로 일정한 간격으로 관리하고 보호하려는 경우 앱을 개별적으로 정의하는 것이 좋습니다.

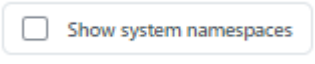
단계

1. 클러스터 페이지에서 클러스터를 선택합니다.

2. Namespaces* 탭을 선택합니다.

3. 관리하려는 앱 리소스가 포함된 네임스페이스의 작업 메뉴를 선택하고 * 응용 프로그램으로 정의 * 를 선택합니다.

여러 응용 프로그램을 정의하려면 네임스페이스 목록에서 선택하고 왼쪽 위 모서리에 있는 * 작업 * 버튼을 선택한 다음 * 응용 프로그램으로 정의 * 를 선택합니다. 이렇게 하면 개별 네임스페이스에 여러 개의 개별 응용 프로그램이 정의됩니다. 다중 네임스페이스 응용 프로그램의 경우 를 참조하십시오 앱으로 관리할 리소스를 정의합니다.

기본적으로 앱 관리에 사용되지 않는 시스템 네임스페이스를 표시하려면 * Show system namespaces * 확인란을 선택합니다. 

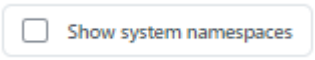
"자세히 보기".

프로세스가 완료되면 네임스페이스와 연결된 응용 프로그램이 에 나타납니다 `Associated applications` 열.

시스템 네임스페이스는 어떻습니까?

Astra Control은 Kubernetes 클러스터에서 시스템 네임스페이스를 검색합니다. 기본적으로 이러한 시스템 네임스페이스는 표시되지 않습니다. 시스템 앱 리소스를 백업해야 하는 경우는 드뭅니다.

선택한 클러스터의 Namespaces 탭에서 * Show system namespaces * 확인란을 선택하여 시스템 네임스페이스를 표시할 수 있습니다.



Astra Control Center는 기본적으로 관리할 수 있는 애플리케이션으로 표시되지 않지만 다른 Astra Control Center 인스턴스를 사용하여 Astra Control Center 인스턴스를 백업 및 복원할 수 있습니다.

예: 다른 릴리즈에 대한 별도의 보호 정책

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

DevOps 팀의 Kubernetes 관리자가 레이블을 추가합니다 `deployment=stable` 안정적인 분리 포드로. 팀에서 라벨을 추가합니다 `deployment=canary` 캔리 분리 포드로.

이 팀의 안정적인 릴리즈에는 시간별 스냅샷 및 일일 백업에 대한 요구 사항이 포함됩니다. 카나리아 릴리스는 수명이 짧기 때문에 레이블이 지정된 모든 것에 대해 공격적이고 단기적인 보호 정책을 만들고자 합니다 `deployment=canary`.

데이터 충돌을 방지하기 위해 관리자는 "Canary" 릴리스용 앱과 "Stable" 릴리즈용 앱을 두 개 만듭니다. 이렇게 하면 두 Kubernetes 객체 그룹에 대해 백업, 스냅샷 및 클론 작업이 분리됩니다.

자세한 내용을 확인하십시오

- "Astra Control API를 사용합니다"
- "앱 관리를 취소합니다"

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".

또한 재해 복구에 대비하여 애플리케이션을 원격 클러스터로 복제할 수 있습니다.

애플리케이션 보호 워크플로우

다음 예제 워크플로를 사용하여 앱 보호를 시작할 수 있습니다.

[1개] 모든 앱을 보호합니다

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] 보호 정책을 조정합니다

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

[네] 앱을 원격 클러스터로 복제합니다

"애플리케이션 복제" NetApp SnapMirror 기술을 사용하여 원격 클러스터로 Astra Control은 스냅샷을 원격 클러스터에 복제하여 비동기식 재해 복구 기능을 제공합니다.

[다섯] 재해가 발생할 경우 최신 백업 또는 복제를 사용하여 원격 시스템으로 앱을 복구합니다

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). 또는 원격 시스템에 복제를 사용할 수 있습니다.

스냅샷 및 백업으로 애플리케이션 보호

자동화된 보호 정책을 사용하거나 필요에 따라 스냅샷 및 백업을 수행하여 모든 애플리케이션을 보호합니다. Astra Control Center UI 또는 를 사용할 수 있습니다 "Astra Control API" 앱을 보호합니다.

이 작업에 대해

- * 앱 배포 *: 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 클러스터에만 해당) 정책 추가 *: 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:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

앱 데이터 보호와 관련된 다음 작업을 수행할 수 있습니다.

- 보호 정책을 구성합니다

- 스냅샷을 생성합니다
- 백업을 생성합니다
- 스냅샷 및 백업을 봅니다
- 스냅샷을 삭제합니다
- 백업을 취소합니다
- 백업을 삭제합니다

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.

시간당 한 번 이상 백업 또는 스냅샷을 자주 실행해야 하는 경우 를 수행할 수 있습니다 "Astra Control REST API를 사용하여 스냅샷과 백업을 생성합니다".

백업 및 복제 일정을 오프셋하여 일정이 겹치지 않도록 합니다. 예를 들어, 매시간 맨 위에서 백업을 수행하고 5분 오프셋 및 10분 간격으로 복제를 시작하도록 예약합니다.

앱이 에서 지원하는 저장소 클래스를 사용하는 경우 `ontap-nas-economy` 드라이버, 보호 정책을 사용할 수 없습니다. 백업 및 스냅샷을 예약하려면 Astra Control에서 지원하는 스토리지 클래스로 마이그레이션합니다.

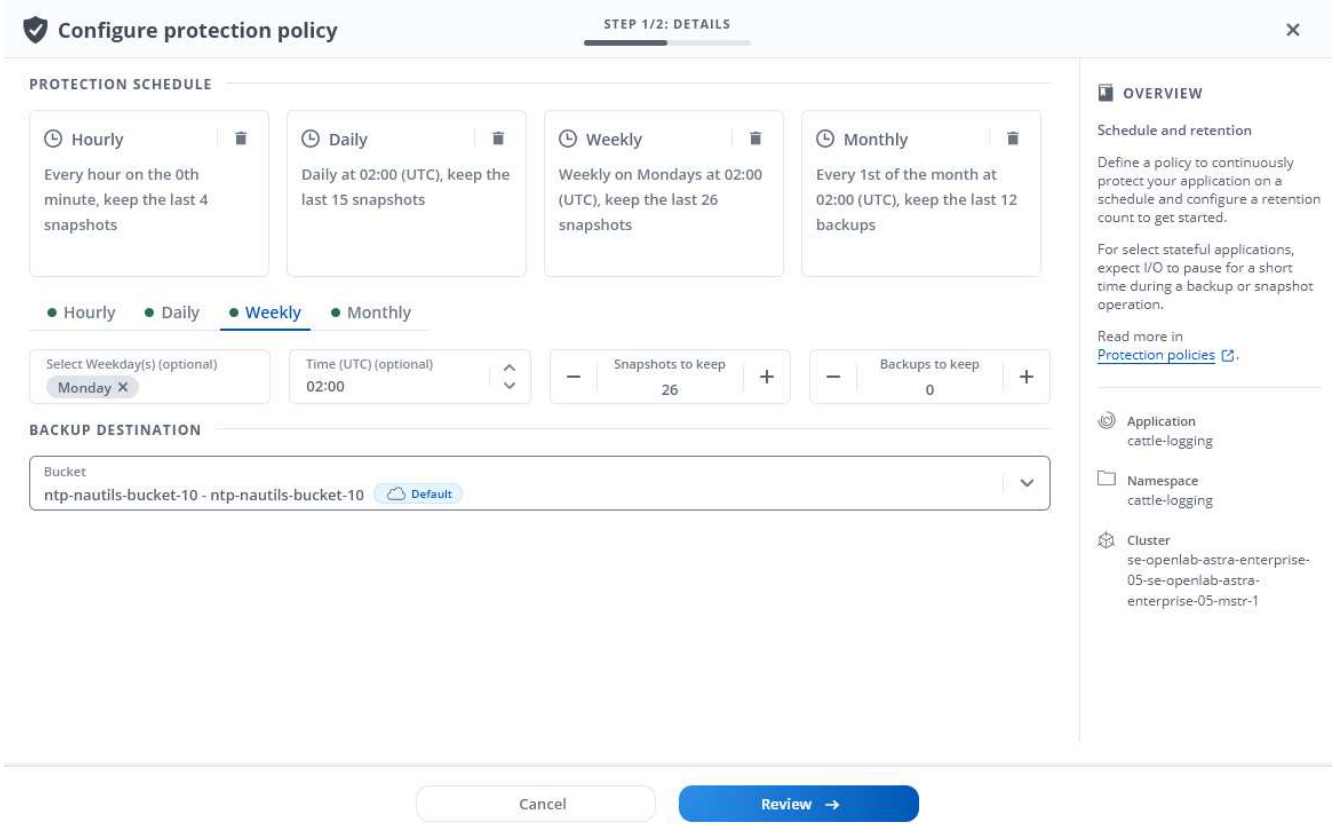
단계

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.

   백업의 보존 레벨을 설정할 때 백업을 저장할 버킷을 선택할 수 있습니다.

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



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

결과

Astra Control은 정의한 스케줄 및 보존 정책을 사용하여 스냅샷 및 백업을 생성하고 유지함으로써 데이터 보호 정책을 구현합니다.

스냅샷을 생성합니다

언제든지 주문형 스냅샷을 생성할 수 있습니다.

> 앱이 에서 지원하는 저장소 클래스를 사용하는 경우 `ontap-nas-economy` 드라이버, 스냅샷을 생성할 수 없습니다. 스냅샷에 대체 스토리지 클래스를 사용합니다.

단계

1. 응용 프로그램 * 을 선택합니다.
2. 원하는 앱의 * Actions * 열에 있는 옵션 메뉴에서 * Snapshot * 을 선택합니다.
3. 스냅샷 이름을 사용자 지정하고 * 다음 * 을 선택합니다.
4. 스냅샷 요약을 검토하고 * Snapshot * 을 선택합니다.

결과

스냅샷 프로세스가 시작됩니다. 데이터 보호 * > * 스냅샷 * 페이지의 * 상태 * 열에서 상태가 * 정상 * 인 경우 스냅샷이 성공합니다.

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.

> 앱이 에서 지원하는 저장소 클래스를 사용하는 경우 `ontap-nas-economy` 드라이버, 을(를) 정의했는지 확인하십시오 `backendType` 매개 변수 을 선택합니다 "Kubernetes 스토리지 오브젝트입니다" 을 값으로 사용합니다 `ontap-nas-economy` 보호 작업을 수행하기 전에 에서 지원하는 앱의 백업 `ontap-nas-economy` 백업 작업이 완료될 때까지 애플리케이션을 사용할 수 없게 됩니다.

단계

1. 응용 프로그램 * 을 선택합니다.
2. 원하는 앱의 * Actions * 열에 있는 옵션 메뉴에서 * Back Up * 을 선택합니다.
3. 백업 이름을 사용자 지정합니다.
4. 기존 스냅샷에서 앱을 백업할지 여부를 선택합니다. 이 옵션을 선택하면 기존 스냅샷 목록에서 선택할 수 있습니다.
5. 스토리지 버킷 목록에서 백업할 대상 버킷을 선택합니다.
6. 다음 * 을 선택합니다.
7. 백업 요약을 검토하고 * 백업 * 을 선택합니다.

결과

Astra Control은 앱 백업을 생성합니다.

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

> 실행 중인 백업을 취소해야 하는 경우 의 지침을 따릅니다 백업을 취소합니다. 백업을 삭제하려면 백업이 완료될 때까지 기다린 다음 의 지침을 따르십시오 백업을 삭제합니다.

> 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.

> 현재 복제 중인 스냅샷은 삭제할 수 없습니다.

단계

1. 응용 프로그램 * 을 선택한 다음 관리되는 응용 프로그램의 이름을 선택합니다.
2. 데이터 보호 * 를 선택합니다.
3. 원하는 스냅샷에 대한 * Actions * 열의 Options 메뉴에서 * Delete snapshot * 을 선택합니다.
4. 삭제를 확인하려면 "delete"라는 단어를 입력하고 * Yes, Delete snapshot * 을 선택합니다.

결과

Astra Control이 스냅샷을 삭제합니다.

백업을 취소합니다

진행 중인 백업을 취소할 수 있습니다.

> 백업을 취소하려면 백업이 에 있어야 합니다 `Running` 상태. 에 있는 백업은 취소할 수 없습니다 `Pending` 상태.

단계

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 데이터 보호 * 를 선택합니다.
3. Backups * 를 선택합니다.
4. 원하는 백업에 대한 * Actions * 열의 Options 메뉴에서 * Cancel * 을 선택합니다.
5. 작업을 확인하려면 "취소"라는 단어를 입력하고 * 예, 백업 취소 * 를 선택합니다.

백업을 삭제합니다

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

> 실행 중인 백업을 취소해야 하는 경우 의 지침을 따릅니다 백업을 취소합니다. 백업을 삭제하려면 백업이 완료될 때까지 기다린 다음 이 지침을 따르십시오.

단계

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 데이터 보호 * 를 선택합니다.
3. Backups * 를 선택합니다.
4. 원하는 백업에 대한 * Actions * 열의 Options 메뉴에서 * Delete backup * 을 선택합니다.

5. 삭제를 확인하려면 "delete"라는 단어를 입력하고 * Yes, Delete backup * 을 선택합니다.

결과

Astra Control이 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 API를 참조하십시오" 앱을 복원합니다.

이 작업에 대해

- * 앱을 먼저 보호 *: 복원하기 전에 응용 프로그램의 스냅샷 또는 백업을 수행하는 것이 좋습니다. 이렇게 하면 복구에 실패한 경우 스냅샷 또는 백업에서 클론을 생성할 수 있습니다.
- * 대상 볼륨 확인 *: 다른 스토리지 클래스로 복원하는 경우 스토리지 클래스가 동일한 영구 볼륨 액세스 모드(예: ReadWriteMany)를 사용하는지 확인합니다. 대상 영구 볼륨 액세스 모드가 다르면 복원 작업이 실패합니다. 예를 들어, 소스 영구 볼륨에서 rwx 액세스 모드를 사용하는 경우 Azure Managed Disks, AWS EBS, Google Persistent Disk 또는 와 같이 rwx를 제공할 수 없는 대상 스토리지 클래스를 선택합니다 `ontap-san`, 복구 작업이 실패합니다. 영구 볼륨 액세스 모드에 대한 자세한 내용은 를 참조하십시오 "쿠버네티스" 문서화:
- * 공간 요구사항 계획 *: NetApp ONTAP 스토리지를 사용하는 애플리케이션의 데이터 이동 없이 복원을 수행할 경우 복원된 앱에서 사용하는 공간이 두 배로 증가할 수 있습니다. 데이터 이동 없이 복구를 수행한 후 복구된 애플리케이션에서 원치 않는 스냅샷을 모두 제거하여 스토리지 공간을 확보합니다.
- * (OpenShift 클러스터에만 해당) 정책 추가 *: 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:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

- * H제어 응용 프로그램 배포 *: Helm 3으로 배포된 응용 프로그램(또는 Helm 2에서 Helm 3으로 업그레이드)이 완벽하게 지원됩니다. Helm 2와 함께 배포된 앱은 지원되지 않습니다.

다른 앱과 리소스를 공유하는 앱에서 데이터 이동 없이 복원 작업을 수행하면 의도하지 않은 결과가 발생할 수 있습니다. 앱 간에 공유되는 모든 리소스는 앱 중 하나에서 데이터 이동 없이 복원이 수행될 때 교체됩니다. 자세한 내용은 을 참조하십시오 이 예는 다음과 같습니다.

단계

1. 응용 프로그램 * 을 선택한 다음 앱 이름을 선택합니다.
2. 작업 열의 옵션 메뉴에서 * 복원 * 을 선택합니다.
3. 복원 유형 선택:
   - * 원래 네임스페이스로 복원 *: 이 절차를 사용하여 원래 클러스터로 응용 프로그램을 원래 상태로 복원할 수 있습니다.

앱이 에서 지원하는 저장소 클래스를 사용하는 경우 `ontap-nas-economy` 드라이버, 원래 저장소 클래스를 사용하여 앱을 복원해야 합니다. 앱을 동일한 네임스페이스로 복원하는 경우 다른 스토리지 클래스를 지정할 수 없습니다.

i. 앱을 원래 상태로 복원하는 데 사용할 스냅샷 또는 백업을 선택합니다. 그러면 앱이 이전 버전으로 되돌아갑니다.

ii. 다음 * 을 선택합니다.

이전에 삭제된 네임스페이스에 복원하는 경우 복원 프로세스의 일부로 동일한 이름의 새 네임스페이스가 만들어집니다. 이전에 삭제된 네임스페이스에서 앱을 관리할 권한이 있는 사용자는 새로 다시 생성된 네임스페이스에 대한 권한을 수동으로 복원해야 합니다.

◦ * 새 네임스페이스로 복원 *: 이 절차를 사용하여 응용 프로그램을 다른 클러스터나 소스의 다른 네임스페이스로 복원할 수 있습니다.

이 절차를 사용하여 다음 작업을 수행할 수 있습니다 에서 지원하는 스토리지 클래스로 `ontap-nas` 에서 지원하는 저장소 클래스가 있는 다른 클러스터에 앱을 * 또는 * 같은 클러스터에서 복사합니다 `ontap-nas-economy` 드라이버.

i. 복원된 앱의 이름을 지정합니다.

ii. 복원하려는 앱의 대상 클러스터를 선택합니다.

iii. 앱과 연결된 각 소스 네임스페이스의 대상 네임스페이스를 입력합니다.

Astra Control은 이 복원 옵션의 일부로 새 대상 네임스페이스를 만듭니다. 지정한 대상 네임스페이스가 대상 클러스터에 이미 있으면 안 됩니다.

iv. 다음 * 을 선택합니다.

v. 앱을 복원하는 데 사용할 스냅샷 또는 백업을 선택합니다.

vi. 다음 * 을 선택합니다.

vii. 다음 중 하나를 선택합니다.

- * 원래 스토리지 클래스를 사용하여 복원 *: 대상 클러스터에 없는 경우 응용 프로그램은 원래 연결된 스토리지 클래스를 사용합니다. 이 경우 클러스터의 기본 스토리지 클래스가 사용됩니다.
- * 다른 스토리지 클래스를 사용하여 복구 *: 타겟 클러스터에 존재하는 스토리지 클래스를 선택합니다. 원래 연결된 스토리지 클래스에 관계없이 모든 애플리케이션 볼륨은 복구의 일부로 이 서로 다른 스토리지 클래스로 마이그레이션됩니다.

viii. 다음 * 을 선택합니다.

4. 필터링할 자원 선택:

◦ * 모든 리소스 복원 *: 원래 앱과 연결된 모든 리소스를 복원합니다.

◦ * 필터 리소스 *: 원래 응용 프로그램 리소스의 하위 집합을 복원하는 규칙을 지정합니다.

i. 복원된 응용 프로그램에서 리소스를 포함하거나 제외하도록 선택합니다.

ii. 포함 규칙 추가 * 또는 * 제외 규칙 추가 * 를 선택하고 응용 프로그램 복원 중에 올바른 리소스를 필터링하도록 규칙을 구성합니다. 규칙을 편집하거나 제거하고 구성이 올바른지 때까지 규칙을 다시 만들 수 있습니다.

포함 및 제외 규칙 구성에 대한 자세한 내용은 을 참조하십시오 응용 프로그램 복원 중에 리소스를 필터링합니다.

5. 다음 * 을 선택합니다.
6. 복원 작업에 대한 세부 정보를 주의 깊게 검토하고 "restore"를 입력하고(메시지가 나타나면) * Restore * 를 선택합니다.

결과

Astra Control은 사용자가 제공한 정보를 기반으로 앱을 복원합니다. 앱을 제자리에 복원한 경우 기존 영구 볼륨의 콘텐츠가 복원된 앱의 영구 볼륨 콘텐츠로 바뀝니다.

데이터 보호 작업(클론, 백업 또는 복원)과 후속 영구 볼륨 크기 조정 후 웹 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.

네임스페이스 이름/ID 또는 네임스페이스 레이블에 의해 네임스페이스 제한이 있는 구성원 사용자는 동일한 클러스터 또는 조직 계정의 다른 클러스터에 있는 새 네임스페이스에 앱을 클론 복제하거나 복원할 수 있습니다. 그러나 동일한 사용자가 새 네임스페이스에서 복제되거나 복원된 앱에 액세스할 수 없습니다. 클론 또는 복원 작업을 통해 새 네임스페이스를 생성한 후 계정 관리자/소유자는 구성원 사용자 계정을 편집하고 영향을 받는 사용자의 역할 제약 조건을 업데이트하여 새 네임스페이스에 대한 액세스 권한을 부여할 수 있습니다.

응용 프로그램 복원 중에 리소스를 필터링합니다

에 필터 규칙을 추가할 수 있습니다 "복원" 복원된 응용 프로그램에서 포함하거나 제외할 기존 응용 프로그램 리소스를 지정하는 작업입니다. 지정된 네임스페이스, 레이블 또는 GVK(GroupVersionKind)를 기반으로 리소스를 포함하거나 제외할 수 있습니다.

포함 및 제외 시나리오에 대한 자세한 내용은 를 확장합니다

- * 원본 네임스페이스가 있는 포함 규칙(원본 위치 복원) * 을 선택합니다. 규칙에 정의된 기존 응용 프로그램 리소스는 삭제되며 복구에 사용하는 선택한 스냅숏 또는 백업의 리소스로 대체됩니다. 포함 규칙에 지정하지 않은 모든 리소스는 변경되지 않습니다.
- * 새 네임스페이스가 있는 포함 규칙 선택 *: 이 규칙을 사용하여 복원된 응용 프로그램에서 원하는 특정 리소스를 선택합니다. 포함 규칙에 지정하지 않은 리소스는 복원된 응용 프로그램에 포함되지 않습니다.
- * 원본 네임스페이스가 있는 제외 규칙(원본 위치 복원) * 선택: 제외하도록 지정한 리소스는 복원되지 않고 변경되지 않습니다. 제외하도록 지정하지 않은 리소스는 스냅샷 또는 백업에서 복구됩니다. 해당 StatefulSet 이 필터링된 리소스의 일부인 경우 영구 볼륨의 모든 데이터가 삭제되고 다시 생성됩니다.
- * 새 네임스페이스가 있는 제외 규칙을 선택합니다. * : 규칙을 사용하여 복원된 응용 프로그램에서 제거할 특정 리소스를 선택합니다. 제외하도록 지정하지 않은 리소스는 스냅샷 또는 백업에서 복구됩니다.

규칙은 포함 또는 제외 유형입니다. 자원 포함과 제외 를 결합하는 규칙은 사용할 수 없습니다.

단계

1. 리소스를 필터링하도록 선택하고 앱 복원 마법사에서 포함 또는 제외 옵션을 선택한 후 * 포함 규칙 추가 * 또는 * 제외 규칙 추가 * 를 선택합니다.

Astra Control에 의해 자동으로 포함되는 클러스터 범위 리소스는 제외할 수 없습니다.

2. 필터 규칙 구성:

적어도 하나의 네임스페이스, 레이블 또는 GVK를 지정해야 합니다. 필터 규칙을 적용한 후 유지하는 리소스가 복원된 응용 프로그램을 양호한 상태로 유지하는 데 충분한지 확인합니다.

a. 규칙의 특정 네임스페이스를 선택합니다. 선택하지 않으면 모든 네임스페이스가 필터에 사용됩니다.

응용 프로그램에 원래 여러 네임스페이스가 포함되어 있고 이를 새 네임스페이스로 복원하면 리소스에 포함되지 않은 네임스페이스도 모두 만들어집니다.

b. (선택 사항) 리소스 이름을 입력합니다.

c. (선택 사항) * 라벨 선택기 *: 포함 "라벨 선택기" 규칙에 추가합니다. 레이블 선택기는 선택한 레이블과 일치하는 자원만 필터링하는 데 사용됩니다.

d. (선택 사항) 추가 필터링 옵션을 사용하려면 GVK(GroupVersionKind) SET * 를 선택하여 리소스 * 를 필터링합니다.

GVK 필터를 사용하는 경우 버전 및 종류를 지정해야 합니다.

i. (선택 사항) * Group *: 드롭다운 목록에서 Kubernetes API 그룹을 선택합니다.

ii. * Kind *: 드롭다운 목록에서 필터에 사용할 Kubernetes 리소스 유형에 대한 오브젝트 스키마를 선택합니다.

iii. * 버전 *: Kubernetes API 버전을 선택합니다.

3. 항목에 따라 만들어진 규칙을 검토합니다.

4. 추가 * 를 선택합니다.

원하는 만큼 리소스 포함 및 제외 규칙을 만들 수 있습니다. 작업을 시작하기 전에 복원 애플리케이션 요약에 규칙이 나타납니다.

ONTAP-NAS-이코노미 스토리지에서 **ONTAP-NAS** 스토리지로 마이그레이션

Astra Control을 사용할 수 있습니다 "애플리케이션 복원" 또는 "애플리케이션 클론" 에서 지원하는 스토리지 클래스에서 애플리케이션 볼륨을 마이그레이션하는 작업입니다 `ontap-nas-economy`에서는 제한된 애플리케이션 보호 옵션을 에서 지원하는 스토리지 클래스에 허용합니다 `ontap-nas` Astra Control 보호 옵션을 모두 갖추고 있습니다. 클론 또는 복원 작업은 를 사용하는 Qtree 기반 볼륨을 마이그레이션합니다 `ontap-nas-economy` 에서 지원하는 표준 볼륨에 백엔드를 제공합니다 `ontap-nas`. 볼륨에 대한 모든 정보가 포함됩니다 `ontap-nas-economy` 백업만 또는 혼합으로 타겟 스토리지 클래스로 마이그레이션됩니다. 마이그레이션이 완료된 후에는 보호 옵션이 더 이상 제한되지 않습니다.

다른 앱과 리소스를 공유하는 앱의 데이터 이동 없이 복원 복잡성

다른 앱과 리소스를 공유하고 의도하지 않은 결과를 생성하는 앱에서 현재 위치 복원 작업을 수행할 수 있습니다. 앱 간에 공유되는 모든 리소스는 앱 중 하나에서 데이터 이동 없이 복원이 수행될 때 교체됩니다.

다음은 복원에 NetApp SnapMirror 복제를 사용할 때 바람직하지 않은 상황을 만드는 예제 시나리오입니다.

1. 애플리케이션을 정의합니다 `app1` 네임스페이스 사용 `ns1`.
2. 에 대한 복제 관계를 구성합니다 `app1`.
3. 애플리케이션을 정의합니다 `app2` 네임스페이스 사용 `ns1` 및 `ns2`.
4. 에 대한 복제 관계를 구성합니다 `app2`.
5. 에 대한 역방향 복제를 수행합니다 `app2`. 이렇게 하면 가 발생합니다 `app1` 비활성화할 소스 클러스터의 앱.

SnapMirror 기술을 사용하여 스토리지 백엔드 간에 앱을 복제합니다

Astra Control을 사용하면 NetApp SnapMirror 기술의 비동기식 복제 기능을 사용하여 낮은 RPO(복구 시점 목표) 및 낮은 RTO(복구 시간 목표)로 애플리케이션에 대한 비즈니스 연속성을 구축할 수 있습니다. 이 기능을 구성하면 애플리케이션에서 한 스토리지 백엔드에서 다른 스토리지 백엔드, 동일한 클러스터 또는 서로 다른 클러스터 간에 데이터 및 애플리케이션 변경 사항을 복제할 수 있습니다.

백업/복구와 복제를 비교하려면 을 참조하십시오 "데이터 보호 개념".

다음과 같은 사내 전용, 하이브리드 및 멀티 클라우드 시나리오와 같은 다양한 시나리오에서 앱을 복제할 수 있습니다.

- 사내 사이트 A에서 사내 사이트 A로
- 사내 사이트 A에서 사내 사이트 B로
- Cloud Volumes ONTAP를 사용하여 사내에서 클라우드로 전환
- Cloud Volumes ONTAP를 사용하는 클라우드를 사내에서 운영
- Cloud Volumes ONTAP를 사용하는 클라우드(동일한 클라우드 공급자 내의 서로 다른 지역 또는 다른 클라우드 공급자 간)

Astra Control은 사내 클러스터, 사내 클러스터, 클라우드(Cloud Volumes ONTAP 사용) 또는 클라우드 간(Cloud Volumes ONTAP에서 Cloud Volumes ONTAP로) 애플리케이션을 복제할 수 있습니다.

> 다른 앱을 반대 방향으로 동시에 복제할 수 있습니다. 예를 들어, 애플리케이션 A, B, C를 데이터 센터 1에서 데이터 센터 2로 복제하고 애플리케이션 X, Y, Z를 데이터 센터 2에서 데이터 센터 1로 복제할 수 있습니다.

Astra Control을 사용하면 애플리케이션 복제와 관련된 다음 작업을 수행할 수 있습니다.

- 복제 관계를 설정합니다
- 대상 클러스터에서 복제된 앱을 온라인 상태로 전환(페일오버)
- 페일오버된 복제 다시 동기화
- 애플리케이션 복제를 역으로 수행합니다
- 애플리케이션을 원래 소스 클러스터로 페일백합니다
- 애플리케이션 복제 관계를 삭제합니다

복제 사전 요구 사항

Astra Control 애플리케이션 복제를 시작하려면 먼저 다음과 같은 사전 요구 사항을 충족해야 합니다.

- * ONTAP 클러스터 *:
  - * Astra Trident *: Astra Trident 버전 22.10 이상이 ONTAP를 백엔드로 사용하는 소스 및 대상 Kubernetes 클러스터 모두에 있어야 합니다.
  - * 라이센스 *: 소스 및 대상 ONTAP 클러스터 모두에서 데이터 보호 번들을 사용하는 ONTAP SnapMirror 비동기 라이센스를 활성화해야 합니다. 을 참조하십시오 "ONTAP의 SnapMirror 라이센스 개요" 를 참조하십시오.
- * 피어링 *:
  - * 클러스터 및 SVM *: ONTAP 스토리지 백엔드를 피어링해야 합니다. 을 참조하십시오 "클러스터 및 SVM 피어링 개요" 를 참조하십시오.

    두 ONTAP 클러스터 간의 복제 관계에 사용되는 SVM 이름이 고유한지 확인합니다.

  - * Astra Trident 및 SVM *: 대상 클러스터의 Astra Trident에서 피어링된 원격 SVM을 사용할 수 있어야 합니다.
- * Astra Control Center *:

  "Astra Control Center를 구축합니다" 원활한 재해 복구를 위한 세 번째 장애 도메인 또는 보조 사이트.

  - * 관리형 클러스터 *: Astra Control에 다음 클러스터를 추가하고 관리해야 하며, 이상적으로는 서로 다른 장애 도메인 또는 사이트에서 관리되어야 합니다.
    - 소스 Kubernetes 클러스터
    - 대상 Kubernetes 클러스터
    - 연결된 ONTAP 클러스터
  - * 사용자 계정 *: ONTAP 스토리지 백엔드를 Astra 제어 센터에 추가할 때 "admin" 역할을 사용하여 사용자 자격 증명을 적용합니다. 이 역할에는 액세스 방법이 있습니다 `http` 및 `ontapi` ONTAP 소스 클러스터와 대상 클러스터 모두에서 사용하도록 설정되었습니다. 을 참조하십시오 "ONTAP 설명서에서 사용자 계정을 관리합니다" 를 참조하십시오.
- * Astra Trident/ONTAP 구성 *: Astra Control Center를 사용하려면 소스 및 대상 클러스터 모두에 대한 복제를 지원하는 스토리지 백엔드를 하나 이상 구성해야 합니다. 소스 및 대상 클러스터가 동일한 경우 대상 애플리케이션은 최상의 복원력을 위해 소스 애플리케이션과 다른 스토리지 백엔드를 사용해야 합니다.

Astra Control 복제는 단일 스토리지 클래스를 사용하는 애플리케이션을 지원합니다. 네임스페이스에 앱을 추가하는 경우 네임스페이스에서 다른 앱과 동일한 저장소 클래스가 앱에 있는지 확인합니다. 복제된 앱에 PVC를 추가할 때 새로운 PVC의 저장 클래스가 네임스페이스의 다른 PVC와 동일한지 확인하십시오.

복제 관계를 설정합니다

복제 관계를 설정하려면 다음을 수행해야 합니다.

- Astra Control에서 앱 스냅샷을 얼마나 자주 생성할지 선택(앱의 Kubernetes 리소스 및 각 앱의 볼륨에 대한 볼륨

스냅샷 포함)

- 복제 일정 선택(Kubernetes 리소스 및 영구 볼륨 데이터 포함)
- 스냅샷을 생성할 시간을 설정합니다

단계

1. Astra Control 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 데이터 보호 * > * 복제 * 탭을 선택합니다.
3. Configure replication policy * 를 선택합니다. 또는 애플리케이션 보호 상자에서 작업 옵션을 선택하고 * 복제 정책 구성 * 을 선택합니다.
4. 다음 정보를 입력하거나 선택합니다.
   - * 대상 클러스터 *: 대상 클러스터를 입력합니다(소스 클러스터와 같을 수 있음).
   - * 대상 스토리지 클래스 *: 대상 ONTAP 클러스터에서 피어링된 SVM을 사용하는 스토리지 클래스를 선택하거나 입력합니다. 모범 사례로서, 대상 스토리지 클래스는 소스 스토리지 클래스와 다른 스토리지 백엔드를 가리켜야 합니다.
   - * 복제 유형 *: `Asynchronous` 은 현재 사용 가능한 유일한 복제 유형입니다.
   - * 대상 네임스페이스 *: 대상 클러스터에 대한 새 또는 기존 대상 네임스페이스를 입력합니다.
   - (선택 사항) * 네임스페이스 추가 * 를 선택하고 드롭다운 목록에서 네임스페이스를 선택하여 네임스페이스를 추가합니다.
   - * 복제 빈도 *: Astra Control이 스냅샷을 촬영하여 대상에 복제할 빈도를 설정합니다.
   - * Offset *: Astra Control에서 스냅샷을 생성할 시간(분)을 설정합니다. 다른 예약된 작업과 일치하지 않도록 오프셋을 사용할 수 있습니다.

     백업 및 복제 일정을 오프셋하여 일정이 겹치지 않도록 합니다. 예를 들어, 매시간 맨 위에서 백업을 수행하고 5분 오프셋 및 10분 간격으로 복제를 시작하도록 예약합니다.
5. 다음 * 을 선택하고 요약을 검토하고 * 저장 * 을 선택합니다.

   첫 번째 일정이 발생하기 전에 상태가 "APP-MIRROR"로 표시됩니다.

   Astra Control은 복제에 사용되는 애플리케이션 스냅샷을 생성합니다.
6. 응용 프로그램 스냅샷 상태를 보려면 * 응용 프로그램 * > * 스냅샷 * 탭을 선택합니다.

   스냅샷 이름은 의 형식을 사용합니다 `replication-schedule-<string>`. Astra Control은 복제에 사용된 마지막 스냅샷을 보존합니다. 복제를 성공적으로 완료한 후에는 이전의 모든 복제 스냅샷이 삭제됩니다.

결과

그러면 복제 관계가 생성됩니다.

Astra Control은 관계를 수립함으로써 다음과 같은 조치를 수행합니다.

- 대상에서 네임스페이스 생성(없는 경우)
- 소스 앱의 PVC에 해당하는 대상 네임스페이스에 PVC를 생성합니다.

- 애플리케이션 정합성이 보장되는 초기 스냅샷을 생성합니다.
- 초기 스냅샷을 사용하여 영구 볼륨의 SnapMirror 관계를 설정합니다.

데이터 보호 * 페이지에는 복제 관계 상태 및 상태가 표시됩니다.
<Health status> | <Relationship life cycle state>

예를 들면 다음과 같습니다.
정상|설정됨

이 항목의 끝에 있는 복제 상태 및 상태에 대해 자세히 알아보십시오.

대상 클러스터에서 복제된 앱을 온라인 상태로 전환(페일오버)

Astra Control을 사용하면 복제된 애플리케이션을 대상 클러스터로 페일오버할 수 있습니다. 이 절차는 복제 관계를 중지하고 대상 클러스터에서 앱을 온라인으로 전환합니다. 이 절차를 수행해도 소스 클러스터에서 앱이 중지되지 않습니다.

단계

1. Astra Control 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 데이터 보호 * > * 복제 * 탭을 선택합니다.
3. Actions 메뉴에서 * Fail Over * 를 선택합니다.
4. 페일오버 페이지에서 정보를 검토하고 * 페일오버 * 를 선택합니다.

결과

페일오버 절차로 인해 다음 작업이 수행됩니다.

- 대상 앱은 최근 복제된 스냅샷을 기반으로 시작됩니다.
- 소스 클러스터와 앱(작동 중인 경우)이 중지되지 않고 계속 실행됩니다.
- 복제 상태가 "페일오버 중"으로 변경되고, 완료되면 "페일오버 실패"로 변경됩니다.
- 소스 앱의 보호 정책은 페일오버 시 소스 앱에 있는 일정에 따라 대상 앱에 복사됩니다.
- 소스 앱에 복원 후 실행 후크가 하나 이상 활성화된 경우 해당 실행 후크가 대상 앱에 대해 실행됩니다.
- Astra Control은 소스 및 대상 클러스터와 해당 상태 모두에서 앱을 표시합니다.

페일오버된 복제 다시 동기화

재동기화 작업은 복제 관계를 다시 설정합니다. 관계의 소스를 선택하여 소스 또는 타겟 클러스터에 데이터를 유지할 수 있습니다. 이 작업은 SnapMirror 관계를 다시 설정하여 원하는 방향으로 볼륨 복제를 시작합니다.

이 프로세스는 복제를 다시 설정하기 전에 새 대상 클러스터에서 앱을 중지합니다.

재동기화 프로세스 중에 수명 주기 상태가 "설정 중"으로 표시됩니다.

단계

1. Astra Control 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 데이터 보호 * > * 복제 * 탭을 선택합니다.

3. 작업 메뉴에서 * 재동기화 * 를 선택합니다.
4. 재동기화 페이지에서 보존할 데이터가 포함된 소스 또는 대상 앱 인스턴스를 선택합니다.

대상의 데이터를 덮어쓰므로 재동기화 소스를 신중하게 선택합니다.

5. 계속하려면 * 재동기화 * 를 선택하십시오.
6. "resync"를 입력하여 확인합니다.
7. 예, 재동기화 * 를 선택하여 완료합니다.

결과

- 복제 페이지에는 복제 상태로 "설정 중"이 표시됩니다.
- Astra Control은 새 대상 클러스터에서 애플리케이션을 중지합니다.
- Astra Control은 SnapMirror 재동기화를 사용하여 선택한 방향으로 영구 볼륨 복제를 다시 설정합니다.
- 복제 페이지에는 업데이트된 관계가 표시됩니다.

애플리케이션 복제를 역으로 수행합니다

원래 소스 스토리지 백엔드로 계속 복제하면서 애플리케이션을 대상 스토리지 백엔드로 이동하기 위한 계획된 작업입니다. Astra Control은 대상 앱으로 페일오버하기 전에 소스 애플리케이션을 중지하고 데이터를 대상에 복제합니다.

이 경우 소스와 대상을 스와핑합니다.

단계

1. Astra Control 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 데이터 보호 * > * 복제 * 탭을 선택합니다.
3. Actions 메뉴에서 * Reverse replication * 을 선택합니다.
4. 역방향 복제 페이지에서 정보를 검토하고 계속하려면 * 역방향 복제 * 를 선택합니다.

결과

역방향 복제의 결과로 다음 작업이 수행됩니다.

- 원본 소스 앱의 Kubernetes 리소스에 대한 스냅샷이 생성됩니다.
- 앱의 Kubernetes 리소스를 삭제하여 원본 소스 앱의 Pod를 정상적으로 중지할 수 있습니다(PVC 및 PVS를 그대로 둡니다).
- 포드가 종료된 후 앱 볼륨의 스냅샷이 촬영되고 복제됩니다.
- SnapMirror 관계가 끊어져 타겟 볼륨이 읽기/쓰기 준비가 되었습니다.
- 앱의 Kubernetes 리소스는 원래 소스 애플리케이션이 종료된 후 복제된 볼륨 데이터를 사용하여 사전 종료 스냅샷에서 복구됩니다.
- 복제는 반대 방향으로 다시 설정됩니다.

애플리케이션을 원래 소스 클러스터로 페일백합니다

Astra Control을 사용하면 다음 작업 시퀀스를 사용하여 장애 조치 작업 후 "장애 복구"를 수행할 수 있습니다. 이 워크플로우에서 원래 복제 방향을 복구하기 위해 Astra Control은 복제 방향을 바꾸기 전에 애플리케이션 변경 사항을 원래 소스 애플리케이션으로 복제(재동기화)합니다.

이 프로세스는 대상에 대한 페일오버를 완료한 관계로부터 시작되며 다음 단계를 포함합니다.

- 페일오버된 상태로 시작합니다.
- 관계를 다시 동기화합니다.
- 복제를 역으로 수행합니다.

단계

1. Astra Control 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 데이터 보호 * > * 복제 * 탭을 선택합니다.
3. 작업 메뉴에서 * 재동기화 * 를 선택합니다.
4. 페일백 작업의 경우 페일오버된 앱을 재동기화 작업의 소스로 선택합니다(기록된 모든 데이터 유지 사후 페일오버).
5. "resync"를 입력하여 확인합니다.
6. 예, 재동기화 * 를 선택하여 완료합니다.
7. 재동기화가 완료되면 데이터 보호 > 복제 탭의 동작 메뉴에서 * 역방향 복제 * 를 선택합니다.
8. 역방향 복제 페이지에서 정보를 검토하고 * 역방향 복제 * 를 선택합니다.

결과

이렇게 하면 "재동기화" 및 "역관계" 작업의 결과가 결합되어 원래 소스 클러스터에서 애플리케이션이 온라인 상태가 되고 복제가 원래 대상 클러스터로 다시 시작됩니다.

애플리케이션 복제 관계를 삭제합니다

관계를 삭제하면 두 개의 별도 앱이 서로 관계가 없습니다.

단계

1. Astra Control 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 데이터 보호 * > * 복제 * 탭을 선택합니다.
3. 애플리케이션 보호 상자 또는 관계 다이어그램에서 * 복제 관계 삭제 * 를 선택합니다.

결과

복제 관계를 삭제하면 다음과 같은 작업이 수행됩니다.

- 관계가 설정되었지만 대상 클러스터에서 앱이 아직 온라인 상태가 되지 않은 경우(장애 발생) Astra Control은 초기화 중에 생성된 PVC를 유지하고 "비어 있는" 관리 앱을 대상 클러스터에 남겨두고 생성된 백업을 유지할 수 있도록 대상 앱을 유지합니다.
- 대상 클러스터에서 앱이 온라인 상태가 된 경우(장애 발생), Astra Control은 PVC 및 대상 앱을 유지합니다. 이제 소스 및 대상 앱이 독립 앱으로 취급됩니다. 백업 스케줄은 두 애플리케이션 모두에 유지되지만 서로 연결되지 않습니다.

복제 관계 상태 및 관계 수명 주기 상태입니다

Astra Control은 복제 관계의 관계 상태와 수명 주기의 상태를 표시합니다.

복제 관계 상태

다음 상태는 복제 관계의 상태를 나타냅니다.

- * 정상 *: 관계가 설정되었거나 설정되었으며 최근 스냅샷이 성공적으로 전송되었습니다.
- * 경고 *: 관계가 페일오버되었거나 페일오버되었습니다(따라서 소스 앱을 더 이상 보호하지 않음).
- * 심각 *
  - 관계가 설정 또는 페일오버되고 마지막 조정 시도가 실패했습니다.
  - 관계가 성립되고 새로운 PVC의 추가를 조정하기 위한 마지막 시도가 실패합니다.
  - 관계가 설정되지만(따라서 성공한 스냅샷이 복제되고 페일오버가 가능함) 가장 최근의 스냅샷이 실패했거나 복제하지 못했습니다.

복제 수명 주기 상태입니다

다음 상태는 복제 주기의 여러 단계를 반영합니다.

- * 설정 *: 새 복제 관계가 생성됩니다. Astra Control은 필요한 경우 네임스페이스를 생성하고, 대상 클러스터의 새 볼륨에 지속적인 PVC(Volume Claim)를 생성하여 SnapMirror 관계를 생성합니다. 이 상태는 복제가 재동기화 중이거나 복제 재동기화 중임을 나타낼 수도 있습니다.
- * 설정됨 *: 복제 관계가 있습니다. Astra Control은 주기적으로 PVC가 사용 가능한지 확인하고, 복제 관계를 확인하고, 정기적으로 앱 스냅샷을 생성하고, 앱에서 새로운 PVC 소스를 식별합니다. 이 경우 Astra Control은 복제에 포함할 리소스를 생성합니다.
- * 페일오버 *: Astra Control은 SnapMirror 관계를 중단시키고 마지막으로 성공적으로 복제된 앱 스냅샷에서 앱의 Kubernetes 리소스를 복원합니다.
- * 페일오버됨 *: Astra Control은 소스 클러스터에서 복제를 중지하고, 대상에서 최근(성공한) 복제 앱 스냅샷을 사용하여 Kubernetes 리소스를 복원합니다.
- * 재동기화 *: Astra Control SnapMirror 재동기화를 사용하여 재동기화 소스의 새 데이터를 재동기화 대상으로 재동기화합니다. 이 작업은 동기화 방향에 따라 대상의 일부 데이터를 덮어쓸 수 있습니다. Astra Control은 대상 네임스페이스에서 실행 중인 앱을 중지하고 Kubernetes 앱을 제거합니다. 재동기화 프로세스 중에 상태가 "설정 중"으로 표시됩니다.
- * 후진 *: 은 원래 소스 클러스터로 계속 복제하면서 애플리케이션을 대상 클러스터로 이동하기 위한 계획된 작업입니다. Astra Control은 소스 클러스터에서 애플리케이션을 중지하고, 대상 클러스터에 앱을 페일오버하기 전에 데이터를 대상에 복제합니다. 역방향 복제 중에 상태가 "설정 중"으로 표시됩니다.
- * 삭제 *:
  - 복제 관계가 설정되었지만 아직 페일오버되지 않은 경우 Astra Control은 복제 중에 생성된 PVC를 제거하고 대상 관리 앱을 삭제합니다.
  - 복제가 이미 실패한 경우 Astra Control은 PVC 및 대상 앱을 유지합니다.

애플리케이션 클론 복제 및 마이그레이션

기존 앱을 클론 복제하여 동일한 Kubernetes 클러스터 또는 다른 클러스터에 중복 앱을 생성할

수 있습니다. Astra Control은 앱을 클론할 때 애플리케이션 구성 및 영구 스토리지의 클론을 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 Control Center UI 또는 를 사용할 수 있습니다 "Astra Control API를 참조하십시오" 앱을 클론 복제 및 마이그레이션합니다.

시작하기 전에

- * 대상 볼륨 확인 *: 다른 스토리지 클래스에 클론을 생성하는 경우 스토리지 클래스가 동일한 영구 볼륨 액세스 모드(예: ReadWriteMany)를 사용하는지 확인합니다. 대상 영구 볼륨 액세스 모드가 다르면 클론 작업이 실패합니다. 예를 들어, 소스 영구 볼륨에서 rwx 액세스 모드를 사용하는 경우 Azure Managed Disks, AWS EBS, Google Persistent Disk 또는 와 같이 rwx를 제공할 수 없는 대상 스토리지 클래스를 선택합니다 `ontap-san`, 클론 작업이 실패합니다. 영구 볼륨 액세스 모드에 대한 자세한 내용은 를 참조하십시오 "쿠버네티스" 문서화:
- 앱을 다른 클러스터에 클론 복제하려면 소스 및 대상 클러스터가 포함된 클라우드 인스턴스(동일하지 않은 경우)에 기본 버킷이 있는지 확인해야 합니다. 각 클라우드 인스턴스에 대해 기본 버킷을 할당해야 합니다.
- 클론 작업 중에 IngressClass 리소스 또는 Webhook가 필요한 애플리케이션에는 대상 클러스터에 이미 정의된 리소스가 없어야 합니다.

> OpenShift 환경에서 앱을 복제하는 동안, Astra Control Center는 OpenShift가 볼륨을 마운트하고 파일 소유권을 변경할 수 있도록 허용해야 합니다. 따라서 이러한 작업을 허용하려면 ONTAP 볼륨 내보내기 정책을 구성해야 합니다. 다음 명령을 사용하여 이 작업을 수행할 수 있습니다.
>
> 1. `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -superuser sys`
> 2. `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -anon 65534`

클론 제한 사항

- * 명시적 스토리지 클래스 *: 스토리지 클래스가 명시적으로 설정된 앱을 배포하고 앱을 복제해야 하는 경우 타겟 클러스터에 원래 지정된 스토리지 클래스가 있어야 합니다. 명시적으로 설정된 스토리지 클래스를 가진 애플리케이션을 동일한 스토리지 클래스가 없는 클러스터로 클론 복제하면 실패합니다.
- * ONTAP - NAS - 경제적인 스토리지 클래스 *: 앱이 에서 지원하는 스토리지 클래스를 사용하는 경우 `ontap-nas-economy` 드라이버, 클론 작업의 백업 부분은 중단이 발생합니다. 백업이 완료될 때까지 소스 애플리케이션을 사용할 수 없습니다. 클론 작업의 복원 부분은 무중단으로 수행됩니다.
- * 클론 및 사용자 제약 조건 *: 네임스페이스 이름/ID 또는 네임스페이스 레이블에 의해 네임스페이스 제한이 있는 구성원 사용자는 동일한 클러스터의 새 네임스페이스 또는 조직 계정의 다른 클러스터에 앱을 클론 복제하거나 복원할 수 있습니다. 그러나 동일한 사용자가 새 네임스페이스에서 복제되거나 복원된 앱에 액세스할 수 없습니다. 클론 또는 복원 작업을 통해 새 네임스페이스를 생성한 후 계정 관리자/소유자는 구성원 사용자 계정을 편집하고 영향을 받는 사용자의 역할 제약 조건을 업데이트하여 새 네임스페이스에 대한 액세스 권한을 부여할 수 있습니다.
- * 클론은 기본 버킷 사용 *: 애플리케이션 백업 또는 애플리케이션 복구 중에 버킷 ID를 선택적으로 지정할 수 있습니다. 그러나 애플리케이션 클론 작업에서는 항상 정의된 기본 버킷을 사용합니다. 클론의 버킷을 변경할 수 있는 옵션은 없습니다. 어떤 버킷이 사용되는지 제어하려는 경우 이 두 가지 방법을 사용할 수 있습니다 "버킷 기본값을 변경합니다" 또는 을 수행합니다 "백업" 뒤에 가 있습니다 "복원" 별도.
- * Jenkins CI * 사용: Jenkins CI의 운영자 배포 인스턴스를 복제하는 경우 영구 데이터를 수동으로 복원해야 합니다. 이는 앱 배포 모델의 제한 사항입니다.
- * S3 버킷 포함 *: 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.

- * 특정 버전의 PostgreSQL * 사용: 동일한 클러스터 내의 앱 클론은 Bitnami PostgreSQL 11.5.0 차트와 함께 일관되게 실패합니다. 클론을 성공적으로 생성하려면 이전 또는 이후 버전의 차트를 사용하십시오.

OpenShift 고려 사항

- * 클러스터 및 OpenShift 버전 *: 클러스터 간에 앱을 복제하는 경우 소스 및 대상 클러스터는 OpenShift의 배포와 동일해야 합니다. 예를 들어 OpenShift 4.7 클러스터에서 앱을 클론하는 경우 OpenShift 4.7인 대상 클러스터를 사용합니다.
- * 프로젝트 및 UID *: OpenShift 클러스터에서 앱을 호스팅하기 위한 프로젝트를 생성하면 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:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

단계

1. 응용 프로그램 * 을 선택합니다.
2. 다음 중 하나를 수행합니다.
   - 원하는 앱의 * Actions * 열에서 Options 메뉴를 선택합니다.
   - 원하는 앱의 이름을 선택하고 페이지 오른쪽 상단의 상태 드롭다운 목록을 선택합니다.
3. 클론 * 을 선택합니다.
4. 클론의 세부 정보 지정:
   - 이름을 입력합니다.
   - 클론의 대상 클러스터를 선택합니다.
   - 클론의 대상 네임스페이스를 입력합니다. 앱과 연결된 각 소스 네임스페이스는 사용자가 정의하는 대상 네임스페이스에 매핑됩니다.

     Astra Control은 클론 작업의 일부로 새 대상 네임스페이스를 생성합니다. 지정한 대상 네임스페이스가 대상 클러스터에 이미 있으면 안 됩니다.

   - 다음 * 을 선택합니다.
   - 앱과 연결된 원래 저장소 클래스를 유지하거나 다른 저장소 클래스를 선택합니다.

     앱의 스토리지 클래스를 네이티브 클라우드 공급자 스토리지 클래스 또는 기타 지원되는 스토리지 클래스로 마이그레이션할 수 있습니다. 에서 지원하는 스토리지 클래스로 `ontap-nas` 또는 에서 지원하는 저장소 클래스가 있는 다른 클러스터로 앱을 복사합니다 `ontap-nas-economy` 드라이버.

     다른 스토리지 클래스를 선택했고 복원 시 이 스토리지 클래스가 존재하지 않는 경우 오류가 반환됩니다.

5. 다음 * 을 선택합니다.

6. 클론에 대한 정보를 검토하고 * Clone * 을 선택합니다.

결과

Astra Control은 사용자가 제공한 정보를 기반으로 앱을 복제합니다. 새 애플리케이션 클론이 에 있을 때 클론 작업이 성공적으로 수행됩니다 `Healthy` 상태를 표시합니다.

클론 또는 복원 작업을 통해 새 네임스페이스를 생성한 후 계정 관리자/소유자는 구성원 사용자 계정을 편집하고 영향을 받는 사용자의 역할 제약 조건을 업데이트하여 새 네임스페이스에 대한 액세스 권한을 부여할 수 있습니다.

> 데이터 보호 작업(클론, 백업 또는 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.

앱 실행 후크 관리

실행 후크는 관리되는 앱의 데이터 보호 작업과 함께 실행되도록 구성할 수 있는 사용자 지정 작업입니다. 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.

실행 후크 유형

Astra Control은 실행 가능 시점을 기준으로 다음과 같은 유형의 실행 후크를 지원합니다.

- 사전 스냅샷
- 사후 스냅샷
- 사전 백업
- 백업 후
- 사후 복원
- 장애 조치 후

실행 후크 필터

실행 후크를 응용 프로그램에 추가하거나 편집할 때 실행 후크에 필터를 추가하여 후크가 일치시킬 컨테이너를 관리할 수 있습니다. 필터는 모든 컨테이너에서 동일한 컨테이너 이미지를 사용하는 응용 프로그램에 유용하지만 각 이미지를 다른 용도(예: Elasticsearch)로 사용할 수 있습니다. 필터를 사용하면 실행 후크가 실행되는 시나리오를 만들 수 있습니다. 단, 모든 동일한 컨테이너를 실행하는 것은 아닙니다. 단일 실행 후크에 대해 여러 개의 필터를 만들면 논리적 필터 및 연산자와 결합됩니다. 실행 후크당 최대 10개의 활성 필터를 사용할 수 있습니다.

실행 후크에 추가하는 각 필터는 클러스터의 컨테이너와 일치시키기 위해 정규식을 사용합니다. 후크가 컨테이너와 일치하면 후크는 해당 컨테이너에서 연결된 스크립트를 실행합니다. 필터에 대한 정규식은 정규식 2(RE2) 구문을 사용합니다. 이 구문은 일치 목록에서 컨테이너를 제외하는 필터를 만드는 것을 지원하지 않습니다. Astra Control이 실행 후크 필터의 정규식에 대해 지원하는 구문에 대한 자세한 내용은 을 참조하십시오 "정규식 2(RE2) 구문 지원".

> 복원 또는 클론 작업 후에 실행되는 실행 후크에 네임스페이스 필터를 추가하고 복원 또는 클론 소스와 대상이 서로 다른 네임스페이스에 있는 경우 네임스페이스 필터는 대상 네임스페이스에만 적용됩니다.

사용자 정의 실행 후크에 대한 중요 참고 사항

앱에 대한 실행 후크를 계획할 때 다음 사항을 고려하십시오.

> 실행 후크는 실행 중인 응용 프로그램의 기능을 줄이거나 완전히 비활성화하기 때문에 사용자 지정 실행 후크가 실행되는 시간을 최소화해야 합니다.
> 연결된 실행 후크와 함께 백업 또는 스냅샷 작업을 시작한 다음 취소하면 백업 또는 스냅샷 작업이 이미 시작된 경우에도 후크를 실행할 수 있습니다. 즉, 백업 후 실행 후크에 사용되는 논리는 백업이 완료된 것으로 가정할 수 없습니다.

- 실행 후크는 스크립트를 사용하여 작업을 수행해야 합니다. 많은 실행 후크가 동일한 스크립트를 참조할 수 있습니다.
- Astra Control에는 실행 후크가 실행 가능한 셸 스크립트 형식으로 기록하는 데 사용하는 스크립트가 필요합니다.
- 스크립트 크기는 96KB로 제한됩니다.
- Astra Control은 실행 후크 설정과 모든 일치 기준을 사용하여 스냅샷, 백업 또는 복구 작업에 적용할 수 있는 후크를 결정합니다.
- 모든 실행 후크 장애는 소프트 장애이며, 후크가 실패하더라도 다른 후크와 데이터 보호 작업은 계속 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.
- 임시 데이터 보호 작업의 경우 모든 후크 이벤트가 생성되고 * Activity * 페이지 이벤트 로그에 저장됩니다. 그러나 예약된 데이터 보호 작업의 경우 후크 장애 이벤트만 이벤트 로그에 기록됩니다(예약된 데이터 보호 작업 자체에서 생성되는 이벤트는 계속 기록됨).
- Astra Control Center가 복제된 소스 앱을 대상 앱으로 페일오버하면 페일오버가 완료된 후 소스 앱에 대해 활성화된 장애 조치 후 실행 후크가 대상 앱에 대해 실행됩니다.

> Astra Control Center 23.04와 함께 복원 후 후크를 실행하고 Astra Control Center를 23.07로 업그레이드한 경우 페일오버 복제 후 복원 후 실행 후크가 더 이상 실행되지 않습니다. 앱을 위한 새로운 장애 조치 후 실행 후크를 만들어야 합니다. 또는 "사후 복원"에서 "사후 페일오버"로 페일오버하기 위한 기존 복원 후 후크의 작업 유형을 변경할 수 있습니다.

실행 순서

데이터 보호 작업이 실행되면 실행 후크 이벤트가 다음 순서로 발생합니다.

1. 해당되는 모든 사용자 정의 사전 작업 실행 후크는 해당 컨테이너에서 실행됩니다. 필요한 만큼 사용자 지정 사전 작업 후크를 만들고 실행할 수 있지만, 이 후크의 실행 순서는 보장되거나 구성할 수 없습니다.
2. 데이터 보호 작업이 수행됩니다.
3. 해당되는 모든 사용자 지정 작업 후 실행 후크는 해당 컨테이너에서 실행됩니다. 필요한 만큼 사용자 지정 사후 작업 후크를 만들고 실행할 수 있지만 작업 후 후크의 실행 순서는 보장되거나 구성할 수 없습니다.

같은 유형의 실행 후크를 여러 개 생성하는 경우(예: 사전 스냅샷) 해당 후크의 실행 순서는 보장되지 않습니다. 그러나 다른 유형의 후크를 실행하는 순서는 보장됩니다. 예를 들어, 서로 다른 모든 유형의 후크가 있는 구성의 실행 순서는 다음과 같습니다.

1. 예비 후크가 실행되었습니다
2. 사전 스냅샷 후크가 실행되었습니다
3. 사후 스냅샷 후크가 실행되었습니다
4. 백업 후 후크가 실행되었습니다
5. 복원 후 후크가 실행되었습니다

시나리오 번호 2에서 이 구성의 예를 볼 수 있습니다 후크가 실행될지 여부를 결정합니다.

운영 환경에서 실행 후크 스크립트를 사용하려면 항상 해당 스크립트를 테스트해야 합니다. 'kubbeck exec' 명령을 사용하여 스크립트를 편리하게 테스트할 수 있습니다. 운영 환경에서 실행 후크를 사용하도록 설정한 후 결과 스냅샷과 백업을 테스트하여 정합성이 보장되는지 확인합니다. 앱을 임시 네임스페이스에 클론 복제하고, 스냅샷 또는 백업을 복원한 다음 앱을 테스트하여 이 작업을 수행할 수 있습니다.

후크가 실행될지 여부를 결정합니다

다음 표를 사용하여 사용자 지정 실행 후크가 앱에 대해 실행되는지 여부를 확인할 수 있습니다.

모든 상위 수준 앱 작업은 스냅샷, 백업 또는 복원의 기본 작업 중 하나를 실행하는 것으로 구성됩니다. 시나리오에 따라 클론 작업은 이러한 작업의 다양한 조합으로 구성되므로 클론 작업이 실행되는 실행 후크는 달라집니다.

데이터 이동 없이 복원 작업을 수행하려면 기존 스냅샷 또는 백업이 필요하므로 이러한 작업은 스냅샷 또는 백업 후크를 실행하지 않습니다.

를 시작한 다음 스냅샷이 포함된 백업을 취소하고 연결된 실행 후크가 있는 경우 일부 후크가 실행될 수 있고 그렇지 않은 백업이 있을 수 있습니다. 즉, 백업 후 실행 후크는 백업이 완료된 것으로 가정할 수 없습니다. 연결된 실행 후크와 함께 취소된 백업의 경우 다음 사항에 유의하십시오.

- 예비 백업 및 예비 후크는 항상 실행됩니다.
- 백업에 새 스냅샷이 포함되어 있고 스냅샷이 시작된 경우 사전 스냅샷 및 사후 스냅샷 후크가 실행됩니다.
- 스냅샷을 시작하기 전에 백업을 취소하면 사전 스냅샷 및 사후 스냅샷 후크가 실행되지 않습니다.

시나리오	작동	기존 스냅샷	더 많은 워크로드 추가/제거	네임스페이스	클러스터	스냅샷 후크가 실행됩니다	백업 후크가 실행됩니다	후크 실행을 복원합니다	페일오버 후크가 실행됩니다
1	복제	해당 없음	해당 없음	신규	동일합니다	예	해당 없음	예	해당 없음
2	복제	해당 없음	해당 없음	신규	다릅니다	예	예	예	해당 없음
3	복제 또는 복원	예	해당 없음	신규	동일합니다	해당 없음	해당 없음	예	해당 없음
4	복제 또는 복원	해당 없음	예	신규	동일합니다	해당 없음	해당 없음	예	해당 없음
5	복제 또는 복원	예	해당 없음	신규	다릅니다	해당 없음	해당 없음	예	해당 없음

시나리오	작동	기존 스냅샷	더 많은 워크로드 추가/제거	네임스페이스	클러스터	스냅샷 후크가 실행됩니다	백업 후크가 실행됩니다	후크 실행을 복원합니다	페일오버 후크가 실행됩니다
6	복제 또는 복원	해당 없음	예	신규	다릅니다	해당 없음	해당 없음	예	해당 없음
7	복원	예	해당 없음	기존	동일합니다	해당 없음	해당 없음	예	해당 없음
8	복원	해당 없음	예	기존	동일합니다	해당 없음	해당 없음	예	해당 없음
9	스냅샷	해당 없음	해당 없음	해당 없음	해당 없음	예	해당 없음	해당 없음	해당 없음
10	백업	해당 없음	해당 없음	해당 없음	해당 없음	예	예	해당 없음	해당 없음
11	백업	예	해당 없음	해당 없음	해당 없음	해당 없음	해당 없음	해당 없음	해당 없음
12	페일오버	예	해당 없음	복제에 의해 생성되었습니다	다릅니다	해당 없음	해당 없음	해당 없음	예
13	페일오버	예	해당 없음	복제에 의해 생성되었습니다	동일합니다	해당 없음	해당 없음	해당 없음	예

실행 후크 예

를 방문하십시오 "NetApp Verda GitHub 프로젝트" Apache Cassandra 및 Elasticsearch와 같은 인기 있는 앱의 실제 실행 후크를 다운로드하려면 다음을 수행합니다. 예제를 보고 사용자 지정 실행 후크를 구조화하는 아이디어를 얻을 수도 있습니다.

기존 실행 후크를 봅니다

앱의 기존 사용자 지정 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, 일치하는 컨테이너 수, 생성 시간 및 실행 시간(사전 또는 사후 작업)을 확인할 수 있습니다. 를 선택할 수 있습니다 + 실행할 컨테이너 목록을 확장하려면 후크 이름 옆에 있는 아이콘을 클릭합니다. 이 응용 프로그램의 실행 후크를 둘러싼 이벤트 로그를 보려면 * Activity * 탭으로 이동하십시오.

기존 스크립트 보기

업로드된 기존 스크립트를 볼 수 있습니다. 또한 이 페이지에서 사용 중인 스크립트와 해당 스크립트를 사용하는 후크를 확인할 수 있습니다.

단계

1. 계정 * 으로 이동합니다.
2. 스크립트 * 탭을 선택합니다.

   이 페이지에서는 업로드된 기존 스크립트 목록을 볼 수 있습니다. Used By* 열에는 각 스크립트를 사용하는 실행 후크가 표시됩니다.

스크립트를 추가합니다

각 실행 후크는 스크립트를 사용하여 작업을 수행해야 합니다. 실행 후크가 참조할 수 있는 스크립트를 하나 이상 추가할 수 있습니다. 많은 실행 후크가 동일한 스크립트를 참조할 수 있으므로 하나의 스크립트만 변경하여 여러 실행 후크를 업데이트할 수 있습니다.

단계

1. 계정 * 으로 이동합니다.
2. 스크립트 * 탭을 선택합니다.
3. 추가 * 를 선택합니다.
4. 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.
     v. Save script * 를 선택합니다.
   - 클립보드에서 사용자 정의 스크립트를 붙여 넣습니다.
     i. 붙여넣기 또는 형식 * 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.
5. Save script * 를 선택합니다.

결과

새 스크립트가 * 스크립트 * 탭의 목록에 나타납니다.

스크립트를 삭제합니다

스크립트가 더 이상 필요하지 않고 실행 후크에서 사용되지 않는 경우 시스템에서 스크립트를 제거할 수 있습니다.

단계

1. 계정 * 으로 이동합니다.
2. 스크립트 * 탭을 선택합니다.
3. 제거할 스크립트를 선택하고 * Actions * 열에서 메뉴를 선택합니다.

4. 삭제 * 를 선택합니다.

> 스크립트가 하나 이상의 실행 후크에 연결되어 있으면 * 삭제 * 작업을 사용할 수 없습니다. 스크립트를 삭제하려면 먼저 연결된 실행 후크를 편집하여 다른 스크립트에 연결합니다.

사용자 지정 실행 후크를 만듭니다

앱에 대한 사용자 정의 실행 후크를 생성하여 Astra Control에 추가할 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 특정 명령을 실행하거나 실행 파일에 대한 전체 경로를 제공하지 않는 한 파일 시작 부분에 적절한 셸을 지정해야 합니다.

단계

1. 응용 프로그램 * 을 선택한 다음 관리되는 응용 프로그램의 이름을 선택합니다.
2. Execution hook * 탭을 선택합니다.
3. 추가 * 를 선택합니다.
4. 후크 세부 정보 * 영역에서:
   a. 작업 * 드롭다운 메뉴에서 작업 유형을 선택하여 후크를 언제 실행해야 하는지 결정합니다.
   b. 후크의 고유한 이름을 입력합니다.
   c. (선택 사항) 실행 중에 후크에 전달할 인수를 입력하고 각 인수 뒤에 Enter 키를 눌러 각 인수를 기록합니다.
5. (선택 사항) * Hook Filter Details * 영역에서 실행 후크가 실행되는 컨테이너를 제어하는 필터를 추가할 수 있습니다.
   a. 필터 추가 * 를 선택합니다.
   b. Hook filter type * 열의 드롭다운 메뉴에서 필터링할 특성을 선택합니다.
   c. Regex * 열에 필터로 사용할 정규식을 입력합니다. Astra Control은 를 사용합니다 "정규식 2(RE2) regex 구문".

      > 정규식 필드에 다른 텍스트가 없는 특성(예: pod 이름)의 정확한 이름을 필터링하면 부분 문자열 일치가 수행됩니다. 정확한 이름과 해당 이름만 일치시키려면 정확한 문자열 일치 구문(예: `^exact_podname$`)를 클릭합니다.

   d. 필터를 더 추가하려면 * 필터 추가 * 를 선택합니다.

      > 실행 후크에 대한 여러 필터가 논리 및 연산자와 결합됩니다. 실행 후크당 최대 10개의 활성 필터를 사용할 수 있습니다.

6. 완료되면 * Next * 를 선택합니다.
7. Script * 영역에서 다음 중 하나를 수행합니다.
   - 새 스크립트를 추가합니다.
     i. 추가 * 를 선택합니다.
     ii. 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.
  V. Save script * 를 선택합니다.
- 클립보드에서 사용자 정의 스크립트를 붙여 넣습니다.
  I. 붙여넣기 또는 형식 * 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.

◦ 목록에서 기존 스크립트를 선택합니다.

이렇게 하면 실행 후크에 이 스크립트를 사용하도록 지시합니다.

8. 다음 * 을 선택합니다.
9. 실행 후크 구성을 검토합니다.
10. 추가 * 를 선택합니다.

실행 후크의 상태를 확인합니다

스냅샷, 백업 또는 복원 작업이 실행된 후에 작업의 일부로 실행된 실행 후크의 상태를 확인할 수 있습니다. 이 상태 정보를 사용하여 실행 후크를 유지할지, 수정하거나 삭제할 것인지 결정할 수 있습니다.

단계

1. 응용 프로그램 * 을 선택한 다음 관리되는 응용 프로그램의 이름을 선택합니다.
2. 데이터 보호 * 탭을 선택합니다.
3. 스냅샷 * 을 선택하여 실행 중인 스냅샷을 보거나 * 백업 * 을 선택하여 실행 중인 백업을 확인합니다.

   후크 상태 * 는 작업이 완료된 후 실행 후크의 상태를 표시합니다. 상태 위로 마우스를 가져가면 자세한 정보를 볼 수 있습니다. 예를 들어, 스냅샷 중에 실행 후크 오류가 발생한 경우 해당 스냅샷의 후크 상태 위로 마우스를 이동하면 실패한 실행 후크 목록이 표시됩니다. 각 오류의 원인을 확인하려면 왼쪽 탐색 영역의 * Activity * 페이지를 확인하십시오.

스크립트 사용을 봅니다

Astra Control 웹 UI에서 특정 스크립트를 사용하는 실행 후크를 확인할 수 있습니다.

단계

1. 계정 * 을 선택합니다.
2. 스크립트 * 탭을 선택합니다.

   스크립트 목록의 * Used By * 열에 목록의 각 스크립트를 사용하는 후크에 대한 세부 정보가 포함되어 있습니다.

3. 관심 있는 스크립트에 대해 * Used By *(사용 대상 *) 열에서 정보를 선택합니다.

   스크립트를 사용하는 후크의 이름 및 스크립트를 실행하도록 구성된 작업 유형과 함께 더 자세한 목록이 나타납니다.

실행 후크를 편집합니다

실행 후크를 편집하여 속성, 필터 또는 사용하는 스크립트를 변경할 수 있습니다. 실행 후크를 편집하려면 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. 편집할 후크의 경우 * Actions * 열에서 옵션 메뉴를 선택합니다.
4. 편집 * 을 선택합니다.
5. 필요한 사항을 변경하고 각 섹션을 완료한 후 * 다음 * 을 선택합니다.
6. 저장 * 을 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. 비활성화할 후크의 경우 * Actions * 열에서 옵션 메뉴를 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 후크의 경우 * Actions * 열에서 옵션 메뉴를 선택합니다.
4. 삭제 * 를 선택합니다.
5. 결과 대화 상자에 "delete"를 입력하여 확인합니다.
6. 예, 실행 후크 삭제 * 를 선택합니다.

를 참조하십시오

- "NetApp Verda GitHub 프로젝트"

Astra Control Center를 사용하여 Astra Control Center를 보호합니다

Astra Control Center가 실행 중인 Kubernetes 클러스터에서 심각한 오류로부터 복원력을 개선하려면 Astra Control Center 애플리케이션 자체를 보호합니다. 보조 Astra Control Center 인스턴스를 사용하여 Astra Control Center를 백업 및 복원하거나 기본 스토리지에서 ONTAP를 사용하는 경우 Astra 복제를 사용할 수 있습니다.

이 시나리오에서는 Astra Control Center의 두 번째 인스턴스가 다른 오류 도메인에 구축 및 구성되었으며 1차 Astra Control Center 인스턴스와 다른 두 번째 Kubernetes 클러스터에서 실행됩니다. 두 번째 Astra Control 인스턴스는 운영 Astra Control Center 인스턴스를 백업하고 복원하는 데 사용됩니다. 복원되거나 복제된 Astra Control Center 인스턴스는 애플리케이션 클러스터 애플리케이션에 대한 애플리케이션 데이터 관리를 계속 제공하며 이러한 애플리케이션의 백업 및 스냅샷에 대한 액세스 권한을 복원합니다.

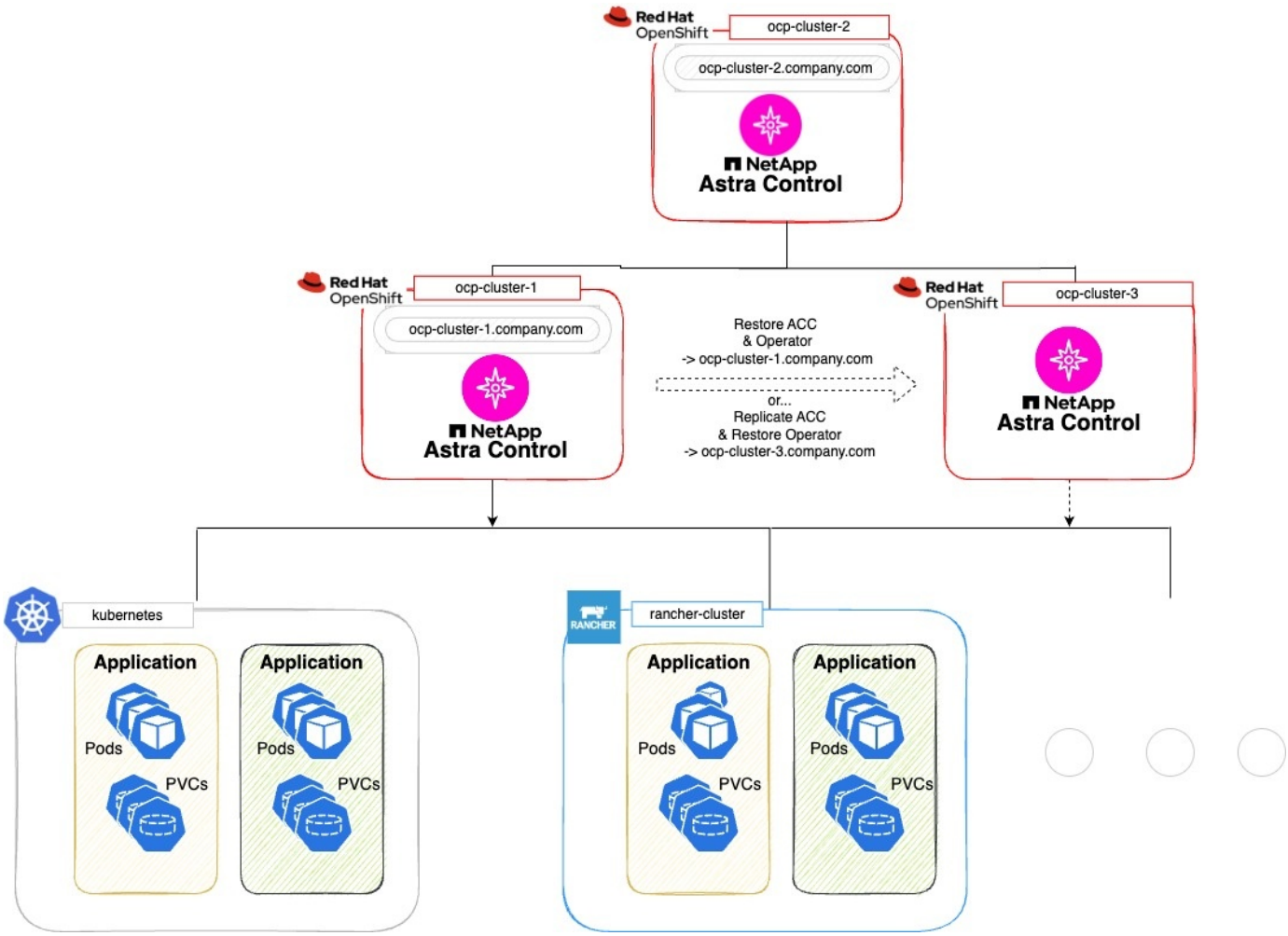
시작하기 전에

Astra Control Center에 대한 보호 시나리오를 설정하기 전에 다음 사항이 있는지 확인하십시오.

- * 1차 Astra Control Center 인스턴스를 실행하는 Kubernetes 클러스터 *: 이 클러스터는 애플리케이션 클러스터를 관리하는 1차 Astra Control Center 인스턴스를 호스팅합니다.
- * 보조 Astra Control Center 인스턴스를 실행하는 기본 시스템과 동일한 Kubernetes 배포 유형의 두 번째 Kubernetes 클러스터 *: 이 클러스터는 기본 Astra Control Center 인스턴스를 관리하는 Astra Control Center 인스턴스를 호스팅합니다.
- * 기본 Kubernetes 배포 유형이 동일한 세 번째 Kubernetes 클러스터 *: 이 클러스터는 Astra Control Center의 복원되거나 복제된 인스턴스를 호스팅합니다. 현재 운영 사이트에 구축되어 있는 것과 동일한 Astra Control Center 네임스페이스를 사용할 수 있어야 합니다. 예를 들어, Astra Control Center가 네임스페이스에 구축된 경우 `netapp-acc` 소스 클러스터에서 네임스페이스입니다 `netapp-acc` 사용 가능하고 대상 Kubernetes 클러스터의 어떤 애플리케이션에서도 사용하지 않아야 합니다.
- * S3 호환 버킷 *: 각 Astra Control Center 인스턴스에는 액세스 가능한 S3 호환 오브젝트 스토리지 버킷이 있습니다.
- * 구성된 로드 밸런서 *: 로드 밸런서는 Astra에 대한 IP 주소를 제공하며 애플리케이션 클러스터와 두 S3 버킷 모두에 대한 네트워크 연결이 있어야 합니다.
- * 클러스터가 Astra Control Center 요구 사항을 충족함 *: Astra Control Center 보호에 사용되는 각 클러스터가 충족함 "일반 Astra Control Center 요구사항".

이 작업에 대해

다음 절차에서는 다음 중 하나를 사용하여 Astra Control Center를 새 클러스터로 복원하는 데 필요한 단계를 설명합니다 백업 및 복원 또는 복제. 단계는 여기에 설명된 예제 구성을 기반으로 합니다.



이 예제 구성에서는 다음과 같이 표시됩니다.

- * 1차 Astra Control Center 인스턴스를 실행하는 Kubernetes 클러스터 *:
  - OpenShift 클러스터: `ocp-cluster-1`
  - Astra Control Center 1차 인스턴스: `ocp-cluster-1.company.com`
  - 이 클러스터는 애플리케이션 클러스터를 관리합니다.
- * 보조 Astra Control Center 인스턴스를 실행하는 기본 Kubernetes 배포 유형이 동일한 두 번째 Kubernetes 클러스터 *:
  - OpenShift 클러스터: `ocp-cluster-2`
  - Astra Control Center 2차 인스턴스: `ocp-cluster-2.company.com`
  - 이 클러스터는 기본 Astra Control Center 인스턴스를 백업하거나 다른 클러스터(이 예에서는)에 대한 복제를 구성하는 데 사용됩니다 `ocp-cluster-3` 클러스터).
- * 복원 작업에 사용될 기본 Kubernetes 배포 유형이 동일한 세 번째 Kubernetes 클러스터 *:
  - OpenShift 클러스터: `ocp-cluster-3`
  - Astra Control Center 3번째 인스턴스: `ocp-cluster-3.company.com`
  - 이 클러스터는 Astra Control Center 복원 또는 복제 페일오버에 사용됩니다.

이상적으로는, 애플리케이션 클러스터는 위의 이미지에서 Kubernetes 및 Rancher 클러스터에 설명된 대로 Astra Control Center 클러스터 3개 외부에 위치해야 합니다.

다이어그램에 표시되지 않음:

- 모든 클러스터에는 Trident가 설치된 ONTAP 백 엔드가 있습니다.
- 이 구성에서 OpenShift 클러스터는 로드 밸런서로 MetalLB를 사용합니다.
- 스냅샷 컨트롤러와 VolumeSnapshotClass도 에 설명된 대로 모든 클러스터에 설치됩니다 "필수 구성 요소".

1단계 옵션**: Astra Control Center** 백업 및 복원

이 절차에서는 백업 및 복원을 사용하여 Astra Control Center를 새 클러스터로 복원하는 데 필요한 단계를 설명합니다.

이 예에서는 Astra Control Center가 항상 아래에 설치됩니다 `netapp-acc` 네임스페이스 및 연산자는 아래에 설치됩니다 `netapp-acc-operator` 네임스페이스.

설명한 것은 아니지만 Astra Control Center 운영자는 Astra CR과 동일한 네임스페이스에 구축할 수도 있습니다.

시작하기 전에

- 클러스터에 운영 Astra Control Center를 설치했습니다.
- 보조 Astra Control Center를 다른 클러스터에 설치했습니다.

단계

1. 에서 실행되는 2차 Astra Control Center 인스턴스에서 운영 Astra Control Center 애플리케이션 및 타겟 클러스터를 관리합니다 `ocp-cluster-2` 클러스터):
   a. 보조 Astra Control Center 인스턴스에 로그인합니다.
   b. "1차 Astra Control Center 클러스터를 추가합니다" (`ocp-cluster-1`)를 클릭합니다.
   c. "대상 세 번째 클러스터를 추가합니다" (`ocp-cluster-3`)를 선택합니다.
2. 보조 Astra Control Center에서 Astra Control Center 및 Astra Control Center 운영자:
   a. 응용 프로그램 페이지에서 * 정의 * 를 선택합니다.
   b. Define application * (애플리케이션 정의 *) 창에서 새 애플리케이션 이름을 입력합니다 (`netapp-acc`)를 클릭합니다.
   c. 1차 Astra Control Center를 실행 중인 클러스터를 선택합니다 (`ocp-cluster-1`)를 선택합니다.
   d. 를 선택합니다 `netapp-acc` Namespace * 드롭다운 목록에서 Astra Control Center의 네임스페이스입니다.
   e. 클러스터 리소스 페이지에서 * 추가 클러스터 범위 리소스 포함 * 을 선택합니다.
   f. 포함 규칙 추가 * 를 선택합니다.
   g. 다음 항목을 선택하고 * 추가 * 를 선택합니다.
      - 라벨 선택기: acc-CRD
      - 그룹: apiextensions.k8s.io
      - 버전: v1

- 종류: CustomResourceDefinition

h. 응용 프로그램 정보를 확인합니다.

i. 정의 * 를 선택합니다.

정의 * 를 선택한 후 연산자에 대해 애플리케이션 정의 프로세스를 반복합니다 `netapp-acc-operator`)를 선택하고 를 선택합니다 `netapp-acc-operator` 응용 프로그램 정의 마법사의 네임스페이스입니다.

3. Astra Control Center 및 운영자 백업:

a. 보조 Astra Control Center에서 애플리케이션 탭을 선택하여 애플리케이션 페이지로 이동합니다.

b. "백업하다" Astra Control Center 애플리케이션 (`netapp-acc`)를 클릭합니다.

c. "백업하다" 오퍼레이터 (`netapp-acc-operator`)를 클릭합니다.

4. Astra Control Center와 운영자를 백업한 후 를 통해 DR(재해 복구) 시나리오를 시뮬레이션합니다 "Astra Control Center 제거 중" 운영 클러스터에서

Astra Control Center를 새 클러스터(이 절차에서 설명하는 세 번째 Kubernetes 클러스터)에 복원하고 새로 설치된 Astra Control Center의 운영 클러스터와 동일한 DNS를 사용합니다.

5. 보조 Astra Control Center를 사용하여 "복원" Astra Control Center 애플리케이션의 1차 인스턴스:

a. 응용 프로그램 * 을 선택한 다음 Astra Control Center 응용 프로그램의 이름을 선택합니다.

b. 작업 열의 옵션 메뉴에서 * 복원 * 을 선택합니다.

c. 복원 유형으로 * Restore to new namespaces * 를 선택합니다.

d. 복원 이름을 입력합니다 (`netapp-acc`)를 클릭합니다.

e. 대상 세 번째 클러스터를 선택합니다 (`ocp-cluster-3`)를 클릭합니다.

f. 원본 네임스페이스와 동일한 네임스페이스가 되도록 대상 네임스페이스를 업데이트합니다.

g. Restore Source 페이지에서 복구 소스로 사용할 애플리케이션 백업을 선택합니다.

h. Restore using original storage classes * 를 선택합니다.

i. Restore all resources * 를 선택합니다.

j. 복원 정보를 검토한 다음 * Restore * 를 선택하여 Astra Control Center를 대상 클러스터로 복원하는 복원 프로세스를 시작합니다 (`ocp-cluster-3`)를 클릭합니다. 애플리케이션이 들어가면 복구가 완료됩니다 `available` 상태.

6. 대상 클러스터에서 Astra Control Center 구성:

a. 터미널을 열고 kubeconfig를 사용하여 대상 클러스터에 연결합니다 (`ocp-cluster-3`) 복원된 Astra Control Center가 포함되어 있습니다.

b. 를 확인합니다 `ADDRESS` Astra Control Center 구성의 열은 운영 시스템의 DNS 이름을 참조합니다.

```
kubectl get acc -n netapp-acc
```

응답:

```
NAME  UUID                                   VERSION    ADDRESS
READY
astra 89f4fd47-0cf0-4c7a-a44e-43353dc96ba8 23.07.0-24 ocp-cluster-
1.company.com           True
```

a. 를 누릅니다 `ADDRESS` 위 응답의 필드에 기본 Astra Control Center 인스턴스의 FQDN이 없습니다. Astra Control Center DNS를 참조하도록 구성을 업데이트하십시오.

```
kubectl edit acc -n netapp-acc
```

i. 를 변경합니다 `astraAddress` 에서 `spec:` FQDN으로 이동합니다 (`ocp-cluster-1.company.com` 이 예에서는 기본 Astra Control Center 인스턴스의

ii. 구성을 저장합니다.

iii. 주소가 업데이트되었는지 확인합니다.

```
kubectl get acc -n netapp-acc
```

b. 로 이동합니다 Astra Control Center Operator를 복원합니다 섹션을 참조하십시오.

1단계 옵션**:** 복제를 사용하여 **Astra Control Center** 보호

이 절차에서는 를 구성하는 데 필요한 단계를 설명합니다 "Astra Control Center 복제" 1차 Astra Control Center 인스턴스를 보호하기 위해

이 예에서는 Astra Control Center가 항상 아래에 설치됩니다 `netapp-acc` 네임스페이스 및 연산자는 아래에 설치됩니다 `netapp-acc-operator` 네임스페이스.

시작하기 전에

- 클러스터에 운영 Astra Control Center를 설치했습니다.
- 보조 Astra Control Center를 다른 클러스터에 설치했습니다.

단계

1. 보조 Astra Control Center 인스턴스에서 운영 Astra Control Center 애플리케이션 및 타겟 클러스터 관리:
   a. 보조 Astra Control Center 인스턴스에 로그인합니다.
   b. "1차 Astra Control Center 클러스터를 추가합니다" (`ocp-cluster-1`)를 클릭합니다.
   c. "대상 세 번째 클러스터를 추가합니다" (`ocp-cluster-3`)를 사용하여 복제됩니다.
2. 보조 Astra Control Center에서 Astra Control Center 및 Astra Control Center 운영자:
   a. 클러스터 * 를 선택하고 기본 Astra Control Center가 포함된 클러스터를 선택합니다 (`ocp-cluster-1`)를 클릭합니다.
   b. Namespaces* 탭을 선택합니다.
   c. 를 선택합니다 `netapp-acc` 및 `netapp-acc-operator` 네임스페이스.

d. 작업 메뉴를 선택하고 * 응용 프로그램으로 정의 * 를 선택합니다.

e. 정의된 애플리케이션을 보려면 * 애플리케이션에서 보기 * 를 선택합니다.

3. 복제를 위한 백엔드 구성:

> 복제를 수행하려면 운영 Astra Control Center 클러스터와 대상 클러스터가 필요합니다 (`ocp-cluster-3`) 다른 피어링된 ONTAP 스토리지 백엔드를 사용합니다.
> 각 백엔드가 피어링되어 Astra Control에 추가되면 백엔드가 백엔드 페이지의 * 검색됨 * 탭에 표시됩니다.

a. "피어링된 백엔드를 추가합니다" 운영 클러스터의 Astra Control Center로 전환

b. "피어링된 백엔드를 추가합니다" 대상 클러스터의 Astra Control Center로 전송

4. 복제 구성:

a. Applications(응용 프로그램) 화면에서 을 선택합니다 `netapp-acc` 응용 프로그램.

b. Configure replication policy * 를 선택합니다.

c. 를 선택합니다 `ocp-cluster-3` 대상 클러스터 역할을 합니다.

d. 스토리지 클래스를 선택합니다.

e. 를 입력합니다 `netapp-acc` 대상 네임스페이스로 사용됩니다.

f. 원하는 경우 복제 빈도를 변경합니다.

g. 다음 * 을 선택합니다.

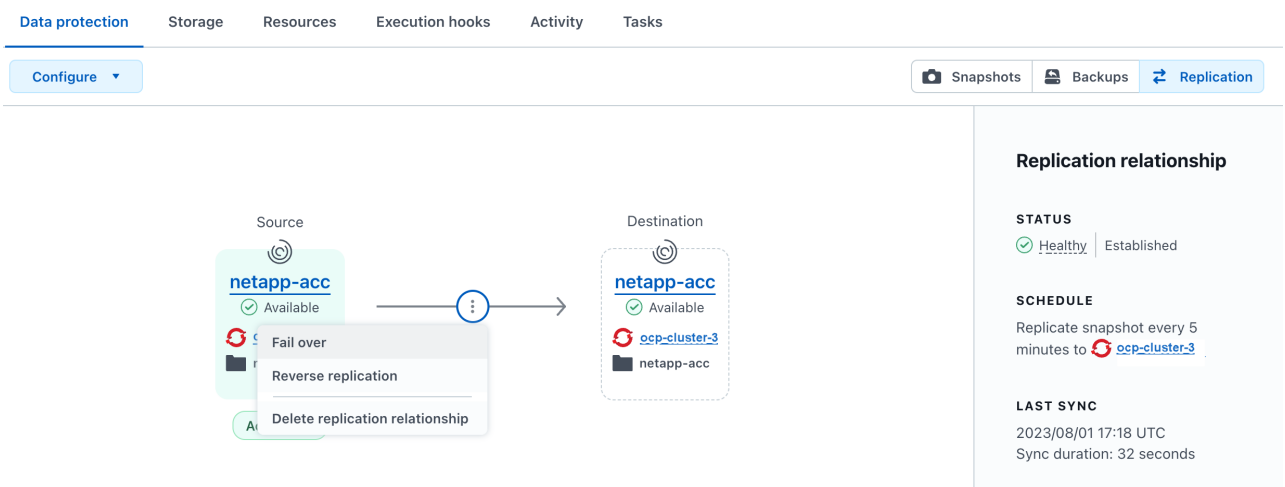
h. 구성이 올바른지 확인하고 * 저장 * 을 선택합니다.

에서 복제 관계가 전환됩니다 `Establishing` 를 선택합니다 `Established`. 활성 상태인 경우 이 복제는 복제 구성이 삭제될 때까지 5분마다 수행됩니다.

5. 운영 시스템이 손상되었거나 더 이상 액세스할 수 없는 경우 다른 클러스터로 복제를 페일오버합니다.

> 성공적인 페일오버를 보장하기 위해 대상 클러스터에 Astra Control Center가 설치되어 있지 않은지 확인합니다.

a. 수직 타원 아이콘을 선택하고 * Fail Over * 를 선택합니다.



b. 세부 정보를 확인하고 * Fail Over * 를 선택하여 페일오버 프로세스를 시작합니다.

   복제 관계 상태가 로 변경됩니다 `Failing over` 그리고 나서 `Failed over` 완료 시.

6. 페일오버 구성을 완료합니다.

   a. 터미널을 열고 세 번째 클러스터의 kubeconfig를 사용하여 연결합니다 (`ocp-cluster-3`)를 클릭합니다. 이제 이 클러스터에 Astra Control Center가 설치되었습니다.

   b. 세 번째 클러스터에서 Astra Control Center FQDN을 확인합니다 (`ocp-cluster-3`)를 클릭합니다.

   c. Astra Control Center DNS를 참조하도록 구성을 업데이트합니다.

```
kubectl edit acc -n netapp-acc
```

      i. 를 변경합니다 `astraAddress` 에서 `spec:` FQDN을 사용합니다 (`ocp-cluster-3.company.com`대상 세 번째 클러스터의).

      ii. 구성을 저장합니다.

      iii. 주소가 업데이트되었는지 확인합니다.

```
kubectl get acc -n netapp-acc
```

   d. 필요한 모든 traefik CRD가 있는지 확인합니다.

```
kubectl get crds | grep traefik
```

   필수 traefik CRD:

```
ingressroutes.traefik.containo.us
ingressroutes.traefik.io
ingressroutetcps.traefik.containo.us
ingressroutetcps.traefik.io
ingressrouteudps.traefik.containo.us
ingressrouteudps.traefik.io
middlewares.traefik.containo.us
middlewares.traefik.io
middlewaretcps.traefik.containo.us
middlewaretcps.traefik.io
serverstransports.traefik.containo.us
serverstransports.traefik.io
tlsoptions.traefik.containo.us
tlsoptions.traefik.io
tIsstores.traefik.containo.us
tIsstores.traefik.io
traefikservices.traefik.containo.us
traefikservices.traefik.io
```

a. 위의 CRD 중 일부가 누락된 경우:

i. 로 이동합니다 "Traefik 설명서".

ii. "정의" 영역을 파일로 복사합니다.

iii. 변경 내용 적용:

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

iv. Traefik 다시 시작:

```
kubectl get pods -n netapp-acc | grep -e "traefik" | awk '{print
$1}' | xargs kubectl delete pod -n netapp-acc"
```

b. 로 이동합니다 Astra Control Center Operator를 복원합니다 섹션을 참조하십시오.

2단계**: Astra Control Center Operator**를 복원합니다

보조 Astra Control Center를 사용하여 백업에서 기본 Astra Control Center 운영자를 복원합니다. 대상 네임스페이스는 소스 네임스페이스와 같아야 합니다. Astra Control Center가 운영 소스 클러스터에서 삭제된 경우에도 동일한 복원 단계를 수행하기 위한 백업은 계속 존재합니다.

단계

1. 응용 프로그램 * 을 선택한 다음 운영자 앱의 이름을 선택합니다 (`netapp-acc-operator`)를 클릭합니다.
2. 작업 열의 옵션 메뉴에서 * 복원 * 을 선택합니다

3. 복원 유형으로 * Restore to new namespaces * 를 선택합니다.
4. 대상 세 번째 클러스터를 선택합니다 (`ocp-cluster-3`)를 클릭합니다.
5. 네임스페이스를 운영 소스 클러스터에 연결된 네임스페이스와 동일하게 변경합니다 (`netapp-acc-operator`)를 클릭합니다.
6. 이전에 수행한 백업을 복구 소스로 선택합니다.
7. Restore using original storage classes * 를 선택합니다.
8. Restore all resources * 를 선택합니다.
9. 세부 정보를 검토한 후 * Restore * 를 클릭하여 복원 프로세스를 시작합니다.

   Applications 페이지에는 대상 세 번째 클러스터로 복구 중인 Astra Control Center 운영자가 표시됩니다 (`ocp-cluster-3`)를 클릭합니다. 프로세스가 완료되면 상태가 로 표시됩니다 `Available`. 10분 이내에 DNS 주소가 페이지에서 확인되어야 합니다.

결과

Astra Control Center, 등록된 클러스터, 스냅샷과 백업이 포함된 관리형 애플리케이션을 이제 타겟 세 번째 클러스터에서 사용할 수 있습니다 (`ocp-cluster-3`)를 클릭합니다. 원본에서 사용했던 보호 정책도 새 인스턴스에도 그대로 유지됩니다. 예약된 백업 또는 필요 시 백업 및 스냅샷을 계속 생성할 수 있습니다.

문제 해결

시스템 상태 및 보호 프로세스가 성공적인지 확인합니다.

- * Pod가 실행되지 않음 *: 모든 Pod가 실행 중인지 확인합니다.

  ```
  kubectl get pods -n netapp-acc
  ```

  에 일부 Pod가 있는 경우 `CrashLookBackOff` 다음과 같이 말하고 다시 시작하면 로 전환됩니다 `Running` 상태.

- * 시스템 상태 확인 *: Astra Control Center 시스템이 입력되었는지 확인합니다 `ready` 상태:

  ```
  kubectl get acc -n netapp-acc
  ```

  응답:

  ```
  NAME  UUID                                   VERSION    ADDRESS
  READY
  astra 89f4fd47-0cf0-4c7a-a44e-43353dc96ba8 23.07.0-24 ocp-cluster-
  1.company.com           True
  ```

- * 배포 상태 확인 * : Astra Control Center 배포 정보를 표시하여 이를 확인합니다 `Deployment State` 있습니다 `Deployed`.

```
kubectl describe acc astra -n netapp-acc
```

- *복원된 Astra Control Center UI가 404 오류를 반환합니다. * : 선택한 경우 이 오류가 발생합니다 `AccTraefik` 수신 옵션으로 을(를) 점검하십시오 Traefik CRD를 참조하십시오 모두 설치되었는지 확인합니다.

앱 및 클러스터 상태를 모니터링합니다

앱 및 클러스터 상태 요약 보기

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

이것들은 단순히 정적 숫자나 상태만이 아니라, 각 상태로부터 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.

클러스터 상태를 보고 스토리지 클래스를 관리합니다

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

클러스터의 기본 스토리지 클래스를 변경할 수도 있습니다.

클러스터 상태 및 세부 정보 보기

클러스터의 위치, 작업자 노드, 영구 볼륨 및 스토리지 클래스와 같은 클러스터에 대한 세부 정보를 볼 수 있습니다.

단계

1. Astra Control Center UI에서 * Clusters * 를 선택합니다.
2. 클러스터 * 페이지에서 세부 정보를 확인할 클러스터를 선택합니다.

> 클러스터가 인 경우 `removed` 클러스터 및 네트워크 연결이 양호해 보이지만(Kubernetes API를 사용하여 클러스터에 액세스하려는 외부 시도가 성공한 경우), Astra Control에 제공한 kubeconfig는 더 이상 유효하지 않을 수 있습니다. 클러스터의 인증서 순환 또는 만료 때문일 수 있습니다. 이 문제를 해결하려면 을 사용하여 Astra Control의 클러스터와 연결된 자격 증명을 업데이트하십시오 "Astra Control API를 참조하십시오".

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.

기본 스토리지 클래스를 변경합니다

클러스터의 기본 스토리지 클래스를 변경할 수 있습니다. Astra Control이 클러스터를 관리할 때 클러스터의 기본 스토리지 클래스를 추적합니다.

> kubbeck 명령을 사용하여 스토리지 클래스를 변경하지 마십시오. 대신 이 절차를 사용하십시오. kubectl을 사용하면 Astra Control이 변경 사항을 되돌립니다.

단계

1. Astra Control Center 웹 UI에서 * Clusters * 를 선택합니다.
2. 클러스터 * 페이지에서 변경할 클러스터를 선택합니다.
3. Storage * 탭을 선택합니다.
4. 스토리지 클래스 * 범주를 선택합니다.
5. 기본값으로 설정할 스토리지 클래스에 대해 * Actions * 메뉴를 선택합니다.
6. Set as default * 를 선택합니다.

앱의 상태 및 세부 정보를 봅니다

앱 관리를 시작한 후 Astra Control은 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. 정보를 검토합니다.
   - * App Status *: 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 *: 앱 보호 상태 제공:
     - * 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 포드의 상태에 대한 정보입니다.
   - * 데이터 보호 *: 데이터 보호 정책을 구성하고 기존 스냅샷 및 백업을 볼 수 있습니다.
   - * 스토리지 *: 앱 레벨 영구 볼륨을 표시합니다. 영구 볼륨의 상태는 Kubernetes 클러스터의 관점에서 나옵니다.
   - * 리소스 *: 백업 및 관리되는 리소스를 확인할 수 있습니다.
   - * 활동 *: 앱과 관련된 활동을 표시합니다.

> 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

로컬 사용자 및 역할 관리

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

LDAP를 사용하여 선택한 사용자에 대한 인증을 수행할 수도 있습니다.

LDAP를 사용합니다

LDAP는 분산된 디렉터리 정보에 액세스하기 위한 업계 표준 프로토콜이며 엔터프라이즈 인증에 널리 사용되는 프로토콜입니다. Astra Control Center를 LDAP 서버에 연결하여 선택한 Astra Control 사용자에 대한 인증을 수행할 수 있습니다. 이 구성에는 Astra와 LDAP를 통합하고 LDAP 정의에 해당하는 Astra Control 사용자 및 그룹을 정의하는 작업이 포함됩니다. Astra Control API 또는 웹 UI를 사용하여 LDAP 인증과 LDAP 사용자 및 그룹을 구성할 수 있습니다. 자세한 내용은 다음 설명서를 참조하십시오.

- "Astra Control API를 사용하여 원격 인증 및 사용자를 관리합니다"
- "Astra Control UI를 사용하여 원격 사용자 및 그룹을 관리합니다"
- "Astra Control UI를 사용하여 원격 인증을 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 가지며 앱 및 클러스터를 관리하고, 앱을 관리하고, 스냅샷 및 백업을 삭제할 수 있습니다.
   - 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. 멤버 또는 뷰어 역할이 있는 사용자에게 제약 조건을 추가하려면 * 제약 조건으로 역할 제한 * 확인란을 활성화합니다.

   제약 조건 추가에 대한 자세한 내용은 을 참조하십시오 "로컬 사용자 및 역할 관리".

7. 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 * 열의 옵션 메뉴에서 * 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. 작업 * 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.

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 사용자의 행에서 확인란을 선택합니다.
3. Actions * 열의 Options 메뉴에서 * 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.

역할을 관리합니다

네임스페이스 제약 조건을 추가하고 이러한 제약 조건에 대한 사용자 역할을 제한하여 역할을 관리할 수 있습니다. 이렇게 하면 조직 내의 리소스에 대한 액세스를 제어할 수 있습니다. Astra Control UI 또는 를 사용할 수 있습니다 "Astra Control API를 참조하십시오" 역할을 관리합니다.

역할에 네임스페이스 제약 조건을 추가합니다

관리자 또는 소유자 사용자는 구성원 또는 뷰어 역할에 네임스페이스 제약 조건을 추가할 수 있습니다.

단계

1. 계정 관리 * 탐색 영역에서 * 계정 * 을 선택합니다.
2. 사용자 * 탭을 선택합니다.

3. Actions * 열에서 Member 또는 Viewer 역할을 가진 사용자의 메뉴 버튼을 선택합니다.
4. 역할 편집 * 을 선택합니다.
5. 제약 조건으로 역할 제한 * 확인란을 활성화합니다.

   이 확인란은 구성원 또는 뷰어 역할에만 사용할 수 있습니다. 역할 * 드롭다운 목록에서 다른 역할을 선택할 수 있습니다.

6. 구속 조건 추가 * 를 선택합니다.

   네임스페이스 또는 네임스페이스 레이블별로 사용 가능한 제약 조건 목록을 볼 수 있습니다.

7. 네임스페이스 구성 방법에 따라 * 제약 조건 유형 * 드롭다운 목록에서 * Kubernetes 네임스페이스 * 또는 * Kubernetes 네임스페이스 레이블 * 을 선택합니다.
8. 목록에서 하나 이상의 네임스페이스 또는 레이블을 선택하여 해당 네임스페이스로 역할을 제한하는 제약 조건을 구성합니다.
9. Confirm * 을 선택합니다.

   역할 편집 * 페이지에는 이 역할에 대해 선택한 제약 조건 목록이 표시됩니다.

10. Confirm * 을 선택합니다.

    계정 * 페이지의 * 역할 * 열에서 구성원 또는 뷰어 역할에 대한 제약 조건을 볼 수 있습니다.

> 역할에 대한 제약 조건을 설정하고 제약 조건을 추가하지 않고 * 확인 * 을 선택하면 역할이 전체 제한 사항으로 간주됩니다(역할에 네임스페이스가 할당된 리소스에 대한 액세스가 거부됨).

역할에서 네임스페이스 제약 조건을 제거합니다

관리자 또는 소유자 사용자는 역할에서 네임스페이스 제약 조건을 제거할 수 있습니다.

단계

1. 계정 관리 * 탐색 영역에서 * 계정 * 을 선택합니다.
2. 사용자 * 탭을 선택합니다.
3. Actions * 열에서 활성 제약 조건이 있는 Member 또는 Viewer 역할을 가진 사용자의 메뉴 버튼을 선택합니다.
4. 역할 편집 * 을 선택합니다.

   역할 편집 * 대화 상자에 해당 역할에 대한 활성 제약 조건이 표시됩니다.

5. 제거할 구속 조건의 오른쪽에 있는 * X * 를 선택합니다.
6. Confirm * 을 선택합니다.

를 참조하십시오

- "사용자 역할 및 네임스페이스"

원격 인증을 관리합니다

LDAP는 분산된 디렉터리 정보에 액세스하기 위한 업계 표준 프로토콜이며 엔터프라이즈 인증에 널리 사용되는 프로토콜입니다. Astra Control Center를 LDAP 서버에 연결하여 선택한 Astra Control 사용자에 대한 인증을 수행할 수 있습니다.

이 구성에는 Astra와 LDAP를 통합하고 LDAP 정의에 해당하는 Astra Control 사용자 및 그룹을 정의하는 작업이 포함됩니다. Astra Control API 또는 웹 UI를 사용하여 LDAP 인증과 LDAP 사용자 및 그룹을 구성할 수 있습니다.

> Astra Control Center는 원격 인증이 활성화될 때 구성된 사용자 로그인 속성을 사용하여 원격 사용자를 검색하고 추적합니다. Astra Control Center에 표시하고자 하는 원격 사용자의 경우 이메일 주소("메일") 또는 사용자 주체 이름("userPrincipalName")의 속성이 이 필드에 있어야 합니다. 이 속성은 인증을 위한 Astra Control Center의 사용자 이름과 원격 사용자를 검색하는 데 사용됩니다.

LDAPS 인증을 위한 인증서를 추가합니다

LDAPS 연결을 사용할 때 Astra Control Center가 LDAP 서버를 인증할 수 있도록 LDAP 서버에 대한 개인 TLS 인증서를 추가합니다. 이 작업은 한 번만 수행하거나 설치한 인증서가 만료되면 수행해야 합니다.

단계

1. 계정 * 으로 이동합니다.
2. 인증서 * 탭을 선택합니다.
3. 추가 * 를 선택합니다.
4. 를 업로드하거나 `.pem` 클립보드에서 파일의 내용을 파일 또는 붙여 넣습니다.
5. 신뢰할 수 있는 * 확인란을 선택합니다.
6. 인증서 추가 * 를 선택합니다.

원격 인증을 사용합니다

LDAP 인증을 설정하고 Astra Control과 원격 LDAP 서버 간의 연결을 구성할 수 있습니다.

시작하기 전에

LDAPS를 사용하려는 경우 Astra Control Center에서 LDAP 서버를 인증할 수 있도록 LDAP 서버의 개인 TLS 인증서가 Astra Control Center에 설치되어 있는지 확인합니다. 을 참조하십시오 LDAPS 인증을 위한 인증서를 추가합니다 를 참조하십시오.

단계

1. 계정 > 연결 * 으로 이동합니다.
2. Remote Authentication* 창에서 구성 메뉴를 선택합니다.
3. Connect * 를 선택합니다.
4. 서버 IP 주소, 포트 및 기본 설정 연결 프로토콜(LDAP 또는 LDAPS)을 입력합니다.

> 가장 좋은 방법은 LDAP 서버와 연결할 때 LDAPS를 사용하는 것입니다. LDAPS에 연결하기 전에 Astra Control Center에 LDAP 서버의 개인 TLS 인증서를 설치해야 합니다.

5. 서비스 계정 자격 증명을 이메일 형식(administrator@example.com 입력합니다. Astra Control은 LDAP 서버에 연결할 때 이러한 자격 증명을 사용합니다.
6. 사용자 일치 * 섹션에서 다음을 수행합니다.
   a. LDAP 서버에서 사용자 정보를 검색할 때 사용할 기본 DN과 적절한 사용자 검색 필터를 입력합니다.
   b. (선택 사항) 디렉터리에서 사용자 로그인 속성을 사용하는 경우 `userPrincipalName` 대신 `mail`를 입력합니다 `userPrincipalName` 사용자 로그인 속성 * 필드의 올바른 속성
7. 그룹 일치 * 섹션에서 그룹 검색 기준 DN과 적절한 사용자 지정 그룹 검색 필터를 입력합니다.

   올바른 기본 DN(고유 이름)과 * 사용자 일치 * 및 * 그룹 일치 * 에 대한 적절한 검색 필터를 사용해야 합니다. 기본 DN은 Astra Control에 검색을 시작할 디렉토리 트리의 수준을 알리고 검색 필터는 디렉토리 트리 Astra Control의 검색 부분을 제한합니다.

8. 제출 * 을 선택합니다.

결과

원격 인증 * 창 상태는 * Pending * 으로 이동한 다음 LDAP 서버 연결이 설정되면 * Connected * 로 이동합니다.

원격 인증을 비활성화합니다

LDAP 서버에 대한 활성 연결을 일시적으로 해제할 수 있습니다.

LDAP 서버에 대한 연결을 비활성화하면 모든 설정이 저장되고 해당 LDAP 서버에서 Astra Control에 추가된 모든 원격 사용자 및 그룹은 유지됩니다. 언제든지 이 LDAP 서버에 다시 연결할 수 있습니다.

단계

1. 계정 > 연결 * 으로 이동합니다.
2. Remote Authentication* 창에서 구성 메뉴를 선택합니다.
3. 비활성화 * 를 선택합니다.

결과

원격 인증* 창 상태가 * 사용 안 함 * 으로 이동합니다. 모든 원격 인증 설정, 원격 사용자 및 원격 그룹이 보존되며 언제든지 연결을 다시 활성화할 수 있습니다.

원격 인증 설정을 편집합니다

LDAP 서버에 대한 연결을 해제했거나 * 원격 인증 * 창이 "연결 오류" 상태인 경우 구성 설정을 편집할 수 있습니다.

원격 인증* 창이 "사용 안 함" 상태이면 LDAP 서버 URL 또는 IP 주소를 편집할 수 없습니다. 다음 작업을 수행해야 합니다 원격 인증 연결을 끊습니다 먼저,

단계

1. 계정 > 연결 * 으로 이동합니다.
2. Remote Authentication* 창에서 구성 메뉴를 선택합니다.
3. 편집 * 을 선택합니다.

4. 필요한 내용을 변경하고 * Edit * 를 선택합니다.

원격 인증 연결을 끊습니다

LDAP 서버에서 연결을 끊고 Astra Control에서 구성 설정을 제거할 수 있습니다.

LDAP 사용자인 경우 연결을 끊으면 세션이 즉시 종료됩니다 LDAP 서버에서 연결을 끊으면 해당 LDAP 서버에 대한 모든 구성 설정이 Astra Control에서 제거되고 해당 LDAP 서버에서 추가된 모든 원격 사용자 및 그룹이 제거됩니다.

단계

1. 계정 > 연결 * 으로 이동합니다.
2. Remote Authentication* 창에서 구성 메뉴를 선택합니다.
3. Disconnect * 를 선택합니다.

결과

원격 인증 * 창 상태가 * 연결 끊김 * 으로 이동합니다. 원격 인증 설정, 원격 사용자 및 원격 그룹은 Astra Control에서 제거됩니다.

원격 사용자 및 그룹 관리

Astra Control 시스템에서 LDAP 인증을 활성화한 경우 LDAP 사용자 및 그룹을 검색하여 승인된 시스템 사용자에게 포함시킬 수 있습니다.

원격 사용자를 추가합니다

계정 소유자와 관리자는 Astra Control에 원격 사용자를 추가할 수 있습니다. Astra Control Center는 최대 10,000명의 LDAP 원격 사용자를 지원합니다.

Astra Control Center는 원격 인증이 활성화될 때 구성된 사용자 로그인 속성을 사용하여 원격 사용자를 검색하고 추적합니다. Astra Control Center에 표시하고자 하는 원격 사용자의 경우 이메일 주소("메일") 또는 사용자 주체 이름("userPrincipalName")의 속성이 이 필드에 있어야 합니다. 이 속성은 인증을 위한 Astra Control Center의 사용자 이름과 원격 사용자를 검색하는 데 사용됩니다.

시스템에 동일한 이메일 주소("메일" 또는 "사용자 기본 이름" 속성에 기반함)를 가진 로컬 사용자가 이미 있는 경우 원격 사용자를 추가할 수 없습니다. 사용자를 원격 사용자로 추가하려면 먼저 시스템에서 로컬 사용자를 삭제합니다.

단계

1. 계정 * 영역으로 이동합니다.
2. 사용자 및 그룹 * 탭을 선택합니다.
3. 페이지 맨 오른쪽에서 * 원격 사용자 * 를 선택합니다.
4. 추가 * 를 선택합니다.
5. 선택적으로 * Filter by email * 필드에 사용자의 이메일 주소를 입력하여 LDAP 사용자를 검색합니다.
6. 목록에서 한 명 이상의 사용자를 선택합니다.

7. 사용자에게 역할을 할당합니다.

   사용자와 사용자 그룹에 서로 다른 역할을 할당하면 더 많은 권한을 허용하는 역할이 우선합니다.

8. 필요한 경우 이 사용자에게 하나 이상의 네임스페이스 제약 조건을 할당하고 * 제약 조건으로 역할 제한 * 을 선택하여 해당 제약 조건을 적용합니다. 제약 조건 추가 * 를 선택하여 새 네임스페이스 제약 조건을 추가할 수 있습니다.

   사용자가 LDAP 그룹 구성원 자격을 통해 여러 역할을 할당하면 가장 허용 가능한 역할의 제약 조건만 적용됩니다. 예를 들어, 로컬 뷰어 역할을 가진 사용자가 멤버 역할에 바인딩된 세 개의 그룹에 참여하는 경우 멤버 역할의 제약 조건의 합계가 적용되고 뷰어 역할의 모든 제약 조건은 무시됩니다.

9. 추가 * 를 선택합니다.

결과

새 사용자가 원격 사용자 목록에 나타납니다. 이 목록에서 사용자의 활성 제약 조건을 확인하고 * Actions * 메뉴에서 사용자를 관리할 수 있습니다.

원격 그룹을 추가합니다

한 번에 많은 원격 사용자를 추가하려면 계정 소유자와 관리자가 Astra Control에 원격 그룹을 추가할 수 있습니다. 원격 그룹을 추가하면 해당 그룹의 모든 원격 사용자가 Astra Control에 로그인할 수 있으며 그룹과 동일한 역할을 상속합니다.

Astra Control Center는 최대 5,000개의 LDAP 원격 그룹을 지원합니다.

단계

1. 계정 * 영역으로 이동합니다.
2. 사용자 및 그룹 * 탭을 선택합니다.
3. 페이지 맨 오른쪽에서 * 원격 그룹 * 을 선택합니다.
4. 추가 * 를 선택합니다.

   이 창에서는 Astra Control이 디렉토리에서 검색한 LDAP 그룹의 공통 이름과 고유 이름 목록을 볼 수 있습니다.

5. 선택적으로 * Filter by common name * 필드에 그룹의 공통 이름을 입력하여 LDAP 그룹을 검색합니다.
6. 목록에서 그룹을 하나 이상 선택합니다.
7. 그룹에 역할을 할당합니다.

   선택한 역할은 이 그룹의 모든 사용자에게 할당됩니다. 사용자와 사용자 그룹에 서로 다른 역할을 할당하면 더 많은 권한을 허용하는 역할이 우선합니다.

8. 필요한 경우 이 그룹에 하나 이상의 네임스페이스 제약 조건을 할당하고 * 제약 조건으로 역할 제한 * 을 선택하여 해당 제약 조건을 적용합니다. 제약 조건 추가 * 를 선택하여 새 네임스페이스 제약 조건을 추가할 수 있습니다.

사용자가 LDAP 그룹 구성원 자격을 통해 여러 역할을 할당하면 가장 허용 가능한 역할의 제약 조건만 적용됩니다. 예를 들어, 로컬 뷰어 역할을 가진 사용자가 멤버 역할에 바인딩된 세 개의 그룹에 참여하는 경우 멤버 역할의 제약 조건의 합계가 적용되고 뷰어 역할의 모든 제약 조건은 무시됩니다.

9. 추가 * 를 선택합니다.

결과

원격 그룹 목록에 새 그룹이 나타납니다. 이 그룹의 원격 사용자는 각 원격 사용자가 로그인할 때까지 원격 사용자 목록에 나타나지 않습니다. 이 목록에서 그룹에 대한 세부 정보를 볼 수 있을 뿐 아니라 * Actions * 메뉴에서 그룹을 관리할 수 있습니다.

알림을 보고 관리합니다

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

인터페이스의 오른쪽 상단에서 이러한 알림을 관리할 수 있습니다.

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 kubecononfig 파일을 만드는 경우 해당 파일에 * 하나의 * 컨텍스트 요소만 정의해야 합니다. 을 참조하십시오 "Kubernetes 문서" kubecononfig 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 * 을 선택합니다.
2. 자격 증명 * 탭을 선택합니다.
3. 제거할 자격 증명에 대한 * 상태 * 열의 옵션 메뉴를 선택합니다.
4. 제거 * 를 선택합니다.
5. 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 Kubernetes 클러스터를 관리하고, Astra Control이 Cloud Insights에 연결된 경우, Astra Control은 이벤트 로그를 Cloud Insights로 보냅니다. POD 배포 및 PVC 첨부 파일에 대한 정보를 포함한 로그 정보가 Astra Control Activity 로그에 표시됩니다. 이 정보를 사용하여 관리하고 있는 Kubernetes 클러스터의 문제를 식별할 수 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 Control Center UI 또는 를 사용할 수 있습니다 "Astra Control API를 참조하십시오" 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.

를 참조하십시오

- "Astra Control Center 라이센스"

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 Amazon S3(Simple Storage Service) 버킷 공급자 중 하나를 사용하십시오.

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Microsoft Azure를 참조하십시오

- 일반 S3

AWS(Amazon Web Services) 및 GCP(Google Cloud Platform)는 일반 S3 버킷 유형을 사용합니다.

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 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 지침은 을 참조하십시오 "Astra 자동화 및 API 정보".

버킷 관리와 관련된 다음 작업을 수행할 수 있습니다.

- "버킷을 추가합니다"
- 버킷을 편집합니다
- 기본 버킷을 설정합니다
- 버킷 자격 증명을 회전하거나 제거합니다
- 버킷을 탈거하십시오

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.

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 탐색 창에서 * Bucket * 을 선택합니다.
2. Actions * 열의 메뉴에서 * Edit * 를 선택합니다.
3. 버킷 유형 이외의 모든 정보를 변경합니다.

버킷 유형을 수정할 수 없습니다.

4. Update * 를 선택합니다.

기본 버킷을 설정합니다

클러스터 간에 클론을 수행할 경우 Astra Control에 기본 버킷이 필요합니다. 다음 단계에 따라 모든 클러스터의 기본 버킷을 설정합니다.

단계

1. 클라우드 인스턴스 * 로 이동합니다.
2. 목록에서 클라우드 인스턴스의 * 작업 * 열에 있는 메뉴를 선택합니다.
3. 편집 * 을 선택합니다.
4. Bucket * 목록에서 기본값으로 사용할 버킷을 선택합니다.
5. 저장 * 을 선택합니다.

버킷 자격 증명을 회전하거나 제거합니다

Astra Control은 버킷 자격 증명을 사용하여 액세스 권한을 얻고 S3 버킷에 대한 비밀 키를 제공하여 Astra Control Center가 버킷과 통신할 수 있도록 합니다.

버킷 자격 증명을 회전합니다

자격 증명을 회전하는 경우 백업이 진행 중인 상태(예약 또는 필요 시)가 없을 때 유지 관리 창에서 자격 증명을 회전합니다.

자격 증명을 편집하고 회전하는 단계입니다

1. 왼쪽 탐색 창에서 * Bucket * 을 선택합니다.
2. Actions * 열의 Options 메뉴에서 * Edit * 를 선택합니다.
3. 새 자격 증명을 생성합니다.
4. Update * 를 선택합니다.

버킷 자격 증명을 제거합니다

버킷에 새 자격 증명이 적용된 경우 또는 버킷이 더 이상 사용되지 않는 경우에만 버킷 자격 증명을 제거해야 합니다.

Astra Control에 추가하는 첫 번째 자격 증명 세트는 항상 사용 중입니다. Astra Control은 자격 증명을 사용하여 백업 버킷을 인증하기 때문입니다. 버킷이 사용 중인 경우 이러한 자격 증명을 제거하지 마십시오. 이 경우 백업 실패 및 백업 가용성 손실이 발생할 수 있습니다.

활성 버킷 자격 증명을 제거하는 경우 를 참조하십시오 "버킷 자격 증명 제거 문제 해결".

Astra Control API를 사용하여 S3 자격 증명을 제거하는 방법에 대한 지침은 을 참조하십시오 "Astra 자동화 및 API 정보".

버킷을 탈거하십시오

더 이상 사용하지 않거나 상태가 불량한 버킷을 제거할 수 있습니다. 오브젝트 저장소 구성을 단순하고 최신 상태로 유지하기 위해 이 작업을 수행할 수 있습니다.

기본 버킷을 제거할 수 없습니다. 해당 버킷을 제거하려면 먼저 다른 버킷을 기본값으로 선택하십시오.

시작하기 전에

- 시작하기 전에 이 버킷에 대해 실행 중이거나 완료된 백업이 없는지 확인해야 합니다.
- 버킷이 활성 보호 정책에서 사용되고 있지 않은지 확인해야 합니다.

있는 경우 계속할 수 없습니다.

단계

1. 왼쪽 탐색에서 * Bucket * 을 선택합니다.
2. Actions * 메뉴에서 * Remove * 를 선택합니다.

   Astra Control은 먼저 버킷에 백업을 사용하는 스케줄 정책이 없고 제거할 버킷에 활성 백업이 없음을 보장합니다.

3. 작업을 확인하려면 "remove"를 입력합니다.
4. 예, 버킷 제거 * 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 백엔드 인증 세부 정보를 편집합니다
- 검색된 스토리지 백엔드를 관리합니다
- 스토리지 백엔드의 관리를 취소합니다

- 스토리지 백엔드를 제거합니다

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 대시보드의 스토리지 백엔드 패널을 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

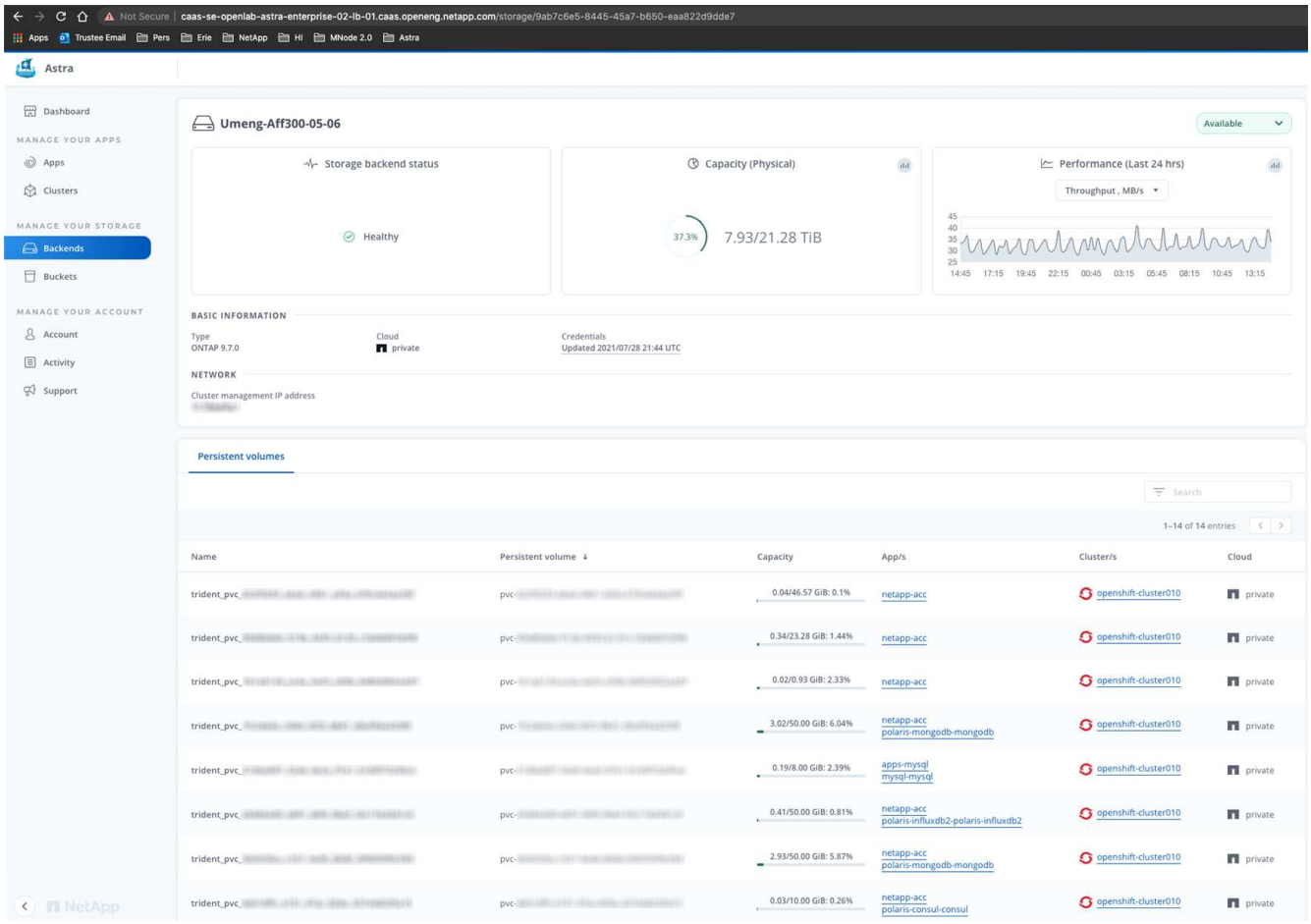
백엔드 상태, 용량 및 성능(IOPS 처리량 및/또는 지연 시간)에 대한 정보를 봅니다.

Kubernetes 앱이 사용 중인 볼륨을 볼 수 있습니다. 볼륨은 선택한 스토리지 백엔드에 저장됩니다. Cloud Insights를 사용하면 추가 정보를 볼 수 있습니다. 을 참조하십시오 "Cloud Insights 설명서".

단계

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

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



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

스토리지 백엔드 인증 세부 정보를 편집합니다

Astra Control Center는 ONTAP 백엔드를 인증하는 두 가지 모드를 제공합니다.

- * 자격 증명 기반 인증 *: 필요한 권한이 있는 ONTAP 사용자의 사용자 이름 및 암호입니다. ONTAP 버전과의 호환성을 최대화하려면 admin과 같이 미리 정의된 보안 로그인 역할을 사용해야 합니다.
- * 인증서 기반 인증 *: Astra Control Center는 백엔드에 설치된 인증서를 사용하여 ONTAP 클러스터와 통신할 수도 있습니다. 클라이언트 인증서, 키 및 신뢰할 수 있는 CA 인증서를 사용해야 합니다(권장).

기존 백엔드를 업데이트하여 한 가지 인증 유형에서 다른 방법으로 이동할 수 있습니다. 한 번에 하나의 인증 방법만 지원됩니다.

인증서 기반 인증 활성화에 대한 자세한 내용은 을 참조하십시오 "ONTAP 스토리지 백엔드에서 인증을 설정합니다".

단계

1. 왼쪽 탐색에서 * backends * 를 선택합니다.
2. 스토리지 백엔드를 선택합니다.
3. 자격 증명 필드에서 * 편집 * 아이콘을 선택합니다.
4. 편집 페이지에서 다음 중 하나를 선택합니다.
    - * 관리자 자격 증명 사용 *: ONTAP 클러스터 관리 IP 주소와 관리 자격 증명을 입력합니다. 자격 증명은

클러스터 전체의 자격 증명이어야 합니다.

> 여기에 자격 증명을 입력한 사용자에게는 가 있어야 합니다 `ontapi` ONTAP 클러스터의 ONTAP System Manager에서 활성화된 사용자 로그인 액세스 방법입니다. SnapMirror 복제를 사용하려는 경우 액세스 방법이 있는 "admin" 역할의 사용자 자격 증명을 적용하십시오 `ontapi` 및 `http`, 소스 및 대상 ONTAP 클러스터 모두에서. 을 참조하십시오 "ONTAP 설명서에서 사용자 계정을 관리합니다" 를 참조하십시오.

- * 인증서 사용 *: 인증서를 업로드합니다 `.pem` 파일, 인증서 키입니다 `.key` 파일 및 인증 기관 파일(옵션)을 선택합니다.

5. 저장 * 을 선택합니다.

검색된 스토리지 백엔드를 관리합니다

관리되지 않지만 검색된 스토리지 백엔드를 관리하도록 선택할 수 있습니다. 스토리지 백엔드를 관리할 때 Astra Control은 인증 인증서가 만료되었는지 여부를 나타냅니다.

단계

1. 왼쪽 탐색에서 * backends * 를 선택합니다.
2. 검색된 * 옵션을 선택합니다.
3. 스토리지 백엔드를 선택합니다.
4. Actions * 열의 Options 메뉴에서 * Manage * 를 선택합니다.
5. 변경 사항을 적용합니다.
6. 저장 * 을 선택합니다.

스토리지 백엔드의 관리를 취소합니다

백엔드의 관리를 해제할 수 있습니다.

단계

1. 왼쪽 탐색에서 * backends * 를 선택합니다.
2. 스토리지 백엔드를 선택합니다.
3. Actions * 열의 Options 메뉴에서 * Unmanage * 를 선택합니다.
4. "unmanage"를 입력하여 작업을 확인합니다.
5. Yes, unmanage storage backend * 를 선택합니다.

스토리지 백엔드를 제거합니다

더 이상 사용되지 않는 스토리지 백엔드를 제거할 수 있습니다. 구성을 간단하고 최신 상태로 유지하기 위해 이 작업을 수행할 수 있습니다.

시작하기 전에

- 스토리지 백엔드가 관리되지 않는 상태인지 확인합니다.
- 스토리지 백엔드에 클러스터와 연결된 볼륨이 없는지 확인합니다.

단계

1. 왼쪽 탐색에서 * backends * 를 선택합니다.
2. 백엔드가 관리되는 경우 관리를 해제합니다.
   a. Managed * 를 선택합니다.
   b. 스토리지 백엔드를 선택합니다.
   c. Actions * 옵션에서 * Unmanage * 를 선택합니다.
   d. "unmanage"를 입력하여 작업을 확인합니다.
   e. Yes, unmanage storage backend * 를 선택합니다.
3. 검색된 * 를 선택합니다.
   a. 스토리지 백엔드를 선택합니다.
   b. Actions * 옵션에서 * Remove * 를 선택합니다.
   c. 작업을 확인하려면 "remove"를 입력합니다.
   d. Yes, remove storage backend * 를 선택합니다.

자세한 내용을 확인하십시오

- "Astra Control API를 사용합니다"

실행 중인 작업을 모니터링합니다

지난 24시간 동안 Astra Control에서 완료, 실패 또는 취소된 작업 및 실행 작업에 대한 세부 정보를 볼 수 있습니다. 예를 들어 실행 중인 백업, 복원 또는 클론 작업의 상태를 보고 완료율 및 남은 예상 시간과 같은 세부 정보를 볼 수 있습니다. 가 실행된 예약된 작업의 상태 또는 수동으로 시작한 작업을 볼 수 있습니다.

실행 중이거나 완료된 작업을 보는 동안 작업 세부 정보를 확장하여 각 하위 작업의 상태를 볼 수 있습니다. 진행 중이거나 완료된 작업의 경우 작업 진행률 표시줄이 녹색이고, 취소된 작업의 경우 파란색이고, 오류로 인해 실패한 작업의 경우 빨간색입니다.

클론 작업의 경우 작업 하위 작업은 스냅샷과 스냅샷 복구 작업으로 구성됩니다.

실패한 작업에 대한 자세한 내용은 을 참조하십시오 "계정 활동을 모니터링합니다".

단계

1. 작업을 실행하는 동안 * 응용 프로그램 * 으로 이동합니다.
2. 목록에서 응용 프로그램의 이름을 선택합니다.
3. 응용 프로그램의 세부 정보에서 * 작업 * 탭을 선택합니다.

   현재 또는 과거 작업의 세부 정보를 보고 작업 상태별로 필터링할 수 있습니다.

태스크는 최대 24시간 동안 * 작업 * 목록에 유지됩니다. 을 사용하여 이 제한 및 기타 작업 모니터 설정을 구성할 수 있습니다 "Astra Control API를 참조하십시오".

Cloud Insights, Prometheus 또는 Fluentd 연결을 통해 인프라를 모니터링합니다

Astra Control Center 환경을 향상시키기 위해 몇 가지 선택적 설정을 구성할 수 있습니다. 전체 인프라를 모니터링하고 통찰력을 얻기 위해 NetApp Cloud Insights에 대한 연결을 생성하거나 Prometheus를 구성하거나 Fluentd 연결을 추가합니다.

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

- Cloud Insights에 연결합니다
- Prometheus에 연결하세요
- Fluentd에 연결합니다

Cloud Insights 또는 NetApp Support 사이트에 연결할 프록시 서버를 추가합니다

Astra Control Center를 실행 중인 네트워크에 인터넷에 연결하기 위한 프록시가 필요한 경우(NetApp Support 사이트에 지원 번들을 업로드하거나 Cloud Insights에 연결을 설정하려면), 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.

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.

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 Insights을 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.

Astra Control Center가 Cloud Insights와 페어링된 경우 Cloud Insights에서 * 배포 수정 * 옵션을 사용하면 안 됩니다.

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

시작하기 전에

- 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.

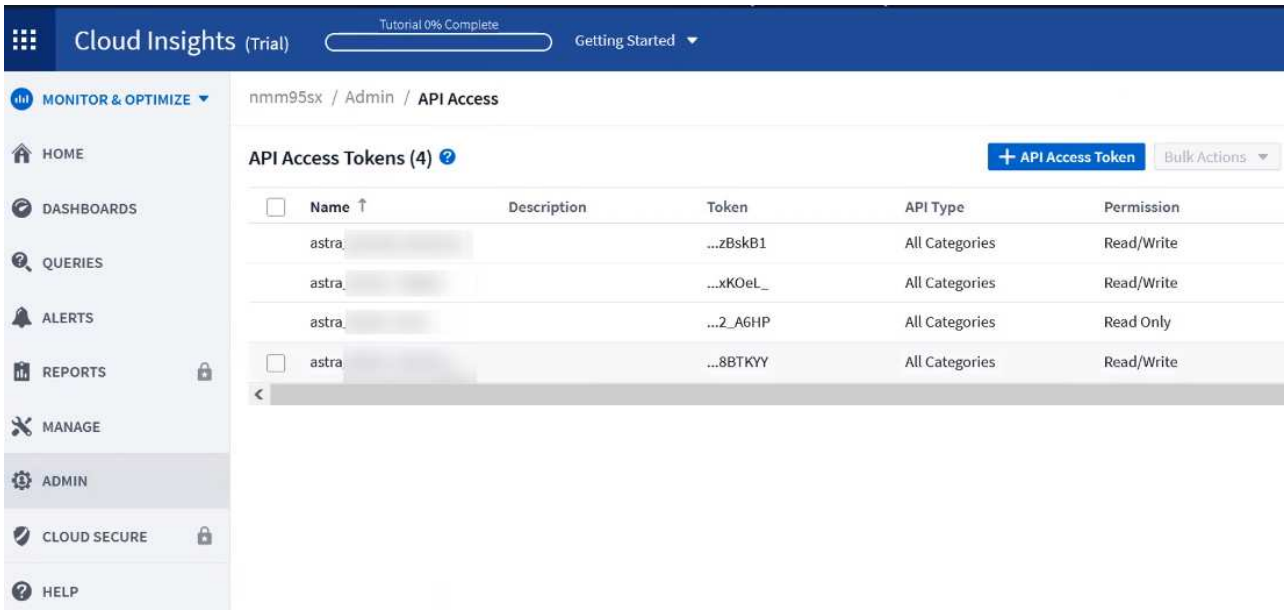
4. Cloud Insights API 토큰 및 테넌트 URL을 입력합니다. 테넌트 URL의 형식은 다음과 같습니다.

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

Cloud Insights 라이센스가 있으면 테넌트 URL을 가져옵니다. 테넌트 URL이 없는 경우 을 참조하십시오 "Cloud Insights 설명서".

a. 를 다운로드하십시오 "API 토큰"에서 Cloud Insights 테넌트 URL에 로그인합니다.

b. Cloud Insights에서 * 관리자 * > * API 액세스 * 를 클릭하여 * 읽기/쓰기 * 와 * 읽기 전용 * API 액세스 토큰을 모두 생성합니다.



c. 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.

d. 읽기/쓰기 * 키를 복사합니다. Astra Control Center * Connect Cloud Insights * 창에 붙여 넣어야 합니다. 읽기/쓰기 API 액세스 토큰 키 권한에 대해 데이터 수집, 로그 수집, 획득 장치 및 데이터 수집 을 선택합니다.

> 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.

e. Cloud Insights에서 복사한 키를 Astra Control Center에 붙여 넣습니다.

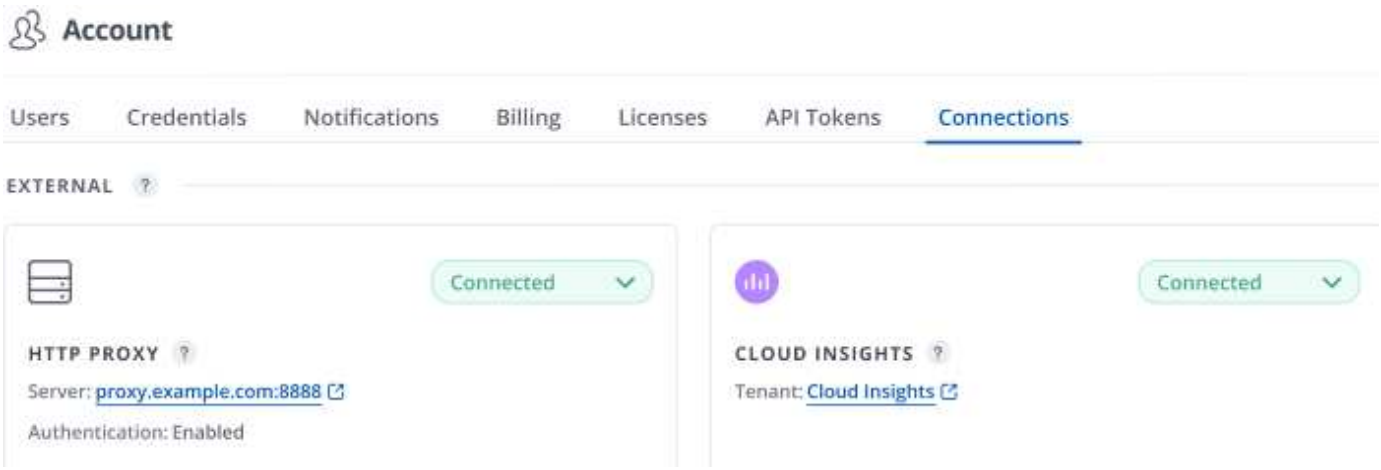
5. Connect * 를 선택합니다.

> 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.

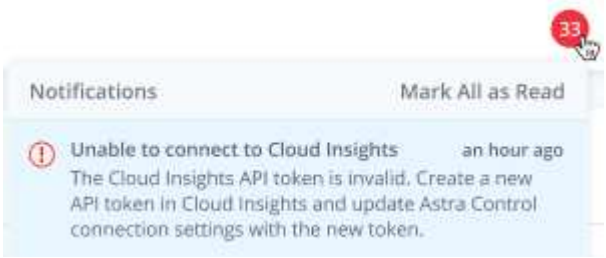
> Astra Control Center와 Cloud Insights UI 사이를 쉽게 오갈 수 있도록 두 가지 모두에 로그인했는지 확인하십시오.

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.

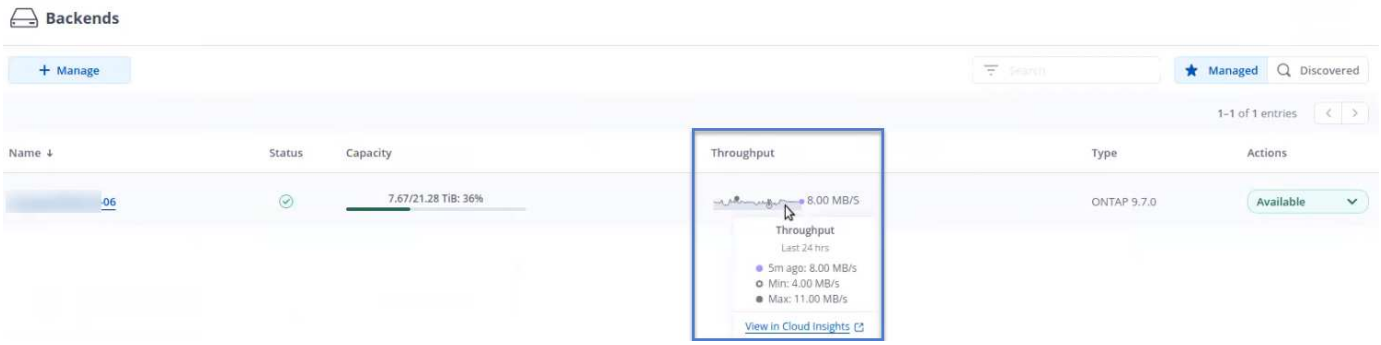


어떤 이유로 연결에 실패한 경우 상태가 * 실패 * 로 표시됩니다. UI 오른쪽 상단의 * 알림 * 에서 실패 원인을 찾을 수 있습니다.



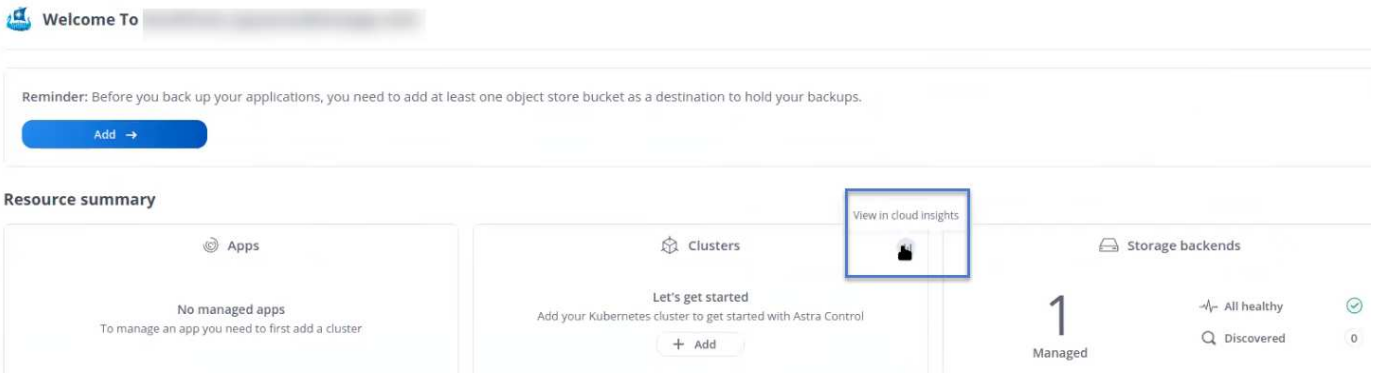
계정 * > * 알림 * 에서 동일한 정보를 찾을 수도 있습니다.

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.

Prometheus에 연결하세요

Prometheus로 Astra Control Center 데이터를 모니터링할 수 있습니다. Kubernetes 클러스터 메트릭 엔드포인트에서 메트릭을 수집하도록 Prometheus를 구성할 수 있으며 Prometheus를 사용하여 메트릭 데이터를 시각화할 수도 있습니다.

Prometheus 사용에 대한 자세한 내용은 에서 해당 설명서를 참조하십시오 "Prometheus 시작".

필요한 것

Astra Control Center 클러스터나 Astra Control Center 클러스터와 통신할 수 있는 다른 클러스터에 Prometheus 패키지를 다운로드하여 설치했는지 확인하십시오.

의 공식 설명서에 있는 지침을 따르십시오 "Prometheus를 설치합니다".

Prometheus는 Astra Control Center Kubernetes 클러스터와 통신할 수 있어야 합니다. Prometheus가 Astra Control Center 클러스터에 설치되어 있지 않은 경우 Astra Control Center 클러스터에서 실행 중인 메트릭 서비스와 통신할 수 있는지 확인해야 합니다.

Prometheus를 구성합니다

Astra Control Center는 Kubernetes 클러스터의 TCP 포트 9090에 메트릭 서비스를 제공합니다. 이 서비스에서 메트릭을 수집하려면 Prometheus를 구성해야 합니다.

단계

1. Prometheus 서버에 로그인합니다.
2. 에 클러스터 항목을 추가합니다 `prometheus.yml` 파일. 에 있습니다 `yml` 파일에서 의 클러스터에 대해 다음과 유사한 항목을 추가합니다 `scrape_configs section`:

```
job_name: '<Add your cluster name here. You can abbreviate. It just
needs to be a unique name>'
  metrics_path: /accounts/<replace with your account ID>/metrics
  authorization:
     credentials: <replace with your API token>
  tls_config:
     insecure_skip_verify: true
  static_configs:
    - targets: ['<replace with your astraAddress. If using FQDN, the
prometheus server has to be able to resolve it>']
```

> 를 설정하는 경우 `tls_config insecure_skip_verify` 를 선택합니다 `true`, TLS 암호화 프로토콜이 필요하지 않습니다.

3. Prometheus 서비스를 다시 시작합니다.

```
sudo systemctl restart prometheus
```

Prometheus에 액세스하십시오

Prometheus URL에 액세스합니다.

단계

1. 브라우저에서 포트 9090이 있는 Prometheus URL을 입력합니다.
2. 상태 * > * 대상 * 을 선택하여 연결을 확인합니다.

Prometheus에서 데이터를 봅니다

Prometheus를 사용하여 Astra Control Center 데이터를 볼 수 있습니다.

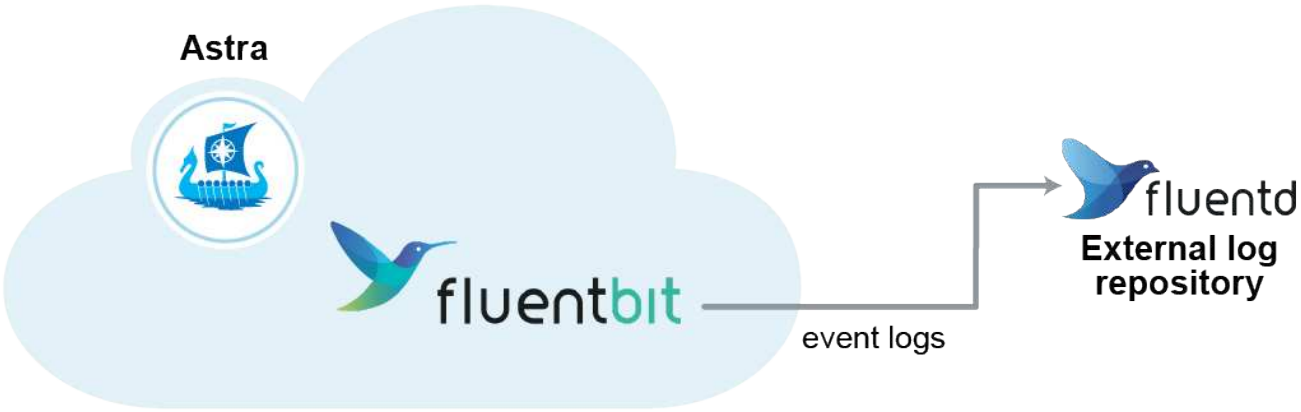
단계

1. 브라우저에 Prometheus URL을 입력합니다.
2. Prometheus 메뉴에서 * Graph * 를 선택합니다.
3. 메트릭 탐색기를 사용하려면 * Execute * 옆에 있는 아이콘을 선택합니다.
4. 를 선택합니다 `scrape_samples_scraped` 를 선택하고 * 실행 * 을 선택합니다.
5. 시간에 따른 샘플 스크레핑을 보려면 * Graph * 를 선택합니다.

여러 클러스터 데이터가 수집되면 각 클러스터의 메트릭이 서로 다른 색으로 표시됩니다.

Fluentd에 연결합니다

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



관리되는 클러스터의 이벤트 로그만 Fluentd로 전달됩니다.

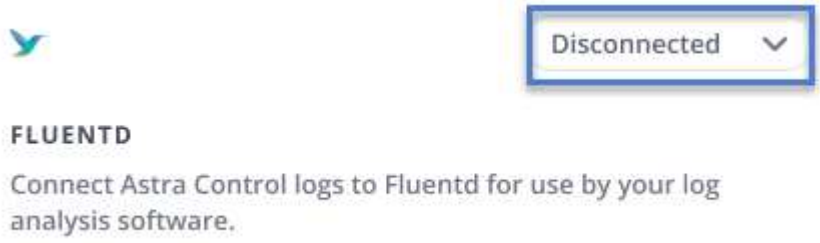
시작하기 전에

- 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.



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 에서 로그를 사용할 수 있는지 확인해야 합니다 `/var/log/containers/`.

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.

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.

앱 관리를 취소하는 경우:

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

단계

1. 왼쪽 탐색 모음에서 * 응용 프로그램 * 을 선택합니다.
2. 앱을 선택합니다.
3. 작업 열의 옵션 메뉴에서 * 관리 취소 * 를 선택합니다.
4. 정보를 검토합니다.
5. "unmanage"를 입력하여 확인합니다.
6. 예, 응용 프로그램 관리 취소 * 를 선택합니다.

결과

Astra Control Center가 앱 관리를 중지합니다.

클러스터 관리를 취소합니다

Astra Control Center에서 더 이상 관리하지 않으려는 클러스터 관리를 중지합니다.

클러스터를 관리하기 전에 클러스터와 연결된 앱의 관리를 해제해야 합니다.

클러스터 관리를 취소하는 경우:

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

단계

1. 왼쪽 탐색 모음에서 * 클러스터 * 를 선택합니다.
2. 더 이상 관리하지 않으려는 클러스터의 확인란을 선택합니다.
3. Actions * 열의 Options 메뉴에서 * Unmanage * 를 선택합니다.

4. 클러스터 관리를 해제할지 확인한 다음 * 예, 클러스터 관리 취소 * 를 선택합니다.

결과

클러스터의 상태가 * Removing * 으로 변경됩니다. 그 이후에는 클러스터가 * Clusters * 페이지에서 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 연결된 경우 * 클러스터를 관리하지 않으면 만 삭제됩니다 `fluentbit` 및 `event-exporter` Pod를 클릭합니다.

Astra Control Center를 업그레이드합니다

Astra Control Center를 업그레이드하려면 NetApp Support 사이트에서 설치 번들을 다운로드하고 다음 지침을 완료하십시오. 이 절차를 사용하여 인터넷에 연결되거나 공기가 연결된 환경에서 Astra Control Center를 업그레이드할 수 있습니다.

시작하기 전에

업그레이드하기 전에 환경이 에 맞는지 확인하십시오 "Astra Control Center 구축을 위한 최소 요구 사항". 환경에 다음이 있어야 합니다.

- 지원되는 Astra Trident 버전

  실행 중인 Trident 버전 확인:

  ```
  kubectl get tridentversion -n trident
  ```

  을 참조하십시오 "Astra Trident 문서" 이전 버전에서 업그레이드하는 경우.

  > Kubernetes 1.25로 업그레이드하기 전에 * Astra Trident 22.10 * 으로 업그레이드해야 합니다.

- 지원되는 Kubernetes 배포

  실행 중인 Kubernetes 버전 확인:

  ```
  kubectl get nodes -o wide
  ```

- 충분한 클러스터 리소스

사용 가능한 클러스터 리소스 결정:

```
kubectl describe node <node name>
```

- Astra Control Center 이미지를 푸시 및 업로드하는 데 사용할 수 있는 레지스트리입니다
- 기본 스토리지 클래스입니다

기본 스토리지 클래스 확인:

```
kubectl get storageclass
```

- 사용 가능한 정상 API 서비스

모든 API 서비스가 정상 상태이며 사용 가능한지 확인합니다.

```
kubectl get apiservices
```

- (OpenShift에만 해당) 사용 가능한 정상 클러스터 운영자

모든 클러스터 운영자가 양호한 상태이며 사용 가능한지 확인합니다.

```
kubectl get clusteroperators
```

이 절차의 일부로 해야 합니다 Astra Control Center를 업그레이드하는 경우 이 업데이트된 연산자를 사용하여 이전 버전의 Astra Control Center로 업그레이드할 수 없습니다.

이 작업에 대해

Astra Control Center 업그레이드 프로세스는 다음과 같은 고급 단계를 안내합니다.

업그레이드를 시작하기 전에 Astra Control Center UI에서 로그아웃하십시오.

- Astra Control Center를 다운로드하고 압축을 풉니다
- NetApp Astra kubtl 플러그인을 제거하고 다시 설치합니다
- 이미지를 로컬 레지스트리에 추가합니다

- 업데이트된 Astra Control Center 운영자를 설치합니다
- Astra Control Center를 업그레이드합니다
- 시스템 상태를 확인합니다

Astra Control Center 운영자를 삭제하지 마십시오(예: `kubectl delete -f astra_control_center_operator_deploy.yaml`) 포드가 삭제되지 않도록 Astra Control Center 업그레이드 또는 작업 중 언제든지.

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

Astra Control Center를 다운로드하고 압축을 풉니다

1. 로 이동합니다 "Astra Control Center 제품 다운로드 페이지" 를 방문하십시오. 드롭다운 메뉴에서 최신 버전 또는 원하는 다른 버전을 선택할 수 있습니다.
2. (권장되지만 선택 사항) Astra Control Center용 인증서 및 서명 번들을 다운로드합니다 (`astra-control-center-certs-[version].tar.gz`)를 클릭하여 번들 서명을 확인합니다.

   자세한 내용을 보려면 를 확장합니다

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

   ```
   openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
   -signature certs/astra-control-center-[version].tar.gz.sig astra-
   control-center-[version].tar.gz
   ```

   출력이 표시됩니다 `Verified OK` 확인 성공 후.

3. Astra Control Center 번들에서 이미지를 추출합니다.

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

NetApp Astra kubtl 플러그인을 제거하고 다시 설치합니다

NetApp Astra kubtl 명령줄 플러그인을 사용하여 이미지를 로컬 Docker 저장소로 푸시할 수 있습니다.

1. 플러그인이 설치되어 있는지 확인합니다.

   ```
   kubectl astra
   ```

2. 다음 작업 중 하나를 수행합니다.

- 플러그인이 설치되어 있는 경우 kubeck 플러그인 도움말을 반환해야 하며 kubbctl-Astra의 기존 버전을 제거할 수 있습니다. `delete /usr/local/bin/kubectl-astra`.
- 명령이 오류를 반환하면 플러그인이 설치되지 않은 것이므로 다음 단계를 수행하여 설치할 수 있습니다.

3. 플러그인 설치:

   a. 사용 가능한 NetApp Astra kubectl 플러그인 바이너리를 나열하고 운영 체제 및 CPU 아키텍처에 필요한 파일 이름을 적어 주십시오.

   > kubbeck 플러그인 라이브러리는 tar 번들의 일부이며 폴더에 압축이 풀립니다 `kubectl-astra`.

```
ls kubectl-astra/
```

   a. 올바른 바이너리를 현재 경로로 이동하고 이름을 로 변경합니다 `kubectl-astra`:

```
cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
```

이미지를 로컬 레지스트리에 추가합니다

1. 용기 엔진에 적합한 단계 시퀀스를 완료합니다.

Docker 를 참조하십시오

1. 타볼의 루트 디렉토리로 변경합니다. 가 표시됩니다 `acc.manifest.bundle.yaml` 파일 및 다음 디렉토리:

   ```
   acc/
   kubectl-astra/
   acc.manifest.bundle.yaml
   ```

2. Astra Control Center 이미지 디렉토리의 패키지 이미지를 로컬 레지스트리에 밀어 넣습니다. 를 실행하기 전에 다음 대체 작업을 수행합니다 `push-images` 명령:
   - <BUNDLE_FILE>를 Astra Control 번들 파일의 이름으로 바꿉니다 (`acc.manifest.bundle.yaml`)를 클릭합니다.
   - <MY_FULL_REGISTRY_PATH>를 Docker 저장소의 URL로 바꿉니다. 예를 들어, "<a href="https://<docker-registry>"" class="bare">https://<docker-registry>".
   - <MY_REGISTRY_USER>를 사용자 이름으로 바꿉니다.
   - <MY_REGISTRY_TOKEN>를 레지스트리에 대한 인증된 토큰으로 바꿉니다.

     ```
     kubectl astra packages push-images -m <BUNDLE_FILE> -r
     <MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
     <MY_REGISTRY_TOKEN>
     ```

팟맨

1. 타볼의 루트 디렉토리로 변경합니다. 이 파일과 디렉토리가 표시됩니다.

   ```
   acc.manifest.bundle.yaml
   acc/
   ```

2. 레지스트리에 로그인합니다.

   ```
   podman login <YOUR_REGISTRY>
   ```

3. 사용하는 Podman 버전에 맞게 사용자 지정된 다음 스크립트 중 하나를 준비하고 실행합니다. <MY_FULL_REGISTRY_PATH>를 모든 하위 디렉토리가 포함된 리포지토리의 URL로 대체합니다.

   ```
   <strong>Podman 4</strong>
   ```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.07.0-25
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.07.0-25
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

레지스트리 구성에 따라 스크립트가 만드는 이미지 경로는 다음과 같아야 합니다.

```
https://netappdownloads.jfrog.io/docker-astra-control-
prod/netapp/astra/acc/23.07.0-25/image:version
```

업데이트된 **Astra Control Center** 운영자를 설치합니다

1. 디렉토리를 변경합니다.

```
cd manifests
```

2. Astra Control Center 운영자 배포 YAML을 편집합니다 (`astra_control_center_operator_deploy.yaml`)를 클릭하여 로컬 레지스트리 및 암호를 참조합니다.

```
vim astra_control_center_operator_deploy.yaml
```

   a. 인증이 필요한 레지스트리를 사용하는 경우 의 기본 줄을 바꾸거나 편집합니다 `imagePullSecrets: []` 다음 포함:

```
imagePullSecrets: [{name: astra-registry-cred}]
```

   b. 변경 `ASTRA_IMAGE_REGISTRY` 의 경우 `kube-rbac-proxy` 이미지를 에서 푸시한 레지스트리 경로로 이미지 이전 단계.

   c. 변경 `ASTRA_IMAGE_REGISTRY` 의 경우 `acc-operator` 이미지를 에서 푸시한 레지스트리 경로로 이미지 이전 단계.

   d. 에 다음 값을 추가합니다 `env` 섹션:

```
- name: ACCOP_HELM_UPGRADETIMEOUT
  value: 300m
```

샘플 **Astra_control_center_operator_deploy.YAML:**

```
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
  strategy:
    type: Recreate
  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: ASTRA_IMAGE_REGISTRY/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
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_UPGRADETIMEOUT
          value: 300m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:23.07.25
        imagePullPolicy: IfNotPresent
        livenessProbe:
          httpGet:
```

```
            path: /healthz
            port: 8081
          initialDelaySeconds: 15
          periodSeconds: 20
        name: manager
        readinessProbe:
          httpGet:
            path: /readyz
            port: 8081
          initialDelaySeconds: 5
          periodSeconds: 10
        resources:
          limits:
            cpu: 300m
            memory: 750Mi
          requests:
            cpu: 100m
            memory: 75Mi
        securityContext:
          allowPrivilegeEscalation: false
      imagePullSecrets: []
      securityContext:
        runAsUser: 65532
      terminationGracePeriodSeconds: 10
```

3. 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.as
tra.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
```

4. Pod가 실행 중인지 확인합니다.

```
kubectl get pods -n netapp-acc-operator
```

Astra Control Center를 업그레이드합니다

1. Astra Control Center 사용자 지정 리소스(CR) 편집:

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

2. Astra 버전 번호를 변경합니다 (`astraVersion` 의 내부 `spec`)를 업그레이드할 버전:

```
spec:
  accountName: "Example"
  astraVersion: "[Version number]"
```

3. 이미지 레지스트리 경로가 에서 이미지를 푸시한 레지스트리 경로와 일치하는지 확인합니다 이전 단계. 업데이트 `imageRegistry` 의 내부 `spec` 마지막 설치 이후 레지스트리가 변경된 경우

```
  imageRegistry:
    name: "[your_registry_path]"
```

4. 에 다음을 추가합니다 `crds` 의 내부 구성 `spec`:

```
crds:
  shouldUpgrade: true
```

5. 에 다음 행을 추가합니다 `additionalValues` 의 내부 `spec` Astra Control Center CR에서 다음을 수행합니다.

```
additionalValues:
    nautilus:
      startupProbe:
        periodSeconds: 30
        failureThreshold: 600
    polaris-keycloak:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
```

6. 파일 편집기를 저장하고 종료합니다. 변경 사항이 적용되고 업그레이드가 시작됩니다.
7. (선택 사항) Pod가 종료되어 다시 사용할 수 있는지 확인합니다.

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

8. Astra Control 상태 조건이 업그레이드가 완료되어 준비되었음을 나타낼 때까지 기다립니다 (`True`):

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

응답:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f   23.07.0-25
10.111.111.111  True
```

작업 중에 업그레이드 상태를 모니터링하려면 다음 명령을 실행합니다. `kubectl get AstraControlCenter -o yaml -n [netapp-acc or custom namespace]`

Astra Control Center 운영자 로그를 검사하려면 다음 명령을 실행하십시오.
`kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-operator -c manager -f`

시스템 상태를 확인합니다

1. Astra Control Center에 로그인합니다.
2. 버전이 업그레이드되었는지 확인합니다. UI의 * 지원 * 페이지를 참조하십시오.
3. 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.

설치 제거에 문제가 있는 경우 를 참조하십시오 제거 문제 해결.

시작하기 전에

1. "모든 앱 관리를 취소합니다" 클러스터에서.
2. "모든 클러스터의 관리를 취소합니다".

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.yaml -n netapp-acc
```

결과:

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

2. 다음 명령을 사용하여 를 삭제합니다 `netapp-acc` (또는 사용자 지정 이름) 네임스페이스:

```
kubectl delete ns [netapp-acc or custom namespace]
```

예제 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
```

제거 문제 해결

다음 해결 방법을 사용하여 Astra Control Center를 제거할 때 발생하는 문제를 해결하십시오.

Astra Control Center를 제거해도 관리 클러스터의 모니터링 운영자 포드가 정리되지 않습니다

Astra Control Center를 제거하기 전에 클러스터를 관리하지 않았다면 NetApp 모니터링 네임스페이스 및 네임스페이스에서 Pod를 수동으로 삭제할 수 있습니다. 이러한 명령은 다음과 같습니다.

단계

1. 삭제 `acc-monitoring` 에이전트:

```
kubectl delete agents acc-monitoring -n netapp-monitoring
```

결과:

```
agent.monitoring.netapp.com "acc-monitoring" deleted
```

2. 네임스페이스 삭제:

```
kubectl delete ns netapp-monitoring
```

결과:

```
namespace "netapp-monitoring" deleted
```

3. 제거된 리소스 확인:

```
kubectl get pods -n netapp-monitoring
```

결과:

```
No resources found in netapp-monitoring namespace.
```

4. 모니터링 에이전트 제거 확인:

```
kubectl get crd|grep agent
```

샘플 결과:

```
agents.monitoring.netapp.com                    2021-07-21T06:08:13Z
```

5. 사용자 정의 리소스 정의(CRD) 정보 삭제:

```
kubectl delete crds agents.monitoring.netapp.com
```

결과:

```
customresourcedefinition.apiextensions.k8s.io
"agents.monitoring.netapp.com" deleted
```

Astra Control Center를 제거해도 **Traefik CRD**가 정리되지 않습니다

Traefik CRD를 수동으로 삭제할 수 있습니다. CRD는 글로벌 리소스이며 CRD를 삭제하면 클러스터의 다른 애플리케이션에 영향을 줄 수 있습니다.

단계

1. 클러스터에 설치된 Traefik CRD 나열:

```
kubectl get crds |grep -E 'traefik'
```

응답

```
ingressroutes.traefik.containo.us             2021-06-23T23:29:11Z
ingressroutetcps.traefik.containo.us          2021-06-23T23:29:11Z
ingressrouteudps.traefik.containo.us          2021-06-23T23:29:12Z
middlewares.traefik.containo.us               2021-06-23T23:29:12Z
middlewaretcps.traefik.containo.us            2021-06-23T23:29:12Z
serverstransports.traefik.containo.us         2021-06-23T23:29:13Z
tlsoptions.traefik.containo.us                2021-06-23T23:29:13Z
tlsstores.traefik.containo.us                 2021-06-23T23:29:14Z
traefikservices.traefik.containo.us           2021-06-23T23:29:15Z
```

2. CRD 삭제:

```
kubectl delete crd ingressroutes.traefik.containo.us
ingressroutetcps.traefik.containo.us
ingressrouteudps.traefik.containo.us middlewares.traefik.containo.us
serverstransports.traefik.containo.us tlsoptions.traefik.containo.us
tlsstores.traefik.containo.us traefikservices.traefik.containo.us
middlewaretcps.traefik.containo.us
```

자세한 내용을 확인하십시오

- "제거 관련 알려진 문제입니다"