Astra Control Center 22.08ドキュメント

Astra Control Center

NetApp
November 21, 2023

This PDF was generated from https://docs.netapp.com/ja-jp/astra-control-center-2208/index.html on November 21, 2023. Always check docs.netapp.com for the latest.

目次

Astra Control Center 22.08ドキュメント 1
リリースノート 2
- このリリースの Astra Control Center の新機能 2
- 既知の問題 4
- 既知の制限 8

概念 12
- Astra Controlの詳細をご確認ください 12
- アーキテクチャとコンポーネント 15
- データ保護 17
- ライセンス 20
- アプリケーション管理について理解する 21
- ストレージクラスと永続的ボリュームサイズ 22
- ユーザロールとネームスペース 23

はじめに 25
- Astra Control Center の要件 25
- Astra Control Center のクイックスタート 32
- インストールの概要 33
- Astra Control Center をセットアップします 82
- Astra Control Center に関するよくある質問 102

Astra を使用する 105
- アプリの管理を開始します 105
- アプリを保護します 109
- アプリケーションとクラスタの健常性を監視 143
- アカウントを管理します 146
- バケットを管理する 157
- ストレージバックエンドを管理します 160
- Cloud Insights 接続とFluentd接続でインフラストラクチャを監視します 166
- アプリケーションとクラスタの管理を解除します 173
- Astra Control Center をアップグレードします 174
- Astra Control Center をアンインストールします 187

REST API で自動化 191
- Astra Control REST API による自動化 191

知識とサポート 192
- トラブルシューティング 192
- ヘルプを表示します 192

以前のバージョンの Astra Control Center ドキュメント 195

法的通知 196
- 著作権 196
- 商標 196

特許 196
プライバシーポリシー 196
オープンソース 196
Astra Control API ライセンス 196

Astra Control Center 22.08ドキュメント

リリースノート

最新リリースのAstra Control Centerを発表しました。

- "このリリースの Astra Control Center の内容"
- "既知の問題"
- "Astra Data Store およびこの Astra Control Center リリースでの既知の問題"
- "既知の制限"

Twitter で @NetAppDoc をフォローしてください。を作成し、ドキュメントに関するフィードバックを送信します "GitHub の貢献者" または、 doccomments@netapp.com に電子メールを送信します。

このリリースの **Astra Control Center** の新機能

最新リリースのAstra Control Centerを発表しました。

2022年9月8日（22.08.1）

このパッチリリース（22.08.1）for Astra Control Center（22.08.0）では、NetApp SnapMirrorを使用したアプリケーションレプリケーションの小さなバグが修正されています。

2022年8月10日（22.08.0）

新機能とサポート

- "NetApp SnapMirrorテクノロジを使用したアプリケーションのレプリケーション"
- "アプリ管理ワークフローの改善"
- "拡張された独自の実行フック機能"

> ネットアップが提供している、特定のアプリケーションのデフォルトのPre-snapshot実行フックとPost-Snapshot実行フックは、このリリースでは削除されています。このリリースにアップグレードし、スナップショットの実行フックを独自に提供しない場合、Astra Controlはクラッシュコンシステントスナップショットのみを作成します。にアクセスします "ネットアップのVerda" GitHubリポジトリ：サンプルの実行フックスクリプトを使用します。環境に合わせて変更できます。

- "VMware Tanzu Kubernetes Grid Integrated Edition（TKGI）のサポート"
- "Google Anthosに対応しています"
- "LDAP設定（Astra Control API経由）"

既知の問題および制限事項

- "このリリースの既知の問題"
- "Astra Data Store およびこの Astra Control Center リリースでの既知の問題"
- "このリリースの既知の制限事項は以下のとおりです"

2022 年 4 月 26 日（ 22.04.0 ）

詳細

新機能とサポート

- "Astra Control CenterからAstraデータストアを導入"
- "ネームスペースのロールベースアクセス制御（ RBAC ）"
- "Cloud Volumes ONTAP のサポート"
- "Astra Control Center の一般的な入力イネーブルメント"
- "Astra Control からバケットを取り外す"
- "VMware Tanzu ポートフォリオのサポート"

既知の問題および制限事項

- "このリリースの既知の問題"
- "Astra Data Store およびこの Astra Control Center リリースでの既知の問題"
- "このリリースの既知の制限事項は以下のとおりです"

2021 年 12 月 14 日（ 21.12 ）

詳細

新機能とサポート

- "アプリケーションのリストア"
- "実行フック"
- "ネームスペースを対象とした演算子を使用して展開されたアプリケーションのサポート"
- "アップストリーム Kubernetes と Rancher もサポートしています"
- "Astra データストアプレビューバックエンドの管理と監視"
- "Astra Control Center のアップグレード"
- "インストール用の Red Hat OperatorHub オプションです"

解決済みの問題

- "このリリースの解決済みの問題"

既知の問題および制限事項

- "このリリースの既知の問題"
- "Astra データストアのプレビューおよびこの Astra Control Center リリースでの既知の問題"
- "このリリースの既知の制限事項は以下のとおりです"

2021 年 8 月 5 日（ 21.08 ）

詳細

Astra Control Center の初回リリース。

- "それは何であるか"
- "アーキテクチャとコンポーネントを理解する"
- "開始には何が必要ですか"
- "をインストールします" および "セットアップ（ Setup ）"
- "管理" および "保護" アプリケーション
- "バケットを管理する" および "ストレージバックエンド"
- "アカウントを管理"
- "API による自動化"

詳細については、こちらをご覧ください

- "このリリースの既知の問題"
- "このリリースの既知の制限事項は以下のとおりです"
- "Astra データストアのドキュメント"
- "以前のバージョンの Astra Control Center ドキュメント"

既知の問題

既知の問題は、このリリースの製品を正常に使用できない可能性のある問題を特定します。

現在のリリースに影響する既知の問題は次のとおりです。

アプリケーション

- アプリケーションをリストアすると、 PV のサイズが元の PV よりも大きくなります
- 特定のバージョンの PostgreSQL を使用すると、アプリケーションクローンが失敗します
- サービスアカウントレベルの OCP セキュリティコンテキスト制約（ SCC ）を使用すると、アプリケーションのクローンが失敗する
- [ストレージクラスを設定してアプリケーションを導入すると、アプリケーションのクローンが失敗する]
- クラスタの管理後にボリュームnapshotclassを追加すると、アプリケーションのバックアップとSnapshotが失敗します

クラスタ

- デフォルトの kubeconfig ファイルに複数のコンテキストが含まれている場合、 Astra Control Center を使用したクラスタの管理が失敗します

その他の問題

- Astra Trident がオフラインの場合、 Internal Service Error （ 500 ）によりアプリケーションデータ管理処理が失敗する
- Snapshot コントローラバージョン 4.2.0 では、 Snapshot が失敗することがあります

アプリケーションをリストアすると、 PV のサイズが元の PV よりも大きくなります

バックアップの作成後に永続ボリュームのサイズを変更し、そのバックアップからリストアすると、永続ボリュームのサイズはバックアップのサイズではなく PV の新しいサイズと一致します。

特定のバージョンの PostgreSQL を使用すると、アプリケーションクローンが失敗します

Bitnami PostgreSQL 11.5.0 チャートを使用すると、同じクラスタ内のアプリケーションクローンは一貫して失敗します。正常にクローニングするには、以前のバージョンのグラフを使用してください。

サービスアカウントレベルの OCP セキュリティコンテキスト制約（ SCC ）を使用すると、アプリケーションのクローンが失敗する

OpenShift Container Platform クラスタのネームスペース内のサービスアカウントレベルで元のセキュリティコンテキストの制約が設定されていると、アプリケーションクローンが失敗する場合があります。アプリケーションクローンが失敗すると、Astra Control Centerの管理対象アプリケーション領域にステータスとともに表示されます `Removed`。を参照してください "技術情報アーティクル" を参照してください。

クラスタの管理後にボリューム**napshotclass**を追加すると、アプリケーションのバックアップと**Snapshot**が失敗します

でバックアップとSnapshotの作成が失敗する `UI 500 error` このシナリオでは、回避策 として、アプリリストを更新します。

ストレージクラスを設定してアプリケーションを導入すると、アプリケーションのクローンが失敗する

ストレージクラスを明示的に設定してアプリケーションを導入したあと（例： `helm install …-set global.storageClass=netapp-cvs-perf-extreme`その後、アプリケーションのクローニングを実行するには、ターゲットクラスタに元のストレージクラスが指定されている必要があります。ストレージクラスを明示的に設定したアプリケーションを、同じストレージクラスを含まないクラスタにクローニングすると、失敗します。このシナリオではリカバリ手順はありません。

デフォルトの kubeconfig ファイルに複数のコンテキストが含まれている場合、 Astra Control Center を使用したクラスタの管理が失敗します

複数のクラスタおよびコンテキストで kubeconfig を使用することはできません。を参照してください "技術情報アーティクル" を参照してください。

Astra Trident がオフラインの場合、 Internal Service Error （ 500 ）によりアプリケーションデータ管理処理が失敗する

アプリケーションクラスタの Astra Trident がオフラインになり（オンラインに戻った）、 500 件の内部サービスエラーが発生した場合に、アプリケーションデータ管理を試みると、アプリケーションクラスタ内のすべての Kubernetes ノードを再起動して機能を復旧します。

Snapshot コントローラバージョン 4.2.0 では、 Snapshot が失敗することがあります

Kubernetes 1.20 または 1.21 で Kubernetes snapshot-controller （別名 external-snapshotter ）バージョン 4.2.0 を使用すると、 Snapshot が失敗することがあります。これを防ぐには、別のを使用してください "サポートされているバージョン" バージョン 4.2.1 などの外部 Snapshot データ。 Kubernetes バージョン 1.20 または 1.21 で使用。

1. POST呼び出しを実行して、更新されたkubeconfigファイルをに追加します `/credentials` エンドポイントを選択し、割り当てられているを取得し `id` 応答の本文から：
2. からPUT呼び出しを実行します `/clusters` 適切なクラスタIDを使用してエンドポイントを設定します `credentialID` に移動します `id` 前の手順で確認した値。

これらの手順が完了すると、クラスタに関連付けられているクレデンシャルが更新され、クラスタを再接続して状態をに更新できるようになります `available`。

詳細については、こちらをご覧ください

- "Astra データストアのプレビューおよびこの Astra Control Center リリースでの既知の問題"
- "既知の制限"

Astra Data Store およびこの Astra Control Center リリースでの既知の問題

既知の問題は、このリリースの製品を正常に使用できない可能性のある問題を特定します。

"追加のAstraデータストアの既知の問題を確認してください" これは、Astra Control Centerの最新リリースでAstraデータストアの管理に影響を与える可能性があります。

Astraデータストアボリュームの詳細は、**Astra Control Center UI**の**Storage Backends**ページに表示されません

容量やスループットなどの詳細はUIに表示されません。この問題 が発生した場合は、ストレージバックエンドの管理を解除してから再度追加してください。

Astraデータストアを使用してクラスタの管理を解除するには、最初に管理対象システムアプリケーションを削除する必要があります

Astraデータストアを含むクラスタをAstra Control Centerクラスタに追加した場合、デフォルトではAstraデータストアシステムアプリケーションが非表示アプリケーションとして管理されます。クラスタの管理を解除するには、まず、astrs-systemアプリケーションの管理を解除する必要があります。Astra Control Center UIを使用してこのタイプのアプリケーションを管理解除することはできません。代わりに、Astra Control API要求を使用して手動でアプリケーションを削除します。

詳細

手順

1. 次のAPIを使用して、管理対象クラスタのIDを取得します。

```
/accounts/{account_id}/topology/v1/managedClusters
```

対応：

```
{
"items": [
{
"type": "application/astra-managedCluster",
"version": "1.1",
"id": "123ab987-0bc0-00d0-a00a-1234567abd8d",
"name": "astrads-cluster-1234567",
...
```

2. マネージド宇宙システムアプリIDを取得：

```
/accounts/{account_id}/topology/v2/managedClusters/{managedCluster_id}/apps
```

対応：

```
{
  "items": [
    [
      "1b011d11-bb88-40c7-a1a1-ab1234c123d3",
      "astrads-system",
      "ready"
    ]
  ],
  "metadata": {}
}
```

3. 前の手順で取得したアプリケーションIDを使用して、宇宙システムアプリを削除します (`1b011d11-bb88-40c7-a1a1-ab1234c123d3`) 。

```
/accounts/{account_id}/k8s/v2/apps/{astrads-system_app_id}
```

詳細については、こちらをご覧ください

- "既知の問題"
- "既知の制限"

既知の制限

既知の制限事項は、このリリースの製品でサポートされていないプラットフォーム、デバイス、機能、または製品と正しく相互運用できない機能を特定します。これらの制限事項を慎重に確認してください

クラスタ管理の制限事項

- 2 つの Astra Control Center インスタンスで同じクラスタを管理することはできません
- Astra Control Center は、同じ名前の 2 つのクラスタを管理できません

Role-Based Access Control （ RBAC ；ロールベースアクセス制御）の制限事項があります

- ネームスペースの RBAC に制約があるユーザは、クラスタの追加と管理解除を行うことができます
- [名前空間の制約を持つメンバは、管理者が名前空間を制約に追加するまで、クローンまたは復元されたアプリケーションにアクセスできません]

アプリケーション管理の制限

- [パスバイリファレンス演算子を使用してインストールされたアプリケーションのクローンが失敗することがあります]
- 証明書マネージャを使用するアプリケーションの In Place リストア処理はサポートされていません
- OLM 対応およびクラスタ対象のオペレータ展開アプリケーションはサポートされていません
- Helm 2 で展開されたアプリケーションはサポートされていません

一般的な制限事項

- Astra Control Center の S3 バケットは、使用可能容量を報告しません
- Astra Control Center は、プロキシサーバー用に入力した詳細を検証しません
- Postgres ポッドへの既存の接続が原因で障害が発生します
- Astra Control Center インスタンスの削除中にバックアップとスナップショットが保持されない場合があります

2 つの **Astra Control Center** インスタンスで同じクラスタを管理することはできません

別の Astra Control Center インスタンスでクラスタを管理する場合は、最初にを実行する必要があります "クラスタの管理を解除します" 別のインスタンスで管理する前に、管理対象のインスタンスから管理します。管理対象からクラスタを削除したら、次のコマンドを実行してクラスタが管理対象外であることを確認します。

```
oc get pods n -netapp-monitoring
```

そのネームスペースでポッドを実行していないことを確認するか、ネームスペースを存在させないようにします。どちらかが true の場合、クラスタは管理対象外です。

Astra Control Center は、同じ名前の 2 つのクラスタを管理できません

既存のクラスタと同じ名前のクラスタを追加しようとすると、処理に失敗します。この問題は、 Kubernetes 構成ファイルでクラスタ名のデフォルトを変更していない場合、通常は標準の Kubernetes 環境で発生します。

回避策として、次の手順を実行します。

1. kubeadm -config 構成マップを編集します。

```
kubectl edit configmaps -n kube-system kubeadm-config
```

2. を変更します `clusterName` フィールド値の開始値 `kubernetes` （Kubernetesのデフォルト名）を一意のカスタム名に変更します。
3. kubeconfigを編集します (`.kube/config`）。
4. からクラスタ名を更新します `kubernetes` を使用して一意のカスタム名を指定します (`xyz-cluster` は、以下の例で使用されています）。両方で更新を行います `clusters` および `contexts` 次の例に示すように、セクションを示します。

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data:
ExAmPLERb2tCcjZ5K3E2Njk4eQotLExAMpLEORCBDRVJUSUZJQ0FURS0txxxxXX==
    server: https://x.x.x.x:6443
  name: xyz-cluster
contexts:
- context:
    cluster: xyz-cluster
    namespace: default
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
```

ネームスペースの RBAC に制約があるユーザは、クラスタの追加と管理解除を行うことができます

ネームスペースの RBAC に制限があるユーザは、クラスタの追加または管理解除を行うことができません。現在の制限により、 Astra は、このようなユーザによるクラスタの管理解除を妨げません。

名前空間の制約を持つメンバは、管理者が名前空間を制約に追加するまで、クローンまたは復元されたアプリケーションにアクセスできません

任意 `member` ネームスペース名/ IDによるRBACの制約があるユーザは、同じクラスタまたは組織のアカウントにある他のクラスタの新しいネームスペースにアプリケーションをクローニングまたはリストアできます。

ただし、同じユーザが、クローニングまたはリストアされたアプリケーションに新しいネームスペースからアクセスすることはできません。クローンまたはリストア処理によって新しいネームスペースが作成されると、アカウントの管理者または所有者はを編集できるようになります `member` 影響を受けるユーザーが新しい名前空間へのアクセスを許可するためのユーザーアカウントの制約を更新します。

パスバイリファレンス演算子を使用してインストールされたアプリケーションのクローンが失敗することがあります

Astra Control は、名前空間を対象とした演算子でインストールされたアプリケーションをサポートします。これらの演算子は、一般に「パスバイリファレンス」アーキテクチャではなく「パスバイ値」で設計されています。これらのパターンに続くいくつかのオペレータアプリを次に示します。

- "Apache K8ssandra"

> K8ssandra では、 In Place リストア処理がサポートされます。新しいネームスペースまたはクラスタにリストアするには、アプリケーションの元のインスタンスを停止する必要があります。これは、ピアグループ情報がインスタンス間通信を行わないようにするためです。アプリケーションのクローニングはサポートされていません。

- "Jenkins CI"
- "Percona XtraDB クラスタ"

Astra Controlでは、「パスバイリファレンス」アーキテクチャ（CockroachDBオペレータなど）で設計されたオペレータをクローニングできない場合があります。クローニング処理では、クローニング処理の一環として独自の新しいシークレットが存在する場合でも、クローニングされたオペレータがソースオペレータから Kubernetes シークレットを参照しようとします。Astra Control がソースオペレータの Kubernetes シークレットを認識しないため、クローニング処理が失敗する場合があります。

証明書マネージャを使用するアプリケーションの **In Place** リストア処理はサポートされていません

このリリースの Astra Control Center では、証明書マネージャを使用したアプリのインプレースリストアはサポートされていません。別のネームスペースへのリストア処理とクローニング処理がサポートされています。

OLM 対応およびクラスタ対象のオペレータ展開アプリケーションはサポートされていません

Astra Control Center は、クラスタを対象としたオペレータによるアプリケーション管理アクティビティをサポートしません。

Helm 2 で展開されたアプリケーションはサポートされていません

Helm を使用してアプリケーションを展開する場合、 Astra Control Center には Helm バージョン 3 が必要です。Helm 3 （または Helm 2 から Helm 3 にアップグレード）を使用して展開されたアプリケーションの管理とクローニングが完全にサポートされています。詳細については、を参照してください "Astra Control Center の要件"。

Astra Control Center の **S3** バケットは、使用可能容量を報告しません

Astra Control Center で管理されているアプリケーションのバックアップまたはクローニングを行う前に、

ONTAP または StorageGRID 管理システムでバケット情報を確認します。

Astra Control Center は、プロキシサーバー用に入力した詳細を検証しません

実行することを確認してください "正しい値を入力します" 接続を確立するとき。

Postgres ポッドへの既存の接続が原因で障害が発生します

Postgres ポッドで操作を実行する場合は、 psql コマンドを使用するためにポッド内で直接接続しないでください。Astra Control では、 psql にアクセスしてデータベースをフリーズし、解凍する必要があります。既存の接続がある場合、スナップショット、バックアップ、またはクローンは失敗します。

Astra Control Center インスタンスの削除中にバックアップとスナップショットが保持されない場合があります

評価用ライセンスをお持ちの場合は、 Astra Control Center に障害が発生したときに ASUP を送信していないときにデータが失われないように、アカウント ID を必ず保存してください。

詳細については、こちらをご覧ください

- "既知の問題"
- "Astra Data Store およびこの Astra Control Center リリースでの既知の問題"

概念

Astra Controlの詳細をご確認ください

Astra Control は、 Kubernetes アプリケーションデータライフサイクル管理解決策で、ステートフルアプリケーションの運用を簡易化します。Kubernetesワークロードの保護、バックアップ、複製、移行を簡易化し、作業アプリケーションのクローンを瞬時に作成できます。

の機能

Astra Control は、 Kubernetes アプリケーションデータのライフサイクル管理に不可欠な機能を提供

- 永続的ストレージを自動的に管理
- アプリケーション対応のオンデマンドの Snapshot とバックアップを作成
- ポリシーベースのスナップショットおよびバックアップ操作を自動化します
- NetApp SnapMirrorテクノロジを使用して、アプリケーションをリモートシステムにレプリケートします
- Kubernetes クラスタ間でアプリケーションとデータを移行
- 本番環境からステージング環境にアプリケーションを簡単にクローニングできます
- アプリケーションの稼働状態と保護状態を視覚化します
- バックアップと移行のワークフローを実装するには、ユーザインターフェイスまたは API を使用してください

導入モデル

Astra Control には、次の 2 つの導入モデルがあります。

- * Astra Control Service * ： Google Kubernetes Engine （ GKE ）および Azure Kubernetes Service （ AKS ）で Kubernetes クラスタのアプリケーション対応データ管理を提供する、ネットアップが管理するサービス。
- * Astra Control Center * ：オンプレミス環境で実行される Kubernetes クラスタのアプリケーション対応データ管理を提供する、自己管理ソフトウェアです。

	Astra 制御サービス	**Astra Control Center** の略
どのような方法で提供されますか？	ネットアップのフルマネージドクラウドサービス	ソフトウェアとしてダウンロード、インストール、および管理します
ホストされているのはどこですか？	ネットアップが選択したパブリッククラウドで実現	指定した Kubernetes クラスタで実行します
更新方法	管理はネットアップが行います	更新を管理します

	Astra 制御サービス	Astra Control Center の略
アプリケーションデータ管理機能とは何ですか？	ストレージバックエンドと外部サービスを除く両方のプラットフォームで同じ機能を利用できます	ストレージバックエンドと外部サービスを除く両方のプラットフォームで同じ機能を利用できます
ストレージバックエンドでサポートされるもの	ネットアップのクラウドサービス	• NetApp ONTAP AFF および FAS システム • ストレージバックエンドとしての Astra データストア • Cloud Volumes ONTAP ストレージバックエンド

Astra Control Service の仕組み

Astra Control Service は、常時稼働し、最新の機能で更新される、ネットアップが管理するクラウドサービスです。複数のコンポーネントを利用して、アプリケーションデータのライフサイクル管理を実現します。

Astra Control Service の概要は次のように機能します。

- Astra Control Service の利用を開始するには、クラウドプロバイダをセットアップし、 Astra アカウントに登録します。
  - GKE クラスタでは、 Astra Control Service はを使用します "NetApp Cloud Volumes Service for Google Cloud" または、永続ボリューム用のストレージバックエンドとして Google Persistent Disk を使用します。
  - AKS クラスタの場合、 Astra Control Service はを使用します "Azure NetApp Files の特長" または、永続ボリューム用のストレージバックエンドとして Azure Disk Storage を選択します。
  - Amazon EKSクラスタの場合、Astra Control Serviceはを使用します "Amazon Elastic Block Store" または "NetApp ONTAP 対応の Amazon FSX" 永続ボリュームのストレージバックエンドとして。
- 最初の Kubernetes コンピューティングを Astra Control サービスに追加します。Astra Control Service は、次の処理を実行します。
  - バックアップコピーが格納されるクラウドプロバイダアカウントにオブジェクトストアを作成します。

    Azure では、 Astra Control Service によって、 BLOB コンテナ用のリソースグループ、ストレージアカウント、およびキーも作成されます。
  - クラスタに新しい admin ロールと Kubernetes サービスアカウントを作成します。
  - 新しい admin ロールを使用してインストールします "Astra Trident" をクリックして、 1 つ以上のストレージクラスを作成します。
  - Azure NetApp Files または NetApp Cloud Volumes Service for Google Cloud をストレージバックエンドとして使用している場合、 Astra Control Service は Astra Trident を使用して、アプリケーション用の永続的ボリュームをプロビジョニングします。
- この時点で、アプリケーションをクラスタに追加できます。永続ボリュームは、新しいデフォルトのストレージクラスでプロビジョニングされます。
- 次に、 Astra Control Service を使用してこれらのアプリケーションを管理し、スナップショット、バック

アップ、クローンの作成を開始します。

Astra Control の無料プランを使用すると、最大 10 個のアプリをアカウントで管理できます。10 以上のアプリを管理する場合は、無料プランからプレミアムプランにアップグレードして請求を設定する必要があります。

Astra Control Center の仕組み

Astra Control Center は、お客様のプライベートクラウドでローカルに実行されます。

Astra Control Centerは、次のものでKubernetesクラスタをサポートします。

- Trident ストレージバックエンドは ONTAP 9.5 以降で構成されています
- Astra データストアストレージバックエンド

クラウド接続環境では、 Cloud Insights を使用して高度なモニタリングとテレメトリを提供します。Cloud Insights 接続がない場合、 Astra Control Center では、限定的な（ 7 日間の指標）監視と計測データを使用できます。また、オープン指標エンドポイントを介して Kubernetes の標準の監視ツール（ Prometheus や Grafana など）にエクスポートすることもできます。

Astra Control Center は、 AutoSupport と Active IQ のエコシステムに完全に統合されており、ユーザとネットアップサポートにトラブルシューティングと使用に関する情報を提供します。

Astra Control Center を試用するには、 90 日間の評価版ライセンスを使用します。評価版は、 E メールとコミュニティ（ Slack チャンネル）のオプションでサポートされています。また、製品内サポートダッシュボードから技術情報アーティクルやドキュメントにアクセスすることもできます。

Astra Control Center をインストールして使用するには、一定の要件を満たす必要があります "要件"。

Astra Control Center の概要は次のように機能します。

- Astra Control Center は、ローカル環境にインストールします。方法の詳細については、こちらをご覧ください "Astra Control Center をインストールします"。
- 次のようなセットアップタスクを実行したとします。
  - ライセンスをセットアップする
  - 最初のクラスタを追加します。
  - クラスタを追加したときに検出されたストレージバックエンドを追加します。
  - アプリケーションバックアップを格納するオブジェクトストアバケットを追加します。

方法の詳細については、こちらをご覧ください "Astra Control Center をセットアップします"。

Astra Control Center は、次のことを行います。

- ネームスペースを含むクラスタの詳細を検出し、アプリケーションを定義して保護できます。
- では、管理対象として選択したクラスタに Astra Trident または Astra データストア構成が検出され、ストレージバックエンドを監視できます。

クラスタにアプリケーションを追加できます。また、管理対象のクラスタにすでにアプリケーションがある場合は、Astra Control Centerを使用してそれらを管理できます。次に、Astra Control Centerを使用して、スナ

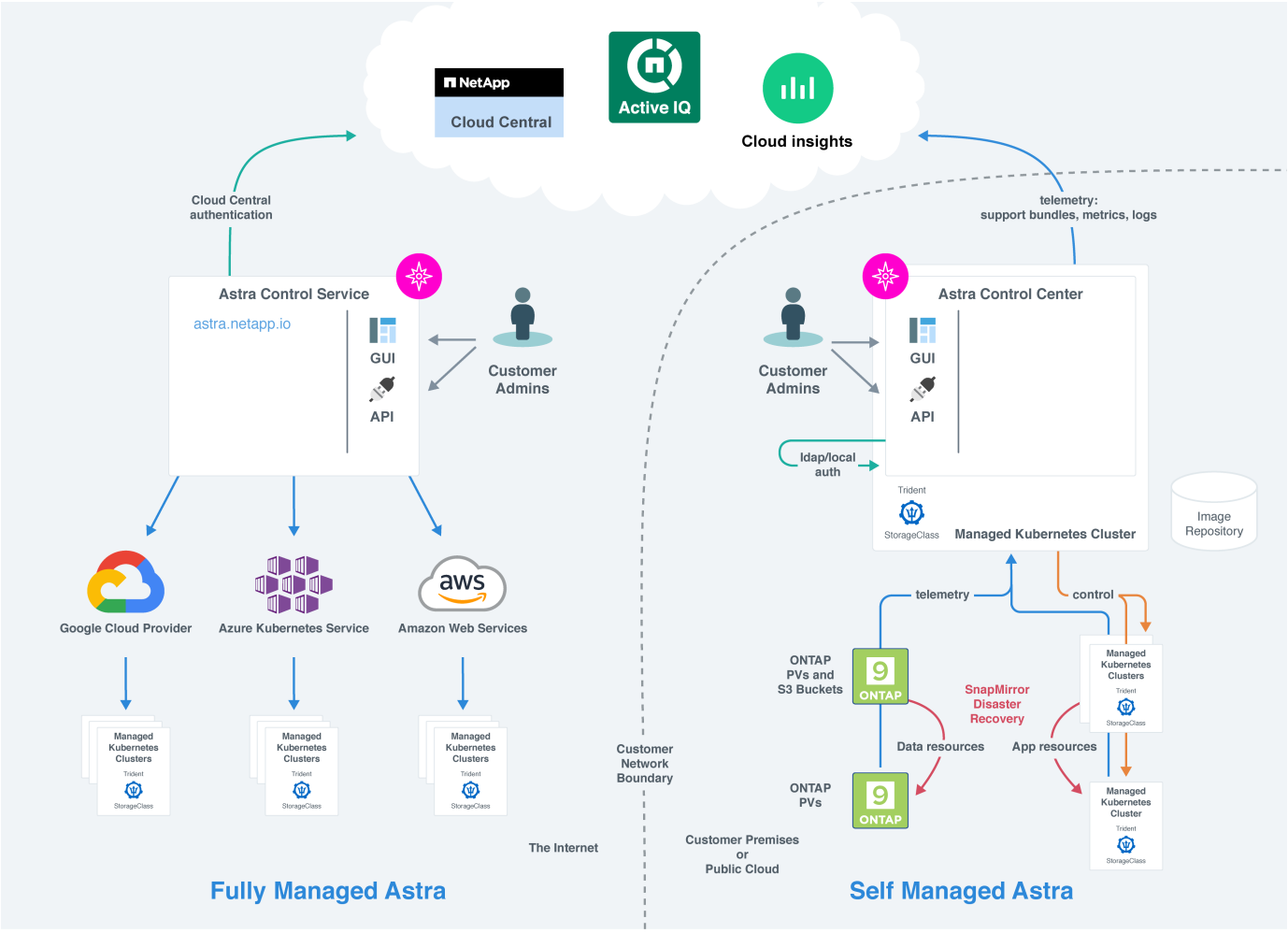
ップショット、バックアップ、クローン、およびレプリケーション関係を作成します。

を参照してください。

- "Astra Control Service のマニュアル"
- "Astra Control Center のドキュメント"
- "Astra データストアのドキュメント"
- "Astra Trident のドキュメント"
- "Astra Control API を使用"
- "Cloud Insights のドキュメント"
- "ONTAP のドキュメント"

アーキテクチャとコンポーネント

ここでは、 Astra Control 環境のさまざまなコンポーネントの概要を示します。



Astra Control コンポーネント

- * Kubernetes クラスタ * ： Kubernetes は、コンテナ化されたワークロードとサービスを管理するため

の、ポータブルで拡張性に優れたオープンソースプラットフォームであり、宣言型の設定と自動化の両方を促進します。Astra は、 Kubernetes クラスタでホストされているアプリケーションに管理サービスを提供します。

- * Trident * ：ネットアップが管理する、完全にサポートされているオープンソースのストレージプロビジョニングおよびオーケストレーションツールである Trident を使用すると、 Docker と Kubernetes で管理するコンテナ化されたアプリケーション用のストレージボリュームを作成できます。Trident を Astra Control Center とともに導入すると、設定済みの ONTAP ストレージバックエンドが含まれます。
- * ストレージバックエンド * ：
  - Astra Control Serviceは、次のストレージバックエンドを使用します。
    - "NetApp Cloud Volumes Service for Google Cloud" または、GKEクラスタのストレージバックエンドとしてGoogle Persistent Diskを使用します
    - "Azure NetApp Files の特長" またはAzure Managed DisksをAKSクラスタのストレージバックエンドとして使用します。
    - "Amazon Elastic Block Store（EBS）" または "NetApp ONTAP 対応の Amazon FSX" EKSクラスタのバックエンドストレージオプションとして使用できます。
  - Astra Control Center は、次のストレージバックエンドを使用します。
    - ONTAP AFF およびFAS 。ONTAP は、ストレージソフトウェアおよびハードウェアプラットフォームとして、コアストレージサービス、複数のストレージアクセスプロトコルのサポート、Snapshot やミラーリングなどのストレージ管理機能を提供します。
    - Cloud Volumes ONTAP
- * Cloud Insights * ：ネットアップのクラウドインフラ監視ツールである Cloud Insights を使用すると、Astra Control Center で管理された Kubernetes クラスタのパフォーマンスと利用率を監視できます。Cloud Insights ：ストレージ使用率とワークロードの相関関係を示します。Cloud Insights 接続を Astra コントロールセンターで有効にすると、テレメータの情報が Astra コントロールセンターの UI ページに表示されます。

Astra Control インターフェイス

さまざまなインターフェイスを使用してタスクを完了できます。

- * ウェブユーザーインタフェース（ UI ） * ： Astra Control Service と Astra Control Center の両方が、同じ Web ベースの UI を使用して、アプリケーションの管理、移行、保護を行うことができます。また、UI を使用してユーザアカウントと設定を管理することもできます。
- * API* ： Astra Control Service と Astra Control Center は、どちらも同じ Astra Control API を使用します。API を使用するタスクは、 UI を使用するタスクと同じです。

Astra Control Center を使用すると、 VM 環境内で実行される Kubernetes クラスタを管理、移行、保護することもできます。

を参照してください。

- "Astra Control Service のマニュアル"
- "Astra Control Center のドキュメント"
- "Astra Trident のドキュメント"
- "Astra Control API を使用"

- "Cloud Insights のドキュメント"
- "ONTAP のドキュメント"

データ保護

Astra Control Center で使用可能なデータ保護の種類と、それらを使用してアプリケーションを保護する最適な方法について説明します。

Snapshot 、バックアップ、保護のポリシー

a_snapshot_は 、アプリケーションと同じプロビジョニングボリュームに格納されるアプリケーションのポイントインタイムコピーです。通常は高速です。ローカル Snapshot を使用して、アプリケーションを以前の時点にリストアできます。スナップショットは高速クローンに便利です。スナップショットには、構成ファイルを含む、アプリケーションのすべての Kubernetes オブジェクトが含まれます。

a_backup_ は外部のオブジェクトストアに格納されます。ローカル Snapshot と比較すると時間がかかることがあります。アプリケーションのバックアップを同じクラスタにリストアすることも、バックアップを別のクラスタにリストアして移行することもできます。バックアップの保持期間を延長することもできます。バックアップは外部のオブジェクトストアに格納されるため、サーバで障害が発生したりデータが失われたりした場合に備えて、 Snapshot よりも優れた保護機能を提供できます。

a_protection policy__ は、アプリケーション用に定義したスケジュールに従って、スナップショット、バックアップ、またはその両方を自動的に作成することで、アプリケーションを保護する方法です。保護ポリシーでは、スケジュールで保持する Snapshot とバックアップの数も選択できます。バックアップとスナップショットを保護ポリシーを使用して自動化することは、組織のニーズに応じて各アプリケーションを確実に保護する最善の方法です。

> _ 最新のバックアップがあるまで、完全に保護することはできません _ 。これは、永続ボリュームから離れたオブジェクトストアにバックアップが格納されるために重要です。障害または事故によってクラスタとその永続的ストレージが消去された場合は、バックアップをリカバリする必要があります。Snapshot を使用してリカバリすることはできません。

クローン

a_clone_は 、アプリケーション、その構成、永続的ストレージの完全な複製です。クローンは、同じ Kubernetes クラスタまたは別のクラスタに手動で作成できます。アプリケーションとストレージを Kubernetes クラスタ間で移動する必要がある場合は、アプリケーションをクローニングすると便利です。

リモートクラスタへのレプリケーション

Astra Controlを使用すると、NetApp SnapMirrorテクノロジの非同期レプリケーション機能を使用して、RPO（目標復旧時点）とRTO（目標復旧時間）の低いアプリケーションのビジネス継続性を構築できます。設定が完了すると、アプリケーションはデータやアプリケーションの変更をクラスタ間でレプリケートできるようになります。

Astra Controlは、アプリケーションのSnapshotコピーをリモートクラスタに非同期でレプリケートします。レプリケーションプロセスには、SnapMirrorでレプリケートされた永続ボリュームのデータと、Astra Controlで保護されたアプリケーションメタデータが含まれます。

アプリケーションのレプリケーションは、次のようにアプリケーションのバックアップとリストアとは異なり

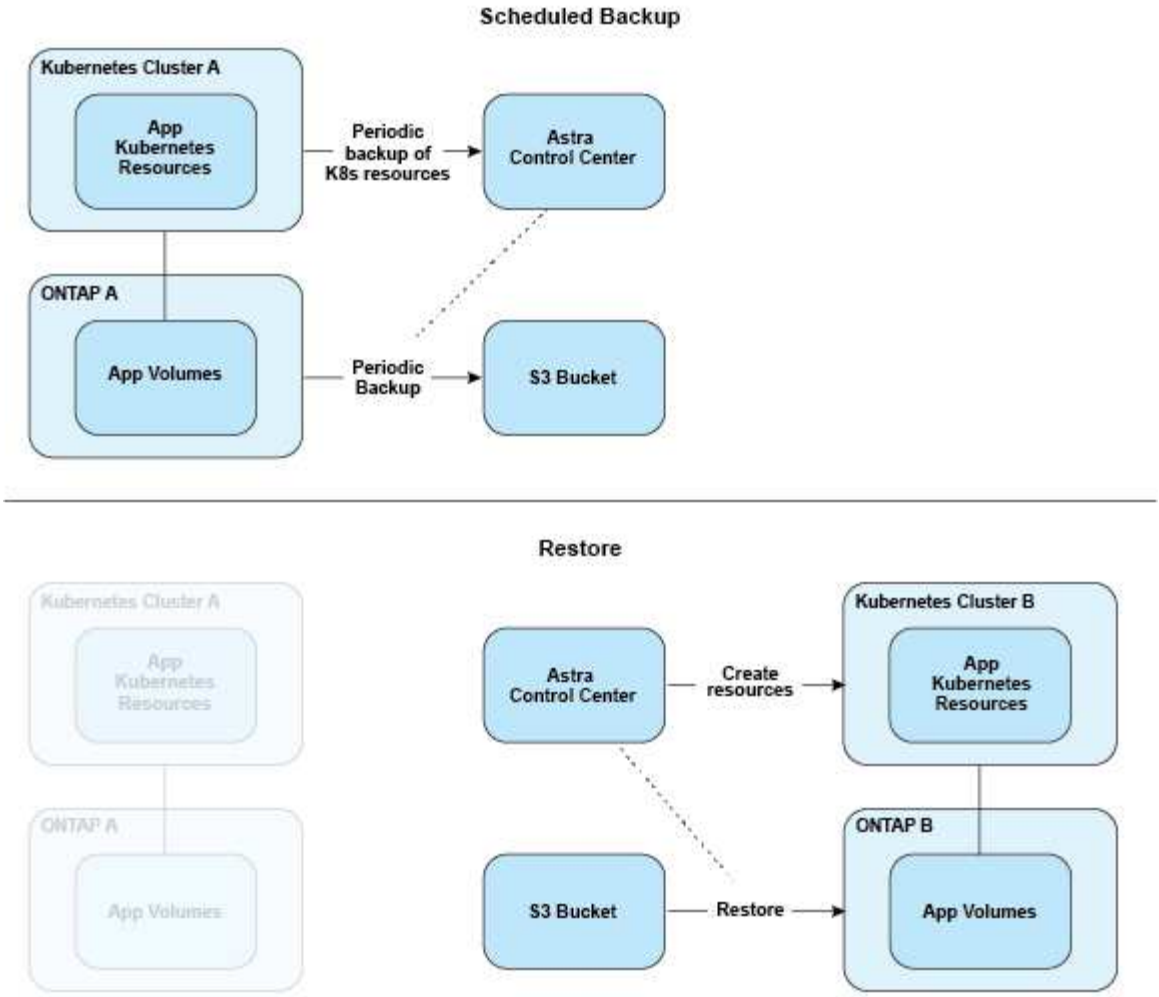
ます。

- アプリケーションレプリケーション：Astra Controlでは、NetApp SnapMirrorを有効にするように構成されたそれぞれのONTAP ストレージバックエンドで、ソースとデスティネーションのKubernetesクラスタを使用および管理する必要があります。Astra Controlは、ポリシーベースのアプリケーションSnapshotを作成して、リモートクラスタにレプリケートします。永続ボリュームのデータのレプリケートには、NetApp SnapMirrorテクノロジを使用します。フェイルオーバーのために、デスティネーションONTAP クラスタ上のレプリケートされたボリュームを含むデスティネーションKubernetesクラスタにアプリケーションオブジェクトを再作成することで、レプリケーションされたアプリケーションをオンラインにすることができます。永続ボリュームのデータはデスティネーションONTAP クラスタにすでに存在するため、Astra Controlを使用すると、フェイルオーバーのためのリカバリ時間を短縮できます。
- アプリケーションのバックアップとリストア：アプリケーションのバックアップ時に、Astra Controlはアプリケーションデータのスナップショットを作成し、オブジェクトストレージバケットに格納します。リストアが必要な場合は、バケット内のデータをONTAP クラスタ上の永続ボリュームにコピーする必要があります。バックアップ/リストア処理では、セカンダリKubernetes / ONTAPクラスタを使用可能にして管理する必要はありませんが、データコピーを追加するとリストア時間が長くなる可能性があります。

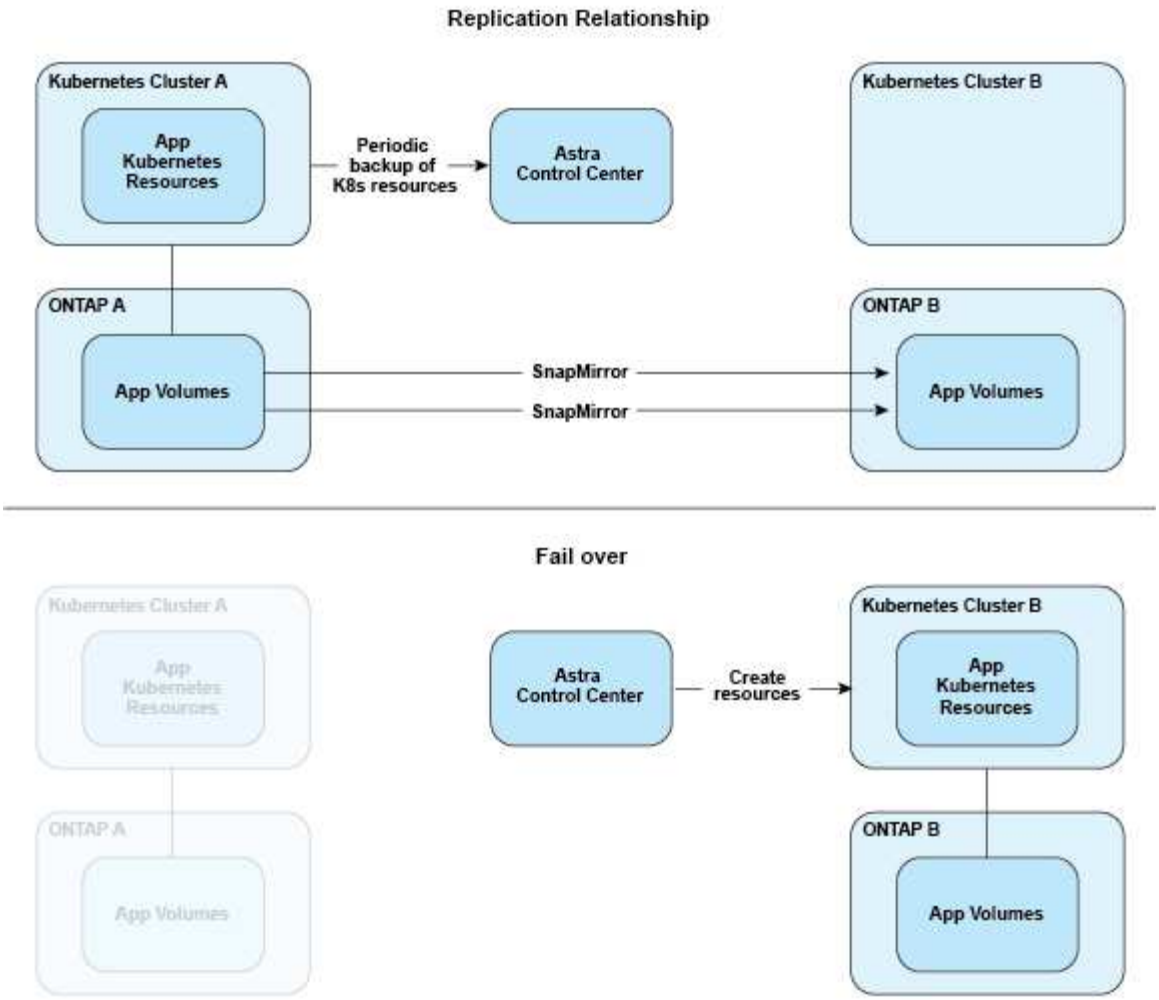
アプリケーションをレプリケートする方法については、を参照してください "SnapMirrorテクノロジを使用してアプリケーションをリモートシステムにレプリケート"。

次の図は、スケジュールされたバックアップおよびリストアのプロセスをレプリケーションプロセスと比較したものです。

バックアッププロセスでは、S3バケットにデータをコピーし、S3バケットからリストアします。



一方、レプリケーションはONTAP にレプリケートされ、その後フェイルオーバーによってKubernetesリソースが作成されます。



ライセンス

Astra Control Centerを有効にするには、フルアプリケーションデータ管理機能を有効にするためのライセンスが必要です。Astra Control Center をライセンスなしで導入すると、システム機能が制限されていることを示すバナーが Web UI に表示されます。

アプリケーションとデータを保護するにはライセンスが必要です。「Astra Control Center」を参照してください "の機能" を参照してください。

製品を購入すると、シリアル番号とライセンスが届きます。からネットアップライセンスファイル（NLF）を生成できます "NetApp Support Site"。

また、 Astra Control Center に評価ライセンスをお試しいただくこともできます。このライセンスは、 Astra Control Center をダウンロードした日から 90 日間使用できます。詳細については、を参照してください "要件"。

ONTAP ストレージバックエンドに必要なライセンスの詳細については、を参照してください "サポートされるストレージバックエンド"。

> クラスタの追加、バケットの追加、ストレージバックエンドの管理をライセンスなしで実行できます。

ライセンス消費量の計算方法

新しいクラスタを Astra Control Center に追加しても、クラスター上で実行されているアプリケーションの少なくとも 1 つが Astra Control Center によって管理されるまで、使用済みのライセンスにはカウントされません。

クラスタ上でアプリケーションの管理を開始すると、そのクラスタのすべてのCPUユニットがAstra Control Centerライセンス消費量に含まれます。

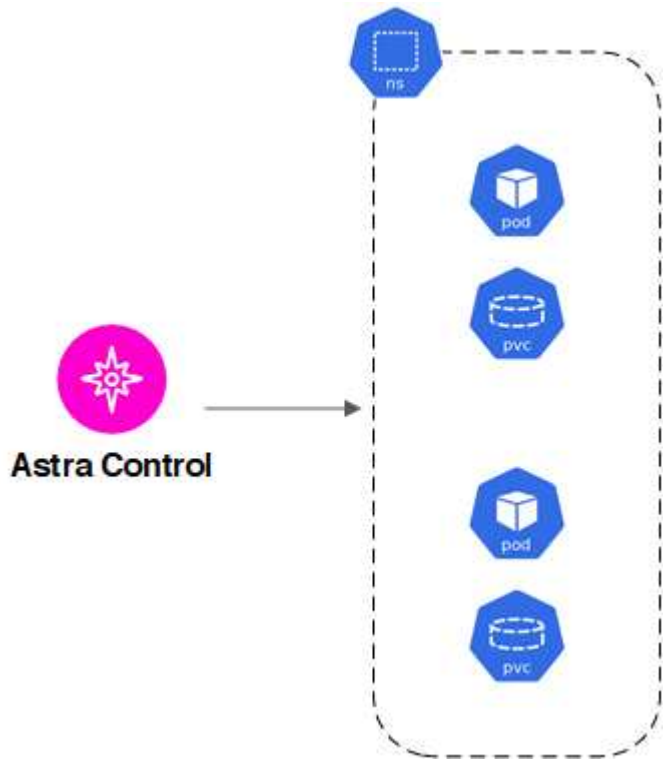
詳細については、こちらをご覧ください

- "既存のライセンスを更新する"

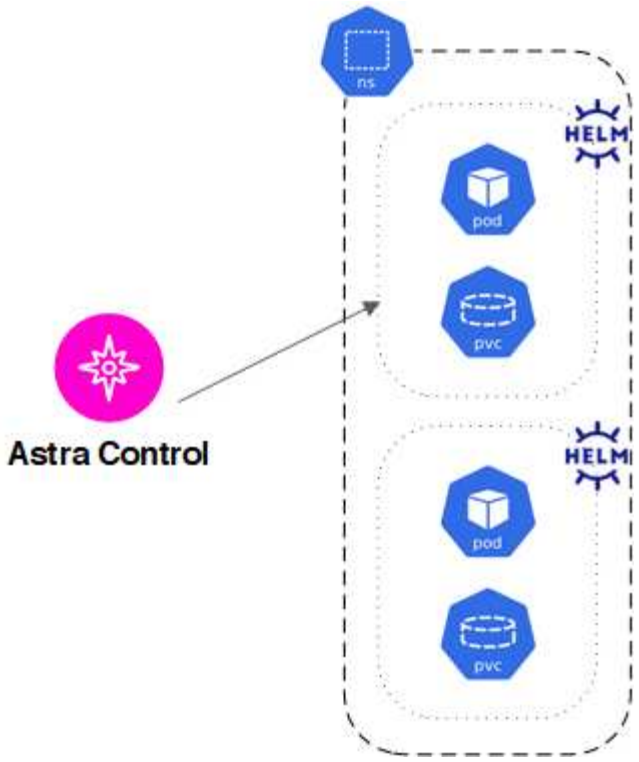
アプリケーション管理について理解する

Astra Controlがクラスタを検出すると、それらのクラスタ上のアプリケーションは、管理方法を選択するまで管理されません。Astra Control のマネージドアプリケーションには、次のいずれかを使用できます。

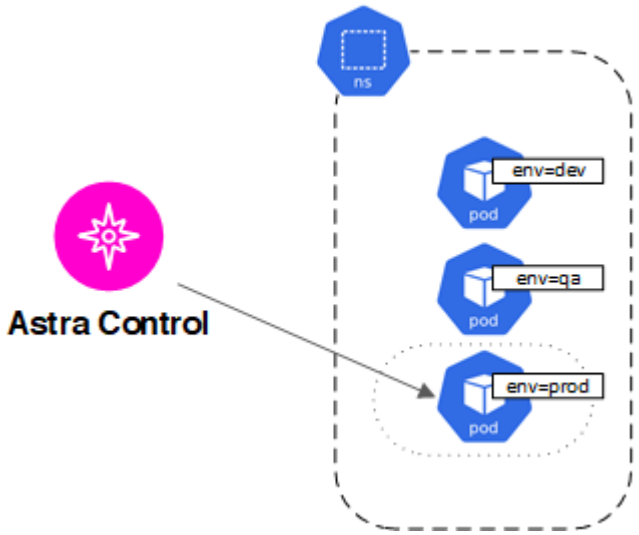
- ネームスペース。ネームスペース内のすべてのリソースを含みます



- ネームスペース内に展開された個々のアプリケーション（この例では、helm3が使用されます）



- ネームスペース内のKubernetesラベルで識別されるリソースのグループ



ストレージクラスと永続的ボリュームサイズ

Astra Control Center は、 ONTAP または Astra データストアをストレージバックエンドとしてサポートします。

概要

Astra Control Center は、次の機能をサポートします。

- *Trident ストレージクラスは Astra Data Store ストレージ * をサポート： 1 つ以上の Astra データストア

クラスタを手動でインストールした場合、 Astra Control Center では、これらをインポートしてトポロジ（ノード、ディスク）とさまざまなステータスを取得することができます。

Astra Control Center には、 Astra Data Store の構成、 Kubernetes クラスタが属するクラウド、 Astra Data Store でプロビジョニングされた永続ボリューム、対応する内部ボリュームの名前、永続ボリュームを使用するアプリケーション、およびアプリケーションを含むクラスタから、基盤となる Kubernetes クラスタが表示されます。

- * ONTAP ストレージ * がサポートする Trident ストレージクラス： ONTAP バックエンドを使用している場合、 Astra Control Center では、 ONTAP バックエンドをインポートしてさまざまな監視情報をレポートすることができます。

> Trident のストレージクラスは、 Astra Control Center の外部で事前に設定しておく必要があります。

ストレージクラス

Astra Control Center にクラスタを追加すると、そのクラスタで以前に設定したストレージクラスをデフォルトのストレージクラスとして選択するように求められます。このストレージクラスは、永続ボリューム要求（ PVC ）でストレージクラスが指定されていない場合に使用されます。デフォルトのストレージクラスは、 Astra Control Center 内でいつでも変更できます。また、 PVC または Helm チャート内のストレージクラスの名前を指定することで、任意のストレージクラスをいつでも使用できます。Kubernetes クラスタにデフォルトのストレージクラスが 1 つだけ定義されていることを確認します。

Astra Control Center を Astra データストアストレージバックエンドと統合して使用する場合、インストール後にストレージクラスは定義されません。Trident のデフォルトストレージクラスを作成し、ストレージバックエンドに適用する必要があります。を参照してください "Astra データストア入門" をクリックして、デフォルトの Astra データストアストレージクラスを作成します。

を参照してください。

- "Astra Trident のドキュメント"

ユーザロールとネームスペース

Astra Control のユーザロールとネームスペースについて説明し、それらを使用して組織内のリソースへのアクセスを制御する方法を説明します。

ユーザロール

ロールを使用して、ユーザが Astra Control のリソースまたは機能にアクセスできるように制御できます。Astra Control のユーザロールは次のとおりです。

- * Viewer * はリソースを表示できます。
- メンバー * には、ビューア・ロールの権限があり、アプリとクラスタの管理、アプリの管理解除、スナップショットとバックアップの削除ができます。
- **Admin** にはメンバーの役割権限があり、 Owner 以外の他のユーザーを追加および削除できます。
- * Owner * には Admin ロールの権限があり、任意のユーザーアカウントを追加および削除できます。

メンバーまたはビューアユーザーに制約を追加して、ユーザーを 1 つ以上に制限できます [ネームスペース]。

ネームスペース

ネームスペースは、 Astra Control によって管理されるクラスタ内の特定のリソースに割り当てることができるスコープです。Astra Control では、 Astra Control にクラスタを追加すると、クラスタのネームスペースが検出されます。検出されたネームスペースは、ユーザに制約として割り当てることができます。そのリソースを使用できるのは、そのネームスペースにアクセスできるメンバーだけです。名前空間を使用すると、組織に適したパラダイム（たとえば、会社内の物理的なリージョンや部門）を使用して、リソースへのアクセスを制御できます。ユーザに制約を追加する場合は、そのユーザにすべてのネームスペースへのアクセス権を設定するか、特定のネームスペースのセットのみを設定できます。ネームスペースラベルを使用して、ネームスペースの制約を割り当てることもできます。

詳細については、こちらをご覧ください

"ロールの管理"

はじめに

Astra Control Center の要件

運用環境、アプリケーションクラスタ、アプリケーション、ライセンス、 Web ブラウザの準備ができているかどうかを検証します。

- [運用環境の要件]
- [サポートされるストレージバックエンド]
- [アプリケーションクラスタの要件]
- [アプリケーション管理の要件]
- [レプリケーションの前提条件]
- [インターネットにアクセスできます]
- [使用許諾]
- オンプレミス Kubernetes クラスタへの入力
- [ネットワーク要件]
- サポートされている Web ブラウザ

運用環境の要件

Astra Control Centerは、次のタイプの運用環境で検証済みです。

- Google Anthos 1.10または1.11
- Kubernetes 1.22〜1.24
- Rancher Kubernetes Engine（RKE）：
  - Rancher 2.5.12およびRKE 1.3.3（2.6.3）付きRKE 1.2.16
  - Rancher 2.6.3を使用したRKE 2（v1.23.6 + rke2r2）
- Red Hat OpenShift Container Platform 4.8、4.9、または4.10
- VMware Tanzu Kubernetes Grid 1.4または1.5
- VMware Tanzu Kubernetes Grid Integrated Edition 1.12.2または1.13

Astra Control Center をホストするオペレーティングシステムが、環境の公式ドキュメントに記載されている基本的なリソース要件を満たしていることを確認します。Astra Control Center では、環境のリソース要件に加え、次のリソースが必要です。

コンポーネント	要件
ストレージバックエンドの容量	500GB以上の容量があります
ワーカーノード	少なくとも 3 つのワーカーノードがあり、それぞれ 4 つの CPU コアと 12GB の RAM が搭載されています
FQDN アドレス	Astra Control Center の FQDN アドレス

コンポーネント	要件
Astra Trident	Trident 21.10.1以降がインストールされ、SnapMirrorベースのアプリケーションレプリケーション用にAstra 22.07以降がインストールされて設定されている必要があります

これらの要件は、運用環境で実行されている唯一のアプリケーションが Astra Control Center であることを前提としています。環境で追加のアプリケーションを実行している場合は、それに応じてこれらの最小要件を調整します。

- * イメージレジストリ * ： Astra Control Center ビルドイメージをプッシュできる、既存のプライベート Docker イメージレジストリが必要です。イメージをアップロードするイメージレジストリの URL を指定する必要があります。
- * Astra Trident / ONTAP 構成 * ： Astra Control Center では、ストレージクラスを作成してデフォルトのストレージクラスとして設定する必要があります。Astra Control Center は、 Astra Trident が提供する次の ONTAP ドライバをサポートしています。
  - ONTAP - NAS
  - ONTAP - SAN
  - ONTAP - SAN - 経済性

OpenShift 環境でのアプリケーションのクローニングでは、 Astra Control Center が OpenShift でボリュームをマウントし、ファイルの所有権を変更できるようにする必要があります。そのため、これらの処理を許可するには、 ONTAP ボリュームのエクスポートポリシーを設定する必要があります。次のコマンドを使用して実行できます。

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`

管理対象のコンピューティングリソースとして 2 つ目の OpenShift 運用環境を追加する場合は、 Astra Trident ボリューム Snapshot 機能を有効にする必要があります。Astra Trident でボリューム Snapshot を有効にしてテストするには、 "Astra Trident の公式ガイドをご覧ください"。

VMware Tanzu Kubernetes Grid クラスタの要件

VMware Tanzu Kubernetes Grid （ TKG ）または Tanzu Kubernetes Grid Integrated Edition （ TKGi ）クラスタで Astra Control Center をホストする場合、次の点に注意してください。

- TKG または TKGi のデフォルト・ストレージ・クラス・エンフォースメントは、 Astra Control によって管理されるすべてのアプリケーション・クラスタで無効にします。これを行うには、を編集します `TanzuKubernetesCluster` ネームスペースクラスタ上のリソース。
- Astra Control Centerのインストールの一部として、PoDセキュリティポリシー（PSP）の制限された環境で次のリソースが作成されます。
  - ポッドセキュリティポリシー

- RBACロール
- RBACロールRBACロールとRoleBindingリソースがに作成されます `netapp-acc` ネームスペース：

- TKG または TKGi 環境に Astra Control Center を導入する際には、 Astra Trident の特定の要件に注意してください。詳細については、を参照してください "Astra Trident のドキュメント"。

> デフォルトの VMware TKG および TKGi 設定ファイルトークンの有効期限は、展開後 10 時間です。Tanzu ポートフォリオ製品を使用する場合は、 Astra Control Center と管理対象アプリケーションクラスタ間の接続の問題を回避するために、期限切れにならないトークンを含む Tanzu Kubernetes Cluster 構成ファイルを生成する必要があります。手順については、を参照してください "VMware NSX -T Data Center 製品ドキュメント"

Google Anthosクラスタの要件

Google AnthosクラスタでAstra Control Centerをホストする場合、Google AnthosにはデフォルトでMetalLBロードバランサとIstio入力ゲートウェイサービスが含まれているため、インストール時にAstra Control Centerの一般的な入力機能を使用するだけで済みます。を参照してください "Astra Control Center を設定します" を参照してください。

サポートされるストレージバックエンド

Astra Control Center は、次のストレージバックエンドをサポートします。

- NetApp ONTAP 9.5 以降の AFF および FAS システム
- NetApp ONTAP 9.8以降AFF およびFAS システム：SnapMirrorベースのアプリケーションレプリケーションに使用
- NetApp Cloud Volumes ONTAP の略

Astra Control Centerを使用するには、必要な機能に応じて、次のONTAP ライセンスがあることを確認します。

- FlexClone
- SnapMirror：オプション。SnapMirrorテクノロジを使用してリモートシステムにレプリケートする場合にのみ必要です。を参照してください "SnapMirrorのライセンス情報"。
- S3ライセンス：オプション。ONTAP S3バケットにのみ必要です

ONTAP システムに必要なライセンスがあるかどうかを確認できます。を参照してください "ONTAP ライセンスを管理します"。

アプリケーションクラスタの要件

Astra Control Center には、 Astra Control Center から管理するクラスタに対する次の要件があります。これらの要件は、管理するクラスタが Astra Control Center をホストする運用環境クラスタである場合にも適用されます。

- Kubernetes の最新バージョン "Snapshot コントローラコンポーネント" がインストールされている
- Astra Trident "volumesnapshotclass オブジェクト" は管理者によって定義されています
- クラスタにはデフォルトの Kubernetes ストレージクラスが存在します

- Astra Trident を使用するように少なくとも 1 つのストレージクラスが設定されている

> アプリケーションクラスタにが必要です `kubeconfig.yaml` 1つの_context_elementだけを定義するファイル。の Kubernetes のドキュメントを参照してください "kubeconfig ファイルの作成に関する情報"。

> Rancher環境でアプリケーションクラスタを管理する場合は、でアプリケーションクラスタのデフォルトコンテキストを変更します `kubeconfig` rancher APIサーバコンテキストの代わりにコントロールプレーンコンテキストを使用するためにrancherによって提供されるファイル。これにより、 Rancher API サーバの負荷が軽減され、パフォーマンスが向上します。

アプリケーション管理の要件

Astra Control には、次のアプリケーション管理要件があります。

- * ライセンス * ： Astra Control Center を使用してアプリケーションを管理するには、 Astra Control Center ライセンスが必要です。
- * 名前空間 * ： Astra Control では、アプリケーションが複数の名前空間にまたがることはありませんが、名前空間には複数のアプリケーションを含めることができます。
- * StorageClass* ： StorageClass が明示的に設定されたアプリケーションをインストールし、そのアプリケーションをクローニングする必要がある場合、クローン処理のターゲットクラスタに最初に指定された StorageClass が必要です。明示的に StorageClass を設定したアプリケーションを、同じストレージクラスを使用しないクラスタにクローニングすると、失敗します。
- * Kubernetes リソース * ： Astra Control で収集されていない Kubernetes リソースを使用するアプリケーションには、アプリケーションのデータ管理機能がフル装備されていない可能性があります。Astra Control では、次の Kubernetes リソースが収集されます。

クラスタロール	ClusterRoleBinding	ConfigMap
cronjob	CustomResourceDefinition の場合	CustomResource の場合
デモンセット（ DemonSet ）	DeploymentConfig	HorizontalPodAutoscaler のように表示されます
入力	MutingWebhook	ネットワークポリシー
PersistentVolumeClaim のように表示され	ポッド	PodDisruptionBudget （予算の廃止）
PodTemplate	ReplicaSet	ロール
RoleBinding です	ルート	秘密
サービス	サービスアカウント	Stateful役立つ セット
検証 Webhook		

レプリケーションの前提条件

Astra Controlアプリケーションのレプリケーションを開始するには、次の前提条件を満たしている必要があります。

- シームレスな災害復旧を実現するために、第3の障害ドメインまたはセカンダリサイトにAstra Control Centerを導入することをお勧めします。
- アプリケーションのホストKubernetesクラスタとデスティネーションKubernetesクラスタが使用可能であり、2つのONTAP クラスタに接続されている必要があります。理想的には別々の障害ドメインまたはサイトに配置できます。
- ONTAP クラスタとホストSVMをペアリングする必要があります。を参照してください "クラスタと SVM のピアリングの概要"。
- ペアリングされているリモートSVMがデスティネーションクラスタのTridentで使用可能である必要があります。
- ソースとデスティネーションの両方のONTAP クラスタにTridentバージョン22.07以降が存在する必要があります。
- ソースとデスティネーションの両方のONTAP クラスタで、データ保護バンドルを使用したONTAP SnapMirror非同期ライセンスが有効になっている必要があります。を参照してください "ONTAP のSnapMirrorライセンスの概要"。
- ONTAP ストレージバックエンドをAstra Control Centerに追加する場合は、「admin」ロールでユーザクレデンシャルを適用します。このロールにはアクセス方法があります `http` および `ontapi` 両方のONTAP クラスタで有効にしてください。を参照してください "ユーザアカウントを管理する" を参照してください。
- ソースとデスティネーションの両方のKubernetesクラスタとONTAP クラスタをAstra Controlで管理する必要があります。

> （別のクラスタまたはサイトで実行されている）別のアプリケーションを逆方向に同時にレプリケートできます。たとえば、アプリケーションA、B、Cはデータセンター1からデータセンター2にレプリケートでき、アプリケーションX、Y、Zはデータセンター2からデータセンター1にレプリケートできます。

方法をご確認ください "SnapMirrorテクノロジを使用してアプリケーションをリモートシステムにレプリケート"。

サポートされているアプリケーションのインストール方法

Astra Control は、次のアプリケーションインストール方法をサポートしています。

- * マニフェストファイル * ： Astra Control は、 kubectl を使用してマニフェストファイルからインストールされたアプリケーションをサポートします。例：

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

- * Helm 3 * ： Helm を使用してアプリケーションをインストールする場合、 Astra Control には Helm バージョン 3 が必要です。Helm 3 （または Helm 2 から Helm 3 にアップグレード）を使用してインストールされたアプリケーションの管理とクローニングが完全にサポートされています。Helm 2 でインストールされたアプリケーションの管理はサポートされていません。
- * オペレータが導入したアプリケーション * ： Astra Control は、名前空間を対象とした演算子を使用してインストールされたアプリケーションをサポートします。このインストールモデルで検証されたアプリケーションには、次のものがあります。

- "Apache K8ssandra"
- "Jenkins CI"
- "Percona XtraDB クラスタ"

> インストールする演算子とアプリケーションは、同じ名前空間を使用する必要があります。このような名前空間を使用するには、演算子の deployment.yaml ファイルを変更する必要があります。

インターネットにアクセスできます

インターネットに外部からアクセスできるかどうかを確認する必要があります。この処理を行わないと、NetApp Cloud Insights からの監視データや指標データの受信や、へのサポートバンドルの送信など、一部の機能が制限される可能性があります "NetApp Support Site"。

使用許諾

Astra Control Center の全機能を使用するには、 Astra Control Center ライセンスが必要です。評価用ライセンスまたはフルライセンスをネットアップから取得する。アプリケーションとデータを保護するにはライセンスが必要です。を参照してください "Astra Control Centerの機能" を参照してください。

Astra Control Centerには、評価用ライセンスをお試しいただけます。このライセンスは、Astra Control Centerをダウンロードした日から90日間使用できます。登録すると、無償トライアルに登録できます "こちらをご覧ください"。

ONTAP ストレージバックエンドに必要なライセンスの詳細については、を参照してください "サポートされるストレージバックエンド"。

ライセンスの機能の詳細については、を参照してください "ライセンス"。

オンプレミス **Kubernetes** クラスタへの入力

ネットワーク入力アストラコントロールセンターで使用するタイプを選択できます。デフォルトでは、 Astra Control Center は Astra Control Center ゲートウェイ（サービス / traefik ）をクラスタ全体のリソースとして展開します。また、お客様の環境でサービスロードバランサが許可されている場合は、 Astra Control Center でサービスロードバランサの使用もサポートされます。サービスロードバランサを使用する必要があり、設定済みでない場合は、 MetalLB ロードバランサを使用して外部 IP アドレスを自動的にサービスに割り当てることができます。内部 DNS サーバ構成では、 Astra Control Center に選択した DNS 名を、負荷分散 IP アドレスに指定する必要があります。

> Tanzu Kubernetes GridクラスタでAstra Control Centerをホストしている場合は、を使用します `kubectl get nsxlbmonitors -A` 入力トラフィックを受け入れるように設定されたサービスモニタがすでにあるかどうかを確認するコマンド。MetalLB が存在する場合は、既存のサービスモニタが新しいロードバランサ設定を上書きするため、 MetalLB をインストールしないでください。

詳細については、を参照してください "ロードバランシング用の入力を設定します"。

ネットワーク要件

Astra Control Center をホストする運用環境は、次の TCP ポートを使用して通信します。これらのポートがファイアウォールを通過できることを確認し、 Astra ネットワークからの HTTPS 出力トラフィックを許可するようにファイアウォールを設定する必要があります。一部のポートでは、 Astra Control Center をホストする環境と各管理対象クラスタ（該当する場合はメモ）の両方の接続方法が必要です。

> Astra Control Center はデュアルスタック Kubernetes クラスタに導入でき、 Astra Control Center はデュアルスタック操作用に構成されたアプリケーションとストレージバックエンドを管理できます。デュアルスタッククラスタの要件の詳細については、を参照してください "Kubernetes のドキュメント"。

ソース	宛先	ポート	プロトコル	目的
クライアント PC	Astra Control Center の略	443	HTTPS	UI / API アクセス - Astra Control Center をホストしているクラスタと各管理対象クラスタの間で、このポートが双方向に開いていることを確認します
指標利用者	Astra Control Center ワーカーノード	9090	HTTPS	メトリックデータ通信 - 各管理対象クラスタが、アストラコントロールセンターをホストしているクラスタ上のこのポートにアクセスできることを確認します （双方向通信が必要）
Astra Control Center の略	Hosted Cloud Insights サービスの略	443	HTTPS	Cloud Insights 通信
Astra Control Center の略	Amazon S3 ストレージバケットプロバイダ	443	HTTPS	Amazon S3 ストレージ通信
Astra Control Center の略	NetApp AutoSupport	443	HTTPS	NetApp AutoSupport 通信

サポートされている **Web** ブラウザ

Astra Control Center は、最新バージョンの Firefox 、 Safari 、 Chrome をサポートし、解像度は 1280 x 720 以上です。

次のステップ

を表示します "クイックスタート" 概要（ Overview ）：

Astra Control Center のクイックスタート

このページでは、 Astra Control Center の導入に必要な手順の概要を説明します。各ステップ内のリンクから、詳細が記載されたページに移動できます。

ぜひお試しください。Astra Control Center を試す場合は、 90 日間の評価ライセンスを使用できます。を参照してください "ライセンス情報" を参照してください。

1 **Kubernetes** クラスタの要件を確認

- Astra は、 Trident が設定された ONTAP ストレージバックエンドまたは Astra データストアストレージバックエンドを使用して、 Kubernetes クラスタと連携します。
- クラスタが正常な状態で稼働し、少なくとも 3 つのオンラインワーカーノードが必要です。
- クラスタで Kubernetes が実行されている必要があります。

の詳細については、を参照してください "Astra Control Center の要件"。

2 **Astra Control Center**をダウンロードしてインストールします

- から Astra Control Center をダウンロードします "NetApp Support Siteの「 Astra Control Center Downloads 」ページ"。
- Astra Control Center をローカル環境にインストールします。

  必要に応じて、 Red Hat OperatorHub を使用して Astra Control Center をインストールします。

  必要に応じて、Cloud Volumes ONTAP ストレージバックエンドにAstra Control Centerをインストールします。

の詳細を確認してください "Astra Control Centerのインストール"。

3 いくつかの初期セットアップ作業を完了します

- Astra Controlライセンスとサポートする ONTAP ライセンスを追加
- Kubernetes クラスタと Astra Control Center を追加すると、詳細が検出されます。
- ONTAP ストレージバックエンドを追加します。
- 必要に応じて、アプリケーションバックアップを格納するオブジェクトストアバケットを追加します。

の詳細については、を参照してください "初期セットアッププロセス"。

4 **Astra Control Center** を使用

Astra Control Center のセットアップが完了したら、次の手順を実行します。

- アプリを管理します。方法の詳細については、こちらをご覧ください "アプリの管理"。

- アプリの保護ポリシーの設定、リモートシステムへのアプリのレプリケーション、アプリのクローニングと移行により、アプリを保護します。方法の詳細については、こちらをご覧ください "アプリを保護します"。
- アカウントの管理（ユーザ、ロール、ユーザ認証用のLDAP、クレデンシャル、リポジトリ接続など）方法の詳細については、こちらをご覧ください "ユーザを管理します"。
- 必要に応じて、 NetApp Cloud Insights に接続し、 Astra Control Center UI 内のシステム、容量、およびスループットの健全性に関する指標を表示します。の詳細を確認してください "Cloud Insights に接続しています"。

5 このクイックスタートから続行します

"Astra Control Center をインストールします"。

詳細については、こちらをご覧ください

- "Astra Control API を使用"

インストールの概要

次の Astra Control Center のインストール手順のいずれかを選択して実行します。

- "標準の手順で Astra Control Center をインストールします"
- "（ Red Hat OpenShift を使用する場合） OpenShift OperatorHub を使用して Astra Control Center をインストールします"
- "Cloud Volumes ONTAP ストレージバックエンドに Astra Control Center をインストールします"

標準の手順で **Astra Control Center** をインストールします

Astra Control Center をインストールするには、NetApp Support Siteからインストールバンドルをダウンロードし、次の手順を実行して、 Astra Control Center Operator と Astra Control Center を環境にインストールします。この手順を使用して、インターネット接続環境またはエアギャップ環境に Astra コントロールセンターをインストールできます。

Red Hat OpenShift環境では、を使用できます "代替手順" OpenShift OperatorHub を使用して Astra Control Center をインストールします。

必要なもの

- "インストールを開始する前に、 Astra Control Center の導入環境を準備します"。
- 使用環境でポッドセキュリティポリシーを設定または設定したい場合は、ポッドセキュリティポリシーと、それらがAstra Control Centerのインストールに与える影響について理解しておいてください。を参照してください "ポッドのセキュリティポリシーの制限事項を理解します"。
- すべてのクラスタオペレータが正常な状態であり、使用可能であることを確認します。

```
kubectl get clusteroperators
```

- すべての API サービスが正常な状態であり、使用可能であることを確認します。

```
kubectl get apiservices
```

- 使用するネットアップFQDNがこのクラスタにルーティング可能であることを確認します。つまり、内部 DNS サーバに DNS エントリがあるか、すでに登録されているコア URL ルートを使用しています。
- クラスタにcert-managerがすでに存在する場合は、いくつかを実行する必要があります "事前に必要な手順" そのため、Astra Control Centerは独自の証明書管理ツールをインストールしません。

このタスクについて

Astra Control Center のインストールプロセスでは、次のことが実行されます。

- Astraコンポーネントをにインストールします `netapp-acc` （またはカスタム名）ネームスペース。
- デフォルトアカウントを作成します。
- デフォルトの管理ユーザのEメールアドレスとデフォルトのワンタイムパスワードを設定します。このユーザには、UIへの初回ログインに必要なシステムのオーナーロールが割り当てられます。
- Astra Control Center のすべてのポッドが実行されていることを確認するのに役立ちます。
- Astra の UI をインストールします。

> （環境 the Astra Data Store Early Access Program（EAP）リリースのみ）Astra Control Center を使用してAstraデータストアを管理し、VMwareワークフローを有効にする場合は、Astra Control Centerのみをに導入します `pcloud` ではなく、ネームスペースで指定します `netapp-acc` この手順 の手順で説明するネームスペースまたはカスタムネームスペース。

> すべてのAstra Control Centerポッドが削除されないようにするため、インストールプロセス全体で次のコマンドを実行しないでください。 `kubectl delete -f astra_control_center_operator_deploy.yaml`

> Docker Engine の代わりに Red Hat の Podman を使用している場合は、 Docker コマンドの代わりに Podman コマンドを使用できます。

手順

Astra Control Center をインストールするには、次の手順に従います。

- Astra Control Centerバンドルをダウンロードして開梱します
- ネットアップAstra kubectlプラグインをインストール
- [イメージをローカルレジストリに追加します]
- [認証要件を持つレジストリのネームスペースとシークレットを設定します]
- Astra Control Center オペレータを設置します
- Astra Control Center を設定します
- Astra Control Center とオペレータのインストールを完了します
- [システムステータスを確認します]

- [ロードバランシング用の入力を設定します]
- Astra Control Center UI にログインします

Astra Control Centerバンドルをダウンロードして開梱します

1. Astra Control Center バンドルをダウンロードします (`astra-control-center-[version].tar.gz`) をクリックします "NetApp Support Site"。
2. から Astra Control Center 証明書とキーの zip をダウンロードします "NetApp Support Site"。
3. （任意）次のコマンドを使用して、バンドルのシグニチャを確認します。

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

4. 画像を抽出します。

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

ネットアップ**Astra kubectl**プラグインをインストール

ネットアップアストラ `kubectl` コマンドラインプラグインにより、Astra Control Centerの導入とアップグレードに関連する一般的なタスクを実行する時間を節約できます。

必要なもの

ネットアップでは、プラグイン用のバイナリを提供しており、CPUアーキテクチャやオペレーティングシステムが異なる場合はそのプラグインをこのタスクを実行する前に、使用しているCPUとオペレーティングシステムを把握しておく必要があります。LinuxおよびMacオペレーティングシステムでは、を使用できます `uname -a` この情報を収集するためのコマンドです。

手順

1. 使用可能なネットアップAstraを選択します `kubectl` プラグインバイナリ。オペレーティングシステムとCPUアーキテクチャに必要なファイルの名前をメモします。

```
ls kubectl-astra/
```

2. ファイルを標準と同じ場所にコピーします `kubectl` ユーティリティ。この例では、を使用しています `kubectl` ユーティリティはにあります `/usr/local/bin` ディレクトリ。交換してください `<binary-name>` 必要なファイル名を使用して、次の操作を実行します。

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

イメージをローカルレジストリに追加します

1. コンテナエンジンに応じた手順を実行します。

Docker です

1. Astraディレクトリに移動します。

```
cd acc
```

2. [[[[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1></Z1>_image_local_registry_push]]]]]]]]]</Z2>アストラControl Centerイメージディレクトリ内のパッケージイメージをローカルレジストリにプッシュします。</Z3>コマンドを実行する前に、次の置き換えを行ってください。
   - bundle_fileをAstra Controlバンドルファイルの名前に置き換えます（例： `acc.manifest.yaml` ）。
   - my_registryをDockerリポジトリのURLに置き換えます。
   - my_registry_userをユーザー名に置き換えます。
   - my_registry_tokenをレジストリの認証済みトークンに置き換えます。

```
kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY
-u MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
```

ポドマン

1. レジストリにログインします。

```
podman login [your_registry_path]
```

2. 次のスクリプトを実行して、コメントに記載されているように<your _registry>を置き換えます。

```
# You need to be at the root of the tarball.
# You should see these files to confirm correct location:
#   acc.manifest.yaml
#   acc/

# Replace <YOUR_REGISTRY> with your own registry (e.g
registry.customer.com or registry.customer.com/testing, etc..)
export REGISTRY=<YOUR_REGISTRY>
export PACKAGENAME=acc
export PACKAGEVERSION=22.08.1-26
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
  # Load to local cache
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //')

  # Remove path and keep imageName.
  astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')

  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/netapp/astra/${PACKAGENAME}
/${PACKAGEVERSION}/${astraImageNoPath}

  # Push to the local repo.
  podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

認証要件を持つレジストリのネームスペースとシークレットを設定します

1. Astra Control Centerホストクラスタ用のKUBECONFIGをエクスポートします。

```
export KUBECONFIG=[file path]
```

2. 認証が必要なレジストリを使用する場合は、次の手順を実行する必要があります。

   a. を作成します `netapp-acc-operator` ネームスペース：

```
kubectl create ns netapp-acc-operator
```

   対応：

```
namespace/netapp-acc-operator created
```

b. のシークレットを作成します `netapp-acc-operator` ネームスペース：Docker 情報を追加して次のコマンドを実行します。

> プレースホルダ `your_registry_path` 以前にアップロードした画像の場所と一致する必要があります（例： `[Registry_URL]/netapp/astra/astracc/22.08.1-26` ）。

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-operator --docker-server=[your_registry_path] --docker-username=[username] --docker-password=[token]
```

回答例：

```
secret/astra-registry-cred created
```

> シークレットの生成後にネームスペースを削除する場合は、ネームスペースが再作成されたあとにネームスペースのシークレットを再生成する必要があります。

c. を作成します `netapp-acc` （またはカスタムの名前付き）ネームスペース。

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

回答例：

```
namespace/netapp-acc created
```

d. のシークレットを作成します `netapp-acc` （またはカスタムの名前付き）ネームスペース。Docker 情報を追加して次のコマンドを実行します。

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-acc or custom namespace] --docker-server=[your_registry_path] --docker-username=[username] --docker-password=[token]
```

応答

```
secret/astra-registry-cred created
```

a. [[[[sup_kubeconfig _secret]] （オプション）インストール後に Astra Control Center でクラスタを自動的に管理する場合は、このコマンドを使用して展開する Astra Control Center ネームスペース内のシークレットとして kubeconfig を指定する必要があります。

```
kubectl create secret generic [acc-kubeconfig-cred or custom secret
name] --from-file=<path-to-your-kubeconfig> -n [netapp-acc or custom
namespace]
```

Astra Control Center オペレータを設置します

1. ディレクトリを変更します。

```
cd manifests
```

2. Astra Control Centerオペレータ配置YAMLを編集します (`astra_control_center_operator_deploy.yaml`)を参照して、ローカルレジストリとシークレットを参照してください。

```
vim astra_control_center_operator_deploy.yaml
```

> 注釈付きサンプルYAMLは以下の手順に従います。

a. 認証が必要なレジストリを使用する場合は、のデフォルト行を置き換えます `imagePullSecrets: []` 次の条件を満たす場合：

```
imagePullSecrets:
- name: <astra-registry-cred>
```

b. 変更 `[your_registry_path]` をクリックします `kube-rbac-proxy` でイメージをプッシュしたレジストリパスへのイメージ 前の手順。

c. 変更 `[your_registry_path]` をクリックします `acc-operator-controller-manager` でイメージをプッシュしたレジストリパスへのイメージ 前の手順。

d. （Astra データストアプレビューを使用するインストールの場合）この問題に関する既知の情報を参照してください "ストレージクラスのプロビジョニングと YAML に対する追加の変更"。

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

3. Astra Control Center オペレータをインストールします。

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

回答例：

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

4. ポッドが実行中であることを確認します

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

Astra Control Center を設定します

1. Astra Control Centerカスタムリソース（CR）ファイルを編集します (`astra_control_center_min.yaml`)アカウント、AutoSupport 、レジストリ、およびその他の必要な設定を行うには、次の手順を実行します。

> `astra_control_center_min.yaml` はデフォルトのCRで、ほとんどのインストールに適しています。すべてのことをよく理解してください "CRオプションとその可能性のある値" お客様の環境にAstra Control Centerを正しく導入できるようにするため。環境で追加のカスタマイズが必要な場合は、を使用できます `astra_control_center.yaml` 代替CRとして。

```
vim astra_control_center_min.yaml
```

> 許可が不要なレジストリを使用している場合は、を削除する必要があります `secret` ラインの内側 `imageRegistry` または、インストールが失敗します。

a. 変更 `[your_registry_path]` 前の手順でイメージをプッシュしたレジストリパスに移動します。

b. を変更します `accountName` stringには、アカウントに関連付ける名前を指定します。

c. を変更します `astraAddress` ブラウザからAstraにアクセスする際に使用するFQDNを文字列で指定します。使用しないでください `http://` または `https://` をクリックします。この FQDN をコピーしてで使用します 後の手順。

d. を変更します `email` デフォルトの初期管理者アドレスを表す文字列。この E メールアドレスをコピーしてで使用します 後の手順。

e. 変更 `enrolled` を選択しますAutoSupport `false` インターネットに接続されていないか、または保持されているサイト `true` 接続されているサイト用。

f. 外部証明書マネージャを使用する場合は、に次の行を追加します `spec`：

```
spec:
  crds:
    externalCertManager: true
```

g. （オプション）名を追加します `firstName` 姓を入力します `lastName` アカウントに関連付けられているユーザのこの手順は、 UI ですぐに実行することもあとで実行することもできます。

h. （オプション）を変更します `storageClass` インストールに必要な場合、別のTridentストレージクラスリソースへの値。

i. （オプション）インストール後に Astra Control Center でクラスタを自動的に管理する場合は このクラスタの kubeconfig を含むシークレットを作成しましたをクリックし、という名前のYAMLファイルに新しいフィールドを追加して、シークレットの名前を指定します `astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"`

j. 次のいずれかの手順を実行します。

- * その他の入力コントローラ（ ingressType: Generic ） * ：これはアストラコントロールセンターでのデフォルトのアクションです。Astra Control Center を展開したら、 Astra Control Center を URL で公開するように入力コントローラを設定する必要があります。

  デフォルトのAstra Control Centerインストールでは、ゲートウェイがセットアップされます（`service/traefik`）を入力します `ClusterIP`。このデフォルトのインストールでは、トラフィックをルーティングするために Kubernetes IngressController/Ingress を追加で設定する必要があります。入力を使用する場合は、を参照してください "ロードバランシング用の入力を設定します"。

- サービスロードバランサ（**ingressType: AccTrafik**）: IngressControllerをインストールしない場合、または入力リソースを作成しない場合は、を設定します `ingressType` 終了：`AccTraefik`。

  これにより、Astra Control Centerが導入されます `traefik` Gateway as a Kubernetes LoadBalancer type serviceの略。

  Astra Control Centerは、タイプ「LoadBalancer」のサービスを使用します。（`svc/traefik` Astra Control Centerの名前空間）で、アクセス可能な外部IPアドレスが割り当てられている必要があります。お使いの環境でロードバランサが許可されていて、設定されていない場合は、MetalLB または別の外部サービスロードバランサを使用して、外部 IP アドレスをサービスに割り当てることができます。内部 DNS サーバ構成では、 Astra Control Center に選択した DNS 名を、負荷分散 IP アドレスに指定する必要があります。

サービスタイプ「 LoadBalancer 」および入力の詳細については、を参照してください "要件"。

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"
  ingressType: "Generic"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
```

Astra Control Center とオペレータのインストールを完了します

1. 前の手順でまだ行っていない場合は、を作成します `netapp-acc` （またはカスタム）ネームスペース：

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

回答例：

```
namespace/netapp-acc created
```

2. にAstra Control Centerをインストールします `netapp-acc` （またはカスタムの）ネームスペース：

```
kubectl apply -f astra_control_center_min.yaml -n [netapp-acc or custom
namespace]
```

回答例：

```
astracontrolcenter.astra.netapp.io/astra created
```

システムステータスを確認します

> OpenShift を使用する場合は、同等の OC コマンドを検証手順に使用できます。

1. すべてのシステムコンポーネントが正常にインストールされたことを確認します。

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

各ポッドのステータスがになっている必要があります `Running`。システムポッドが展開されるまでに数分かかることがあります。

回答例

```
NAME                                        READY   STATUS    RESTARTS
AGE
acc-helm-repo-6b44d68d94-d8m55              1/1     Running   0
13m
activity-78f99ddf8-hltct                    1/1     Running   0
10m
api-token-authentication-457nl              1/1     Running   0
9m28s
api-token-authentication-dgwsz              1/1     Running   0
9m28s
api-token-authentication-hmqqc              1/1     Running   0
9m28s
asup-75fd554dc6-m6qzh                       1/1     Running   0
9m38s
authentication-6779b4c85d-92gds             1/1     Running   0
8m11s
bucketservice-7cc767f8f8-lqwr8              1/1     Running   0
9m31s
certificates-549fd5d6cb-5kmd6               1/1     Running   0
9m56s
certificates-549fd5d6cb-bkjh9               1/1     Running   0
9m56s
cloud-extension-7bcb7948b-hn8h2             1/1     Running   0
10m
cloud-insights-service-56ccf86647-fgg69     1/1     Running   0
9m46s
composite-compute-677685b9bb-7vgsf          1/1     Running   0
10m
composite-volume-657d6c5585-dnq79           1/1     Running   0
9m49s
credentials-755fd867c8-vrlmt                1/1     Running   0
11m
entitlement-86495cdf5b-nwhh2                1/1     Running   2
10m
features-5684fb8b56-8d6s8                   1/1     Running   0
10m
fluent-bit-ds-rhx7v                         1/1     Running   0
7m48s
fluent-bit-ds-rjms4                         1/1     Running   0
7m48s
fluent-bit-ds-zf5ph                         1/1     Running   0
7m48s
graphql-server-66d895f544-w6hjd             1/1     Running   0
```

```
3m29s
identity-744df448d5-rlcmm                1/1     Running   0
10m
influxdb2-0                              1/1     Running   0
13m
keycloak-operator-75c965cc54-z7csw       1/1     Running   0
8m16s
krakend-798d6df96f-9z2sk                 1/1     Running   0
3m26s
license-5fb7d75765-f8mjg                 1/1     Running   0
9m50s
login-ui-7d5b7df85d-l2s7s                1/1     Running   0
3m20s
loki-0                                   1/1     Running   0
13m
metrics-facade-599b9d7fcc-gtmgl          1/1     Running   0
9m40s
monitoring-operator-67cc74f844-cdplp     2/2     Running   0
8m11s
nats-0                                   1/1     Running   0
13m
nats-1                                   1/1     Running   0
13m
nats-2                                   1/1     Running   0
12m
nautilus-769f5b74cd-k5jxm                1/1     Running   0
9m42s
nautilus-769f5b74cd-kd9gd                1/1     Running   0
8m59s
openapi-84f6ccd8ff-76kvp                 1/1     Running   0
9m34s
packages-6f59fc67dc-4g2f5                1/1     Running   0
9m52s
polaris-consul-consul-server-0           1/1     Running   0
13m
polaris-consul-consul-server-1           1/1     Running   0
13m
polaris-consul-consul-server-2           1/1     Running   0
13m
polaris-keycloak-0                       1/1     Running   0
8m7s
polaris-keycloak-1                       1/1     Running   0
5m49s
polaris-keycloak-2                       1/1     Running   0
5m15s
polaris-keycloak-db-0                    1/1     Running   0
```

```
8m6s
polaris-keycloak-db-1                     1/1     Running   0
5m49s
polaris-keycloak-db-2                     1/1     Running   0
4m57s
polaris-mongodb-0                         2/2     Running   0
13m
polaris-mongodb-1                         2/2     Running   0
12m
polaris-mongodb-2                         2/2     Running   0
12m
polaris-ui-565f56bf7b-zwr8b               1/1     Running   0
3m19s
polaris-vault-0                           1/1     Running   0
13m
polaris-vault-1                           1/1     Running   0
13m
polaris-vault-2                           1/1     Running   0
13m
public-metrics-6d86d66444-2wbzl           1/1     Running   0
9m30s
storage-backend-metrics-77c5d98dcd-dbhg5 1/1     Running   0
9m44s
storage-provider-78c885f57c-6zcv4         1/1     Running   0
9m36s
telegraf-ds-2l2m9                         1/1     Running   0
7m48s
telegraf-ds-qfzgh                         1/1     Running   0
7m48s
telegraf-ds-shrms                         1/1     Running   0
7m48s
telegraf-rs-bjpkt                         1/1     Running   0
7m48s
telemetry-service-6684696c64-qzfdf        1/1     Running   0
10m
tenancy-6596b6c54d-vmpsm                  1/1     Running   0
10m
traefik-7489dc59f9-6mnst                  1/1     Running   0
3m19s
traefik-7489dc59f9-xrkgg                  1/1     Running   0
3m4s
trident-svc-6c8dc458f5-jswcl              1/1     Running   0
10m
vault-controller-6b954f9b76-gz9nm         1/1     Running   0
11m
```

2. （オプション）インストールが完了したことを確認するには、を参照してください `acc-operator` 次のコマンドを使用してログを作成します。

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

> `accHost` クラスタの登録は最後の処理の1つです。登録に失敗しても原因 の導入は失敗しません。ログにクラスタ登録エラーが示された場合は、クラスタ追加ワークフローを通じて再度登録を試行できます "UI で" または API 。

3. すべてのポッドが実行中の場合は、インストールが正常に完了したことを確認します (`READY` はです `True`）を使用してAstra Control Centerにログインする際に使用するワンタイムパスワードを取得します。

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

対応：

```
NAME    UUID                                      VERSION     ADDRESS
READY
astra   ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  22.08.1-26
10.111.111.111  True
```

> UUIDの値をコピーします。パスワードはです `ACC-` 続けてUUIDの値を指定します (`ACC-[UUID]` または、この例では、 `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`）。

ロードバランシング用の入力を設定します

Kubernetes 入力コントローラをセットアップして、クラスタのロードバランシングなどのサービスへの外部アクセスを管理できます。

この手順 では、入力コントローラの設定方法について説明します (`ingressType:Generic`）。これは、Astra Control Center でのデフォルトのアクションです。Astra Control Center を展開したら、 Astra Control Center を URL で公開するように入力コントローラを設定する必要があります。

> 入力コントローラを設定しない場合は、を設定できます `ingressType:AccTraefik`)。Astra Control Centerは、タイプ「LoadBalancer」のサービスを使用します。（`svc/traefik` Astra Control Centerの名前空間）で、アクセス可能な外部IPアドレスが割り当てられている必要があります。お使いの環境でロードバランサが許可されていて、設定されていない場合は、MetalLB または別の外部サービスロードバランサを使用して、外部 IP アドレスをサービスに割り当てることができます。内部 DNS サーバ構成では、 Astra Control Center に選択した DNS 名を、負荷分散 IP アドレスに指定する必要があります。サービスタイプ「 LoadBalancer 」および入力の詳細については、を参照してください "要件"。

この手順は、使用する入力コントローラのタイプによって異なります。

- Istio入力
- nginx 入力コントローラ
- OpenShift 入力コントローラ

必要なもの

- が必要です "入力コントローラ" すでに導入されている必要があります。
- 。 "入力クラス" 入力コントローラに対応するものがすでに作成されている必要があります。
- V1.19 と v1.22 の間で Kubernetes のバージョンを使用している。

Istio Ingressの手順

1. Istio Ingressを設定します。

   この手順 では、「デフォルト」の構成プロファイルを使用してIstioが導入されていることを前提としています。

2. 入力ゲートウェイに必要な証明書と秘密鍵ファイルを収集または作成します。

   CA署名証明書または自己署名証明書を使用できます。共通名はAstraアドレス（FQDN）である必要があります。

   コマンド例：

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout tls.key -out tls.crt
```

3. シークレットを作成します `tls secret name` を入力します `kubernetes.io/tls` でTLS秘密鍵と証明書を使用する場合 `istio-system namespace` TLSシークレットで説明されているように、

   コマンド例：

```
kubectl create secret tls [tls secret name]
--key="tls.key"
--cert="tls.crt" -n istio-system
```

   シークレットの名前はと一致する必要があります `spec.tls.secretName` で提供されます `istio-ingress.yaml` ファイル。

4. 入力リソースをに配置します `netapp-acc` （またはカスタムネームド）ネームスペースで、v1beta1（Kubernetesバージョンで1.22未満で廃止）またはv1リソースタイプを使用して、非推奨または新しいスキーマのいずれかに対応します。

   出力：

```
apiVersion: networking.k8s.io/v1beta1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress
  namespace: istio-system
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC addess>
    secretName: [tls secret name]
  rules:
  - host: [ACC addess]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          serviceName: traefik
          servicePort: 80
```

v1の新しいスキーマについては、次の例を参照してください。

```
kubectl apply -f istio-Ingress.yaml
```

出力：

```
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  namespace: istio-system
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC addess>
    secretName: [tls secret name]
  rules:
  - host: [ACC addess]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: traefik
            port:
              number: 80
```

5. 通常どおりAstra Control Centerを導入します。
6. 入力のステータスを確認します。

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

対応：

```
NAME    CLASS HOSTS              ADDRESS         PORTS   AGE
ingress istio astra.example.com 172.16.103.248  80, 443 1h
```

Nginx Ingress Controller の手順

1. タイプのシークレットを作成します[`kubernetes.io/tls`]をクリックします `netapp-acc` （またはカス

タム名前付き）ネームスペース。を参照してください "TLS シークレット"。

2. 入力リソースをに配置します `netapp-acc` （またはカスタムの名前付き）ネームスペースのいずれかを使用します `v1beta1` （Kubernetesのバージョンが1.22より前の場合は廃止）または `v1` 非推奨または新しいスキーマのリソースタイプ：

   a. に設定します `v1beta1` 廃止されたスキーマの例を次に示します。

```
apiVersion: extensions/v1beta1
Kind: IngressClass
metadata:
  name: ingress-acc
  namespace: [netapp-acc or custom namespace]
  annotations:
    kubernetes.io/ingress.class: [class name for nginx controller]
spec:
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - backend:
        serviceName: traefik
        servicePort: 80
        pathType: ImplementationSpecific
```

   b. をクリックします `v1` 新しいスキーマの例を次に示します。

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC addess>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

OpenShift 入力コントローラの手順

1. 証明書を調達し、 OpenShift ルートで使用できるようにキー、証明書、および CA ファイルを取得します。
2. OpenShift ルートを作成します。

```
oc create route edge --service=traefik
--port=web -n [netapp-acc or custom namespace]
--insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem
```

Astra Control Center UI にログインします

Astra Control Center をインストールした後、デフォルトの管理者のパスワードを変更し、 Astra Control Center UI ダッシュボードにログインします。

手順

1. ブラウザで、で使用したFQDNを入力します `astraAddress` を参照してください `astra_control_center_min.yaml` CR When（時間） Astra Control Center をインストールした。
2. プロンプトが表示されたら、自己署名証明書を受け入れます。

カスタム証明書はログイン後に作成できます。

3. Astra Control Centerのログインページで、に使用した値を入力します `email` インチ `astra_control_center_min.yaml` CR When（時間） Astra Control Center をインストールしたをクリックし、続けてワンタイムパスワードを入力します (`ACC-[UUID]`）。

   誤ったパスワードを 3 回入力すると、管理者アカウントは 15 分間ロックされます。

4. [**Login**] を選択します。
5. プロンプトが表示されたら、パスワードを変更します。

   初めてログインする際にパスワードを忘れた場合、他の管理ユーザアカウントがまだ作成されていないときは、ネットアップのサポートに問い合わせて、パスワードのリカバリに関するサポートを依頼してください。

6. （オプション）既存の自己署名 TLS 証明書を削除して、に置き換えます "認証局（ CA ）が署名したカスタム TLS 証明書"。

インストールのトラブルシューティングを行います

いずれかのサービスがにある場合 `Error` ステータスを確認すると、ログを調べることができます。400 ~ 500 の範囲の API 応答コードを検索します。これらは障害が発生した場所を示します。

手順

1. Astra Control Center のオペレータログを調べるには、次のように入力します。

```
kubectl logs --follow -n netapp-acc-operator $(kubectl get pods -n
netapp-acc-operator -o name) -c manager
```

次のステップ

を実行して導入を完了します "セットアップのタスク"。

=
:allow-uri-read:

ポッドのセキュリティポリシーの制限事項を理解します

Astra Control Centerは、PoDセキュリティポリシー（PSP）による特権制限をサポートします。ポッドセキュリティポリシーを使用すると、ユーザまたはグループがコンテナを実行できる対象や、それらのコンテナに付与できる権限を制限できます。

RKE2などの一部のKubernetesディストリビューションには、デフォルトのポッドセキュリティポリシーが用意されていますが、制限が厳しく、Astra Control Centerのインストール時に問題が発生します。

ここに記載されている情報と例を使用して、Astra Control Centerによって作成されるポッドセキュリティポリシーを理解し、Astra Control Centerの機能を妨げずに必要な保護を提供するポッドセキュリティポリシーを設定できます。

Astra Control Centerによってインストールされた**PSP**

Astra Control Centerは、インストール中にポッドのセキュリティポリシーをいくつか作成します。これらの中には永続的なものもあれば、一部のものは特定の処理中に作成され、処理が完了すると削除されます。

インストール中に作成された**PSP**

Astra Control Centerのインストール中に、Astra Control Centerオペレータは、Astra Control CenterネームスペースへのAstra Control Centerサービスの展開をサポートするために、カスタムのPoDセキュリティポリシー、Roleオブジェクト、およびRoleBindingオブジェクトをインストールします。

新しいポリシーとオブジェクトには次の属性があります。

```
kubectl get psp

NAME                            PRIV    CAPS          SELINUX    RUNASUSER
FSGROUP     SUPGROUP    READONLYROOTFS   VOLUMES
avp-psp                         false                 RunAsAny   RunAsAny
RunAsAny    RunAsAny    false            *
netapp-astra-deployment-psp     false                 RunAsAny   RunAsAny
RunAsAny    RunAsAny    false            *

kubectl get role

NAME                                     CREATED AT
netapp-astra-deployment-role             2022-06-27T19:34:58Z

kubectl get rolebinding

NAME                                     ROLE
AGE
netapp-astra-deployment-rb               Role/netapp-astra-deployment-role
32m
```

バックアップ処理中に作成された**PSP**

バックアップ処理中に、Astra Control Centerは、動的なPODセキュリティポリシー、ClusterRoleオブジェクト、およびRoleBindingオブジェクトを作成します。これらの機能により、別のネームスペースで実行されるバックアッププロセスがサポートされます。

新しいポリシーとオブジェクトには次の属性があります。

```
kubectl get psp

NAME                            PRIV    CAPS
SELINUX    RUNASUSER          FSGROUP     SUPGROUP    READONLYROOTFS
VOLUMES
netapp-astra-backup             false   DAC_READ_SEARCH
RunAsAny   RunAsAny           RunAsAny    RunAsAny    false            *

kubectl get role

NAME                   CREATED AT
netapp-astra-backup    2022-07-21T00:00:00Z

kubectl get rolebinding

NAME                   ROLE                        AGE
netapp-astra-backup    Role/netapp-astra-backup    62s
```

クラスタ管理中に作成された**PSP**

クラスターを管理する場合、Astra Control Centerは管理対象クラスタにNetApp Monitoringオペレータをインストールします。この演算子は、PODセキュリティポリシー、ClusterRoleオブジェクト、およびRoleBindingオブジェクトを作成して、Astra Control Center名前空間にテレメトリサービスを展開します。

新しいポリシーとオブジェクトには次の属性があります。

```
kubectl get psp

NAME                            PRIV    CAPS
SELINUX    RUNASUSER          FSGROUP     SUPGROUP    READONLYROOTFS
VOLUMES
netapp-monitoring-psp-nkmo      true    AUDIT_WRITE,NET_ADMIN,NET_RAW
RunAsAny   RunAsAny           RunAsAny    RunAsAny    false            *

kubectl get role

NAME                                            CREATED AT
netapp-monitoring-role-privileged               2022-07-21T00:00:00Z

kubectl get rolebinding

NAME                                                    ROLE
AGE
netapp-monitoring-role-binding-privileged               Role/netapp-
monitoring-role-privileged              2m5s
```

ネームスペース間のネットワーク通信を有効にします

一部の環境では、NetworkPolicy構造体を使用してネームスペース間のトラフィックを制限します。Astra Control Centerオペレータ、Astra Control Center、およびAstra Plugin for VMware vSphereはすべて異なる名前空間に存在します。これらの異なるネームスペース内のサービスは、相互に通信できる必要があります。この通信をイネーブルにするには、次の手順を実行します。

手順

1. Astra Control Center名前空間に存在するNetworkPolicyリソースをすべて削除します。

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

2. 前のコマンドで返された各NetworkPolicyオブジェクトについて、次のコマンドを使用して削除します。object_name >を、返されたオブジェクトの名前に置き換えます。

```
kubectl delete networkpolicy <OBJECT_NAME> -n netapp-acc
```

3. 次のリソースファイルを適用して、Astra Plugin for VMware vSphereサービスがAstra Control Centerサービスに要求を送信できるように、accネットワーキングポリシーオブジェクトを設定します。角かっこ内の情報を環境内の情報に置き換えます。

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: acc-avp-network-policy
  namespace: <ACC_NAMESPACE_NAME> # REPLACE THIS WITH THE ASTRA CONTROL
CENTER NAMESPACE NAME
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              kubernetes.io/metadata.name: <PLUGIN_NAMESPACE_NAME> #
REPLACE THIS WITH THE ASTRA PLUGIN FOR VMWARE VSPHERE NAMESPACE NAME
```

4. 次のリソースファイルを適用して、Astra Control CenterオペレータがAstra Control Centerサービスと通信できるように、acc-operator-network-policyオブジェクトを設定します。角かっこ内の情報を環境内の情報に置き換えます。

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: acc-operator-network-policy
  namespace: <ACC_NAMESPACE_NAME> # REPLACE THIS WITH THE ASTRA CONTROL
CENTER NAMESPACE NAME
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              kubernetes.io/metadata.name: <NETAPP-ACC-OPERATOR> #
REPLACE THIS WITH THE OPERATOR NAMESPACE NAME
```

リソースの制限を解除します

一部の環境では、ResourceQuotasオブジェクトとLimitRangesオブジェクトを使用して、ネームスペース内のリソースがクラスタ上の使用可能なCPUとメモリをすべて消費しないようにします。Astra Control Centerでは上限が設定されていないため、これらのリソースに準拠していません。Astra Control Centerをインストールするネームスペースから削除する必要があります。

これらのクォータと制限を取得および削除するには、次の手順を実行します。これらの例では、コマンド出力はコマンド出力の直後に表示されます。

手順

1. NetApp-accネームスペース内のリソースクォータを取得します。

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

対応：

```
NAME          AGE   REQUEST                                          LIMIT
pods-high     16s   requests.cpu: 0/20, requests.memory: 0/100Gi
limits.cpu: 0/200, limits.memory: 0/1000Gi
pods-low      15s   requests.cpu: 0/1, requests.memory: 0/1Gi
limits.cpu: 0/2, limits.memory: 0/2Gi
pods-medium   16s   requests.cpu: 0/10, requests.memory: 0/20Gi
limits.cpu: 0/20, limits.memory: 0/200Gi
```

2. 名前別にすべてのリソースクォータを削除します。

```
kubectl delete resourcequota  pods-high -n netapp-acc
```

```
kubectl delete resourcequota  pods-low -n netapp-acc
```

```
kubectl delete resourcequota  pods-medium -n netapp-acc
```

3. NetApp-accネームスペース内の制限範囲を取得します。

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

対応：

```
NAME               CREATED AT
cpu-limit-range    2022-06-27T19:01:23Z
```

4. 制限範囲を名前で削除します。

```
kubectl delete limitrange cpu-limit-range -n netapp-acc
```

=
:allow-uri-read:

OpenShift OperatorHub を使用して Astra Control Center をインストールします

Red Hat OpenShift を使用する場合は、 Red Hat 認定オペレータを使用して Astra Control Center をインストールできます。この手順を使用して、から Astra Control Center をインストールします "Red Hat エコシステムカタログ" または、 Red Hat OpenShift Container Platform を使用します。

この手順を完了したら、インストール手順に戻ってを実行する必要があります "残りのステップ" インストールが成功したかどうかを確認し、ログオンします。

必要なもの

- "インストールを開始する前に、 Astra Control Center の導入環境を準備します"。
- OpenShiftクラスタから、すべてのクラスタオペレータが正常な状態にあることを確認します (`available` はです `true`）：

```
oc get clusteroperators
```

- OpenShiftクラスタから、すべてのAPIサービスが正常な状態であることを確認します (`available` はです

true）：

```
oc get apiservices
```

- データセンターにAstra Control CenterのFQDNアドレスを作成します。
- 説明したインストール手順を実行するために必要な権限を取得し、Red Hat OpenShift Container Platformにアクセスします。
- クラスタにcert-managerがすでに存在する場合は、いくつかを実行する必要があります "事前に必要な手順" そのため、Astra Control Centerは独自の証明書管理ツールをインストールしません。

手順

- Astra Control Centerバンドルをダウンロードして開梱します
- ネットアップAstra kubectlプラグインをインストール
- [イメージをローカルレジストリに追加します]
- [オペレータインストールページを検索します]
- [オペレータをインストールします]
- Astra Control Center をインストールします

Astra Control Centerバンドルをダウンロードして開梱します

1. Astra Control Center バンドルをダウンロードします (`astra-control-center-[version].tar.gz`) をクリックします "NetApp Support Site"。
2. から Astra Control Center 証明書とキーの zip をダウンロードします "NetApp Support Site"。
3. （任意）次のコマンドを使用して、バンドルのシグニチャを確認します。

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

4. 画像を抽出します。

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

ネットアップ**Astra kubectl**プラグインをインストール

ネットアップアストラ `kubectl` コマンドラインプラグインにより、Astra Control Centerの導入とアップグレードに関連する一般的なタスクを実行する時間を節約できます。

必要なもの

ネットアップでは、プラグイン用のバイナリを提供しており、CPUアーキテクチャやオペレーティングシステムが異なる場合はそのプラグインをこのタスクを実行する前に、使用しているCPUとオペレーティングシ

ステムを把握しておく必要があります。LinuxおよびMacオペレーティングシステムでは、を使用できます `uname -a` この情報を収集するためのコマンドです。

手順

1. 使用可能なネットアップAstraを選択します `kubectl` プラグインバイナリ。オペレーティングシステムとCPUアーキテクチャに必要なファイルの名前をメモします。

```
ls kubectl-astra/
```

2. ファイルを標準と同じ場所にコピーします `kubectl` ユーティリティ。この例では、を使用しています `kubectl` ユーティリティはにあります `/usr/local/bin` ディレクトリ。交換してください `<binary-name>` 必要なファイル名を使用して、次の操作を実行します。

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

イメージをローカルレジストリに追加します

1. コンテナエンジンに応じた手順を実行します。

Docker です

1. Astraディレクトリに移動します。

```
cd acc
```

2. [[[[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1></Z1>_image_local_registry_push]]]]]]]]]</Z2>アストラControl Centerイメージディレクトリ内のパッケージイメージをローカルレジストリにプッシュします。</Z3>コマンドを実行する前に、次の置き換えを行ってください。
   - bundle_fileをAstra Controlバンドルファイルの名前に置き換えます（例： `acc.manifest.yaml` ）。
   - my_registryをDockerリポジトリのURLに置き換えます。
   - my_registry_userをユーザー名に置き換えます。
   - my_registry_tokenをレジストリの認証済みトークンに置き換えます。

```
kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY
-u MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
```

ポドマン

1. レジストリにログインします。

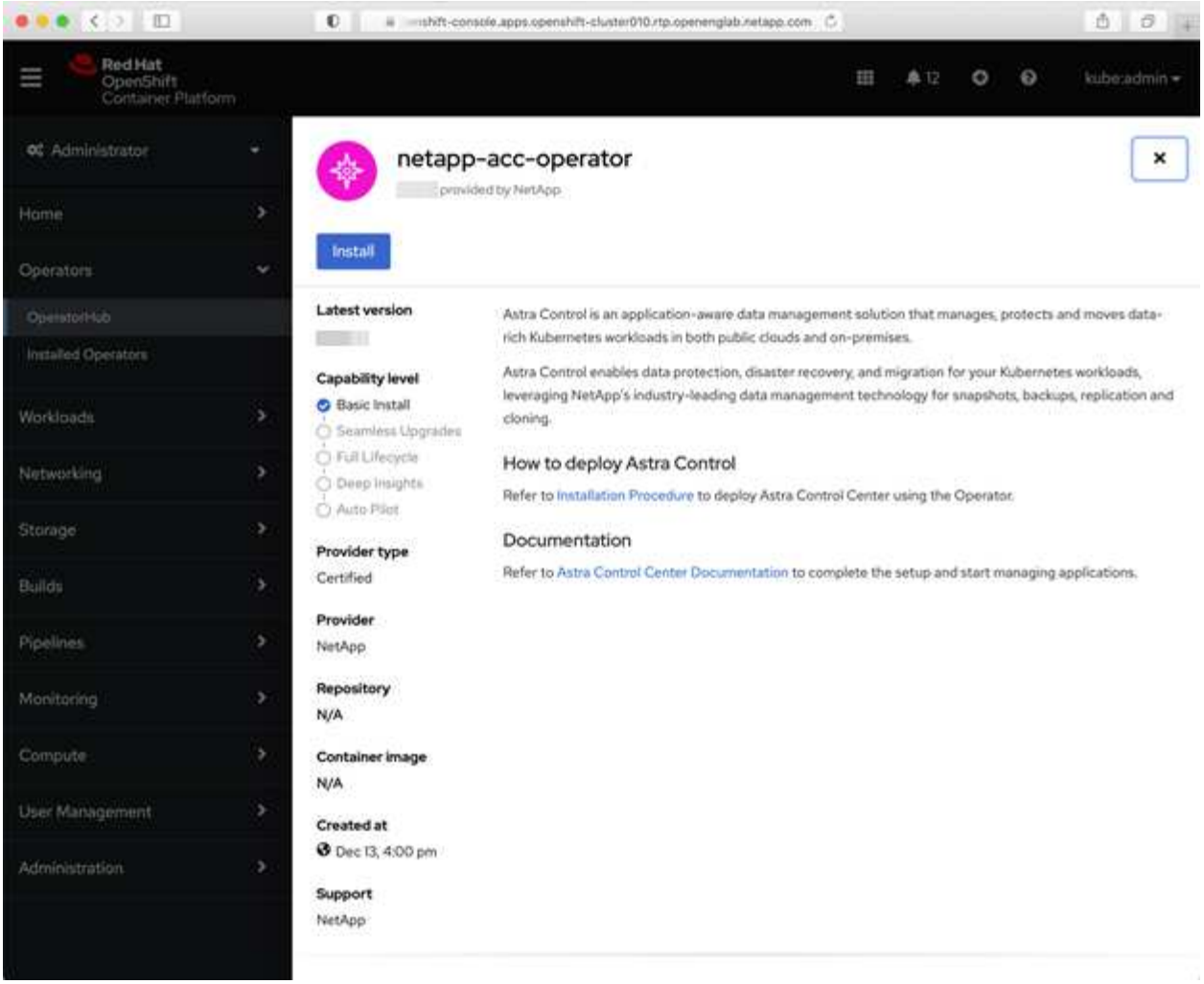
```
podman login [your_registry_path]
```

2. 次のスクリプトを実行して、コメントに記載されているように<your _registry>を置き換えます。

```
# You need to be at the root of the tarball.
# You should see these files to confirm correct location:
#   acc.manifest.yaml
#   acc/

# Replace <YOUR_REGISTRY> with your own registry (e.g
registry.customer.com or registry.customer.com/testing, etc..)
export REGISTRY=<YOUR_REGISTRY>
export PACKAGENAME=acc
export PACKAGEVERSION=22.08.1-26
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
  # Load to local cache
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //')

  # Remove path and keep imageName.
  astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')

  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/netapp/astra/${PACKAGENAME}
/${PACKAGEVERSION}/${astraImageNoPath}

  # Push to the local repo.
  podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

オペレータインストールページを検索します

1. 次のいずれかの手順を実行して、オペレータインストールページにアクセスします。
    - Red Hat OpenShift の Web コンソールから



:

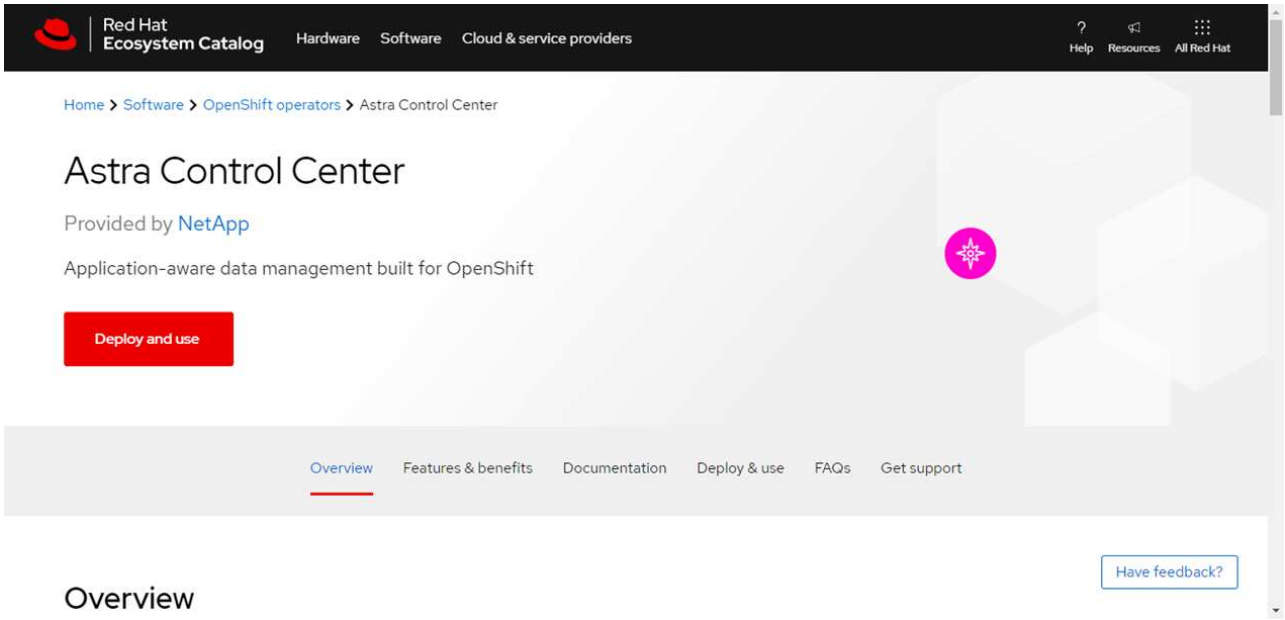
i. OpenShift Container Platform UI にログインします。

ii. サイドメニューから、 * 演算子 > OperatorHub * を選択します。

iii. NetApp Astra Control Center オペレータを選択します。

iv. 「 * Install * 」を選択します。

- Red Hat エコシステムカタログから
:



i. NetApp Astra Control Center を選択します "演算子"。

ii. [Deploy and Use] を選択します。

オペレータをインストールします

1. 「 * インストールオペレータ * 」ページに必要事項を入力し、オペレータをインストールします。

   オペレータはすべてのクラスタネームスペースで使用できます。

   a. operator名前空間またはを選択します `netapp-acc-operator` オペレータのインストールの一環として、名前空間が自動的に作成されます。

   b. 手動または自動の承認方法を選択します。

      手動による承認が推奨されます。1 つのクラスタで実行する演算子インスタンスは 1 つだけです。

   c. 「 * Install * 」を選択します。

      手動承認方式を選択した場合は、このオペレータの手動インストール計画を承認するように求められます。

2. コンソールで、 OperatorHub メニューに移動して、オペレータが正常にインストールされたことを確認します。

Astra Control Center をインストールします

1. コンソールのAstra Control Centerオペレータの詳細ビューで、を選択します `Create instance` を参照してください。

2. を実行します `Create AstraControlCenter` フォームフィールド：

   a. Astra Control Center の名前を保持または調整します。

b. （オプション） AutoSupport を有効または無効にします。Auto Support 機能の保持を推奨します。

c. Astra Control Center のアドレスを入力します。入らないでください `http://` または `https://` をクリックします。

d. Astra Control Center のバージョンを入力します。たとえば、 21.12.60 と入力します。

e. アカウント名、 E メールアドレス、および管理者の姓を入力します。

f. デフォルトのボリューム再利用ポリシーをそのまま使用します。

g. * Image Registry * に、ローカルコンテナイメージのレジストリパスを入力します。入らないでください `http://` または `https://` をクリックします。

h. 認証が必要なレジストリを使用する場合は、シークレットを入力します。

i. 管理者の名を入力します。

j. リソースの拡張を構成する。

k. デフォルトのストレージクラスは保持します。

l. CRD 処理の環境設定を定義します。

3. 選択するオプション `Create`。

次のステップ

Astra Control Center が正しくインストールされたことを確認し、を完了します "残りのステップ" ログインしてください。さらに、の導入も完了します "セットアップのタスク"。

Cloud Volumes ONTAP ストレージバックエンドに Astra Control Center をインストールします

Astra Control Center を使用すると、 Kubernetes クラスタと Cloud Volumes ONTAP インスタンスを自己管理することで、ハイブリッドクラウド環境でアプリケーションを管理できます。Astra Control Center は、オンプレミスの Kubernetes クラスタ、またはクラウド環境内の自己管理型 Kubernetes クラスタのいずれかに導入できます。

これらのいずれかの環境では、 Cloud Volumes ONTAP をストレージバックエンドとして使用して、アプリケーションデータの管理処理を実行できます。バックアップターゲットとして S3 バケットを設定することもできます。

Amazon Web Services（AWS）、Google Cloud Platform（GCP）、およびCloud Volumes ONTAP ストレージバックエンドを使用するMicrosoft AzureにAstra Control Centerをインストールするには、クラウド環境に応じて次の手順を実行します。

- Amazon Web Services に Astra Control Center を導入
- Astra Control CenterをGoogle Cloud Platformに導入
- Microsoft Azure に Astra Control Center を導入

OpenShift Container Platform（OCP）などの自己管理型Kubernetesクラスタを使用して、ディストリビューション内のアプリケーションを管理できます。Astra Control Centerを導入するために検証されるのは、自己管理型のOCPクラスタのみです。

Amazon Web Services に Astra Control Center を導入

Amazon Web Services （ AWS ）パブリッククラウドでホストされる自己管理型の Kubernetes クラスタに Astra Control Center を導入できます。

AWSに必要なもの

AWS に Astra Control Center を導入する前に、次のものが必要です。

- Astra Control Center ライセンス。を参照してください "Astra Control Center のライセンス要件"。
- "Astra Control Center の要件を満たす"。
- NetApp Cloud Central アカウント
- OCPを使用する場合は、Red Hat OpenShift Container Platform（OCP）権限（ポッドを作成するためのネームスペースレベル）
- バケットとコネクタを作成するための権限を持つ AWS クレデンシャル、アクセス ID 、シークレットキー
- AWS アカウント Elastic Container Registry （ ECR ）アクセスおよびログイン
- AWS がホストするゾーンと Route 53 エントリは、 Astra Control UI にアクセスするために必要です

AWS の運用環境の要件

Astra Control Center を使用するには、 AWS 向けに次の運用環境が必要です。

- Red Hat OpenShift Container Platform 4.8 の場合

> Astra Control Center をホストするオペレーティングシステムが、環境の公式ドキュメントに記載されている基本的なリソース要件を満たしていることを確認します。

Astra Control Center では、環境のリソース要件に加え、次のリソースが必要です。

コンポーネント	要件
バックエンドの **NetApp Cloud Volumes ONTAP** ストレージ容量	300GB 以上のデータがあります
ワーカーノード（ **AWS EC2** の要件）	少なくとも 3 つのワーカーノードが必要です。 vCPU コア 4 基、 RAM はそれぞれ 12GB です
ロードバランサ	動作環境クラスタ内のサービスに送信される入力トラフィックに使用できるサービスタイプ「 LoadBalancer 」
FQDN	Astra Control Center の FQDN をロードバランシング IP アドレスに指定する方法
Astra Trident （ NetApp Cloud Manager の Kubernetes クラスタ検出の一部としてインストール）	Trident 21.04 以降がインストールおよび設定され、 NetApp ONTAP バージョン 9.5 以降がストレージバックエンドとしてインストールされている必要があります

コンポーネント	要件
イメージレジストリ	Astra Control Center のビルドイメージをプッシュできる、 AWS Elastic Container Registry などの既存のプライベートレジストリが必要です。イメージをアップロードするイメージレジストリの URL を指定する必要があります。 Restic ベースのイメージを使用してアプリケーションをバックアップおよび復元するには、 Astra Control Center ホストクラスタと管理対象クラスタが同じイメージレジストリにアクセスできる必要があります。
Astra Trident / ONTAP 構成	Astra Control Center を使用するには、ストレージクラスを作成してデフォルトのストレージクラスとして設定する必要があります。Astra Control Center では、 Kubernetes クラスタを NetApp Cloud Manager にインポートする際に作成される次の ONTAP Kubernetes ストレージクラスがサポートされます。Astra Trident によって提供される機能は次のとおりです。 • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

これらの要件は、運用環境で実行されている唯一のアプリケーションが Astra Control Center であることを前提としています。環境で追加のアプリケーションを実行している場合は、それに応じてこれらの最小要件を調整します。

AWS レジストリトークンは 12 時間で期限切れになり、その後 Docker イメージのレジストリシークレットを更新する必要があります。

AWS の導入の概要を参照してください

Cloud Volumes ONTAP をストレージバックエンドとして使用して Astra Control Center for AWS をインストールするプロセスの概要を以下に示します。

これらの各手順については、以下で詳しく説明します。

1. 十分な IAM 権限があることを確認します。
2. AWS に Red Hat OpenShift クラスタをインストールします。
3. AWS を設定します。
4. NetApp Cloud Manager を設定します。

5. Astra Control Center をインストールします。

十分な IAM 権限があることを確認します

Red Hat OpenShiftクラスタとNetApp Cloud Manager Connectorをインストールできる十分な数のIAMロールと権限があることを確認します。

を参照してください "AWS の初期クレデンシャル"。

AWS に Red Hat OpenShift クラスタをインストールします

AWS に Red Hat OpenShift Container Platform クラスタをインストールします。

インストール手順については、を参照してください "AWS で OpenShift Container Platform にクラスタをインストールします"。

AWS を設定します

次に、仮想ネットワークの作成、EC2コンピューティングインスタンスのセットアップ、AWS S3バケットの作成、Astra Control Centerイメージをホストする Elastic Container Register（ECR）の作成、このレジストリへのイメージのプッシュを行うようにAWSを設定します。

AWS のドキュメントに従って次の手順を実行します。を参照してください "AWS インストールドキュメント"。

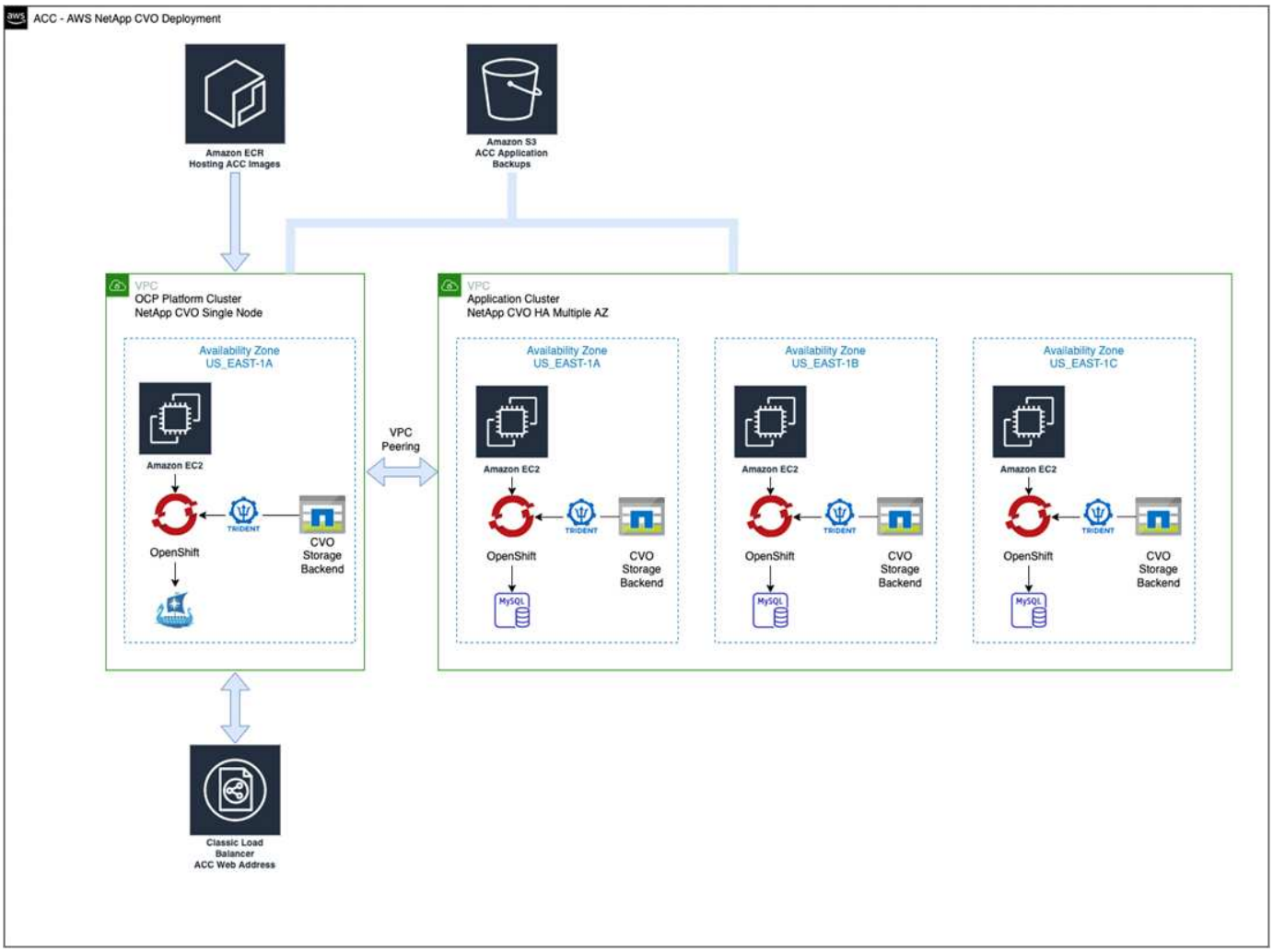
1. AWS仮想ネットワークを作成します。
2. EC2 コンピューティングインスタンスを確認します。AWS ではベアメタルサーバまたは VM を使用できます。
3. インスタンスタイプが、マスターノードとワーカーノードの Astra の最小リソース要件に一致していない場合は、 Astra の要件に合わせて AWS でインスタンスタイプを変更します。 を参照してください "Astra Control Center の要件"。
4. バックアップを格納する AWS S3 バケットを少なくとも 1 つ作成します。
5. すべての ACC イメージをホストする AWS Elastic Container Registry （ ECR ）を作成します。

> ECRを作成しないと、Astra Control Centerは、AWSバックエンドを持つCloud Volumes ONTAP を含むクラスタからモニタリングデータにアクセスできません。問題 は、 Astra Control Center を使用して検出および管理しようとしたクラスタに AWS ECR アクセスがない場合に発生します。

6. ACC イメージを定義済みのレジストリにプッシュします。

> AWS Elastic Container Registry （ ECR ）トークンの有効期限は 12 時間です。有効期限が切れたため、クラスタ間のクローニング処理が失敗します。この問題 は、AWS用に設定されたCloud Volumes ONTAP からストレージバックエンドを管理する場合に発生します。この問題 を修正するには、 ECR で再度認証を行い、クローン操作を再開するための新しいシークレットを生成します。

AWS 環境の例を次に示します。



NetApp Cloud Manager を設定します

Cloud Manager を使用して、ワークスペースの作成、 AWS へのコネクタの追加、作業環境の作成、クラスタのインポートを行います。

Cloud Manager のドキュメントに従って、次の手順を実行します。以下を参照してください。

- "AWS で Cloud Volumes ONTAP を使用するための準備"。
- "Cloud Manager を使用して AWS でコネクタを作成します"

手順

1. Cloud Manager にクレデンシャルを追加します。
2. ワークスペースを作成します。
3. AWS 用のコネクタを追加します。プロバイダとして AWS を選択します。
4. クラウド環境の作業環境を構築
   a. 場所：「 Amazon Web Services （ AWS ）」
   b. 「 Cloud Volumes ONTAP HA 」と入力します。
5. OpenShift クラスタをインポートします。作成した作業環境にクラスタが接続されます。

a. ネットアップクラスタの詳細を表示するには、 * K8s * > * Cluster list * > * Cluster Details * を選択します。

b. 右上隅に Trident のバージョンが表示されていることを確認します。

c. Cloud Volumes ONTAP クラスタのストレージクラスは、プロビジョニングツールとしてネットアップを使用していることに注目してください。

   これにより、 Red Hat OpenShift クラスタがインポートされ、デフォルトのストレージクラスに割り当てられます。ストレージクラスを選択します。Trident は、インポートと検出のプロセスの一環として自動的にインストールされます。

6. このCloud Volumes ONTAP 環境内のすべての永続ボリュームとボリュームをメモします。

> Cloud Volumes ONTAP は、シングルノードまたはハイアベイラビリティとして動作できます。HA が有効になっている場合は、 AWS で実行されている HA ステータスとノード導入ステータスを確認します。

Astra Control Center をインストールします

標準に従ってください "Astra Control Center のインストール手順"。

> AWSでは汎用のS3バケットタイプが使用されます。

Astra Control CenterをGoogle Cloud Platformに導入

Astra Control Centerは、Google Cloud Platform（GCP）パブリッククラウドでホストされる自己管理型のKubernetesクラスタに導入できます。

GCPに必要なもの

GCPでAstra Control Centerを導入する前に、次の項目が必要です。

- Astra Control Center ライセンス。を参照してください "Astra Control Center のライセンス要件"。
- "Astra Control Center の要件を満たす"。
- NetApp Cloud Central アカウント
- OCPを使用している場合は、Red Hat OpenShift Container Platform（OCP）4.10
- OCPを使用する場合は、Red Hat OpenShift Container Platform（OCP）権限（ポッドを作成するためのネームスペースレベル）
- バケットとコネクタの作成を可能にする権限を持つGCPサービスアカウント

GCPの運用環境の要件

> Astra Control Center をホストするオペレーティングシステムが、環境の公式ドキュメントに記載されている基本的なリソース要件を満たしていることを確認します。

Astra Control Center では、環境のリソース要件に加え、次のリソースが必要です。

コンポーネント	要件
バックエンドの **NetApp Cloud Volumes ONTAP** ストレージ容量	300GB 以上のデータがあります
ワーカーノード（**GCP**コンピューティング要件）	少なくとも 3 つのワーカーノードが必要です。 vCPU コア 4 基、 RAM はそれぞれ 12GB です
ロードバランサ	動作環境クラスタ内のサービスに送信される入力トラフィックに使用できるサービスタイプ「 LoadBalancer 」
FQDN（**GCP DNS**ゾーン）	Astra Control Center の FQDN をロードバランシング IP アドレスに指定する方法
Astra Trident （ **NetApp Cloud Manager** の **Kubernetes** クラスタ検出の一部としてインストール）	Trident 21.04 以降がインストールおよび設定され、 NetApp ONTAP バージョン 9.5 以降がストレージバックエンドとしてインストールされている必要があります
イメージレジストリ	Astra Control Centerビルドイメージをプッシュできる、Google Container Registryなどの既存のプライベートレジストリが必要です。イメージをアップロードするイメージレジストリの URL を指定する必要があります。 バックアップ用にリストアイメージを取得するには、匿名アクセスを有効にする必要があります。
Astra Trident / ONTAP 構成	Astra Control Center を使用するには、ストレージクラスを作成してデフォルトのストレージクラスとして設定する必要があります。Astra Control Center では、 Kubernetes クラスタを NetApp Cloud Manager にインポートする際に作成される次の ONTAP Kubernetes ストレージクラスがサポートされます。Astra Trident によって提供される機能は次のとおりです。 • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

これらの要件は、運用環境で実行されている唯一のアプリケーションが Astra Control Center であることを前提としています。環境で追加のアプリケーションを実行している場合は、それに応じてこれらの最小要件を調整します。

GCPの導入の概要

ここでは、Cloud Volumes ONTAP をストレージバックエンドとして使用して、GCP内の自己管理型OCPクラスタにAstra Control Centerをインストールするプロセスの概要を示します。

これらの各手順については、以下で詳しく説明します。

1. GCPにRed Hat OpenShiftクラスタをインストールします。
2. GCPプロジェクトとVirtual Private Cloudを作成します。
3. 十分な IAM 権限があることを確認します。
4. GCPを設定します。
5. NetApp Cloud Manager を設定します。
6. Astra Control Center をインストールして設定します。

GCPにRed Hat OpenShiftクラスタをインストールします

まず、GCPにRedHat OpenShiftクラスタをインストールします。

インストール手順については、次を参照してください。

- "GCPにOpenShiftクラスタをインストールする"
- "GCPサービスアカウントの作成"

GCPプロジェクトとVirtual Private Cloudを作成します

少なくとも1つのGCPプロジェクトとVirtual Private Cloud（VPC）を作成します。

> OpenShift では、独自のリソースグループを作成できます。さらに、GCP VPCも定義する必要があります。OpenShift のドキュメントを参照してください。

プラットフォームクラスタリソースグループおよびターゲットアプリケーション OpenShift クラスタリソースグループを作成できます。

十分な IAM 権限があることを確認します

Red Hat OpenShiftクラスタとNetApp Cloud Manager Connectorをインストールできる十分な数のIAMロールと権限があることを確認します。

を参照してください "GCPの初期資格情報と権限"。

GCPを設定します

次に、VPCの作成、コンピューティングインスタンスのセットアップ、Google Cloud Object Storageの作成、Astra Control Centerイメージのホストに Google Container Registerの作成、このレジストリへのイメージのプッシュを行うようにGCPを設定します。

GCPのドキュメントに従って、次の手順を実行します。「GCPへのOpenShiftクラスタのインストール」を参照してください。

1. GCPでGCPプロジェクトとVPCを作成します。GCPでは、CVOバックエンドでOCPクラスタ用にを使用する予定です。
2. コンピューティングインスタンスを確認します。GCP内のベアメタルサーバまたはVMです。
3. インスタンスタイプが、マスターノードとワーカーノードのAstra最小リソース要件と一致していない場合は、GCPでインスタンスタイプを変更してAstraの要件を満たします。を参照してください "Astra Control Center の要件"。
4. バックアップを保存するGCP Cloud Storageバケットを少なくとも1つ作成します。
5. バケットへのアクセスに必要なシークレットを作成します。
6. すべてのAstra Control Centerイメージをホストするの Google Container Registryを作成します。
7. すべてのAstra Control Centerイメージに対して、Dockerプッシュ/プル用のGoogle Container Registryアクセスを設定します。

   例：次のスクリプトを入力すると、ACCイメージをこのレジストリにプッシュできます。

   ```
   gcloud auth activate-service-account <service account email address>
   --key-file=<GCP Service Account JSON file>
   ```

   このスクリプトには、Astra Control Centerマニフェストファイルと Google Image Registryの場所が必要です。

   例

   ```
   manifestfile=astra-control-center-<version>.manifest
   GCP_CR_REGISTRY=<target image repository>
   ASTRA_REGISTRY=<source ACC image repository>

   while IFS= read -r image; do
       echo "image: $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image"
       root_image=${image%:*}
       echo $root_image
       docker pull $ASTRA_REGISTRY/$image
       docker tag $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image
       docker push $GCP_CR_REGISTRY/$image
   done < astra-control-center-22.04.41.manifest
   ```

8. DNS ゾーンを設定します。

NetApp Cloud Manager を設定します

Cloud Managerを使用して、ワークスペースの作成、GCPへのコネクターの追加、作業環境の作成、クラスタのインポートを行います。

Cloud Manager のドキュメントに従って、次の手順を実行します。を参照してください "GCPでCloud Volumes ONTAP の使用を開始する"。

必要なもの

- 必要なIAM権限と役割を持つGCPサービスアカウントにアクセスします

手順

1. Cloud Manager にクレデンシャルを追加します。を参照してください "GCP アカウントの追加"。
2. GCPのコネクターを追加します。
   a. プロバイダーとして[GCP]を選択します。
   b. GCP資格情報を入力します。を参照してください "Cloud ManagerからGCPでコネクタを作成する"。
   c. コネクタが動作していることを確認し、コネクタに切り替えます。
3. クラウド環境の作業環境を構築
   a. 場所："GCP"
   b. 「 Cloud Volumes ONTAP HA 」と入力します。
4. OpenShift クラスタをインポートします。作成した作業環境にクラスタが接続されます。
   a. ネットアップクラスタの詳細を表示するには、 * K8s * > * Cluster list * > * Cluster Details * を選択します。
   b. 右上隅に Trident のバージョンが表示されていることを確認します。
   c. Cloud Volumes ONTAP クラスタのストレージクラスは、プロビジョニングツールとして「ネットアップ」を使用していることに注目してください。

      これにより、 Red Hat OpenShift クラスタがインポートされ、デフォルトのストレージクラスに割り当てられます。ストレージクラスを選択します。Trident は、インポートと検出のプロセスの一環として自動的にインストールされます。
5. このCloud Volumes ONTAP 環境内のすべての永続ボリュームとボリュームをメモします。

> Cloud Volumes ONTAP は、シングルノードまたはハイアベイラビリティ（HA）で動作します。HAが有効になっている場合は、GCPで実行されているHAステータスとノード導入ステータスを確認します。

Astra Control Center をインストールします

標準に従ってください "Astra Control Center のインストール手順"。

> GCPでは汎用S3バケットタイプが使用されます。

1. Astra Control Centerインストール用のイメージをプルするDocker Secretを生成します。

```
kubectl create secret docker-registry <secret name>
--docker-server=<Registry location>
--docker-username=_json_key
--docker-password="$(cat <GCP Service Account JSON file>)"
--namespace=pcloud
```

Microsoft Azure に Astra Control Center を導入

Microsoft Azure パブリッククラウドでホストされる自己管理型の Kubernetes クラスタに Astra Control Center を導入できます。

Azureに必要なもの

Azure に Astra Control Center を導入する前に、次のものが必要です。

- Astra Control Center ライセンス。を参照してください "Astra Control Center のライセンス要件"。
- "Astra Control Center の要件を満たす"。
- NetApp Cloud Central アカウント
- OCPを使用する場合、Red Hat OpenShift Container Platform（OCP）4.8
- OCPを使用する場合は、Red Hat OpenShift Container Platform（OCP）権限（ポッドを作成するためのネームスペースレベル）
- バケットとコネクタの作成を可能にする権限を持つ Azure クレデンシャル

Azure の運用環境の要件

Astra Control Center をホストするオペレーティングシステムが、環境の公式ドキュメントに記載されている基本的なリソース要件を満たしていることを確認します。

Astra Control Center では、環境のリソース要件に加え、次のリソースが必要です。

を参照してください "Astra Control Center の運用環境要件"。

コンポーネント	要件
バックエンドの **NetApp Cloud Volumes ONTAP** ストレージ容量	300GB 以上のデータがあります
ワーカーノード（**Azure** コンピューティング要件）	少なくとも 3 つのワーカーノードが必要です。 vCPU コア 4 基、 RAM はそれぞれ 12GB です
ロードバランサ	動作環境クラスタ内のサービスに送信される入力トラフィックに使用できるサービスタイプ「 LoadBalancer 」
FQDN （**Azure DNS** ゾーン）	Astra Control Center の FQDN をロードバランシング IP アドレスに指定する方法
Astra Trident （**NetApp Cloud Manager** の **Kubernetes** クラスタ検出の一部としてインストール）	Trident 21.04 以降がインストールおよび設定され、 NetApp ONTAP バージョン 9.5 以降がストレージバックエンドとして使用されます

コンポーネント	要件
イメージレジストリ	Astra Control Center ビルドイメージをプッシュできる、 Azure Container Registry （ACR ）などの既存のプライベートレジストリが必要です。イメージをアップロードするイメージレジストリの URL を指定する必要があります。 バックアップ用にリストアイメージを取得するには、匿名アクセスを有効にする必要があります。
Astra Trident / ONTAP 構成	Astra Control Center を使用するには、ストレージクラスを作成してデフォルトのストレージクラスとして設定する必要があります。Astra Control Center では、 Kubernetes クラスタを NetApp Cloud Manager にインポートする際に作成される次の ONTAP Kubernetes ストレージクラスがサポートされます。Astra Trident によって提供される機能は次のとおりです。 • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

これらの要件は、運用環境で実行されている唯一のアプリケーションが Astra Control Center であることを前提としています。環境で追加のアプリケーションを実行している場合は、それに応じてこれらの最小要件を調整します。

Azure の導入の概要

ここでは、 Astra Control Center for Azure のインストールプロセスの概要を示します。

これらの各手順については、以下で詳しく説明します。

1. Azure に Red Hat OpenShift クラスタをインストールします。
2. Azure リソースグループを作成する。
3. 十分な IAM 権限があることを確認します。
4. Azure を設定。
5. NetApp Cloud Manager を設定します。
6. Astra Control Center をインストールして設定します。

Azure に Red Hat OpenShift クラスタをインストールします

まず、 Azure に Red Hat OpenShift クラスタをインストールします。

インストール手順については、のRedHatのマニュアルを参照してください "AzureにOpenShiftクラスタをインストールしています" および "Azureアカウントをインストールしています"。

Azure リソースグループを作成する

Azure リソースグループを少なくとも 1 つ作成します。

> OpenShift では、独自のリソースグループを作成できます。さらに、 Azure リソースグループも定義する必要があります。OpenShift のドキュメントを参照してください。

プラットフォームクラスタリソースグループおよびターゲットアプリケーション OpenShift クラスタリソースグループを作成できます。

十分な IAM 権限があることを確認します

Red Hat OpenShiftクラスタとNetApp Cloud Manager Connectorをインストールできる十分な数のIAMロールと権限があることを確認します。

を参照してください "Azure のクレデンシャルと権限"。

Azure を設定

次に、仮想ネットワークの作成、コンピューティングインスタンスのセットアップ、Azure Blobコンテナの作成、Astra Control Centerイメージをホストするazure Container Register（ACR）の作成、このレジストリへのイメージのプッシュを行うようにAzureを設定します。

Azure のドキュメントに従って、次の手順を実行します。を参照してください "Azure への OpenShift クラスタのインストール"。

1. Azure Virtual Networkの作成
2. コンピューティングインスタンスを確認します。Azure の場合、ベアメタルサーバまたは VM を使用できます。
3. インスタンスタイプがまだマスターノードとワーカーノードの Astra 最小リソース要件に一致していない場合は、 Azure でインスタンスタイプを変更して Astra の要件を満たします。を参照してください "Astra Control Center の要件"。
4. バックアップを格納するAzure BLOBコンテナを少なくとも1つ作成します。
5. ストレージアカウントを作成します。Astra Control Center でバケットとして使用するコンテナを作成するには、ストレージアカウントが必要です。
6. バケットへのアクセスに必要なシークレットを作成します。
7. Azure Container Registry （ ACR ）を作成して、すべての Astra Control Center イメージをホストします。
8. ACR アクセスを設定して Docker プッシュ / プルをすべての Astra Control Center イメージに適用します。
9. 次のスクリプトを入力して、 ACC イメージをこのレジストリにプッシュします。

```
az acr login -n <AZ ACR URL/Location>
This script requires ACC manifest file and your Azure ACR location.
```

- 例 * ：

```
manifestfile=astra-control-center-<version>.manifest
AZ_ACR_REGISTRY=<target image repository>
ASTRA_REGISTRY=<source ACC image repository>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRYY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < astra-control-center-22.04.41.manifest
```

10. DNS ゾーンを設定します。

NetApp Cloud Manager を設定します

Cloud Manager を使用して、ワークスペースの作成、 Azure へのコネクタの追加、作業環境の作成、クラスタのインポートを行います。

Cloud Manager のドキュメントに従って、次の手順を実行します。を参照してください "Azure で Cloud Manager を使用する準備をしています"。

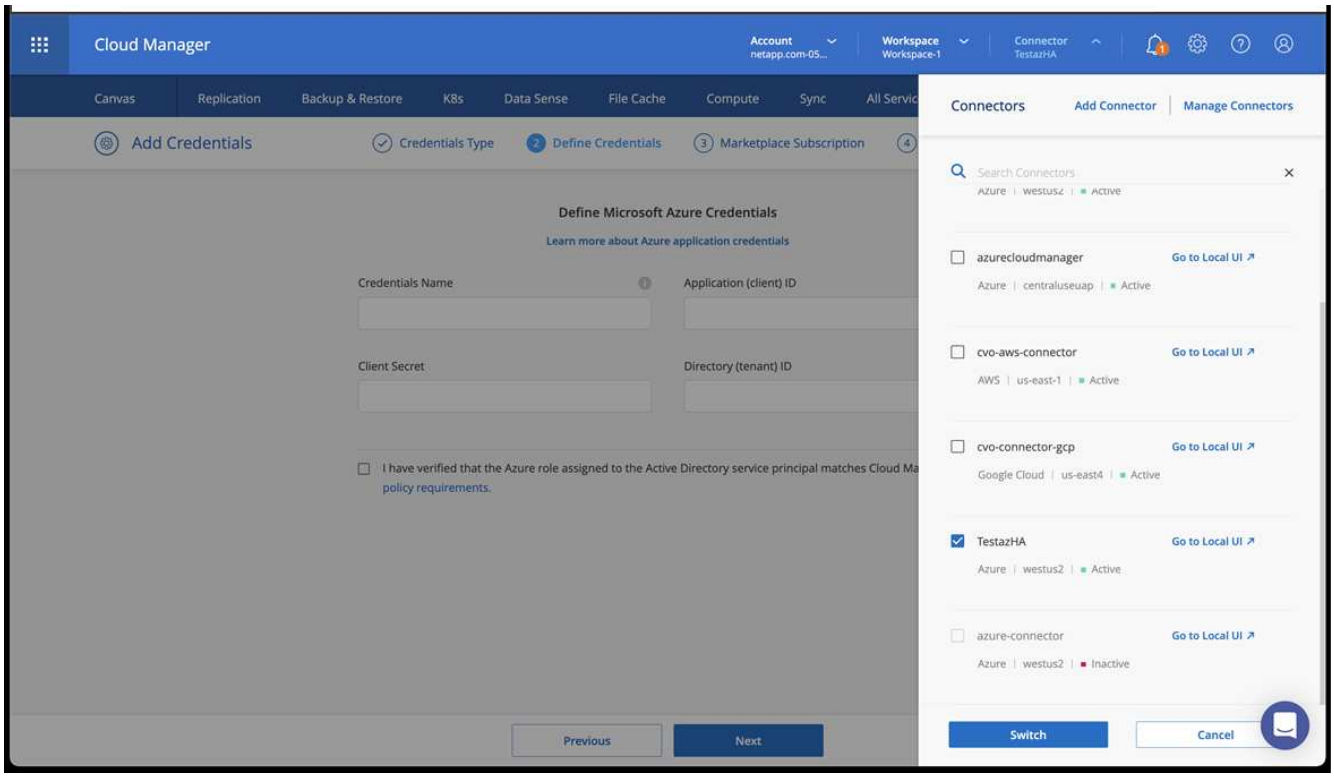
必要なもの

必要な IAM 権限とロールを持つ Azure アカウントにアクセスします

手順

1. Cloud Manager にクレデンシャルを追加します。
2. Azure 用のコネクタを追加します。を参照してください "Cloud Manager のポリシー"。
   a. プロバイダとして「 * Azure * 」を選択します。
   b. アプリケーション ID 、クライアントシークレット、ディレクトリ（テナント） ID など、 Azure クレデンシャルを入力します。

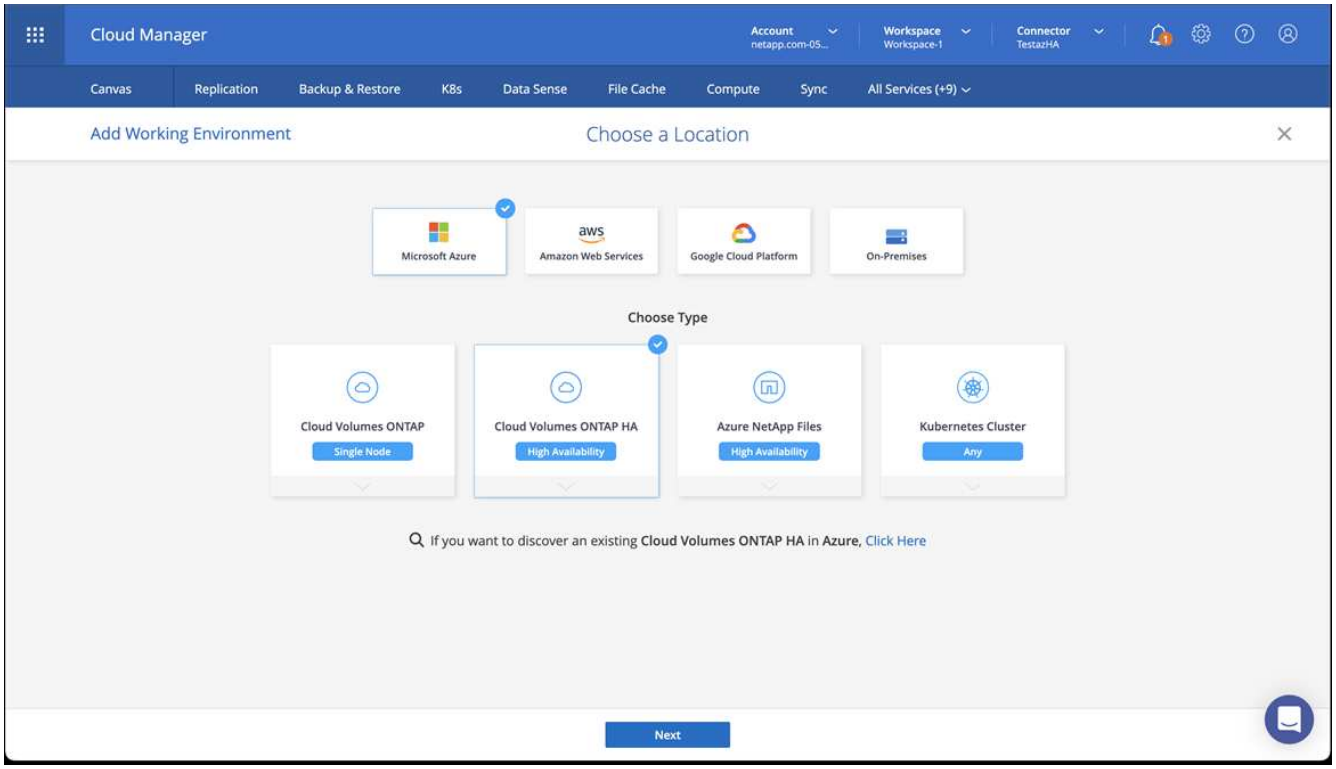
      を参照してください "Cloud Manager から Azure にコネクタを作成する"。
3. コネクタが動作していることを確認し、コネクタに切り替えます。



4. クラウド環境の作業環境を構築

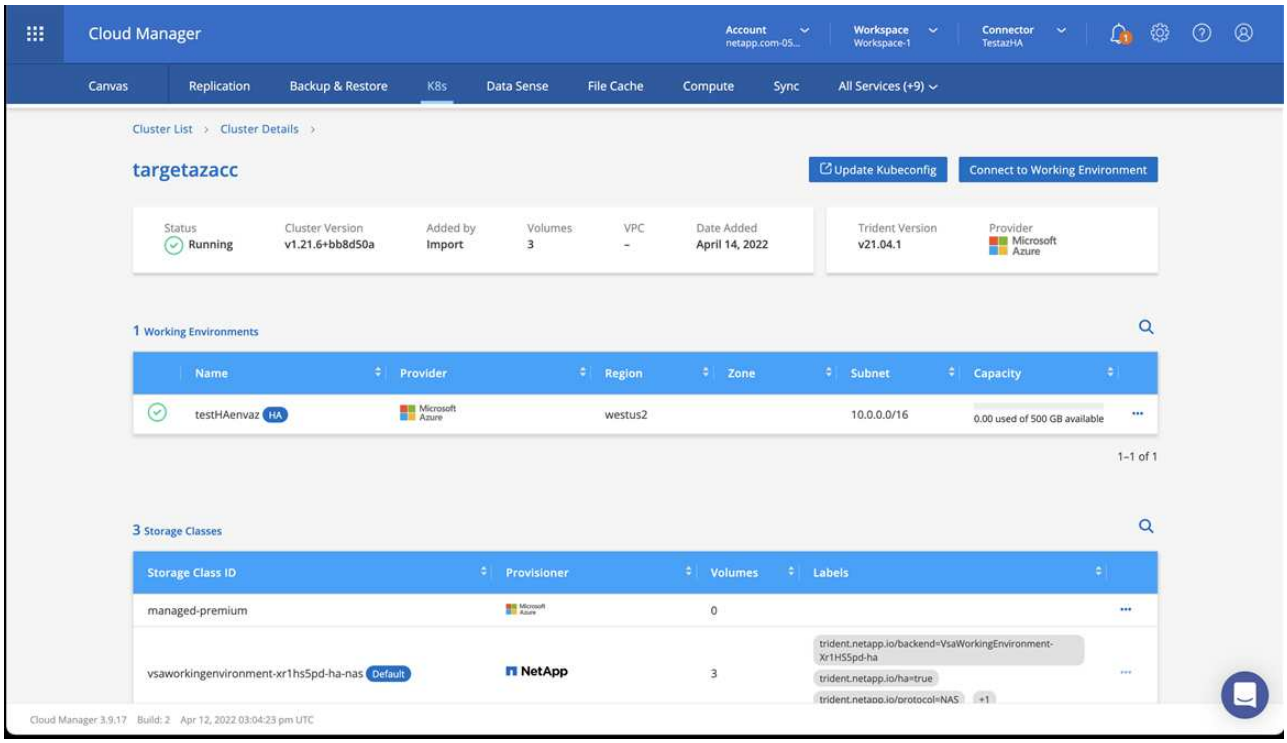
   a. 場所：「 Microsoft Azure 」。

   b. 「 Cloud Volumes ONTAP HA 」と入力します。



5. OpenShift クラスタをインポートします。作成した作業環境にクラスタが接続されます。

a. ネットアップクラスタの詳細を表示するには、 * K8s * > * Cluster list * > * Cluster Details * を選択します。



b. 右上隅に Trident のバージョンが表示されていることを確認します。

c. Cloud Volumes ONTAP クラスタのストレージクラスは、プロビジョニングツールとしてネットアップを使用していることに注目してください。

これにより、 Red Hat OpenShift クラスタがインポートされ、デフォルトのストレージクラスが割り当てられます。ストレージクラスを選択します。Trident は、インポートと検出のプロセスの一環として自動的にインストールされます。

6. このCloud Volumes ONTAP 環境内のすべての永続ボリュームとボリュームをメモします。

7. Cloud Volumes ONTAP は、シングルノードまたはハイアベイラビリティとして動作できます。HA が有効になっている場合は、 Azure で実行されている HA ステータスとノード導入ステータスを確認します。

Astra Control Center をインストールして設定します

Astra Control Center を標準でインストールします "インストール手順"。

Astra Control Center を使用して、 Azure バケットを追加する。を参照してください "Astra Control Center をセットアップし、バケットを追加する"。

Astra Control Center をセットアップします

Astra Control Center は、 ONTAP と Astra データストアをストレージバックエンドとしてサポートおよび監視します。Astra Control Center をインストールして UI にログインし、パスワードを変更したら、ライセンスの設定、クラスタの追加、ストレージの管理、バケットの追加を行います。

タスク

- Astra Control Center のライセンスを追加します
- [クラスタを追加]
- [ストレージバックエンドを追加します]
- [バケットを追加します]

Astra Control Center のライセンスを追加します

UI またはを使用して新しいライセンスを追加できます "API" Astra Control Center の全機能を利用できます。ライセンスがないと、 Astra Control Center の使用は、ユーザの管理と新しいクラスタの追加に限定されます。

ライセンスの計算方法の詳細については、を参照してください "ライセンス"。

> 既存の評価版またはフルライセンスを更新するには、を参照してください "既存のライセンスを更新する"。

Astra Control Center ライセンスは、 Kubernetes CPU ユニットを使用して CPU リソースを測定します。ライセンスには、管理対象のすべての Kubernetes クラスタのワーカーノードに割り当てられた CPU リソースが含まれている必要があります。ライセンスを追加する前に、からライセンスファイル（ NLF ）を取得する必要があります "NetApp Support Site"。

また、 Astra Control Center に評価ライセンスをお試しいただくこともできます。このライセンスは、 Astra Control Center をダウンロードした日から 90 日間使用できます。登録すると、無償トライアルに登録できます "こちらをご覧ください"。

> インストールがライセンス数を超えると、 Astra Control Center は新しいアプリケーションを管理できなくなります。容量を超えるとアラートが表示されます。

必要なもの

から Astra Control Center をダウンロードした場合 "NetApp Support Site"には、 NetApp License File （ NLF ）もダウンロードします。このライセンスファイルにアクセスできることを確認してください。

手順

1. Astra Control Center UI にログインします。
2. 「 * アカウント * > * ライセンス * 」を選択します。
3. 「 * ライセンスの追加 * 」を選択します。
4. ダウンロードしたライセンスファイル（ NLF ）を参照します。
5. 「 * ライセンスの追加 * 」を選択します。

Account>*License* ページには、ライセンス情報、有効期限、ライセンスシリアル番号、アカウント ID 、および使用されている CPU ユニットが表示されます。

> 評価用ライセンスをお持ちの場合は、 Astra Control Center に障害が発生したときに ASUP を送信していないときにデータが失われないように、アカウント ID を必ず保存してください。

クラスタを追加

アプリケーションの管理を開始するには、 Kubernetes クラスタを追加し、コンピューティングリソースとして管理します。Kubernetes アプリケーションを検出するには、 Astra Control Center のクラスタを追加する必要があります。Astra データストアの場合は、 Astra データストアによってプロビジョニングされたボリュームを使用するアプリケーションを含む Kubernetes アプリケーションクラスタを追加します。

> 他のクラスタを Astra Control Center に追加して管理する前に、 Astra Control Center が最初に導入したクラスタを管理することをお勧めします。指標およびトラブルシューティング用の Kubemetrics データとクラスタ関連データを送信するには、最初のクラスタを管理下に配置する必要があります。Add Cluster * 機能を使用して、 Astra Control Center でクラスタを管理できます。

> Astra Controlは、クラスタを管理する際に、クラスタのデフォルトストレージクラスを追跡します。を使用してストレージクラスを変更する場合は、を使用します `kubectl` コマンドを実行すると、Astra Controlによって変更が元に戻されます。Astra Controlで管理されるクラスタのデフォルトのストレージクラスを変更するには、次のいずれかの方法を使用します。
>
> - Astra Control API を使用 `PUT /managedClusters` エンドポイントを追加し、で別のデフォルトストレージクラスを割り当てます `DefaultStorageClass` パラメータ
> - Astra Control Web UIを使用して、別のデフォルトストレージクラスを割り当てます。を参照してください [デフォルトのストレージクラスを変更する]。

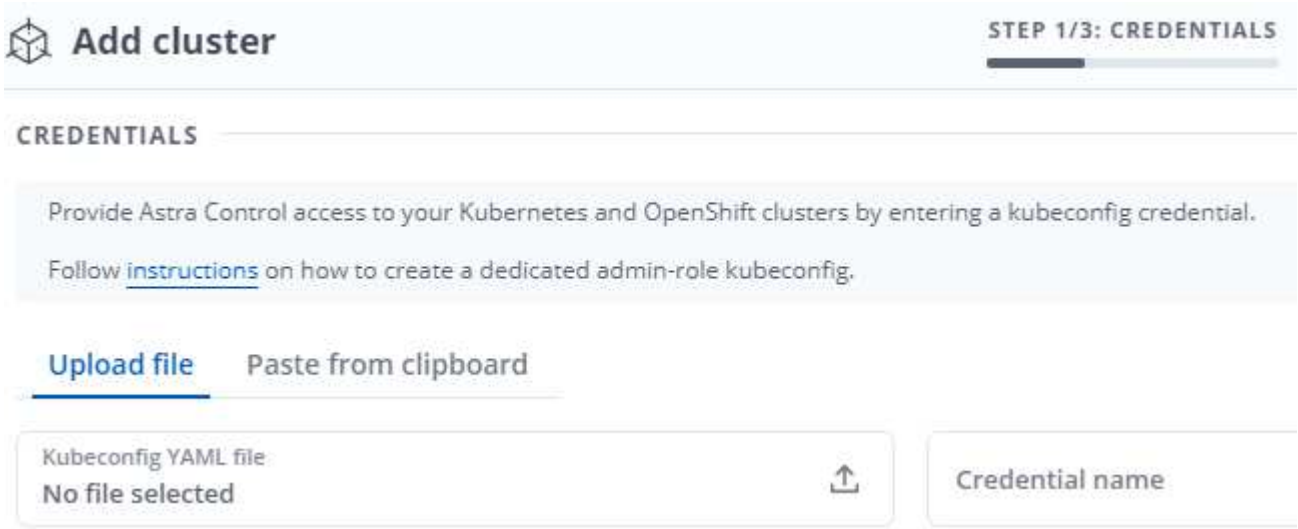
必要なもの

- クラスタを追加する前に、必要なを確認し、実行しておきます "前提条件となるタスク"。

手順

1. Astra Control Center UI の * ダッシュボード * から、クラスタセクションで * 追加 * を選択します。
2. 表示された*クラスタの追加*ウィンドウで、をアップロードします `kubeconfig.yaml` の内容をファイルまたは貼り付けます `kubeconfig.yaml` ファイル。

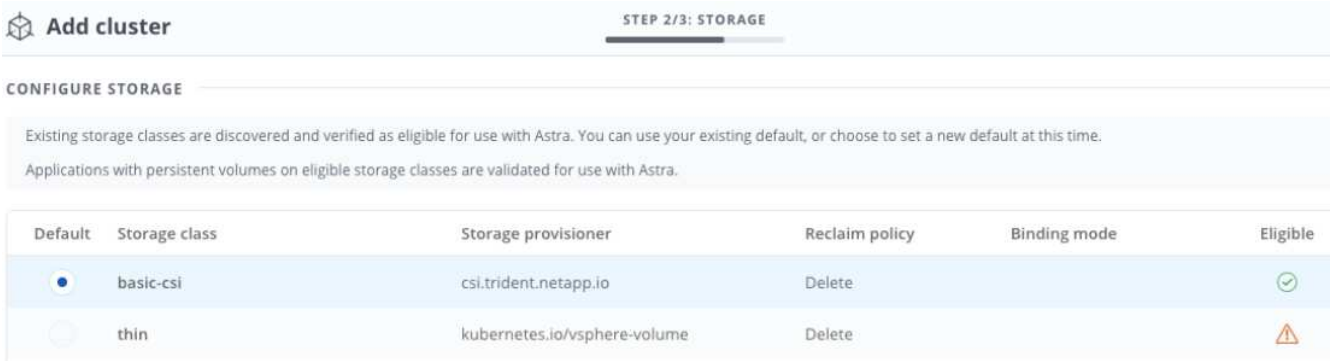
> 。 `kubeconfig.yaml` ファイルには、1つのクラスタのクラスタクレデンシャルのみを含める必要があります*。



> 自分で作成する場合は `kubeconfig` ファイルには、* 1つの*コンテキストエレメントのみを定義する必要があります。を参照してください "Kubernetes のドキュメント" を参照してください `kubeconfig` ファイル。

3. クレデンシャル名を指定します。デフォルトでは、クレデンシャル名がクラスタの名前として自動的に入力されます。
4. 「ストレージの設定」を選択します。
5. この Kubernetes クラスタに使用するストレージクラスを選択し、 * Review * を選択します。

> ONTAP ストレージまたは Astra データストアからバックアップされた Trident ストレージクラスを選択する必要があります。



6. 情報を確認し、問題がない場合は「 * クラスタの追加 * 」を選択します。

結果

クラスタが「 Discovering * 」ステータスになり、「 Running 」に変わります。Kubernetes クラスタが正常に追加され、 Astra Control Center で管理できるようになりました。

> Astra Control Center で管理するクラスタを追加したあと、監視オペレータの配置に数分かかる場合があります。それまでは、通知アイコンが赤に変わり、 * モニタリングエージェントステータスチェック失敗 * イベントが記録されます。この問題は無視してかまいません。問題は、Astra Control Center が正しいステータスを取得したときに解決します。数分経っても問題 が解決しない場合は、クラスタに移動してを実行します `oc get pods -n netapp-monitoring` を開始点として指定します。問題をデバッグするには、監視オペレータのログを調べる必要があります。

ストレージバックエンドを追加します

ストレージバックエンドを追加して、 Astra Control がリソースを管理できるようにすることができます。管理対象クラスタにストレージバックエンドを導入するか、既存のストレージバックエンドを使用できます。

ストレージバックエンドとして Astra Control のストレージクラスタを管理することで、永続ボリューム（ PVS ）とストレージバックエンドの間のリンケージを取得できるだけでなく、追加のストレージ指標も取得できます。

既存の**Astra**データストアの導入に必要なもの

- Kubernetesアプリケーションクラスタと基盤となるコンピューティングクラスタを追加しておきます。

> Astraデータストア用のKubernetesアプリケーションクラスタを追加し、Astra Controlで管理したあと、クラスタはのように表示されます `unmanaged` 検出されたバックエンドのリスト。次に、 Astra データストアを含むコンピューティングクラスタを追加し、Kubernetes アプリケーションクラスタの基盤を構築する必要があります。これは、 UI の * Backends * から実行できます。クラスタのActions（操作）メニューを選択し、を選択します `Manage`および "クラスタを追加"。をクラスタの状態のあとに続けて追加します `unmanaged` Kubernetesクラスタの名前を変更した場合は、バックエンドの追加に進むことができます。

新しい**Astra**データストアの導入に必要なもの

- これで完了です "導入するインストールバンドルのバージョンをアップロードしました" Astra Controlからアクセス可能な場所への移動。
- 導入に使用するKubernetesクラスタを追加しておきます。
- をアップロードしました Astraデータストアライセンス Astra Controlからアクセス可能な場所への導入をサポートします。

オプション（ **Options** ）

- [ストレージリソースを導入]
- [既存のストレージバックエンドを使用する]

ストレージリソースを導入

新しいAstraデータストアを導入して、関連するストレージバックエンドを管理できます。

手順

1. ダッシュボードまたはバックエンドメニューから移動します。
   - ダッシュボードから*：リソースサマリからストレージバックエンドペインからリンクを選択し、バックエンドセクションから*追加*を選択します。
   - バックエンドから * ：
     i. 左側のナビゲーション領域で、 * Backends * を選択します。
     ii. 「 * 追加」を選択します。
2. Deploy タブで Astra Data Store *導入オプションを選択します。
3. 導入するAstraデータストアパッケージを選択：
   a. Astraデータストアアプリケーションの名前を入力します。
   b. 導入するAstraデータストアのバージョンを選択します。

   > 展開するバージョンをまだアップロードしていない場合は、*パッケージの追加*オプションを使用するか、ウィザードを終了してを使用できます "パッケージ管理" インストールバンドルをアップロードします。

4. 以前にアップロードしたAstraデータストアライセンスを選択するか、*ライセンスの追加*オプションを使用して、アプリケーションで使用するライセンスをアップロードします。

> 完全な権限を持つAstra Data StoreライセンスはKubernetesクラスタに関連付けられており、この関連クラスタは自動的に表示されるはずです。管理対象クラスタがない場合は、*クラスタの追加*オプションを選択してAstra Control管理に追加できます。Astra Data Storeライセンスの場合、ライセンスとクラスタの間に関連付けが行われていない場合は、ウィザードの次のページでこの関連付けを定義できます。

5. KubernetesクラスタをAstra Control管理に追加していない場合は、* Kubernetes cluster *ページから追加する必要があります。リストから既存のクラスタを選択するか、「*基盤となるクラスタを追加」を選択してAstra Control管理用にクラスタを追加します。
6. Astraデータストアにリソースを提供するKubernetesクラスタのテンプレートサイズを選択します。次のいずれかを選択できます。
   - をクリックします `Recommended Kubernetes worker node requirements`をクリックし、ライセンスで許可されている内容に基づいて、テンプレートを大規模から小に選択します。
   - をクリックします `Custom Kubernetes worker node requirements`をクリックし、各クラスタノードに必要なコア数と総メモリを選択します。また、コアとメモリの選択基準を満たす、クラスタ内の対応するノード数も表示できます。

> テンプレートを選択する際は、大規模なワークロードにはメモリとコアが多く、小規模なワークロードにはノード数が多い大規模なノードを選択します。ライセンスで許可されている内容に基づいてテンプレートを選択する必要があります。推奨されるテンプレートオプションごとに、各ノードのメモリとコアおよび容量のテンプレートパターンを満たす、適格なノードの数が提示されます。

7. ノードを設定します。
   a. ノードラベルを追加して、このAstraデータストアクラスタをサポートするワーカーノードのプールを特定します。

   > このラベルは、Astraデータストアの導入に使用するクラスタ内の各ノードに追加してからでないと、導入や導入が失敗します。

   b. ノードあたりの容量（GiB）を手動で設定するか、許容される最大ノード容量を選択します。
   c. クラスタで許可される最大ノード数を設定するか、クラスタで許容される最大ノード数を設定します。
8. （Astraデータストアフルライセンスのみ）保護ドメインに使用するラベルのキーを入力します。

> 各ノードのキーに対して、少なくとも3つの一意のラベルを作成します。たとえば、キーがの場合などです `astra.datastore.protection.domain`次のラベルを作成できます。`astra.datastore.protection.domain=domain1,astra.datastore.protection.domain=domain2`および `astra.datastore.protection.domain=domain3`。

9. 管理ネットワークを設定します。
   a. Astraデータストアの内部管理用の管理IPアドレスを入力します。このIPアドレスは、ワーカーノードのIPアドレスと同じサブネットにあります。
   b. 管理ネットワークとデータネットワークで同じNICを使用するか、または個別に設定します。
   c. データネットワークのIPアドレスプール、サブネットマスク、ストレージアクセス用のゲートウェイを入力してください。

10. 設定を確認し、「* Deploy *」を選択してインストールを開始します。

結果

インストールが正常に完了すると、バックエンドがに表示されます `available` バックエンドにアクティブなパフォーマンス情報とともに表示

> バックエンドが表示されるようにページを更新する必要がある場合があります。

既存のストレージバックエンドを使用する

検出されたONTAP またはAstraデータストアのストレージバックエンドをAstra Control Center管理に組み込むことができます。

手順

1. ダッシュボードまたはバックエンドメニューから移動します。
   - ダッシュボードから*：リソースサマリからストレージバックエンドペインからリンクを選択し、バックエンドセクションから*追加*を選択します。
   - バックエンドから * ：
     i. 左側のナビゲーション領域で、 * Backends * を選択します。
     ii. 管理対象クラスタから検出されたバックエンドで* Manage を選択するか、 Add *を選択して追加の既存バックエンドを管理します。
2. [既存の使用（ Use Existing ）] * タブを選択します。
3. バックエンドの種類に応じて、次のいずれかの操作を行います。
   - * Astra データストア * ：
     i. 「* Astra Data Store *」を選択します。
     ii. 管理対象のコンピューティングクラスタを選択し、 * Next * を選択します。
     iii. バックエンドの詳細を確認し、「Add storage backend *」を選択します。
   - * ONTAP * ：
     i. 「* ONTAP 」を選択し、「 Next *」を選択します。
     ii. ONTAP クラスタ管理IPアドレスと管理者クレデンシャルを入力します。

     > ここで入力するクレデンシャルのユーザは、を持っている必要があります `ontapi` ONTAP クラスタのONTAP System Managerで有効になっているユーザログインアクセス方法。SnapMirrorレプリケーションを使用する場合は、アクセス方法を有効にします `ontapi` および `http` 両方のONTAP クラスタ上のユーザに対して設定します。を参照してください "ユーザアカウントを管理する" を参照してください。

     iii. [* Review （レビュー）] を選択します
     iv. バックエンドの詳細を確認し、「Add storage backend *」を選択します。

結果

バックエンドがに表示されます `available` リストに概要情報を表示します。

バックエンドが表示されるようにページを更新する必要がある場合があります。

バケットを追加します

アプリケーションと永続的ストレージをバックアップする場合や、クラスタ間でアプリケーションのクローニングを行う場合は、オブジェクトストアバケットプロバイダの追加が不可欠です。Astra Control は、これらのバックアップまたはクローンを、定義したオブジェクトストアバケットに格納します。

バケットを追加すると、 Astra Control によって、 1 つのバケットがデフォルトのバケットインジケータとしてマークされます。最初に作成したバケットがデフォルトバケットになります。

アプリケーション構成と永続的ストレージを同じクラスタにクローニングする場合、バケットは必要ありません。

次のいずれかのバケットタイプを使用します。

- NetApp ONTAP S3
- NetApp StorageGRID S3 の略
- 汎用 S3

Amazon Web Services（AWS）とGoogle Cloud Platform（GCP）では、汎用のS3バケットタイプを使用します。

- Microsoft Azure

Astra Control Center は Amazon S3 を汎用 S3 バケットプロバイダとしてサポートしていますが、 Astra Control Center は Amazon の S3 サポートを要求するすべてのオブジェクトストアベンダーをサポートしているわけではありません。

- Microsoft Azure

Astra Control API を使用してバケットを追加する手順については、を参照してください "Astra の自動化と API に関する情報"。

手順

1. 左側のナビゲーション領域で、 * バケット * を選択します。
   a. 「 * 追加」を選択します。
   b. バケットタイプを選択します。

   バケットを追加するときは、正しいバケットプロバイダを選択し、そのプロバイダに適したクレデンシャルを指定します。たとえば、タイプとして NetApp ONTAP S3 が許可され、 StorageGRID クレデンシャルが受け入れられますが、このバケットを使用して原因の以降のアプリケーションのバックアップとリストアはすべて失敗します。

   c. 新しいバケット名を作成するか、既存のバケット名とオプションの概要を入力します。

   バケット名と概要は、バックアップを作成するときに後で選択できるバックアップの場所として表示されます。この名前は、保護ポリシーの設定時にも表示されます。

d. S3 エンドポイントの名前または IP アドレスを入力します。

e. このバケットをすべてのバックアップのデフォルトバケットにする場合は、を確認します `Make this bucket the default bucket for this private cloud` オプション

> このオプションは、最初に作成したバケットに対しては表示されません。

f. 追加して続行します クレデンシャル情報。

S3 アクセスクレデンシャルを追加します

S3 アクセスクレデンシャルはいつでも追加できます。

手順

1. バケット（ Buckets ）ダイアログで、 * 追加（ Add ） * または * 既存の * を使用（ Use Existing * ）タブのいずれかを選択します。
   a. Astra Control の他のクレデンシャルと区別するクレデンシャルの名前を入力します。
   b. クリップボードからコンテンツを貼り付けて、アクセス ID とシークレットキーを入力します。

デフォルトのストレージクラスを変更する

クラスタのデフォルトのストレージクラスは変更できます。

手順

1. Astra Control Center Web UIで、[* Clusters]を選択します。
2. [* Clusters]ページで、変更するクラスタを選択します。
3. [* ストレージ *] タブを選択します。
4. 「ストレージクラス」カテゴリを選択します。
5. デフォルトとして設定するストレージクラスの* Actions *メニューを選択します。
6. 「デフォルトに設定」を選択します。

次の手順

Astra Control Center にログインしてクラスタを追加したので、 Astra Control Center のアプリケーションデータ管理機能を使い始めることができます。

- "ユーザを管理します"
- "アプリの管理を開始します"
- "アプリを保護します"
- "アプリケーションをクローニング"
- "通知を管理します"
- "Cloud Insights に接続します"
- "カスタム TLS 証明書を追加します"

詳細については、こちらをご覧ください

- "Astra Control API を使用"
- "既知の問題"

クラスタを追加するための前提条件

クラスタを追加する前に、前提条件が満たされていることを確認する必要があります。また、資格チェックを実行して、アストラコントロールセンターにクラスタを追加する準備ができていることを確認する必要があります。

クラスタを追加する前に必要な作業

クラスタがに記載された要件を満たしていることを確認します "アプリケーションクラスタの要件"。

> 管理対象のコンピューティングリソースとして 2 つ目の OpenShift 4.6 、 4.7 、または 4.8 クラスタを追加する場合は、 Astra Trident ボリュームスナップショット機能が有効になっていることを確認する必要があります。ネットアップの公式 Astra Trident をご覧ください "手順 " Trident を使用して、ボリューム Snapshot を有効にしてテストしてください。

- で構成されたAstra Trident StorageClasses "サポートされるストレージバックエンド" （すべてのタイプのクラスタに必要）
- Astra Control Center を使用してアプリケーションをバックアップおよび復元するためにバックアップ ONTAP システムで設定されたスーパーユーザーおよびユーザー ID 。ONTAP コマンドラインで次のコマンドを実行します。
  `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -superuser sysm --anon 65534`
- Astra Trident `volumesnapshotclass` 管理者によって定義されたオブジェクト。Astra Trident の詳細をご確認ください "手順" Trident を使用して、ボリューム Snapshot を有効にしてテストしてください。
- Kubernetes クラスタにデフォルトのストレージクラスが 1 つだけ定義されていることを確認します。

資格チェックを実行します

次の資格チェックを実行して、 Astra Control Center にクラスタを追加する準備ができていることを確認します。

手順

1. Trident のバージョンを確認

```
kubectl get tridentversions -n trident
```

Trident が存在する場合は、次のような出力が表示されます。

```
NAME       VERSION
trident    21.04.0
```

Trident が存在しない場合は、次のような出力が表示されます。

```
error: the server doesn't have a resource type "tridentversions"
```

> Trident がインストールされていない場合や、インストールされているバージョンが最新でない場合は、次に進む前に最新バージョンの Trident をインストールする必要があります。を参照してください "Trident のドキュメント" 手順については、を参照し

2. サポートされている Trident ドライバをストレージクラスが使用しているかどうかを確認します。プロビジョニング担当者の名前はとします `csi.trident.netapp.io`。次の例を参照してください。

```
kubectl get sc
NAME                    PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
ontap-gold (default)    csi.trident.netapp.io          Delete
Immediate           true                   5d23h
thin                    kubernetes.io/vsphere-volume   Delete
Immediate           false                  6d
```

admin ロールの kubeconfig を作成します

手順を実行する前に、マシンに次のものがあることを確認してください。

- `kubectl` V1.19以降がインストールされている
- アクティブなコンテキストのクラスタ管理者権限があるアクティブな kubeconfig です

手順

1. 次の手順でサービスアカウントを作成します。

   a. という名前のサービスアカウントファイルを作成します `astracontrol-service-account.yaml`。

      名前と名前空間を必要に応じて調整します。ここで変更を行った場合は、以降の手順でも同じ変更を適用する必要があります。

```
<strong>astracontrol-service-account.yaml</strong>
```

+

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astracontrol-service-account
  namespace: default
```

a. サービスアカウントを適用します。

```
kubectl apply -f astracontrol-service-account.yaml
```

2. （オプション）権限付きポッドの作成を許可しない制限付きポッドセキュリティポリシーをクラスタで使用している場合、またはポッドコンテナ内のプロセスをルートユーザとして実行できるようにしていない場合は、 Astra Control でポッドを作成および管理できるように、クラスタ用のカスタムポッドセキュリティポリシーを作成します。手順については、を参照してください "カスタムのポッドセキュリティポリシーを作成します"。
3. 次のようにクラスタ管理者権限を付与します。

a. を作成します `ClusterRoleBinding` という名前のファイルです `astracontrol-clusterrolebinding.yaml`。

必要に応じて、サービスアカウントの作成時に変更した名前と名前空間を調整します。

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

+

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

a. クラスタロールバインドを適用します。

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. サービスアカウントのシークレットを一覧表示します（置き換えます） `<context>` インストールに適したコンテキストを使用して、次の操作を行います。

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

出力の末尾は次のようになります。

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-vhz87"},
{ "name": "astracontrol-service-account-token-r59kr"}
]
```

内の各要素のインデックス `secrets` アレイは0から始まります。上記の例では、のインデックスです `astracontrol-service-account-dockercfg-vhz87` は0、のインデックスです `astracontrol-service-account-token-r59kr` は1です。出力で、 "token" という単語が含まれるサービスアカウント名のインデックスをメモしてください。

5. 次のように kubeconfig を生成します。

   a. を作成します `create-kubeconfig.sh` ファイル。交換してください `TOKEN_INDEX` 次のスクリプトの先頭に正しい値を入力します。

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
```

```
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token
-user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

b. コマンドをソースにし、 Kubernetes クラスタに適用します。

```
source create-kubeconfig.sh
```

6. （* オプション *）クラスタにわかりやすい名前にコベ econfig の名前を変更します。クラスタのクレデンシャルを保護します。

```
chmod 700 create-kubeconfig.sh
mv kubeconfig-sa.txt YOUR_CLUSTER_NAME_kubeconfig
```

次の手順

前提条件が満たされていることを確認したら、次は準備ができています "クラスタを追加"。

詳細については、こちらをご覧ください

- "Trident のドキュメント"
- "Astra Control API を使用"

カスタム TLS 証明書を追加します

既存の自己署名 TLS 証明書を削除して、認証局（ CA ）が署名した TLS 証明書に置き換えることができます。

必要なもの

- Astra Control Center をインストールした Kubernetes クラスタ
- 実行するクラスタ上のコマンドシェルへの管理アクセス `kubectl` コマンド
- CA の秘密鍵ファイルと証明書ファイル

自己署名証明書を削除します

既存の自己署名 TLS 証明書を削除します。

1. SSH を使用して、 Astra Control Center をホストする Kubernetes クラスタに管理ユーザとしてログインします。
2. 次のコマンドを使用して、現在の証明書に関連付けられているTLSシークレットを検索します `<ACC-deployment-namespace>` Astra Control Center導入ネームスペースを使用して、次の作業を行います。

```
kubectl get certificate -n <ACC-deployment-namespace>
```

3. 次のコマンドを使用して、現在インストールされているシークレットと証明書を削除します。

```
kubectl delete cert cert-manager-certificates -n <ACC-deployment-
namespace>
kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
```

新しい証明書を追加します

CA によって署名された新しい TLS 証明書を追加します。

1. 次のコマンドを使用して、 CA の秘密鍵ファイルと証明書ファイルを使用して新しい TLS シークレットを作成し、括弧 <> の引数を適切な情報に置き換えます。

```
kubectl create secret tls <secret-name> --key <private-key-filename>
--cert <certificate-filename> -n <ACC-deployment-namespace>
```

2. 次のコマンドと例を使用して、クラスタカスタムリソース定義（CRD）ファイルを編集し、を変更します `spec.selfSigned` の値 `spec.ca.secretName` 以前に作成したTLSシークレットを参照するには、次の手順を実行します

```
kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
<ACC-deployment-namespace>
....

#spec:
#  selfSigned: {}

spec:
  ca:
    secretName: <secret-name>
```

3. 次のコマンドと出力例を使用して、変更が正しいこと、および交換する証明書をクラスタで検証する準備ができていることを確認します `<ACC-deployment-namespace>` Astra Control Center導入ネームスペースを使用して、次の作業を行います。

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
....

Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. を作成します `certificate.yaml` 次の例を使用してファイルを作成し、括弧<>のプレースホルダ値を適切な情報に置き換えます。

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: <certificate-name>
  namespace: <ACC-deployment-namespace>
spec:
  secretName: <certificate-secret-name>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  - <astra.dnsname.example.com> #Replace with the correct Astra Control
Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. 次のコマンドを使用して証明書を作成します。

```
kubectl apply -f certificate.yaml
```

6. 次のコマンドと出力例を使用して、証明書が正しく作成されていること、および作成時に指定した引数（名前、期間、更新期限、 DNS 名など）を使用していることを確認します。

```
kubectl describe certificate -n <ACC-deployment-namespace>
....

Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. 次のコマンドおよび例を使用して、入力 CRD TLS オプションを編集し、新しい証明書シークレットを指定します。括弧 <> のプレースホルダ値を適切な情報に置き換えます。

```
kubectl edit ingressroutes.traefik.containo.us -n <ACC-deployment-
namespace>
....

# tls:
#    options:
#      name: default
#    secretName: secure-testing-cert
#    store:
#      name: default

 tls:
    options:
      name: default
    secretName: <certificate-secret-name>
    store:
      name: default
```

8. Web ブラウザを使用して、 Astra Control Center の導入 IP アドレスにアクセスします。
9. 証明書の詳細がインストールした証明書の詳細と一致していることを確認します。
10. 証明書をエクスポートし、結果を Web ブラウザの証明書マネージャにインポートします。

カスタムのポッドセキュリティポリシーを作成します

Astra Control では、管理対象のクラスタに Kubernetes ポッドを作成して管理する必要があります。クラスタで、特権ポッドの作成を許可しない制限付きポッドセキュリティポリシーを使用している場合、またはポッドコンテナ内のプロセスをルートユーザとして実行できるように許可していない場合は、制限の少ないポッドセキュリティポリシーを作成して、 Astra Control がこれらのポッドを作成および管理できるようにする必要があります。

手順

1. デフォルトよりも制限の緩いクラスタの PoD セキュリティポリシーを作成してファイルに保存します。
   例：

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: astracontrol
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
```

2. ポッドセキュリティポリシーの新しいロールを作成します。

```
kubectl-admin create role psp:astracontrol \
    --verb=use \
    --resource=podsecuritypolicy \
    --resource-name=astracontrol
```

3. 新しいロールをサービスアカウントにバインドします。

```
kubectl-admin create rolebinding default:psp:astracontrol \
    --role=psp:astracontrol \
    --serviceaccount=astracontrol-service-account:default
```

Astra Control Center に関するよくある質問

この FAQ は、質問に対する簡単な回答を探している場合に役立ちます。

概要

次のセクションでは、 Astra Control Center を使用しているときに発生する可能性のあるその他の質問に対する回答を示します。詳しい説明については、 astra.feedback@netapp.com までお問い合わせください

Astra Control Center へのアクセス

- Astra Control の URL は何であるか。 *

Astra Control Center は、ローカル認証と各環境に固有の URL を使用します。

URL には、ブラウザで、 Astra Control Center をインストールしたときに、 Astra_control_center_min YAML カスタムリソース定義（ CRD ）ファイルの spec.astraatAddress フィールドに設定した完全修飾ドメイン名（ FQDN ）を入力します。電子メールは、 Astra_control_center_min YAML CRD の spec.email フィールドで設定した値です。

ライセンス

- 評価ライセンスを使用しています。フルライセンスに変更する方法を教えてください。 *

ネットアップライセンスファイル（ NLF ）を取得することで、フルライセンスに簡単に変更できます。

- 手順 *
- 左側のナビゲーションから、 * アカウント * > * ライセンス * を選択します。
- 「 * ライセンスの追加 * 」を選択します。
- ダウンロードしたライセンスファイルを参照し、 * 追加 * を選択します。
- 評価ライセンスを使用しています。アプリを管理できますか？ *

はい、評価ライセンスを使用して、管理アプリケーション機能をテストできます。

Kubernetes クラスタを登録しています

- Astra Control に追加したワーカーノードを Kubernetes クラスタに追加する必要があります。どうすればよいですか？ *

新しいワーカーノードを既存のプールに追加できます。これらは Astra Control によって自動的に検出されます。新しいノードが Astra Control に表示されない場合は、新しいワーカーノードでサポートされているイメージタイプが実行されているかどうかを確認します。を使用して、新しいワーカーノードの健全性を確認することもできます `kubectl get nodes` コマンドを実行します

- クラスタの管理を適切に解除するにはどうすればよいですか *
  1. "Astra Control からアプリケーションの管理を解除"。
  2. "Astra Control からクラスタの管理を解除"。

- Kubernetes クラスタを Astra Control から削除した後、アプリケーションとデータはどうなりますか。 *

Astra Control からクラスタを削除しても、クラスタの構成（アプリケーションと永続的ストレージ）は変更されません。このクラスタで作成されたアプリケーションの Snapshot やバックアップを Astra Control で復元することはできません。Astra Control で作成した永続的ストレージのバックアップは Astra Control に残っていますが、リストアには使用できません。

> 他の方法でクラスタを削除する場合は、必ず事前に Astra Control からクラスタを削除してください。Astra Control で管理している間に別のツールを使用してクラスタを削除した場合、原因で Astra Control アカウントに問題が発生する可能性があります。

- NetApp Trident は、管理を解除すると自動的にクラスタからアンインストールされますか？ * Astra Control Center からクラスタを管理を解除しても、 Trident は自動的にクラスタからアンインストールされることはありません。Trident をアンインストールするには、が必要です "Trident のドキュメントでは、次の手順を実行します"。

アプリケーションの管理

- Astra Control はアプリケーションを導入できますか。 *

Astra Control はアプリケーションを導入しない。アプリケーションは Astra Control の外部に導入する必要があります。

- アプリケーションを Astra Control から管理しなくなった後、どうなりますか。 *

既存のバックアップまたは Snapshot がすべて削除されます。アプリケーションとデータは引き続き使用できます。管理対象外のアプリケーション、またはそのアプリケーションに属するバックアップや Snapshot では、データ管理操作を実行できません。

- ネットアップ以外のストレージにあるアプリケーションは Astra Control で管理できますか。 *

いいえネットアップ以外のストレージを使用しているアプリケーションは Astra Control で検出できますが、ネットアップ以外のストレージを使用しているアプリケーションは管理できません。

- Astra Control 自体を管理すべきですか？ * いいえ、 Astra Control 自体は「システムアプリケーション」であるため、管理すべきではありません。
- 正常でないポッドはアプリケーション管理に影響しますか？ * 管理対象アプリケーションにポッドが正常な状態でない場合、 Astra Control は新しいバックアップとクローンを作成できません。

データ管理の操作

- 作成していないスナップショットがアカウントにあります。彼らはどこから来たのですか。 *

一部の状況では、バックアップ、クローン、またはリストアのプロセスの一環として、 Astra Control によってスナップショットが自動的に作成されます。

- アプリケーションは複数の PVS を使用しています。Astra Control は、これらすべての PVC のスナップショットとバックアップを作成しますか。 *

はい。Astra Control によるアプリケーションのスナップショット操作には、アプリケーションの PVC にバインドされているすべての PVS のスナップショットが含まれます。

- Astra Control で取得したスナップショットを、別のインターフェイスやオブジェクトストレージから直接管理できますか。 *

いいえAstra Control で作成したスナップショットとバックアップは、 Astra Control でのみ管理できます。

Astra を使用する

アプリの管理を開始します

お先にどうぞ "Astra Control 管理にクラスタを追加"では、クラスターにアプリケーションをインストールし（Astra Controlの外部）、Astra Controlの[アプリケーション]ページに移動して、アプリケーションとそのリソースの管理を開始できます。

詳細については、を参照してください "アプリケーション管理の要件"。

サポートされているアプリインストール方法

Astra Control は、次のアプリケーションインストール方法をサポートしています。

- * マニフェストファイル * ： Astra Control は、 kubectl を使用してマニフェストファイルからインストールされたアプリケーションをサポートします。例：

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

- * Helm 3 * ： Helm を使用してアプリケーションをインストールする場合、 Astra Control には Helm バージョン 3 が必要です。Helm 3 （または Helm 2 から Helm 3 にアップグレード）を使用してインストールされたアプリケーションの管理とクローニングが完全にサポートされています。Helm 2 でインストールされたアプリケーションの管理はサポートされていません。
- オペレータが導入したアプリケーション：Astra Controlは、ネームスペースを対象とした演算子を使用してインストールされたアプリケーションをサポートしています。一般的には、「パスバイリファレンス」アーキテクチャではなく「パスバイバリュー」アーキテクチャで設計されています。インストールする演算子とアプリケーションは、同じ名前空間を使用する必要があります。このような名前空間を使用するには、演算子の deployment.yaml ファイルを変更する必要があります。

  これらのパターンに続くいくつかのオペレータアプリを次に示します。

  - "Apache K8ssandra"

    > K8ssandra では、 In Place リストア処理がサポートされます。新しいネームスペースまたはクラスタにリストアするには、アプリケーションの元のインスタンスを停止する必要があります。これは、ピアグループ情報がインスタンス間通信を行わないようにするためです。アプリケーションのクローニングはサポートされていません。

  - "Jenkins CI"
  - "Percona XtraDB クラスタ"

Astra Controlでは、「パスバイリファレンス」アーキテクチャ（CockroachDBオペレータなど）で設計されたオペレータをクローニングできない場合があります。クローニング処理では、クローニング処理の一環として独自の新しいシークレットが存在する場合でも、クローニングされたオペレータがソースオペレータから Kubernetes シークレットを参照しようとします。Astra Control がソースオペレータの Kubernetes シークレットを認識しないため、クローニング処理が失敗する場合があります。

クラスタにアプリをインストールします

お先にどうぞ "クラスタが追加されました" Astra Controlを使用すると、アプリケーションをインストールしたり、クラスタ上の既存のアプリケーションを管理したりできます。1つのネームスペースに限定されるアプリケーションはすべて管理できます。

アプリの管理

Astra Controlがクラスタ上のネームスペースを検出したら、管理するアプリケーションを定義できます。を選択できます "ネームスペース全体を単一のアプリケーションとして管理することも、ネームスペース内の1つ以上のアプリケーションを個別に管理することもできます"。データ保護処理に必要な精度のレベルが重要になります。

Astra Controlを使用すると、階層の両方のレベル（ネームスペースとそのネームスペース内のアプリケーション）を別々に管理できますが、いずれか一方を選択することを推奨します。Astra Control で実行したアクションは、ネームスペースレベルとアプリケーションレベルの両方で同時に実行される場合、失敗する可能性があります。

> たとえば、「Maria」に対して、毎週同じ頻度でバックアップを作成するように設定することもできますが、同じネームスペースにある「MariaDB」をバックアップする頻度を高く設定する必要があるとします。これらのニーズに基づいて、アプリケーションを個別に管理する必要があります。また、シングルネームスペースアプリケーションとして管理する必要はありません。

必要なもの

- KubernetesクラスタをAstra Controlに追加。
- クラスタにインストールされているアプリケーションが1つ以上あります。 サポートされているアプリケーションのインストール方法については、こちらをご覧ください。
- アクティブなポッドが1つ以上あります。
- Astra Controlに追加したKubernetesクラスタで指定された名前空間。
- （オプション）すべてのにKubernetesラベル "サポートされるKubernetesリソース"。

> ラベルは、 Kubernetes オブジェクトに割り当てて識別できるキーと値のペアです。ラベルを使用すると、 Kubernetes オブジェクトのソート、整理、検索が簡単になります。Kubernetes のラベルの詳細については、 "Kubernetes の公式ドキュメントを参照してください"。

開始する前に、を理解しておく必要があります "標準ネームスペースとシステムネームスペースの管理"。

Astra Control API を使用してアプリケーションを管理する方法については、を参照してください "Astra の自動化と API に関する情報"。

アプリケーション管理オプション

- [アプリケーションとして管理するリソースを定義します]
- [アプリケーションとして管理するネームスペースを定義します]

追加のアプリケーション管理オプション

- [アプリの管理を解除します]

アプリケーションとして管理するリソースを定義します

を指定できます "アプリケーションを構成するKubernetesリソース" Astra Controlで管理したい。アプリケーションを定義すると、Kubernetesクラスタの要素を1つのアプリケーションにグループ化できます。このKubernetesリソースの集まりは、ネームスペースとラベル選択条件によって分類されます。

アプリケーションを定義することで、クローン、スナップショット、バックアップなどのAstra Control操作に含めるものをより細かく制御できます。

> アプリケーションを定義するときは、保護ポリシーを使用して複数のアプリケーションにKubernetesリソースを含めないようにしてください。Kubernetesリソース上の保護ポリシーが重複していると、原因 のデータが競合する可能性があります。 ベストプラクティスの詳細をご確認ください。

手順

1. [アプリケーション（Applications）]ページで、[定義（ Define）]を選択します
2. [アプリケーションの定義（* Define application）]ウィンドウで、アプリケーション名を入力します。
3. [**Cluster**]ドロップダウン・リストから、アプリケーションが実行されているクラスタを選択します。
4. 「名前空間」ドロップダウンリストからアプリケーションの名前空間を選択します。

> アプリケーションは、単一クラスタ上の指定されたネームスペース内でのみ定義できます。Astra Controlでは、複数のネームスペースまたはクラスタにまたがるアプリケーションの機能はサポートされていません。

5. アプリケーションと名前空間のラベルを入力します。ラベルまたはラベルの選択基準（クエリー）を1つ指定できます。

> Kubernetes のラベルの詳細については、 "Kubernetes の公式ドキュメントを参照してください"。

6. [定義（Define *）]を選択した後、必要に応じて他のアプリケーションについても同じ手順を繰り返します。

アプリケーションの定義が完了すると、アプリケーション（Applications）ページのアプリケーションのリストにアプリケーションが表示されます。クローンを作成し、バックアップとスナップショットを作成できるようになりました。

> 追加したアプリケーションの保護列に警告アイコンが表示されている場合は、バックアップされておらず、まだバックアップのスケジュールが設定されていないことを示しています。

> 特定のアプリケーションの詳細を表示するには、アプリケーション名を選択します。

アプリケーションとして管理するネームスペースを定義します

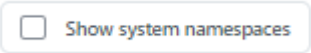
ネームスペースのリソースをアプリケーションとして定義することで、ネームスペース内のすべてのKubernetesリソースをAstra Control管理に追加できます。特定の名前空間内のすべてのリソースを同じよう

な方法で、共通の間隔で管理および保護する場合は、アプリケーションを個別に定義することをお勧めします。

手順

1. クラスタページで、クラスタを選択します。
2. [名前空間]タブを選択します。
3. 管理するアプリケーションリソースを含む名前空間のアクションメニューを選択し、*アプリケーションとして定義*を選択します。

> 複数のネームスペースを管理する場合は、ネームスペースを選択し、左上隅の*アクション*ボタンを選択して*管理*を選択します。

> [システムネームスペースを表示（Show system Namespaces）]チェックボックスを選択して、アプリケーション管理で通常はデフォルトで使用されないシステムネームスペースを表示します。 
>
> "詳細はこちら"。

このプロセスが完了すると、ネームスペースに関連付けられているアプリケーションがに表示されます `Associated applications` 列（Column）：

アプリの管理を解除します

アプリケーションのバックアップ、スナップショット、またはクローンを作成する必要がなくなった場合は、管理を停止できます。

> アプリケーションの管理を解除すると、以前に作成したバックアップやスナップショットは失われます。

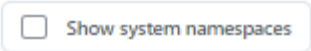
手順

1. 左側のナビゲーションバーから、「 * アプリケーション * 」を選択します。
2. アプリケーションを選択します。
3. [アクション（* Actions ）]列のメニューから、[*管理解除（* Unmanage *）]を選択します。
4. 情報を確認します。
5. 「 unmanage 」と入力して確定します。
6. [はい、アプリケーションの管理を解除 *] を選択します。

システムネームスペースについて教えてください。

Astra Controlは、Kubernetesクラスタ上のシステムネームスペースも検出します。これらのシステムネームスペースはデフォルトでは表示されません。システムアプリケーションリソースのバックアップが必要になることがまれです。

選択したクラスタの[ネームスペース]タブからシステムネームスペースを表示するには、[システムネームスペースを表示]チェックボックスをオンにします。



Astra Control 自体は標準のアプリケーションではなく、「システムアプリケーション」です。Astra Control 自体は管理しないでください。Astra Control 自体は、管理用にデフォルトでは表示されません。

例：リリースごとに保護ポリシーを分ける

この例では、DevOpsチームが「カナリアリリースの導入を管理しています。チームのクラスタにはnginxを実行するポッドが3つあります。そのうちの 2 つのポッドは、安定版リリース専用です。3 番目のポッドはカナリアリリース用です。

DevOpsチームのKubernetes管理者がラベルを追加します `deployment=stable` を使用して、安定版リリースポッドに移動しますチームがラベルを追加します `deployment=canary` カナリアリリースポッドに移動します。

チームの安定版リリースには、 1 時間ごとの Snapshot と日次バックアップの要件が含まれています。カナリアリリースはより一時的なリリースなので、ラベル付きのものは何でも短時間で、よりアグレッシブな保護ポリシーを作成したいと考えています `deployment=canary`。

データの競合を回避するために、管理者は「カナリア」リリース用と「stable」リリース用の2つのアプリケーションを作成します。これにより、 Kubernetes オブジェクトの 2 つのグループに対して、バックアップ、Snapshot 、およびクローニングの処理が分離されます。

詳細については、こちらをご覧ください

- "Astra Control API を使用"

アプリを保護します

保護の概要

Astra Control Center を使用して、アプリケーションのバックアップ、クローン、スナップショット、および保護ポリシーを作成できます。アプリケーションをバックアップすることで、サービスや関連データを可能な限り利用できるようになります。災害時にバックアップからリストアすることで、アプリケーションと関連データを最小限の中断で完全にリカバリできます。バックアップ、クローン、 Snapshot を使用すると、ランサムウェアや偶発的なデータ損失、環境障害などの一般的な脅威からデータを保護できます。 "Astra Control Center で使用可能なデータ保護の種類と、それらを使用するタイミングについて説明します"。

また、ディザスタリカバリに備えてアプリケーションをリモートクラスタにレプリケートすることもできます。

アプリケーション保護のワークフロー

次のワークフロー例を使用して、アプリケーションの保護を開始できます。

[1つ] すべてのアプリケーションを保護

アプリケーションをすぐに保護するには、次の手順を実行します。 "すべてのアプリケーションの手動バックアップを作成する"。

[2 つ] 各アプリケーションの保護ポリシーを設定します

将来のバックアップとスナップショットを自動化するには、 "各アプリケーションの保護ポリシーを設定します"。たとえば、週単位のバックアップと日単位の Snapshot をそれぞれ 1 カ月ずつ保持して開始できます。手動バックアップやスナップショットよりも、保護ポリシーを使用してバックアップとスナップショットを自動化することを強く推奨します。

[3つ] 保護ポリシーを調整します

アプリとその使用パターンが変化したら、必要に応じて保護ポリシーを調整して、最適な保護を実現します。

[4.] アプリケーションをリモートクラスタにレプリケートします

"アプリケーションをレプリケートします" NetApp SnapMirrorテクノロジを使用してリモートクラスタにバックアップする場合Astra Controlは、Snapshotをリモートクラスタにレプリケートし、非同期のディザスタリカバリ機能を提供します。

[5 つ] 災害が発生した場合は、最新のバックアップまたはレプリケーションを使用してアプリケーションをリモートシステムにリストアします

データ損失が発生した場合は、を使用してリカバリできます "最新のバックアップをリストアしています" まず、各アプリケーションについて説明します。その後、最新の Snapshot をリストアできます（使用可能な場合）。または、リモートシステムへのレプリケーションを使用することもできます。

Snapshot とバックアップでアプリケーションを保護

自動保護ポリシーまたはアドホックベースを使用して、スナップショットやバックアップを作成することで、すべてのアプリケーションを保護します。Astra の UI またはを使用できます "Astra Control API" アプリを保護します。

Helm を使用してアプリケーションを展開する場合、 Astra Control Center には Helm バージョン 3 が必要です。Helm 3 （または Helm 2 から Helm 3 にアップグレード）を使用して展開されたアプリケーションの管理とクローニングが完全にサポートされています。Helm 2 で展開されたアプリケーションはサポートされていません。

OpenShift クラスタでアプリケーションをホストするプロジェクトを作成すると、プロジェクト（または Kubernetes ネームスペース）に SecurityContext UID が割り当てられます。Astra Control Center でアプリケーションを保護し、 OpenShift でそのアプリケーションを別のクラスタまたはプロジェクトに移動できるようにするには、アプリケーションを任意の UID として実行できるようにポリシーを追加する必要があります。たとえば、次の OpenShift CLI コマンドは、 WordPress アプリケーションに適切なポリシーを付与します。

```
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
```

アプリケーションデータの保護に関連する次のタスクを実行できます。

- [保護ポリシーを設定します]
- Snapshot を作成します
- [バックアップを作成します]
- Snapshot とバックアップを表示します

- Snapshot を削除します
- [バックアップをキャンセルします]
- [バックアップを削除します]

保護ポリシーを設定します

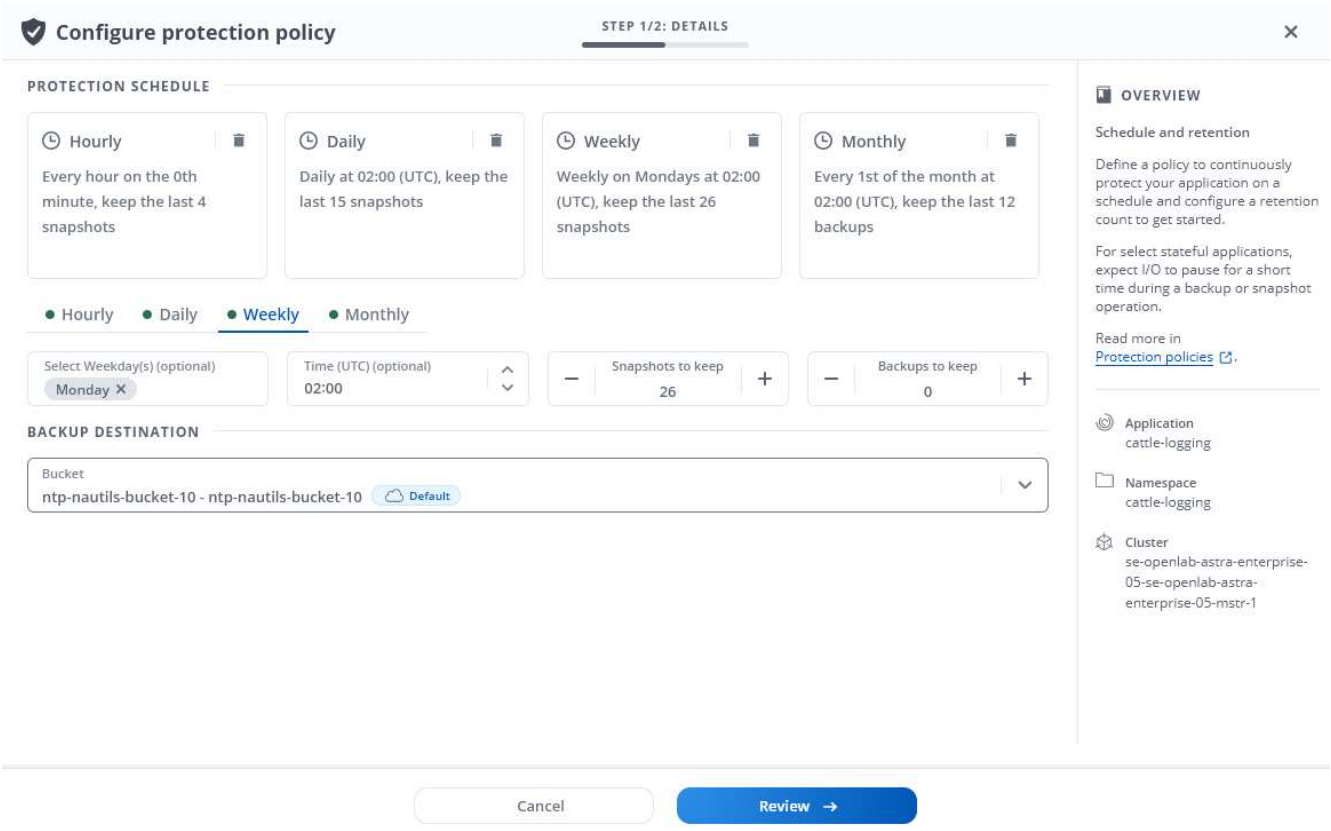
保護ポリシーは、定義されたスケジュールでスナップショット、バックアップ、またはその両方を作成することでアプリケーションを保護します。Snapshot とバックアップを毎時、日次、週次、および月単位で作成し、保持するコピーの数を指定できます。たとえば、保護ポリシーでは、週単位のバックアップと日単位の Snapshot が作成され、 1 カ月間はバックアップと Snapshot が保持されます。スナップショットやバックアップを作成する頻度と、それらを保持する期間は、組織のニーズによって異なります。

手順

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. [* データ保護 *] を選択します。
3. ［ 保護ポリシーの設定 ］ を選択します。
4. 毎時、日次、週次、および月単位で保持する Snapshot とバックアップの数を選択して、保護スケジュールを定義します。

   スケジュールは、毎時、毎日、毎週、および毎月の各スケジュールで同時に定義できます。保持レベルを設定するまで、スケジュールはアクティブになりません。

   次の例では、 Snapshot とバックアップの保護スケジュールとして、毎時、毎日、毎週、毎月の 4 つを設定します。



5. [* Review （レビュー）] を選択します
6. ［ * 保護ポリシーの設定 * ］ を選択します

結果

Astra Control Center は、定義したスケジュールと保持ポリシーを使用して、スナップショットとバックアップを作成し、保持することによって、データ保護ポリシーを実装します。

Snapshot を作成します

オンデマンド Snapshot はいつでも作成できます。

手順

1. 「 * アプリケーション * 」を選択します。
2. 目的のアプリケーションの * アクション * 列のオプションメニューから、 * スナップショット * を選択します。
3. スナップショットの名前をカスタマイズし、 * Review * を選択します。
4. Snapshot の概要を確認し、「 * Snapshot * 」を選択します。

結果

スナップショットプロセスが開始されます。スナップショットは、ステータスが「 * 使用可能 * 」である場合に成功します。この場合、「 * データ保護 * > * スナップショット * 」ページの「 * アクション * 」列に表示されます。

バックアップを作成します

アプリケーションはいつでもバックアップできます。

> Astra Control Center の S3 バケットは、使用可能容量を報告しません。Astra Control Center で管理されているアプリケーションのバックアップまたはクローニングを行う前に、 ONTAP または StorageGRID 管理システムでバケット情報を確認します。

手順

1. 「 * アプリケーション * 」を選択します。
2. 目的のアプリケーションの * アクション * 列のオプションメニューから、 * バックアップ * を選択します。
3. バックアップ名をカスタマイズする。
4. 既存のスナップショットからアプリケーションをバックアップするかどうかを選択します。このオプションを選択すると、既存の Snapshot のリストから選択できます。
5. ストレージバケットのリストから選択して、バックアップのデスティネーションを選択します。
6. [* Review （レビュー）] を選択します
7. バックアップの概要を確認し、「 * Backup * 」を選択します。

結果

Astra Control Center は、アプリケーションのバックアップを作成します。

ネットワークに障害が発生している場合や、処理速度が異常に遅い場合は、バックアップ処理がタイムアウトする可能性があります。その結果、バックアップは失敗します。

実行中のバックアップを停止する方法はありません。バックアップを削除する必要がある場合は、完了するまで待ってから、の手順を実行してください [バックアップを削除します]。失敗したバックアップを削除するには、 "Astra Control API を使用"。

データ保護処理（クローン、バックアップ、リストア）が完了して永続ボリュームのサイズを変更したあと、新しいボリュームのサイズが UI に表示されるまでに最大 20 分かかります。データ保護処理にかかる時間は数分です。また、ストレージバックエンドの管理ソフトウェアを使用してボリュームサイズの変更を確認できます。

Snapshot とバックアップを表示します

アプリケーションのスナップショットとバックアップは、 [データ保護（ Data Protection ）] タブで表示できます。

手順

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. [* データ保護 *] を選択します。

   デフォルトでは、 Snapshot が表示されます。

3. バックアップのリストを表示するには、「 * Backups * 」を選択します。

Snapshot を削除します

不要になったスケジュール済みまたはオンデマンドの Snapshot を削除します。

レプリケート中のSnapshotコピーは削除できません。

手順

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. [* データ保護 *] を選択します。
3. 目的のスナップショットの * アクション * 列のオプションメニューから、 * スナップショットの削除 * を選択します。
4. 削除を確認するために「 delete 」と入力し、「 * はい、 Snapshot を削除します * 」を選択します。

結果

Astra Control Center がスナップショットを削除します。

バックアップをキャンセルします

実行中のバックアップをキャンセルすることができます。

バックアップをキャンセルするには、そのバックアップが実行中の状態である必要があります。Pending状態のバックアップはキャンセルできません。

手順

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. [* データ保護 *] を選択します。
3. 「 * Backups * 」を選択します。
4. 目的のバックアップの[アクション（* Actions ）]列の**[**オプション（**Options**）**]**メニューから、**[***キャンセル（* Cancel *）]を選択します。
5. 削除を確認するために「CANCEL」と入力し、「* Yes、cancel backup *」を選択します。

バックアップを削除します

不要になったスケジュール済みまたはオンデマンドのバックアップを削除します。

実行中のバックアップを停止する方法はありません。バックアップを削除する必要がある場合は、完了するまで待ってから、以下の手順を実行してください。失敗したバックアップを削除するには、 "Astra Control API を使用"。

手順

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. [* データ保護 *] を選択します。
3. 「 * Backups * 」を選択します。
4. 目的のバックアップの [* アクション *] 列の [オプション] メニューから、 [* バックアップの削除 *] を選択します。
5. 削除を確認するために「 delete 」と入力し、「 * はい、バックアップを削除 * 」を選択します。

結果

Astra Control Center はバックアップを削除します。

アプリケーションのリストア

Astra Control を使用すると、スナップショットまたはバックアップからアプリケーションをリストアできます。同じクラスタにアプリケーションをリストアする場合、既存の Snapshot からのリストアは高速です。Astra Control UI またはを使用できます "Astra Control API" アプリを復元するには、

このタスクについて

- アプリケーションをリストアする前に、アプリケーションのスナップショットを作成するか、バックアップすることを強くお勧めします。リストアに失敗した場合に、Snapshotまたはバックアップからクローニングできます。
- Helm を使用してアプリケーションを展開する場合、 Astra Control Center には Helm バージョン 3 が必要です。Helm 3 （または Helm 2 から Helm 3 にアップグレード）を使用して展開されたアプリケーションの管理とクローニングが完全にサポートされています。Helm 2 で展開されたアプリケーションはサポー

トされていません。

- 別のクラスタにリストアする場合は、同じ永続的ボリュームアクセスモード（ ReadWriteMany など）をクラスタで使用していることを確認してください。デスティネーションの永続ボリュームアクセスモードが異なると、リストア処理は失敗します。
- 名前空間の名前 / ID または名前空間ラベルによる名前空間の制約を持つメンバーユーザーは、同じクラスタ上の新しい名前空間、または組織のアカウント内の他のクラスタに対して、アプリケーションのクローンまたはリストアを実行できます。ただし、同じユーザが、クローニングまたはリストアされたアプリケーションに新しいネームスペースからアクセスすることはできません。クローンまたはリストア処理によって新しいネームスペースが作成されると、アカウントの管理者 / 所有者はメンバーユーザアカウントを編集し、該当するユーザに新しいネームスペースへのアクセスを許可するロールの制限を更新できます。
- OpenShift クラスタでアプリケーションをホストするプロジェクトを作成すると、プロジェクト（または Kubernetes ネームスペース）に SecurityContext UID が割り当てられます。Astra Control Center でアプリケーションを保護し、 OpenShift でそのアプリケーションを別のクラスタまたはプロジェクトに移動できるようにするには、アプリケーションを任意の UID として実行できるようにポリシーを追加する必要があります。たとえば、次の OpenShift CLI コマンドは、 WordPress アプリケーションに適切なポリシーを付与します。

```
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
```

手順

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. 「 * データ保護 * 」を選択します。
3. Snapshot からリストアする場合は、 * Snapshots * アイコンを選択したままにします。それ以外の場合は、「 * Backups * 」アイコンを選択してバックアップからリストアします。
4. リストア元のスナップショットまたはバックアップの [* アクション *] 列の [オプション] メニューから、[* アプリケーションのリストア *] を選択します。
5. * リストアの詳細 * ：リストアされたアプリの詳細を指定します。デフォルトでは、現在のクラスタとネームスペースが表示されます。これらの値をそのままにしておくと、アプリがインプレースで復元され、アプリは以前のバージョンのに戻ります。別のクラスタまたはネームスペースにリストアする場合は、これらの値を変更してください。
   - アプリケーションの名前と名前空間を入力します。
   - アプリケーションのデスティネーションクラスタを選択します。
   - [* Review （レビュー）] を選択します

   > 以前に削除したネームスペースにリストアすると、同じ名前の新しいネームスペースがリストアプロセスで作成されます。以前に削除したネームスペースでアプリケーションを管理する権限を持つユーザは、新しく作成したネームスペースに手動で権限を復元する必要があります。

6. * リストアの概要 * ：リストア操作の詳細を確認し、「 restore 」と入力して、 * Restore * を選択します。

結果

Astra Control Center は、指定した情報に基づいてアプリケーションを復元します。アプリケーションをインプレースでリストアした場合、既存の永続ボリュームの内容が、リストアしたアプリケーションの永続ボリュ

ームの内容に置き換えられます。

> データ保護処理（クローン、バックアップ、リストア）が完了して永続ボリュームのサイズが変更されたあと、Web UIに新しいボリュームサイズが表示されるまでに最大20分かかります。データ保護処理にかかる時間は数分です。また、ストレージバックエンドの管理ソフトウェアを使用してボリュームサイズの変更を確認できます。

SnapMirrorテクノロジを使用してアプリケーションをリモートシステムにレプリケート

Astra Controlを使用すると、NetApp SnapMirrorテクノロジの非同期レプリケーション機能を使用して、RPO（目標復旧時点）とRTO（目標復旧時間）の低いアプリケーションのビジネス継続性を構築できます。設定が完了すると、アプリケーションはデータやアプリケーションの変更をクラスタ間でレプリケートできるようになります。

バックアップ/リストアとレプリケーションの比較については、を参照してください "データ保護の概念"。

アプリケーションは、オンプレミスのみ、ハイブリッド、マルチクラウドなど、さまざまなシナリオでレプリケートできます。

- オンプレミスサイトAからオンプレミスサイトBへ
- Cloud Volumes ONTAP を使用してオンプレミスからクラウドに移行できます
- Cloud Volumes ONTAP を使用したクラウドをオンプレミスに移行
- Cloud Volumes ONTAP を使用したクラウドからクラウドへ（同じクラウドプロバイダ内の異なるリージョン間または異なるクラウドプロバイダ間）

Astra Controlを使用すれば、オンプレミスのクラスタからクラウドへ（Cloud Volumes ONTAP を使用）、またはクラウド間（Cloud Volumes ONTAP からCloud Volumes ONTAP へ）にアプリケーションをレプリケートできます。

> （別のクラスタまたはサイトで実行されている）別のアプリケーションを逆方向に同時にレプリケートできます。たとえば、アプリケーションA、B、Cはデータセンター1からデータセンター2にレプリケートでき、アプリケーションX、Y、Zはデータセンター2からデータセンター1にレプリケートできます。

Astra Controlを使用すると、アプリケーションのレプリケーションに関連する次のタスクを実行できます。

- [レプリケーション関係を設定]
- [デスティネーションクラスタでレプリケートされたアプリケーションをオンラインにする（フェイルオーバー）]
- [フェイルオーバーしたレプリケーションを再同期します]
- [アプリケーションのレプリケーションを反転する]
- [アプリケーションを元のソースクラスタにフェイルバックします]
- [アプリケーションレプリケーション関係を削除します]

レプリケーションの前提条件

を参照してください "レプリケーションの前提条件" 始める前に。

レプリケーション関係を設定

レプリケーション関係を設定するには、レプリケーションポリシーを構成する次の作業を行います。

- Astra ControlでのアプリケーションSnapshotの作成頻度の選択（アプリケーションのKubernetesリソースと、アプリケーションの各ボリュームのボリュームSnapshotを含む）
- レプリケーションスケジュールの選択（Kubernetesリソースと永続ボリュームデータを含む）
- Snapshotを作成する時刻を設定します

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. ［アプリケーション］ページで、［データ保護］>［レプリケーション］タブを選択します。
3. ［データ保護］＞［レプリケーション］タブで、［レプリケーションポリシーの設定］を選択します。または、[アプリケーション保護]ボックスから[アクション]オプションを選択し、[レプリケーションポリシーの構成]を選択します。
4. 次の情報を入力または選択します。
   - デスティネーションクラスタ
   - デスティネーションストレージクラス：デスティネーションONTAP クラスタでペアリングされているSVMを使用するストレージクラスを選択または入力します。
   - レプリケーションタイプ：現在使用できるレプリケーションタイプは「非同期」のみです。
   - デスティネーションネームスペース：デスティネーションクラスタの新しいデスティネーションネームスペースまたは既存のデスティネーションネームスペースを入力します。

     選択したネームスペース内で競合するリソースがすべて上書きされます。

   - レプリケーション頻度：Snapshotを作成してデスティネーションにレプリケートする頻度を指定します。
   - オフセット：Astra Controlでスナップショットを作成する時間の上部から分数を設定します。オフセットを使用すると、他のスケジュールされた処理と競合しないようにすることができます。たとえば、10：02から5分ごとにSnapshotを作成する場合は、オフセットの分として「02」を入力します。結果は、10：02、10：07、10：12などになります
5. 「次へ」を選択し、概要を確認して、「保存」を選択します。

   最初に、最初のスケジュールが実行される前にステータスに「app_mirror」と表示されます。

   Astra Control：レプリケーションに使用するアプリケーションSnapshotを作成

6. アプリケーションのスナップショットステータスを表示するには、アプリケーション>*スナップショット*タブを選択します。

   Snapshot名には「replication-schedule -<string>」の形式を使用します。Astra Controlは、レプリケーショ

ンに使用された最後のSnapshotを保持古いレプリケーションSnapshotは、レプリケーションが正常に完了すると削除されます。

結果

これにより、レプリケーション関係が作成されます。

Astra Controlは、関係を確立した結果として次のアクションを実行します。

- デスティネーションにネームスペースを作成します（存在しない場合）。
- 送信元アプリケーションのPVCに対応する宛先ネームスペースにPVCを作成します。
- アプリケーションと整合性のある最初のSnapshotを作成します。
- 初期Snapshotを使用して、永続ボリュームのSnapMirror関係を確立します。

データ保護ページには、レプリケーション関係の状態とステータスが表示されます。<Health status>|<Relationship life cycle state>

たとえば、Normal | Establishedです

レプリケーションの状態とステータスの詳細については、以下を参照してください。

デスティネーションクラスタでレプリケートされたアプリケーションをオンラインにする（フェイルオーバー）

Astra Controlを使用すると、レプリケートされたアプリケーションをデスティネーションクラスタに「フェイルオーバー」できます。この手順 はレプリケーション関係を停止し、デスティネーションクラスタでアプリケーションをオンラインにします。ソースクラスタのアプリケーションが稼働していた場合、この手順 はそのアプリケーションを停止しません。

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. ［アプリケーション］ページで、［データ保護］>［レプリケーション］タブを選択します。
3. [データ保護（Data Protection）]>[複製（Replication）]タブの[アクション（Actions）]メニューから、[フェールオーバー*（フェールオーバー*）]を選択し
4. フェイルオーバーページで、情報を確認し、*フェイルオーバー*を選択します。

結果

手順 のフェイルオーバーでは、次の処理が実行されます。

- デスティネーションクラスタでは、レプリケートされた最新のSnapshotに基づいてアプリケーションが開始されます。
- ソースクラスタとアプリケーション（動作している場合）は停止されず、引き続き実行されます。
- レプリケーションの状態は「フェイルオーバー」に変わり、完了すると「フェイルオーバー」に変わります。
- ソースアプリケーションの保護ポリシーは、フェイルオーバー時にソースアプリケーションに存在するスケジュールに基づいて、デスティネーションアプリケーションにコピーされます。
- Astra Controlには、ソースクラスタとデスティネーションクラスタの両方のアプリケーションと、それぞれの健全性が表示されます。

フェイルオーバーしたレプリケーションを再同期します

再同期処理によってレプリケーション関係が再確立されます。関係のソースを選択して、ソースクラスタまたはデスティネーションクラスタにデータを保持することができます。この処理は、SnapMirror関係を再確立し、ボリュームのレプリケーションを任意の方向に開始します。

レプリケーションを再確立する前に、新しいデスティネーションクラスタ上のアプリケーションが停止されます。

> 再同期プロセスの間、ライフサイクルの状態は「Establishing」と表示されます。

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. ［アプリケーション］ページで、［データ保護］>［レプリケーション］タブを選択します。
3. [データ保護（Data Protection）]>[レプリケーション（Replication）]タブの[アクション（Actions）]メニューから、[*再同期（Resync *）]を
4. 再同期（Resync）ページで、保持するデータを含むソースまたはデスティネーションのアプリケーションインスタンスを選択します。

   > デスティネーションのデータが上書きされるため、再同期元は慎重に選択してください。

5. 続行するには、* Resync *を選択します。
6. 「resync」と入力して確定します。
7. 「* Yes、resync *」を選択して終了します。

結果

- Replication（レプリケーション）ページに、レプリケーションステータスとしてEstablishing（確立）が表示されます。
- Astra Controlは、新しいデスティネーションクラスタのアプリケーションを停止します。
- SnapMirror resyncを使用して、指定した方向に永続的ボリュームのレプリケーションを再確立します。
- [レプリケーション]ページに、更新された関係が表示されます。

アプリケーションのレプリケーションを反転する

元のソースクラスタへのレプリケートを続行したまま、アプリケーションをデスティネーションクラスタに移動する計画的処理です。Astra Controlは、ソースクラスタ上のアプリケーションを停止し、デスティネーションにデータをレプリケートしてから、デスティネーションクラスタにアプリケーションをフェイルオーバーします。

この状況では、ソースとデスティネーションを交換しようとしています。元のソースクラスタが新しいデスティネーションクラスタになり、元のデスティネーションクラスタが新しいソースクラスタになります。

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. ［アプリケーション］ページで、［データ保護］>［レプリケーション］タブを選択します。

3. [データ保護（Data Protection）]>[レプリケーション（Replication）]タブの[アクション（Actions）]メニューから、[レプリケーションを反転（Reverse replication）]を選択します
4. リバース・レプリケーションのページで情報を確認し、「リバース・レプリケーション」を選択して続行します。

結果

リバースレプリケーションの結果、次の処理が実行されます。

- Snapshotは、元のソースアプリケーションのKubernetesリソースから作成されます。
- 元のソースアプリケーションのポッドは、アプリケーションのKubernetesリソースを削除することで正常に停止されます（PVCとPVはそのまま維持されます）。
- ポッドがシャットダウンされると、アプリケーションのボリュームのSnapshotが作成されてレプリケートされます。
- SnapMirror関係が解除され、デスティネーションボリュームが読み取り/書き込み可能な状態になります。
- アプリケーションのKubernetesリソースは、元のソースアプリケーションのシャットダウン後にレプリケートされたボリュームデータを使用して、シャットダウン前のSnapshotからリストアされます。
- 逆方向にレプリケーションが再確立されます。

アプリケーションを元のソースクラスタにフェイルバックします

Astra Controlを使用すると、次の一連の操作を使用して、「フェイルオーバー」操作後に「フェイルバック」を実行できます。このワークフローでは、元のレプリケーション方向を復元するために、レプリケーションの方向を反転する前に、Astra Controlによってアプリケーションの変更が元のソースクラスタにレプリケート（再同期）されます。

このプロセスは、デスティネーションへのフェイルオーバーを完了した関係から開始され、次の手順を実行します。

- フェイルオーバー状態から開始します。
- 関係を再同期します。
- レプリケーションを反転する。

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. ［アプリケーション］ページで、［データ保護］>［レプリケーション］タブを選択します。
3. [データ保護（Data Protection）]>[レプリケーション（Replication）]タブの[アクション（Actions）]メニューから、[*再同期（Resync *）]を
4. フェイルバック処理の場合は、フェイルオーバーしたオーバーアプリケーションを再同期処理のソースとして選択します（フェイルオーバー後に書き込まれたデータは保持します）。
5. 「resync」と入力して確定します。
6. 「* Yes、resync *」を選択して終了します。
7. 再同期が完了したら、[データ保護（Data Protection）]>[レプリケーション（Replication）]タブの[アクション（Actions）]メニューから[*レプリケーションを反転（Reverse replication）]を選択します。
8. リバース・レプリケーションのページで、情報を確認し、*リバース・レプリケーション*を選択します。

結果

このコマンドは、「resync」処理と「reverse relationship」処理の結果を組み合わせて、レプリケーションが再開された元のソースクラスタ上のアプリケーションを元のデスティネーションクラスタにオンラインにします。

アプリケーションレプリケーション関係を削除します

関係を削除すると、2つの異なるアプリケーション間に関係がなくなります。

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. ［アプリケーション］ページで、［データ保護］>［レプリケーション］タブを選択します。
3. [データ保護]>[レプリケーション]タブの[アプリケーション保護]ボックスまたは関係図で、[レプリケーション関係の削除*]を選択します。

結果

レプリケーション関係を削除すると、次の処理が実行されます。

- 関係が確立されていても、アプリケーションがデスティネーションクラスタでオンラインになっていない（フェイルオーバーした）場合、Astra Controlは、初期化中に作成されたPVCを保持し、「空」の管理対象アプリケーションをデスティネーションクラスタに残します。また、作成されたバックアップを保持するためにデスティネーションアプリケーションを保持します。
- アプリケーションがデスティネーションクラスタでオンラインになった（フェイルオーバーした）場合、Astra ControlはPVCと宛先アプリケーションを保持します。ソースとデスティネーションのアプリケーションは、独立したアプリケーションとして扱われるようになりました。バックアップスケジュールは、両方のアプリケーションで維持されますが、相互に関連付けられていません。

レプリケーション関係のヘルスステータスと関係のライフサイクル状態

Astra Controlには、関係の健全性と、レプリケーション関係のライフサイクルの状態が表示されます。

レプリケーション関係のヘルスステータス

レプリケーション関係の健常性は、次のステータスで示されます。

- 正常：関係が確立されているか確立されており、最新のSnapshotが転送されました。
- 警告：関係がフェイルオーバーされているかフェイルオーバーされています（そのためソースアプリは保護されなくなりました）。
- * 重要 *
  - 関係が確立されているか、フェイルオーバーされていて、前回の調整が失敗しました。
  - 関係が確立され、新しいPVCの追加を最後に調整しようとしても失敗しています。
  - 関係は確立されていますが（Snapshotが正常にレプリケートされ、フェイルオーバーが可能になります）、最新のSnapshotはレプリケートに失敗したか、レプリケートに失敗しています。

レプリケーションのライフサイクル状態

次の状態は、レプリケーションのライフサイクルの各段階を表しています。

- * Establishing *：新しいレプリケーション関係を作成中です。Astra Controlは、必要に応じてネームスペースを作成し、デスティネーションクラスタの新しいボリュームにPersistent Volumeクレーム（PVC；永続ボリューム要求）を作成し、SnapMirror関係を作成します。このステータスは、レプリケーションが再同期中であること、またはレプリケーションを反転中であることを示している可能性もあり
- * established *：レプリケーション関係が存在します。Astra Controlは、PVCが使用可能かどうかを定期的にチェックし、レプリケーション関係をチェックし、アプリケーションのSnapshotを定期的に作成し、アプリケーション内の新しいソースPVCを特定します。その場合は、レプリケーションに含めるリソースがAstra Controlによって作成されます。
- フェイルオーバー：SnapMirror関係が解除され、アプリケーションのKubernetesリソースが最後にレプリケートされたアプリケーションのSnapshotからリストアされます。
- *フェイルオーバーした場合：Astra Controlは、ソースクラスタからのレプリケーションを停止し、デスティネーションでレプリケートされた最新の（成功した）アプリケーションSnapshotを使用して、Kubernetesリソースをリストアします。
- * resyncing *：Astra Controlは、SnapMirror resyncを使用して、再同期元の新しいデータを再同期先に再同期します。この処理では、同期の方向に基づいて、デスティネーション上の一部のデータが上書きされる可能性があります。Astra Controlは、デスティネーションネームスペースで実行されているアプリケーションを停止し、Kubernetesアプリケーションを削除します。再同期処理の実行中、ステータスは「Establishing」と表示されます。
- リバース：は、元のソースクラスタへのレプリケーションを続行しながらアプリケーションをデスティネーションクラスタに移動する予定の処理です。Astra Controlは、ソースクラスタ上のアプリケーションを停止し、デスティネーションにデータをレプリケートしてから、デスティネーションクラスタにアプリケーションをフェイルオーバーします。リバースレプリケーションの間、ステータスは「Establishing」と表示されます。
- 削除中：
  - レプリケーション関係が確立されたものの、まだフェイルオーバーされていない場合は、レプリケーション中に作成されたPVCがAstra Controlによって削除され、デスティネーションの管理対象アプリケーションが削除されます。
  - レプリケーションがすでにフェイルオーバーされている場合、Astra ControlはPVCと宛先アプリケーションを保持します。

アプリケーションのクローン作成と移行

既存のアプリケーションをクローニングして、同じ Kubernetes クラスタまたは別のクラスタに重複するアプリケーションを作成する。Astra Control Center は、アプリケーションのクローンを作成するときに、アプリケーション構成と永続的ストレージのクローンを作成します。

Kubernetes クラスタ間でアプリケーションとストレージを移動する必要がある場合は、クローニングが役立ちます。たとえば、 CI / CD パイプラインや Kubernetes ネームスペース間でワークロードを移動できます。Astra の UI またはを使用できます "Astra Control API" アプリケーションのクローン作成と移行を実行します。

必要なもの

アプリケーションを別のクラスタにクローニングするには、デフォルトのバケットが必要です。最初のバケットを追加した時点でデフォルトのバケットになります。

このタスクについて

- StorageClass が明示的に設定されたアプリケーションを展開し、そのアプリケーションをクローニングする必要がある場合、ターゲットクラスタには元の StorageClass が指定されている必要があります。明示的に StorageClass を設定したアプリケーションを、同じストレージクラスを使用しないクラスタにクローニングすると、失敗します。
- オペレータが配置した Jenkins CI のインスタンスをクローニングする場合は、永続データを手動でリストアする必要があります。これは、アプリケーションの展開モデルの制限事項です。
- Astra Control Center の S3 バケットは、使用可能容量を報告しません。Astra Control Center で管理されているアプリケーションのバックアップまたはクローニングを行う前に、 ONTAP または StorageGRID 管理システムでバケット情報を確認します。
- アプリケーションのバックアップやリストア時に、バケット ID を必要に応じて指定することができます。ただし、アプリケーションのクローニング処理では、定義済みのデフォルトバケットが常に使用されます。クローンのバケットを変更するオプションはありません。どのバケットを使用するかを制御する必要がある場合は、どちらかを選択できます "バケットのデフォルト設定を変更する" または、を実行します "バックアップ" その後にを押します "リストア" 個別。
- 名前空間の名前 / ID または名前空間ラベルによる名前空間の制約を持つメンバーユーザーは、同じクラスタ上の新しい名前空間、または組織のアカウント内の他のクラスタに対して、アプリケーションのクローンまたはリストアを実行できます。ただし、同じユーザが、クローニングまたはリストアされたアプリケーションに新しいネームスペースからアクセスすることはできません。クローンまたはリストア処理によって新しいネームスペースが作成されると、アカウントの管理者 / 所有者はメンバーユーザアカウントを編集し、該当するユーザに新しいネームスペースへのアクセスを許可するロールの制限を更新できます。

OpenShift に関する考慮事項

- クラスタ間でアプリケーションをクローニングする場合、ソースクラスタとデスティネーションクラスタは OpenShift の同じディストリビューションである必要があります。たとえば、 OpenShift 4.7 クラスタからアプリケーションをクローニングする場合は、 OpenShift 4.7 でもあるデスティネーションクラスタを使用します。
- OpenShift クラスタでアプリケーションをホストするプロジェクトを作成すると、プロジェクト（または Kubernetes ネームスペース）に SecurityContext UID が割り当てられます。Astra Control Center でアプリケーションを保護し、 OpenShift でそのアプリケーションを別のクラスタまたはプロジェクトに移動できるようにするには、アプリケーションを任意の UID として実行できるようにポリシーを追加する必要があります。たとえば、次の OpenShift CLI コマンドは、 WordPress アプリケーションに適切なポリシーを付与します。

```
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
```

手順

1. 「 * アプリケーション * 」を選択します。
2. 次のいずれかを実行します。
   - 目的のアプリケーションの [* アクション * （ * Actions * ）] 列で [オプション（ Options ）] メニューを選択します。
   - 目的のアプリケーションの名前を選択し、ページの右上にあるステータスドロップダウンリストを選択します。
3. 「 * Clone * 」を選択します。
4. * クローンの詳細 * ：クローンの詳細を指定します。

- 名前を入力します。
- クローンのネームスペースを入力します。
- クローンのデスティネーションクラスタを選択してください。
- 既存の Snapshot からクローンを作成するかバックアップを作成するかを選択します。このオプションを選択しない場合、 Astra Control Center はアプリケーションの現在の状態からクローンを作成します。

5. * 出典 * ：既存のスナップショットまたはバックアップからクローンを作成する場合は、使用するスナップショットまたはバックアップを選択します。
6. [* Review （レビュー）] を選択します
7. * Clone Summary * ：クローンの詳細を確認し、 * Clone * を選択します。

結果

Astra Control Center では、入力した情報に基づいてアプリケーションのクローンを作成します。新しいアプリケーションのクローンがに追加されると、クローニング処理が成功します `Available` 「アプリケーション」ページで説明します。

> データ保護処理（クローン、バックアップ、リストア）が完了して永続ボリュームのサイズを変更したあと、新しいボリュームのサイズが UI に表示されるまでに最大 20 分かかります。データ保護処理にかかる時間は数分です。また、ストレージバックエンドの管理ソフトウェアを使用してボリュームサイズの変更を確認できます。

アプリケーション実行フックを管理します

実行フックは、管理対象アプリケーションのデータ保護操作と組み合わせて実行するように構成できるカスタムアクションです。たとえば、データベースアプリケーションがある場合、実行フックを使用して、スナップショットの前にすべてのデータベーストランザクションを一時停止し、スナップショットの完了後にトランザクションを再開できます。これにより、アプリケーションと整合性のある Snapshot を作成できます。

実行フックのタイプ

Astra Controlは、実行可能なタイミングに基づいて、次の種類の実行フックをサポートします。

- Snapshot前
- Snapshot後
- バックアップ前
- バックアップ後
- リストア後のPOSTコマンドです

カスタム実行フックに関する重要な注意事項

アプリケーションの実行フックを計画するときは、次の点を考慮してください。

- 実行フックは、スクリプトを使用してアクションを実行する必要があります。多くの実行フックは、同じスクリプトを参照できます。

- Astra Controlでは、実行フックが実行可能なシェルスクリプトの形式で記述されるようにするスクリプトが必要です。
- スクリプトのサイズは96KBに制限されています。
- Astra Controlは、実行フックの設定と一致条件を使用して、スナップショット、バックアップ、または復元操作に適用できるフックを決定します。
- 実行フックの障害はすべて'ソフトな障害ですフックが失敗しても'他のフックとデータ保護操作は試行されますただし、フックが失敗すると、 * アクティビティ * ページイベントログに警告イベントが記録されます。
- 実行フックを作成、編集、または削除するには、 Owner 、 Admin 、または Member 権限を持つユーザーである必要があります。
- 実行フックの実行に 25 分以上かかる場合 ' フックは失敗し ' 戻りコードが N/A のイベント・ログ・エントリが作成されます該当する Snapshot はタイムアウトして失敗とマークされ、タイムアウトを通知するイベントログエントリが生成されます。
- アドホックデータ保護操作の場合、すべてのフックイベントが生成され、[**Activity**]ページのイベントログに保存されます。ただし、スケジュールされたデータ保護処理については、フック障害イベントだけがイベントログに記録されます（スケジュールされたデータ保護処理自体によって生成されたイベントは記録されたままです）。

> 実行フックは、実行中のアプリケーションの機能を低下させるか、完全に無効にすることが多いため、カスタム実行フックの実行時間を最小限に抑えるようにしてください。実行フックが関連付けられている状態でバックアップまたはスナップショット操作を開始した後'キャンセルした場合でも'バックアップまたはスナップショット操作がすでに開始されていればフックは実行できますつまり、バックアップ後の実行フックでは、バックアップが完了したとは判断できません。

実行順序

データ保護操作を実行すると、実行フックイベントが次の順序で実行されます。

1. 適用可能なカスタムプリオペレーション実行フックは、適切なコンテナで実行されます。カスタムのプリオペレーションフックは必要なだけ作成して実行できますが、操作前のこれらのフックの実行順序は保証も構成もされていません。
2. データ保護処理が実行されます。
3. 適用可能なカスタムポストオペレーション実行フックは、適切なコンテナで実行されます。必要な数のカスタムポストオペレーションフックを作成して実行できますが、操作後のこれらのフックの実行順序は保証されず、設定もできません。

同じ種類の実行フック（スナップショット前など）を複数作成する場合、これらのフックの実行順序は保証されません。ただし、異なるタイプのフックの実行順序は保証されています。たとえば、5つの異なるタイプのフックをすべて持つ構成の実行順序は、次のようになります。

1. 予備フックが実行されます
2. スナップショット前フックが実行されます
3. スナップショット後フックが実行されます
4. バックアップ後のフックが実行されます
5. 復元後のフックが実行されます

シナリオ番号2のこの設定の例は、の表を参照してください [フックが実行されるかどうかを確認します]。

> 本番環境で実行スクリプトを有効にする前に、必ず実行フックスクリプトをテストしてください。'kubectl exec' コマンドを使用すると、スクリプトを簡単にテストできます。本番環境で実行フックを有効にしたら、作成されたSnapshotとバックアップをテストして整合性があることを確認します。これを行うには、アプリケーションを一時的なネームスペースにクローニングし、スナップショットまたはバックアップをリストアしてから、アプリケーションをテストします。

フックが実行されるかどうかを確認します

次の表を使用して、アプリケーションでカスタム実行フックが実行されるかどうかを判断します。

アプリケーションの高レベルの処理は、すべてスナップショット、バックアップ、またはリストアの基本的な処理のいずれかを実行することで構成されることに注意してください。シナリオによっては、クローニング処理はこれらの処理のさまざまな組み合わせで構成されるため、クローン処理を実行する実行フックはさまざまです。

In Placeリストア処理では既存のSnapshotまたはバックアップが必要になるため、これらの処理ではSnapshotまたはバックアップフックは実行されません。

> 開始してスナップショットを含むバックアップをキャンセルし'実行フックが関連付けられている場合は'一部のフックが実行され'ほかのフックが実行されないことがありますつまり、バックアップ後の実行フックでは、バックアップが完了したとは判断できません。キャンセルしたバックアップに関連する実行フックがある場合は、次の点に注意してください。
>
> - バックアップ前およびバックアップ後のフックは常に実行されます。
> - バックアップに新しいスナップショットが含まれており'スナップショットが開始されている場合は'スナップショット前フックとスナップショット後フックが実行されます
> - スナップショットの開始前にバックアップがキャンセルされた場合は'スナップショット前フックとスナップショット後フックは実行されません

シナリオ（**Scenario**）	操作	既存の**Snapshot**	既存のバックアップ	ネームスペース	クラスタ	スナップショットフックが実行されます	バックアップフックが実行されます	フックを元に戻します
1.	クローン	N	N	新規	同じ	Y	N	Y
2.	クローン	N	N	新規	違う	Y	Y	Y
3.	クローンまたはリストア	Y	N	新規	同じ	N	N	Y
4.	クローンまたはリストア	N	Y	新規	同じ	N	N	Y
5.	クローンまたはリストア	Y	N	新規	違う	N	Y	Y

シナリオ（**Scenario**）	操作	既存の**Snapshot**	既存のバックアップ	ネームスペース	クラスタ	スナップショットフックが実行されます	バックアップフックが実行されます	フックを元に戻します
6.	クローンまたはリストア	N	Y	新規	違う	N	N	Y
7.	リストア	Y	N	既存	同じ	N	N	Y
8.	リストア	N	Y	既存	同じ	N	N	Y
9.	スナップショット	該当なし	該当なし	該当なし	該当なし	Y	該当なし	該当なし
10.	バックアップ	N	該当なし	該当なし	該当なし	Y	Y	該当なし
11.	バックアップ	Y	該当なし	該当なし	該当なし	N	Y	該当なし

既存の実行フックを表示します

アプリケーションの既存のカスタム実行フックを表示できます。

手順

1. 「 * アプリケーション」に移動し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。

   有効または無効になっているすべての実行フックを結果リストに表示できます。フックのステータス、ソース、および実行時刻（プリ/ポストオペレーション）を表示できます。実行フックに関連するイベントログを表示するには、左側のナビゲーション領域の * アクティビティ * ページに移動します。

既存のスクリプトを表示します

アップロードされた既存のスクリプトを表示できます。このページでは、使用中のスクリプトと、使用中のフックを確認することもできます。

手順

1. 「アカウント」に移動します。
2. [スクリプト]タブを選択します。

   このページには、アップロードされた既存のスクリプトのリストが表示されます。[使用者*]列には、各スクリプトを使用している実行フックが表示されます。

スクリプトを追加します

実行フックが参照できるスクリプトを1つ以上追加できます。多くの実行フックは、同じスクリプトを参照できます。これにより、1つのスクリプトのみを変更することで、多数の実行フックを更新できます。

手順

1. 「アカウント」に移動します。
2. [スクリプト]タブを選択します。
3. 「 * 追加」を選択します。
4. 次のいずれかを実行します。
   - カスタムスクリプトをアップロードする。
     i. [ファイルのアップロード（ Upload file ）] オプションを選択します。
     ii. ファイルを参照してアップロードします。
     iii. スクリプトに一意の名前を付けます。
     iv. （オプション）他の管理者がスクリプトについて知っておく必要があるメモを入力します。
     v. 「スクリプトを保存」を選択します。
   - クリップボードからカスタムスクリプトを貼り付けます。
     i. [貼り付け（Paste）]または[タイプ（* type）]オプションを選択する
     ii. テキストフィールドを選択し、スクリプトテキストをフィールドに貼り付けます。
     iii. スクリプトに一意の名前を付けます。
     iv. （オプション）他の管理者がスクリプトについて知っておく必要があるメモを入力します。
5. 「スクリプトを保存」を選択します。

結果

新しいスクリプトが、[スクリプト]タブのリストに表示されます。

スクリプトを削除します

不要になって実行フックで使用されなくなったスクリプトは、システムから削除できます。

手順

1. 「アカウント」に移動します。
2. [スクリプト]タブを選択します。
3. 削除するスクリプトを選択し、「アクション」列のメニューを選択します。
4. 「 * 削除」を選択します。

> スクリプトが1つまたは複数の実行フックに関連付けられている場合、*Delete*アクションは使用できません。スクリプトを削除するには、まず関連する実行フックを編集し、別のスクリプトに関連付けます。

カスタム実行フックを作成します

アプリケーションのカスタム実行フックを作成できます。を参照してください "実行フックの例" フックの例を参照してください。実行フックを作成するには、 Owner 、 Admin 、または Member のいずれかの権限が必要です。

> 実行フックとして使用するカスタムシェルスクリプトを作成する場合は、特定のコマンドを実行するか、実行可能ファイルへの完全パスを指定する場合を除き、ファイルの先頭に適切なシェルを指定するようにしてください。

手順

1. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。
3. 「 * 追加」を選択します。
4. フックの詳細*（* Hook Details ）領域で、*操作（* Operation *）ドロップダウンメニューから操作タイプを選択して、フックを実行するタイミングを決定します。
5. フックの一意の名前を入力します。
6. （オプション）実行中にフックに渡す引数を入力し、各引数を入力した後で Enter キーを押して、それぞれを記録します。
7. [* Container Images * （コンテナイメージ * ）] 領域で、アプリケーションに含まれるすべてのコンテナイメージに対してフックを実行する必要がある場合は、 [* Apply to all container images * （すべてのコンテナイメージに適用 * ）] チェックボックスを有効にします。代わりに、フックが 1 つ以上の指定されたコンテナイメージに対してのみ機能する場合は、 * Container image names to match * フィールドにコンテナイメージ名を入力します。
8. [* スクリプト * （ * Script * ）] 領域で、次のいずれかを実行します。
   - 新しいスクリプトを追加します。
     i. 「 * 追加」を選択します。
     ii. 次のいずれかを実行します。
       - カスタムスクリプトをアップロードする。
         I. [ファイルのアップロード（ Upload file ）] オプションを選択します。
         II. ファイルを参照してアップロードします。
         III. スクリプトに一意の名前を付けます。
         IV. （オプション）他の管理者がスクリプトについて知っておく必要があるメモを入力します。
         V. 「スクリプトを保存」を選択します。
       - クリップボードからカスタムスクリプトを貼り付けます。
         I. [貼り付け（Paste）]または[タイプ（* type）]オプションを選択する
         II. テキストフィールドを選択し、スクリプトテキストをフィールドに貼り付けます。
         III. スクリプトに一意の名前を付けます。
         IV. （オプション）他の管理者がスクリプトについて知っておく必要があるメモを入力します。
   - リストから既存のスクリプトを選択します。

     このスクリプトを使用するように実行フックに指示します。
9. [* フックを追加 *] を選択します。

実行フックの状態を確認します

スナップショット、バックアップ、または復元操作の実行が終了したら、操作の一部として実行された実行フックの状態を確認できます。このステータス情報を使用して、実行フックを保持するか、変更するか、削除するかを決定できます。

手順

1. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
2. [データ保護]タブを選択します。
3. 実行中のSnapshotを表示するには「* Snapshots」を選択し、実行中のバックアップを表示するには「* Backups」を選択します。

   フック状態*は、操作完了後の実行フックランのステータスを示します。状態にカーソルを合わせると、詳細を確認できます。たとえば、スナップショット中に実行フック障害が発生した場合、そのスナップショットのフック状態にカーソルを合わせると、失敗した実行フックのリストが表示されます。各失敗の理由を確認するには、左側のナビゲーション領域の*アクティビティ*ページを確認します。

スクリプトの使用状況を表示します

どの実行フックがAstra Control Web UIの特定のスクリプトを使用しているかを確認できます。

手順

1. 「 * アカウント * 」を選択します。
2. [スクリプト]タブを選択します。

   スクリプトのリストにある* Used by *列には、リスト内の各スクリプトを使用しているフックの詳細が表示されます。

3. 目的のスクリプトの[使用者*]列の情報を選択します。

   より詳細なリストが表示され、スクリプトを使用しているフックの名前と、それらが実行されるように構成されている操作のタイプが示されます。

実行フックを無効にします

アプリケーションのスナップショットの前または後に実行を一時的に禁止する場合は、実行フックを無効にできます。実行フックを無効にするには、 Owner 、 Admin 、または Member のいずれかの権限が必要です。

手順

1. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。
3. 無効にするフックの * アクション * 列のオプションメニューを選択します。
4. [**Disable**] を選択します。

実行フックを削除します

不要になった実行フックは完全に削除できます。実行フックを削除するには、 Owner 、 Admin 、または Member のいずれかの権限が必要です。

手順

1. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。
3. 削除するフックの * アクション * 列のオプションメニューを選択します。
4. 「 * 削除」を選択します。

実行フックの例

次の例を使用して、実行フックの構造を確認してください。これらのフックは、テンプレートまたはテストスクリプトとして使用できます。

シンプルな成功例

次に、成功し、標準出力および標準エラーにメッセージを書き込む単純フックの例を示します。

```
#!/bin/sh

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



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



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

#
```

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


#
# main
#

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

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

シンプルな成功の例（ **bash** バージョン）

次に、 bash 用に書かれた標準出力と標準エラーにメッセージを書き込む単純なフックの例を示します。

```
#!/bin/bash

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


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


#
# Writes the given information message to standard output
#
```

```
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

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



#
# main
#

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

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

単純な成功例（**zsh** バージョン）

これは、成功した単純なフックの例であり、標準出力と標準エラーに Z シェル用に記述されたメッセージを書き込みます。

```
#!/bin/zsh

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



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

```
#
msg() {
    echo "$*"
}


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

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

#
# main
#

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

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

引数を指定した成功の例

次の例は、フックで args を使用する方法を示しています。

```
#!/bin/sh

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

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


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

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


#
# main
#

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


# collect args
arg1=$1
arg2=$2

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

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

スナップショット前 / スナップショット後のフックの例

次の例は、 Snapshot 前フックと Snapshot 後フックの両方に同じスクリプトを使用する方法を示しています。

```
#!/bin/sh

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

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


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


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

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


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

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


#
# main
#

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

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

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

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

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

exit ${rc}
```

失敗の例

次の例は、フックで障害を処理する方法を示しています。

```
#!/bin/sh

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



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



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

```
#
info() {
    msg "INFO: $*"
}

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


#
# main
#

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

argexitcode=$1

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

# exit with specified exit code
exit ${argexitcode}
```

詳細なエラーの例

次の例では ' フックの失敗をより詳細なロギングで処理する方法を示します

```
#!/bin/sh

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


#
# Writes the given message to standard output
#
```

```
# $* - The message to write
#
msg() {
    echo "$*"
}


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

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


#
# main
#

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


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

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

error "exiting with error code 8"
exit 8
```

終了コード例を使用した失敗

次の例は、終了コードを使用したフックの失敗を示しています。

```
#!/bin/sh

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



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



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

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



#
# main
#
```

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

argexitcode=$1

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

# exit with specified exit code
exit ${argexitcode}
```

失敗後の成功の例

次の例では、最初の実行時にフックが失敗していますが、 2 回目の実行後に成功しています。

```
#!/bin/sh

# failure_then_success_sample.sh
#
# A hook script that fails on initial run but succeeds on second run for
testing purposes.
#
# Helpful for testing retry logic for post hooks.
#
# args: None
#

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



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

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


#
# main
#

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


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

アプリケーションとクラスタの健常性を監視

アプリケーションとクラスタの健常性の概要を表示します

ダッシュボード * を選択すると、アプリ、クラスター、ストレージバックエンド、それらのヘルスの概要が表示されます。

これらは静的な数値やステータスだけでなく、それぞれからドリルダウンすることもできます。たとえば、アプリが完全に保護されていない場合は、アイコンの上にカーソルを置くと、完全に保護されていないアプリを特定できます。その理由が含まれます。

アプリケーションタイル

「 * アプリケーション * 」タイルは、次の項目を識別するのに役立ちます。

- Astra で現在管理しているアプリケーションの数。

- それらの管理アプリが正常であるかどうか。
- アプリケーションが完全に保護されているかどうか（最新のバックアップがある場合は保護されます）。
- 検出されたものの、まだ管理されていないアプリケーションの数。

  アプリケーションが検出された後で管理または無視するため、この数はゼロになるのが理想的です。さらに、ダッシュボードで検出されたアプリケーションの数を監視して、開発者がクラスタに新しいアプリケーションを追加するタイミングを特定します。

クラスタタイル

クラスタタイルには、 Astra Control Center を使用して管理しているクラスタの健常性に関する同様の詳細が表示され、ドリルダウンしてアプリと同様に詳細を確認できます。

ストレージバックエンドはタイル張りです

「ストレージバックエンド * 」タイルは、ストレージバックエンドの健全性を特定するための情報を提供します。これには次のものが含まれます。

- 管理対象のストレージバックエンドの数
- これらの管理バックエンドが正常であるかどうか
- バックエンドが完全に保護されているかどうか
- 検出されたがまだ管理されていないバックエンドの数。

クラスタの健常性と詳細を表示します

Astra Control Center で管理するクラスタを追加すると、その場所、ワーカーノード、永続ボリューム、ストレージクラスなど、クラスタに関する詳細を表示できます。

手順

1. Astra Control Center UI で、 [* Clusters] を選択します。
2. [* Clusters] ページで、詳細を表示するクラスタを選択します。

> クラスタの構成 `removed` クラスタとネットワークの接続が正常であると表示される（Kubernetes APIを使用してクラスタに外部からアクセスしようとすると成功する）場合は、Astra Controlに指定したkubeconfigが無効になる可能性があります。クラスタでの証明書のローテーションまたは有効期限が原因の可能性があります。この問題を修正するには、を使用して、 Astra Control のクラスタに関連付けられたクレデンシャルを更新します "Astra Control API の略"。

3. [Overview （概要）] 、 [* Storage （ * ストレージ）] 、 [* Activity * （アクティビティ * ）] タブの情報を表示して、必要な情報を検索します。
   - * 概要 * ：状態を含むワーカーノードの詳細。
   - * ストレージ * ：ストレージクラスと状態を含む、コンピューティングに関連付けられた永続的ボリューム。
   - * アクティビティ * ：クラスタに関連するアクティビティを表示します。

Astra Control Center * Dashboard * から始まるクラスタ情報を表示することもできます。[* クラスタ *] タブの [* リソースサマリ *] で、管理対象クラスタを選択して [* クラスタ *] ページに移動できます。[* Clusters] ページが表示されたら、上記の手順を実行します。

アプリの状態と詳細を表示します

アプリケーションの管理を開始すると、アプリケーションのステータス（正常かどうか）、保護ステータス（障害発生時に完全に保護されているかどうか）、ポッド、永続的ストレージなどを識別できる詳細が Astra から提供されます。

手順

1. Astra Control Center UI で、 * アプリケーション * を選択し、アプリの名前を選択します。
2. お探しの情報を検索してください。

アプリステータス

Kubernetes でのアプリケーションの状態を反映するステータスを提供します。たとえば、ポッドと永続ボリュームはオンラインか？アプリケーションが正常な状態でない場合は、 Kubernetes のログでクラスタの問題を調べてトラブルシューティングする必要があります。Astra は、壊れたアプリケーションの修正に役立つ情報を提供していません。

アプリ保護ステータス

アプリが適切に保護されているかどうかのステータスを表示します。

- * 完全に保護されている * ：アプリにはアクティブなバックアップスケジュールがあり、 1 週間も経過していない正常なバックアップがあります
- * 部分的に保護 * ：アプリケーションには、アクティブなバックアップスケジュール、アクティブなスナップショットスケジュール、または正常なバックアップまたはスナップショットがあります
- * 保護されていない * ：完全に保護されていない、または部分的に保護されていないアプリ

  _ 最新のバックアップがあるまで、完全に保護することはできません _ 。これは、永続ボリュームから離れたオブジェクトストアにバックアップが格納されるために重要です。障害や事故によってクラスタと永続的ストレージが消去された場合は、バックアップをリカバリする必要があります。スナップショットを使用してリカバリすることはできません。

概要

アプリケーションに関連付けられているポッドの状態に関する情報。

データ保護

データ保護ポリシーを設定し、既存の Snapshot とバックアップを表示できます。

ストレージ

アプリケーションレベルの永続ボリュームが表示されます。永続ボリュームの状態は、 Kubernetes クラスタから見たものです。

リソース

バックアップおよび管理対象のリソースを確認できます。

アクティビティ

アプリケーションに関連するアクティビティを表示します。

> Astra Control Center * Dashboard * から始まるアプリ情報を表示することもできます。[* アプリケーション *] タブの [リソースの概要 *] で、管理アプリを選択して [* アプリケーション *] ページに移動できます。[**Applications**] ページが表示されたら、上記の手順に従います。

アカウントを管理します

ユーザを管理します

Astra Control UI を使用して、 Astra Control Center インストールのユーザーを招待、追加、削除、および編集できます。Astra Control UI またはを使用できます "Astra Control API" ユーザを管理するには、を実行

LDAPを使用して、選択したユーザの認証を実行することもできます。

LDAP を使用する

LDAPは、分散ディレクトリ情報にアクセスするための業界標準プロトコルであり、エンタープライズ認証に広く使用されています。Astra Control CenterをLDAPサーバーに接続して、選択したAstraユーザーの認証を実行できます。大まかには、AstraとLDAPを統合し、AstraのユーザとグループをLDAPの定義に対応して定義します。を参照してください "LDAP 認証" を参照してください。

ユーザーを招待します

アカウント所有者と管理者は、 Astra Control Center に新しいユーザを招待できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [**Users**] タブを選択します。
3. [* ユーザーの招待 *] を選択します。
4. ユーザの名前と E メールアドレスを入力します。
5. 適切なシステム権限を持つユーザロールを選択します。

   各ロールには次の権限があります。

   - * Viewer * はリソースを表示できます。
   - メンバー * には、ビューア・ロールの権限があり、アプリとクラスタの管理、アプリの管理解除、スナップショットとバックアップの削除ができます。
   - **Admin** にはメンバーの役割権限があり、 Owner 以外の他のユーザーを追加および削除できます。
   - * Owner * には Admin ロールの権限があり、任意のユーザーアカウントを追加および削除できます。
6. メンバーロールまたはビューアロールを持つユーザーに制約を追加するには、 * 制約へのロールの制限 * チェックボックスをオンにします。

   制約の追加の詳細については、を参照してください "ロールの管理"。

7. [* ユーザーを招待する *] を選択します。

   ユーザーは、 Astra Control Center に招待されたことを通知する電子メールを受信します。このメールには一時的なパスワードが含まれています。このパスワードは初回ログイン時に変更する必要があります。

ユーザを追加します

アカウント所有者と管理者は、 Astra Control Center のインストールにさらにユーザーを追加できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [**Users**] タブを選択します。
3. [ユーザーの追加] を選択します。
4. ユーザ名、 E メールアドレス、および一時パスワードを入力します。

   ユーザは初回ログイン時にパスワードを変更する必要があります。

5. 適切なシステム権限を持つユーザロールを選択します。

   各ロールには次の権限があります。

   - * Viewer * はリソースを表示できます。
   - メンバー * には、ビューア・ロールの権限があり、アプリとクラスタの管理、アプリの管理解除、スナップショットとバックアップの削除ができます。
   - **Admin** にはメンバーの役割権限があり、 Owner 以外の他のユーザーを追加および削除できます。
   - * Owner * には Admin ロールの権限があり、任意のユーザーアカウントを追加および削除できます。
6. メンバーロールまたはビューアロールを持つユーザーに制約を追加するには、 * 制約へのロールの制限 * チェックボックスをオンにします。

   制約の追加の詳細については、を参照してください "ロールの管理"。

7. 「 * 追加」を選択します。

パスワードを管理します

Astra Control Center では、ユーザーアカウントのパスワードを管理できます。

パスワードを変更します

ユーザアカウントのパスワードはいつでも変更できます。

手順

1. 画面の右上にあるユーザアイコンを選択します。
2. * プロファイル * を選択します。
3. [* アクション * （ * Actions * ）] 列の [オプション（ Options ）] メニューから、 [* パスワードの変更 * （ * Change Password ）] を選択します

4. パスワードの要件に準拠するパスワードを入力します。
5. 確認のためパスワードをもう一度入力します。
6. 「 * パスワードの変更 * 」を選択します。

別のユーザのパスワードをリセットします

アカウントに Admin ロールまたは Owner ロールの権限がある場合は、自分だけでなく他のユーザアカウントのパスワードもリセットできます。パスワードをリセットする場合は、ログイン時にユーザが変更しなければならない一時パスワードを割り当てます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [* アクション * （ * Actions * ）] ドロップダウンリストを選択します。
3. 「 * パスワードのリセット * 」を選択します。
4. パスワードの要件に適合する一時パスワードを入力します。
5. 確認のためパスワードをもう一度入力します。

> 次回ユーザがログインするときに、パスワードの変更を求めるプロンプトが表示されます。

6. 「 * パスワードのリセット * 」を選択します。

ユーザのロールを変更します

Owner ロールのユーザはすべてのユーザのロールを変更できますが、 Admin ロールのユーザは Admin 、 Member 、 Viewer のロールを持つユーザのロールを変更できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [* アクション * （ * Actions * ）] ドロップダウンリストを選択します。
3. [役割の編集] を選択します。
4. 新しいロールを選択します。
5. ロールに制約を適用するには、 * 制約へのロールの制限 * チェックボックスを有効にして、リストから制約を選択します。

   拘束がない場合は、拘束を追加できます。詳細については、を参照してください "ロールの管理"。

6. [* 確認 *] を選択します。

結果

Astra Control Center は、選択した新しいロールに基づいてユーザーの権限を更新します。

ユーザを削除します

所有者ロールまたは管理者ロールを持つユーザは、いつでもそのアカウントから他のユーザを削除できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [* ユーザー *] タブで、削除する各ユーザーの行にあるチェックボックスをオンにします。
3. [* アクション * （ * Actions * ）] 列の [オプション（ Options ）] メニューから、 [* ユーザー / 秒を削除（ * Remove user/s * ）] を選択する
4. プロンプトが表示されたら、「 remove 」という単語を入力して削除を確認し、「 * Yes 、 Remove User * 」を選択します。

結果

Astra Control Center は、アカウントからユーザーを削除します。

ロールの管理

ロールを管理するには、ネームスペースの制約を追加し、ユーザロールをその制約に制限します。これにより、組織内のリソースへのアクセスを制御できます。Astra Control UI またはを使用できます "Astra Control API" をクリックしてください。

ロールに名前空間制約を追加します

Admin または Owner ユーザは、ネームスペースの制約を追加できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [**Users**] タブを選択します。
3. [* アクション * （ * Actions * ）] 列で、メンバーまたはビューアーの役割を持つユーザーのメニューボタンを選択します。
4. [役割の編集] を選択します。
5. [ロールを制約に制限する *] チェックボックスをオンにします。

   このチェックボックスは、メンバーロールまたはビューアロールでのみ使用できます。[*Role] ドロップダウン・リストから別のロールを選択できます

6. [* 制約の追加 *] を選択します。

   使用可能な制約の一覧は、ネームスペースまたはネームスペースラベルで確認できます。

7. [制約タイプ * （ Constraint type * ）] ドロップダウンリストで、ネームスペースの構成方法に応じて、 [* Kubernetes namespace] * または [* Kubernetes namespace label*] を選択します。
8. リストから 1 つ以上の名前空間またはラベルを選択して、それらの名前空間にロールを制限する制約を構成します。
9. [* 確認 *] を選択します。

   [役割の編集 *] ページには、この役割に選択した拘束のリストが表示されます。

10. [* 確認 *] を選択します。

    [**Account**] ページでは、 [*Role] 列のメンバまたはビューアの役割の制約を表示できます。

制約を追加せずに役割の制約を有効にし、 * 確認 * を選択すると、役割には完全な制限があると見なされます（役割は、名前空間に割り当てられているリソースへのアクセスを拒否されます）。

ロールから名前空間制約を削除します

管理者または所有者ユーザーは、役割から名前空間の制約を削除できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [**Users**] タブを選択します。
3. [* アクション * （ * Actions * ）] 列で、アクティブな拘束を持つメンバーまたはビューアーの役割を持つユーザーのメニューボタンを選択する。
4. [役割の編集] を選択します。
   - 役割の編集 * （ Edit role * ）ダイアログには、役割のアクティブな拘束が表示されます。
5. 削除する拘束の右側にある * X * を選択します。
6. [* 確認 *] を選択します。

を参照してください。

- "ユーザロールとネームスペース"

通知を表示および管理します

アクションが完了または失敗すると、 Astra から通知が表示されます。たとえば、アプリケーションのバックアップが正常に完了した場合に通知が表示されます。

これらの通知は、インターフェイスの右上から管理できます。

手順

1. 右上の未読通知の数を選択します。
2. 通知を確認し、 [* 既読としてマークする *] または [すべての通知を表示する *] を選択します。

   [すべての通知を表示する *] を選択した場合は、 [通知] ページがロードされます。

3. [* 通知 *] ページで、通知を表示し、既読としてマークする通知を選択し、 [* アクション *] を選択して、 [* 既読としてマークする *] を選択します。

クレデンシャルを追加および削除します

ONTAP S3 、 OpenShift で管理される Kubernetes クラスタ、未管理の Kubernetes クラスタなどのローカルプライベートクラウドプロバイダのクレデンシャルを、お客様のアカウントにいつでも追加、削除できま

す。Astra Control Center は、これらのクレデンシャルを使用して、クラスタ上の Kubernetes クラスタとアプリケーションを検出し、ユーザに代わってリソースをプロビジョニングします。

Astra Control Center のすべてのユーザーが同じ資格情報セットを共有することに注意してください。

クレデンシャルを追加する

クラスターの管理時に、 Astra Control Center に資格情報を追加できます。新しいクラスタを追加してクレデンシャルを追加する手順については、を参照してください "Kubernetes クラスタを追加"。

> 自分で作成する場合は `kubeconfig` ファイルには、* 1つの*コンテキストエレメントのみを定義する必要があります。を参照してください "Kubernetes のドキュメント" を参照してください `kubeconfig` ファイル。

クレデンシャルを削除する

アカウントからのクレデンシャルの削除はいつでも実行できます。クレデンシャルは、のあとに削除してください "関連するすべてのクラスタの管理を解除します"。

> Astra Control Center は、 Astra Control Center の認証情報を使用してバックアップバケットに認証するため、 Astra Control Center に追加する最初の資格情報セットは常に使用されています。これらのクレデンシャルは削除しないことを推奨します。

手順

1. 「 * アカウント * 」を選択します。
2. [*Credentials] タブを選択します。
3. 削除するクレデンシャルの [状態 *] 列で [オプション] メニューを選択します。
4. 「 * 削除」を選択します。
5. 削除を確認するために「削除」と入力し、「はい」、「認証情報を削除」を選択します。

結果

Astra Control Center は、アカウントから資格情報を削除します。

アカウントのアクティビティを監視

Astra Control アカウントのアクティビティの詳細を表示できます。たとえば、新しいユーザを招待したとき、クラスタが追加されたとき、 Snapshot が作成されたときなどです。アカウントアクティビティを CSV ファイルにエクスポートすることもできます。

> KubernetesクラスタをAstra Controlから管理し、Astra ControlをCloud Insights に接続した場合、Astra ControlはCloud Insights にイベントログを送信する。ポッドの導入やPVCの添付ファイルに関する情報などのログ情報が、Astra Control Activityログに記録されます。この情報を使用して、管理しているKubernetesクラスタの問題を特定します。

Astra Control のアカウントアクティビティをすべて表示

1. 「 * Activity * 」を選択します。

2. フィルタを使用してアクティビティのリストを絞り込むか、検索ボックスを使用して探しているものを正確に検索します。
3. アカウントアクティビティを CSV ファイルにダウンロードするには、「 * CSV にエクスポート」を選択します。

特定のアプリケーションのアカウントアクティビティを表示します

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. 「 * Activity * 」を選択します。

クラスタのアカウントアクティビティを表示します

1. 「 * クラスタ」を選択し、クラスタの名前を選択します。
2. 「 * Activity * 」を選択します。

対応が必要なイベントを解決するための操作を実行します

1. 「 * Activity * 」を選択します。
2. 注意が必要なイベントを選択してください。
3. [**Take action**] ドロップダウンオプションを選択します。

   このリストから、実行できる対処方法のほか、問題 に関するドキュメントを参照したり、問題 の解決に役立つサポートを受けたりできます。

既存のライセンスを更新する

評価用ライセンスをフルライセンスに変換したり、既存の評価用ライセンスまたはフルライセンスを新しいライセンスで更新したりできます。フルライセンスがない場合は、ネットアップの営業担当者に連絡して、ライセンスとシリアル番号の全文を入手してください。Astra の UI またはを使用できます "Astra Control API" 既存のライセンスを更新します。

手順

1. にログインします "NetApp Support Site"。
2. Astra Control Center のダウンロードページにアクセスし、シリアル番号を入力して、ネットアップライセンスファイル（ NLF ）をダウンロードする。
3. Astra Control Center UI にログインします。
4. 左側のナビゲーションから、 * アカウント * > * ライセンス * を選択します。
5. [**Account**>*License*] ページで、既存のライセンスのステータスドロップダウンメニューを選択し、[**Replace**] を選択します。
6. ダウンロードしたライセンスファイルを参照します。
7. 「 * 追加」を選択します。

[**Account**>*Licenses*] ページには、ライセンス情報、有効期限、ライセンスシリアル番号、アカウント ID 、および使用されている CPU ユニットが表示されます。

を参照してください。

- "Astra Control Center のライセンス"

リポジトリ接続を管理します

ソフトウェアパッケージのインストールイメージやアーティファクトの参照として使用するリポジトリをAstra Controlに接続できます。ソフトウェアパッケージをインポートすると、Astra Controlは、イメージリポジトリ内のインストールイメージと、アーティファクトリポジトリ内のバイナリおよびその他のアーティファクトを参照します。

必要なもの

- Astra Control Center をインストールした Kubernetes クラスタ
- アクセス可能な稼働中のDockerリポジトリ
- アクセス可能なアーティファクトリポジトリ（Artifactoryなど）が実行されている必要があります

Dockerイメージリポジトリを接続する

Dockerイメージリポジトリを接続して、Astraデータストアなどのパッケージインストールイメージを保持できます。パッケージをインストールすると、Astra Controlはイメージリポジトリからパッケージイメージファイルをインポートします。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [接続（Connections *）]タブを選択します。
3. 「* Docker Image Repository *」セクションで、右上のメニューを選択します。
4. 「 * 接続」を選択します。
5. リポジトリのURLとポートを追加します。
6. リポジトリのクレデンシャルを入力します。
7. 「 * 接続」を選択します。

結果

リポジトリが接続されました。「* Docker Image Repository *」セクションに、リポジトリのステータスが「Connected」になっていることを確認します。

Dockerイメージリポジトリの接続を解除する

不要になったDockerイメージリポジトリへの接続を削除できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [接続（Connections *）]タブを選択します。
3. 「* Docker Image Repository *」セクションで、右上のメニューを選択します。
4. 「切断」を選択します。

5. 「* Yes、disconnect Docker image repository *」を選択します。

結果

リポジトリが切断されました。「* Docker Image Repository *」セクションには、リポジトリのステータスが「Disconnected」になっているはずです。

アーティファクトリポジトリを接続します

アーティファクトリポジトリをソフトウェアパッケージのバイナリなどのホストアーティファクトに接続できます。パッケージをインストールすると、Astra Controlによって、ソフトウェアパッケージのアーティファクトがイメージリポジトリからインポートされます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [接続（Connections *）]タブを選択します。
3. [アーティファクトリポジトリ*]セクションで’右上のメニューを選択します
4. 「 * 接続」を選択します。
5. リポジトリのURLとポートを追加します。
6. 認証が必要な場合は、*認証を使用*チェックボックスを有効にして、リポジトリのクレデンシャルを入力します。
7. 「 * 接続」を選択します。

結果

リポジトリが接続されました。[アーティファクトリポジトリ*]セクションで’リポジトリに[接続済み]ステータスが表示されるはずです

アーティファクトリポジトリの接続を解除します

不要になったアーティファクトリポジトリへの接続を削除できます

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [接続（Connections *）]タブを選択します。
3. [アーティファクトリポジトリ*]セクションで’右上のメニューを選択します
4. 「切断」を選択します。
5. [はい]を選択し’アーティファクト・リポジトリを切断します*

結果

リポジトリが切断されました。[アーティファクトリポジトリ*]セクションで’リポジトリに[接続済み]ステータスが表示されるはずです

詳細については、こちらをご覧ください

- "ソフトウェアパッケージを管理します"

ソフトウェアパッケージを管理します

ネットアップでは、NetApp Support Siteからダウンロード可能なソフトウェアパッケージを使用して、Astra Control Center向けの機能を追加しています。Dockerとアーティファクトのリポジトリを接続したら、パッケージをアップロードしてインポートし、この機能をAstra Control Centerに追加できます。CLIまたはAstra Control CenterのWeb UIを使用して、ソフトウェアパッケージを管理できます。

必要なもの

- Astra Control Center をインストールした Kubernetes クラスタ
- ソフトウェアパッケージイメージを格納するために接続されたDockerイメージリポジトリ。詳細については、を参照してください "リポジトリ接続を管理します"。
- ソフトウェアパッケージのバイナリやアーティファクトを保持するための、接続されたアーティファクトリポジトリ。詳細については、を参照してください "リポジトリ接続を管理します"。
- NetApp Support Siteから提供されるソフトウェアパッケージ

ソフトウェアパッケージのイメージをリポジトリにアップロードします

Astra Control Centerは、接続されたリポジトリ内のパッケージイメージとアーティファクトを参照します。CLIを使用して、リポジトリにイメージとアーティファクトをアップロードできます。

手順

1. NetApp Support Site からソフトウェアパッケージをダウンロードし、を搭載したマシンに保存します `kubectl` ユーティリティがインストールされました
2. 圧縮パッケージファイルを展開し、ディレクトリをAstra Controlバンドルファイルの場所に変更します（例： `acc.manifest.yaml`）。
3. パッケージイメージをDockerリポジトリにプッシュします。次の置換を行います。
   - bundle_fileをAstra Controlバンドルファイルの名前に置き換えます（例： `acc.manifest.yaml`）。
   - my_registryをDockerリポジトリのURLに置き換えます。
   - my_registry_userをユーザー名に置き換えます。
   - my_registry_tokenをレジストリの認証済みトークンに置き換えます。

   ```
   kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY -u
   MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
   ```

4. パッケージにアーティファクトがある場合は'アーティファクトをアーティファクトリポジトリにコピーしますbundle_fileをAstra Controlバンドルファイルの名前に置き換え'network_locationをネットワークロケーションに置き換えて'アーティファクトファイルを次の場所にコピーします

   ```
   kubectl astra packages copy-artifacts -m BUNDLE_FILE -n NETWORK_LOCATION
   ```

ソフトウェアパッケージを追加します

Astra Control Centerバンドルファイルを使用して、ソフトウェアパッケージをインポートできます。これに

より、パッケージがインストールされ、Astra Control Centerで使用できるようになります。

Astra Control Web UIを使用してソフトウェアパッケージを追加

Astra Control Center Web UIを使用して、接続されたリポジトリにアップロードされたソフトウェアパッケージを追加できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [パッケージ]タブを選択します。
3. [*追加（Add *）]ボタンを選択します。
4. ファイル選択ダイアログで、アップロードアイコンを選択します。
5. で、Astra Controlバンドルファイルを選択します `.yaml` 形式を指定して、アップロードします。
6. 「 * 追加」を選択します。

結果

バンドルファイルが有効で、パッケージイメージとアーティファクトが接続されているリポジトリにある場合、パッケージはAstra Control Centerに追加されます。[ステータス*]列のステータスが[使用可能*]に変わったら、パッケージを使用できます。パッケージのステータスにカーソルを合わせると、詳細を確認できます。

> リポジトリ内にパッケージのイメージまたはアーティファクトが1つでも見つからない場合は’そのパッケージのエラーメッセージが表示されます

CLIを使用してソフトウェアパッケージを追加します

CLIを使用して、接続されたリポジトリにアップロードしたソフトウェアパッケージをインポートできます。そのためには、最初にAstra Control CenterのアカウントIDとAPIトークンを記録する必要があります。

手順

1. Webブラウザを使用して、Astra Control Center Web UIにログインします。
2. ダッシュボードの右上にあるユーザアイコンを選択します。
3. [API access*]を選択します。
4. 画面上部のアカウントIDをメモします。
5. [APIトークンの生成]を選択します。
6. 表示されたダイアログで、* APIトークンの生成*を選択します。
7. 生成されたトークンをメモし、*閉じる*を選択します。CLIで、ディレクトリをの場所に変更します `.yaml` 展開されたパッケージの内容に含まれるバンドルファイル。
8. バンドルファイルを使用してパッケージをインポートし、次のように置き換えます。
   - bundle_fileをAstra Controlバンドルファイルの名前に置き換えます。
   - serverをAstra ControlインスタンスのDNS名に置き換えます。
   - account_IDとtokenは、以前に記録したアカウントIDとAPIトークンに置き換えてください。

```
kubectl astra packages import -m BUNDLE_FILE -u SERVER -a ACCOUNT_ID
-k TOKEN
```

結果

バンドルファイルが有効で、パッケージイメージとアーティファクトが接続されているリポジトリにある場合、パッケージはAstra Control Centerに追加されます。

> リポジトリ内にパッケージのイメージまたはアーティファクトが1つでも見つからない場合は’そのパッケージのエラーメッセージが表示されます

ソフトウェアパッケージを削除します

Astra Control Center Web UIを使用して、Astra Control Centerに以前にインポートしたソフトウェアパッケージを削除できます。

手順

1. 「アカウントの管理」ナビゲーション領域で、「 * アカウント * 」を選択します。
2. [パッケージ]タブを選択します。

   このページには、インストールされているパッケージとそのステータスのリストが表示されます。

3. パッケージの*アクション*列で、アクションメニューを開きます。
4. 「 * 削除」を選択します。

結果

パッケージはAstra Control Centerから削除されますが、パッケージのイメージとアーティファクトはリポジトリに残ります。

詳細については、こちらをご覧ください

- "リポジトリ接続を管理します"

バケットを管理する

アプリケーションや永続的ストレージをバックアップする場合や、クラスタ間でアプリケーションをクローニングする場合は、オブジェクトストアバケットプロバイダが不可欠です。Astra Control Center を使用して、オフクラスタのバックアップ先として、アプリケーションのオブジェクトストアプロバイダを追加します。

アプリケーション構成と永続的ストレージを同じクラスタにクローニングする場合、バケットは必要ありません。

次の Amazon Simple Storage Service （ S3 ）バケットプロバイダのいずれかを使用します。

- NetApp ONTAP S3
- NetApp StorageGRID S3 の略
- Microsoft Azure

- 汎用 S3

> Amazon Web Services（AWS）とGoogle Cloud Platform（GCP）では、汎用のS3バケットタイプを使用します。

> Astra Control Center は Amazon S3 を汎用 S3 バケットプロバイダとしてサポートしていますが、 Astra Control Center は Amazon の S3 サポートを要求するすべてのオブジェクトストアベンダーをサポートしているわけではありません。

バケットの状態は次のいずれかになります。

- Pending ：バケットの検出がスケジュールされています。
- Available ：バケットは使用可能です。
- Removed ：バケットには現在アクセスできません。

Astra Control API を使用してバケットを管理する方法については、を参照してください "Astra の自動化と API に関する情報"。

バケットの管理に関連して次のタスクを実行できます。

- "バケットを追加します"
- [バケットを編集する]
- [バケットのクレデンシャルをローテーションするか、削除する]
- [バケットを削除する]

> Astra Control Center の S3 バケットは、使用可能容量を報告しません。Astra Control Center で管理されているアプリケーションのバックアップまたはクローニングを行う前に、 ONTAP または StorageGRID 管理システムでバケット情報を確認します。

バケットを編集する

バケットのアクセスクレデンシャル情報を変更したり、選択したバケットがデフォルトバケットかどうかを変更したりできます。

> バケットを追加するときは、正しいバケットプロバイダを選択し、そのプロバイダに適したクレデンシャルを指定します。たとえば、タイプとして NetApp ONTAP S3 が許可され、StorageGRID クレデンシャルが受け入れられますが、このバケットを使用して原因の以降のアプリケーションのバックアップとリストアはすべて失敗します。を参照してください "リリースノート"。

手順

1. 左側のナビゲーションから、*バケット*を選択します。
2. [* アクション * （ * Actions * ）] 列の [オプション（ Options ）] メニューから、 [* 編集（ * Edit ）] を選択する。
3. バケットタイプ以外の情報を変更します。

バケットタイプは変更できません。

4. 「 * Update * 」を選択します。

バケットのクレデンシャルをローテーションするか、削除する

Astra Controlは、バケットのクレデンシャルを使用してS3バケットにアクセスし、シークレットキーを提供することで、Astra Control Centerがバケットと通信できるようにします。

バケットのクレデンシャルをローテーションする

クレデンシャルのローテーションを行う場合は、バックアップが進行中でないとき（スケジュール設定またはオンデマンド）に、ローテーションを継続して実行してください。

クレデンシャルの編集やローテーションを行う手順

1. 左側のナビゲーションから、*バケット*を選択します。
2. [* アクション * （ * Actions * ）] 列の [オプション（ Options ）] メニューから、 [* 編集（ * Edit ）] を選択する。
3. 新しいクレデンシャルを作成します。
4. 「 * Update * 」を選択します。

バケットのクレデンシャルを削除する

バケットのクレデンシャルを削除するのは、新しいクレデンシャルがバケットに適用されている場合やバケットがアクティブに使用されなくなった場合だけにしてください。

Astra Control に追加する最初のクレデンシャルセットは、 Astra Control がバックアップバケットの認証にクレデンシャルを使用するため、常に使用されています。バケットがアクティブな状態で使用されている場合は、これらのクレデンシャルを削除しないでください。削除すると、バックアップが失敗してバックアップが使用できなくなります。

アクティブなバケットクレデンシャルを削除する場合は、を参照してください "バケットのクレデンシャル削除のトラブルシューティング"。

Astra Control APIを使用してS3クレデンシャルを削除する方法については、を参照してください "Astra の自動化と API に関する情報"。

バケットを削除する

使用されなくなったバケットや正常でないバケットを削除することができます。これは、オブジェクトストアの設定をシンプルかつ最新の状態に保つために役立ちます。

デフォルトバケットを削除することはできません。そのバケットを削除する場合は、最初に別のバケットをデフォルトとして選択します。

必要なもの

- 開始する前に、このバケットの実行中または完了済みのバックアップがないことを確認してください。

- アクティブな保護ポリシーでバケットが使用されていないことを確認する必要があります。

ある場合は、続行できません。

手順

1. 左ナビゲーションから、 * バケット * を選択します。
2. [アクション * （ Actions * ）] メニューから、 [* 削除（ Remove ）] を選択します。

> Astra Control を使用すると、最初にバケットを使用してバックアップを実行するスケジュールポリシーが存在せず、削除しようとしているバケットにアクティブなバックアップが存在しないようにすることができます。

3. 「 remove 」と入力して操作を確認します。
4. 「 * Yes 、 remove bucket * 」を選択します。

詳細については、こちらをご覧ください

- "Astra Control API を使用"

ストレージバックエンドを管理します

ストレージバックエンドとして Astra Control のストレージクラスタを管理することで、永続ボリューム（ PVS ）とストレージバックエンドの間のリンケージを取得できるだけでなく、追加のストレージ指標も取得できます。ストレージ容量と健全性の詳細を監視できます。 Astra Control Center が Cloud Insights に接続されている場合のパフォーマンスも監視できます。

Astra Control API を使用してストレージバックエンドを管理する方法については、を参照してください "Astra の自動化と API に関する情報"。

ストレージバックエンドの管理に関連して、次のタスクを実行できます。

- "ストレージバックエンドを追加します"
- [ストレージバックエンドの詳細を表示します]
- [ストレージバックエンドの管理を解除します]
- Astra Data Storeストレージのバックエンドライセンスを更新
- Astraデータストアストレージバックエンドをアップグレードする
- [ストレージバックエンドを削除します]
- [ストレージバックエンドクラスタにノードを追加します]
- [ストレージバックエンドクラスタからノードを削除します]

ストレージバックエンドの詳細を表示します

ストレージバックエンドの情報は、ダッシュボードまたはバックエンドオプションで確認できます。

Storage Backend Detailsページにあるデータストアの場合は、次の情報が表示されます。

- Astraデータストアクラスタ
  - スループット、IOPS、およびレイテンシ
  - 使用済み容量と総容量の比較
- 各Astraデータストアクラスタボリューム
  - 使用済み容量と総容量の比較
  - スループット

ダッシュボードでストレージバックエンドの詳細を確認します

手順

1. 左側のナビゲーションから、 * ダッシュボード * を選択します。
2. 状態を示す Storage backend セクションを確認します。
   - * 正常でない * ：ストレージが最適な状態ではありません。これは、レイテンシの問題やコンテナの問題が原因でアプリケーションがデグレードした場合などに発生します。
   - * すべて正常 * ：ストレージは管理されており、最適な状態です。
   - * 検出 * ：ストレージは検出されましたが、 Astra Control では管理されていません。

バックエンドからストレージバックエンドの詳細を表示するオプションを選択します

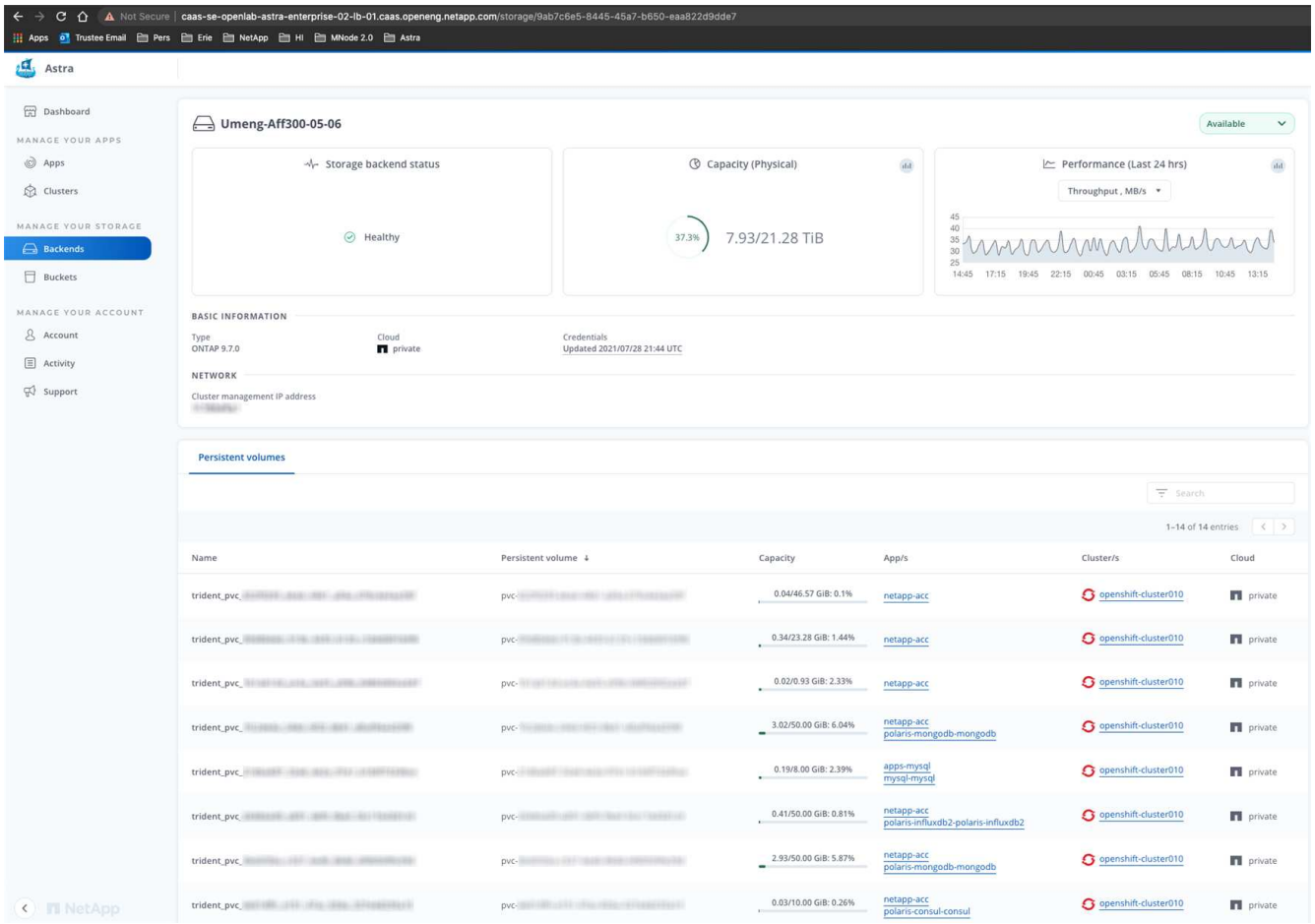
バックエンドの健常性、容量、パフォーマンス（ IOPS スループット、レイテンシ）に関する情報を表示します。

Kubernetesアプリケーションが使用しているボリュームが表示され、選択したストレージバックエンドに格納されます。Cloud Insights を使用すると、追加情報 が表示されます。を参照してください "Cloud Insights のドキュメント"。

手順

1. 左側のナビゲーション領域で、 * Backends * を選択します。
2. ストレージバックエンドを選択します。

NetApp Cloud Insights に接続した場合、 Cloud Insights からの抜粋がバックエンドのページに表示されます。



3. Cloud Insights に直接移動するには、指標画像の横にある * Cloud Insights * アイコンを選択します。

ストレージバックエンドの管理を解除します

バックエンドの管理を解除できます。

手順

1. 左のナビゲーションから、 * Backends * を選択します。
2. ストレージバックエンドを選択します。
3. * アクション * 列のオプションメニューから、 * 管理解除 * を選択します。
4. 「 unmanage 」と入力して操作を確定します。
5. 「 * Yes 、 unmanage storage backend * 」を選択します。

ストレージバックエンドを削除します

使用されなくなったストレージバックエンドを削除できます。これは、設定をシンプルかつ最新の状態に保つために役立ちます。

> Astra データストアバックエンドを削除する場合、 vCenter で作成されていないことが必要です。

必要なもの

- ストレージバックエンドが管理対象外であることを確認します。
- ストレージバックエンドに Astra データストアクラスタに関連付けられたボリュームがないことを確認します。

手順

1. 左ナビゲーションから、 * Backends * を選択します。
2. バックエンドが管理されている場合は、管理を解除します。
   a. [*Managed] を選択します。
   b. ストレージバックエンドを選択します。
   c. [* アクション * （ * Actions * ）] オプションから、 [* アンマネージ * （ * Unmanage * ）] を
   d. 「 unmanage 」と入力して操作を確定します。
   e. 「 * Yes 、 unmanage storage backend * 」を選択します。
3. [* Discovered （検出済み）] を選択
   a. ストレージバックエンドを選択します。
   b. [* アクション * （ * Actions * ）] オプションから、 [* 削除（ * Remove ）] を選択する。
   c. 「 remove 」と入力して操作を確認します。
   d. 「 * Yes 、 remove storage backend * 」を選択します。

Astra Data Storeストレージのバックエンドライセンスを更新

より大規模な導入や拡張機能をサポートするために、 Astra データストアストレージバックエンドのライセンスを更新できます。

必要なもの

- 導入および管理された Astra データストアストレージバックエンド
- Astra データストアライセンスファイル（ネットアップの営業担当者に連絡して Astra データストアライセンスを購入）

手順

1. 左のナビゲーションから、 * Backends * を選択します。
2. ストレージバックエンドの名前を選択します。
3. [基本情報]では、インストールされているライセンスのタイプを確認できます。

   ライセンス情報にカーソルを合わせると、有効期限や使用権の情報などの詳細情報を示すポップアップが表示されます。

4. [* License] で、ライセンス名の横にある編集アイコンを選択します。
5. [ライセンスの更新*]ページで、次のいずれかを実行します。

ライセンスステータス	アクション
Astraデータストアに少なくとも1つのライセンスが追加されている。	リストからライセンスを選択します。

ライセンスステータス	アクション
Astraデータストアにライセンスが追加されていない。	a. [*追加（Add *）]ボタンを選択します。 b. アップロードするライセンスファイルを選択してください。 c. 「*追加」を選択して、ライセンスファイルをアップロードします。

6. 「 * Update * 」を選択します。

Astraデータストアストレージバックエンドをアップグレードする

Astra Data StoreバックエンドをAstra Control Centerからアップグレードできます。アップグレードするには、まずアップグレードパッケージをアップロードする必要があります。Astra Control Centerは、このアップグレードパッケージを使用してAstraデータストアをアップグレードします。

必要なもの

- マネージドAstraデータストアストレージバックエンド
- アップロード済みのAstraデータストアアップグレードパッケージ（を参照） "ソフトウェアパッケージを管理します")

手順

1. [* Backends]を選択します。
2. リストからAstra Data Storeストレージバックエンドを選択し、「* Actions *」列で対応するメニューを選択します。
3. [* Upgrade]を選択します。
4. リストからアップグレードバージョンを選択します。

   リポジトリ内にバージョンが異なる複数のアップグレードパッケージがある場合は'ドロップダウンリストを開いて必要なバージョンを選択できます

5. 「 * 次へ * 」を選択します。
6. [アップグレードの開始]を選択します。

結果

アップグレードが完了するまで、[**Status**]列に[**Upgrading ***](アップグレード中)ステータスが[***Status**](*ステータス)ページに表示されます。

ストレージバックエンドクラスタにノードを追加します

Astra Data Store クラスタにノードを追加できます。このノードは、 Astra Data Store 用にインストールされたライセンスのタイプでサポートされるノード数まで追加できます。

必要なもの

- 導入済みでライセンス供与されている Astra データストアストレージバックエンド
- Astra Data Store ソフトウェアパッケージを Astra Control Center に追加しておきます

- クラスタに追加する 1 つ以上の新しいノード

手順

1. 左のナビゲーションから、 * Backends * を選択します。
2. ストレージバックエンドの名前を選択します。
3. 基本情報では、このストレージバックエンドクラスタ内のノード数を確認できます。
4. [ノード数（ * Nodes ）] で、ノード数の横にある編集アイコンを選択します。
5. [ノードの追加 *] ページで、新しいノードに関する情報を入力します。
   a. 各ノードにノードラベルを割り当てます。
   b. 次のいずれかを実行します。
      - Astra データストアでライセンスに基づいて常に使用可能な最大数のノードを使用する場合は、「常に最大数のノードを使用する」チェックボックスを有効にします。
      - Astra データストアで常に使用可能なノードの最大数を使用しない場合は、使用するノードの合計数を必要な数だけ選択します。
   c. 保護ドメインを有効にした状態で Astra データストアを導入した場合は、新しいノードを保護ドメインに割り当てます。
6. 「 * 次へ * 」を選択します。
7. 新しい各ノードの IP アドレスとネットワーク情報を入力します。1 つの新しいノードに 1 つの IP アドレスを入力するか、複数の新しいノードに 1 つの IP アドレスプールを入力します。

   Astra データストアで導入時に設定した IP アドレスを使用できる場合は、 IP アドレス情報を入力する必要はありません。
8. 「 * 次へ * 」を選択します。
9. 新しいノードの設定を確認します。
10. [ノードの追加] を選択します。

ストレージバックエンドクラスタからノードを削除します

Astraデータストアクラスタからノードを削除できます。正常なノードと障害ノードがあります。

Astraデータストアクラスタからノードを削除すると、そのノードのデータがクラスタ内の他のノードに移動し、Astraデータストアから削除される。

このプロセスには、次の条件が必要です。

- 他のノードにデータを受信するための十分な空きスペースが必要です。
- クラスタにノードが4つ以上必要です。

手順

1. 左のナビゲーションから、 * Backends * を選択します。
2. ストレージバックエンドの名前を選択します。
3. [ノード（Nodes *）]タブを選択します。

4. アクションメニューから*削除*を選択します。
5. 「remove」と入力して削除を確認します。
6. 「はい、ノードを削除します」を選択します。

詳細については、こちらをご覧ください

- "Astra Control API を使用"

Cloud Insights 接続とFluentd接続でインフラストラクチャを監視します

複数のオプション設定を構成して、 Astra Control Center の操作性を高めることができます。インフラ全体を監視して詳細を把握するには、 NetApp Cloud Insights への接続を確立します。Astra Control Center によって監視されているシステムから Kubernetes イベントを収集するには、 Fluentd 接続を追加します。

Astra Control Center を実行しているネットワークで、インターネットに接続するためのプロキシが必要な場合（サポートバンドルをNetApp Support Siteにアップロードする場合、または Cloud Insights への接続を確立する場合）は、 Astra Control Center でプロキシサーバを設定する必要があります。

Astra Data Storeストレージバックエンドのスループット、IOPS、容量をAstra Control Center Storage Backendページから監視することもできます。を参照してください "ストレージバックエンドを管理"。

Cloud InsightまたはNetApp Support Siteに接続するためのプロキシサーバを追加します

Astra Control Center を実行しているネットワークで、インターネットに接続するためのプロキシが必要な場合（サポートバンドルをNetApp Support Siteにアップロードする場合、または Cloud Insights への接続を確立する場合）は、 Astra Control Center でプロキシサーバを設定する必要があります。

> Astra Control Center は、プロキシサーバー用に入力した詳細を検証しません。必ず正しい値を入力してください。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. ドロップダウンリストから [**Connect**] を選択して、プロキシサーバを追加します。

4. プロキシサーバの名前または IP アドレスとプロキシポート番号を入力します。
5. プロキシサーバで認証が必要な場合は、このチェックボックスをオンにしてユーザ名とパスワードを入力します。

6. 「 * 接続」を選択します。

結果

入力したプロキシ情報が保存されている場合は、 **Account**>*Connections* ページの **HTTP Proxy** セクションに、接続されていることが示され、サーバー名が表示されます。

プロキシサーバの設定を編集します

プロキシサーバの設定を編集できます。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. ドロップダウンリストから * Edit * を選択して、接続を編集します。
4. サーバの詳細と認証情報を編集します。
5. [保存（ Save ）] を選択します。

プロキシサーバ接続を無効にします

プロキシサーバ接続を無効にすることができます。他の接続が中断される可能性があることを無効にする前に警告が表示されます。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. 接続を無効にするには、ドロップダウンリストから * 切断 * を選択します。
4. 表示されたダイアログボックスで、処理を確認します。

Cloud Insights に接続します

NetApp Cloud Insights を Astra Control Center インスタンスに接続すると、インフラ全体を監視して詳細に把握できます。Cloud Insights は、 Astra Control Center ライセンスに含まれています。

Cloud Insights には、 Astra Control Center が使用するネットワークから、またはプロキシサーバー経由で間接的にアクセスできる必要があります。

Cloud Insights にアストラコントロールセンターを接続すると、 Acquisition Unit ポッドが作成されます。この

ポッドは、 Astra Control Center で管理されているストレージバックエンドからデータを収集し、 Cloud Insights にプッシュします。このポッドには、 8GB の RAM と 2 つの CPU コアが必要です。

さらに、Cloud Insights に接続されたAstra ControlでAstraデータストアクラスタを管理している場合は、AstraデータストアクラスタごとにAstraデータストアにAcquisition Unitポッドが作成され、Astraデータストアからペアリングされたcloud Insights システムにデータストアが送信されます。各ポッドには、8GB のRAMと2つのCPUコアが必要です。

> Cloud Insights 接続を有効にすると、スループット情報をバックエンド * ページで確認できるほか、ストレージバックエンドを選択したあとにここから Cloud Insights に接続できます。ダッシュボード * の情報はクラスタセクションでも確認できます。また、そこから Cloud Insights に接続できます。

必要なもの

- admin * / * owner * 権限を持つ Astra Control Center アカウント。
- 有効な Astra Control Center ライセンス。
- Astra Control Center を実行しているネットワークで、インターネットに接続するためにプロキシが必要な場合は、プロキシサーバーです。

> Cloud Insights を初めて使用する場合は、の機能について理解しておいてください。を参照してください "Cloud Insights のドキュメント"。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. 接続を追加するには、ドロップダウンリストで * 切断されている * と表示されている * 接続 * を選択します。

オプションを表示して、 Cloud Insights 接続を有効にします。"]

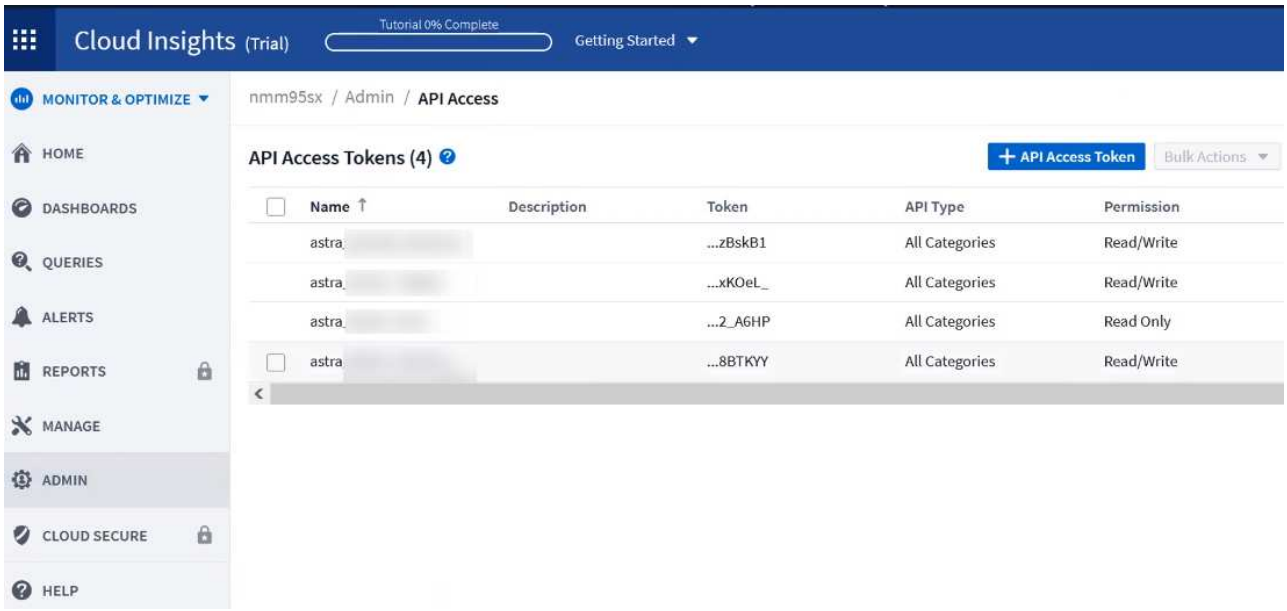
4. Cloud Insights API トークンとテナント URL を入力します。テナント URL の形式は次のようになります。

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

テナント URL は、 Cloud Insights ライセンスを取得すると取得されます。テナント URL がない場合は、を参照してください "Cloud Insights のドキュメント"。

   a. をダウンロードしてください "API トークン"をクリックし、 Cloud Insights テナントの URL にログインします。

b. Cloud Insights で、 * Admin*>* API Access* をクリックして、 * Read/Write * と * Read Only* API Access トークンの両方を生成します。



c. 「 * Read Only * 」キーをコピーします。Cloud Insights 接続を有効にするには、 [Astra Control Center] ウィンドウに貼り付ける必要があります。Read API Access Token Key 権限で、 Assets 、 Alerts 、 Acquisition Unit 、 and Data Collection を選択します。

d. 「 * Read/Write 」キーをコピーします。Astra Control Center * Connect Cloud Insights * ウィンドウに貼り付ける必要があります。Read/Write API Access Token キー権限で、 Assets 、 Data Ingestion 、 Log Ingestion 、 Acquisition Unit を選択します。 および Data Collection 。

> * 読み取り専用 * キーと * 読み取り / 書き込み * キーを生成することを推奨します。両方の目的で同じキーを使用することは推奨しません。デフォルトでは、トークンの有効期限は 1 年に設定されています。トークンが期限切れになるまでの最大期間を指定するために、デフォルトの選択を維持することをお勧めします。トークンの有効期限が切れると、テレメトリが停止します。

e. Cloud Insights からコピーしたキーを Astra コントロールセンターに貼り付けます。

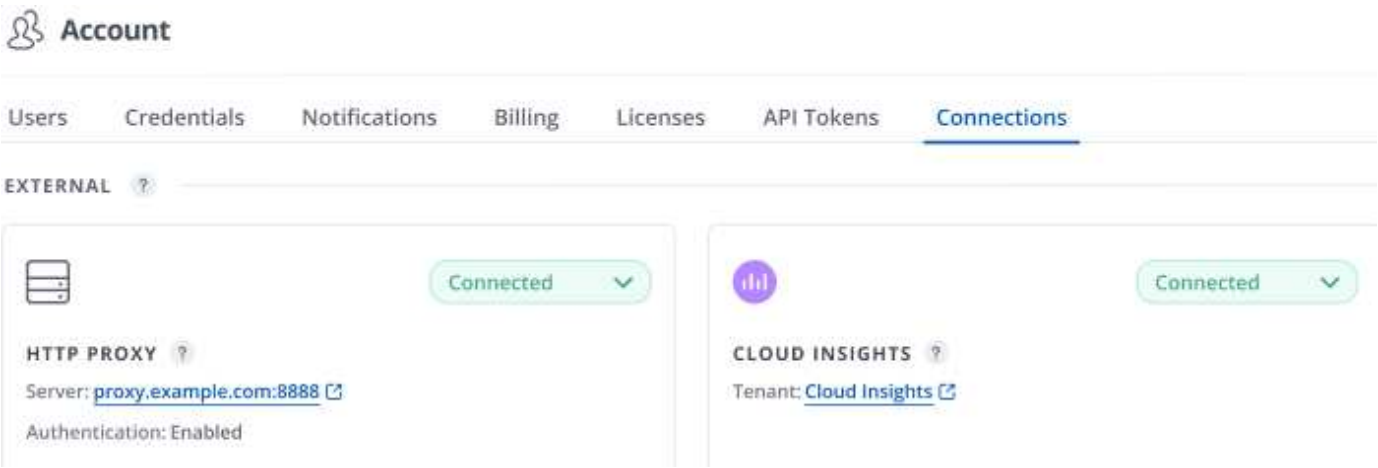
5. 「 * 接続」を選択します。

> [* 接続] を選択すると、 [* アカウント * > * 接続 *] ページの [* Cloud Insights *] セクションで、接続の状態が [* 保留中] に変わります。接続が有効になり、ステータスが * 接続済み * に変わるまで数分かかることがあります。

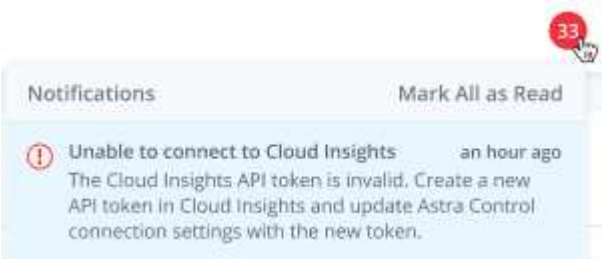
> Astra Control Center と Cloud Insights UI の間を簡単に行き来するには、両方にログインしていることを確認します。

Cloud Insights でデータを表示します

接続に成功した場合は、「 * アカウント * > * 接続 * 」ページの「 * Cloud Insights * 」セクションに接続されていることが示され、テナントの URL が表示されます。Cloud Insights にアクセスして、データが正常に受信されて表示されることを確認できます。

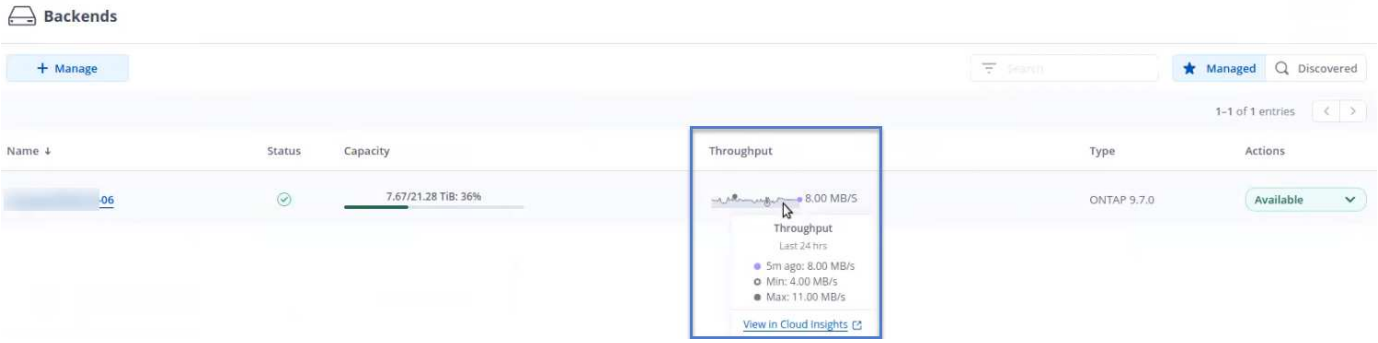


何らかの理由で接続に失敗した場合、ステータスは「 * 失敗 * 」と表示されます。失敗の理由は、 UI の右上にある * Notifications * で確認できます。



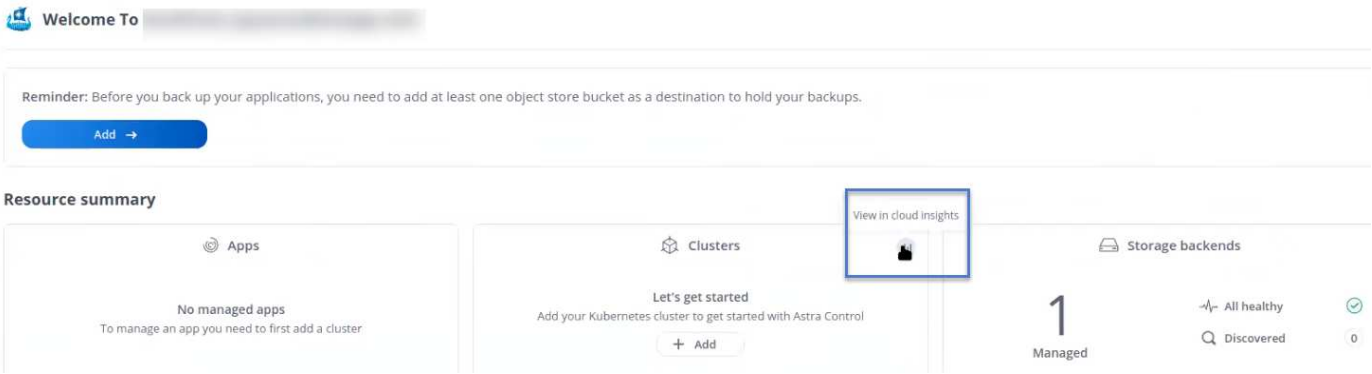
同じ情報は、「 * アカウント * > * 通知 * 」にも記載されています。

Astra Control Center では、スループット情報をバックエンド * ページで表示したり、ストレージバックエンドを選択した後にここから Cloud Insights に接続したりできます。



Cloud Insights に直接移動するには、指標画像の横にある * Cloud Insights * アイコンを選択します。

また、情報は * ダッシュボード * でも確認できます。



Cloud Insights 接続を有効にした後、 Astra Control Center に追加したバックエンドを削除すると、バックエンドは Cloud Insights へのレポートを停止します。

Cloud Insights 接続を編集します

Cloud Insights 接続を編集できます。

編集できるのは API キーのみです。Cloud Insights テナント URL を変更するには、 Cloud Insights 接続を切断して新しい URL に接続することを推奨します。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. ドロップダウンリストから * Edit * を選択して、接続を編集します。
4. Cloud Insights 接続設定を編集します。
5. [保存（ Save ）] を選択します。

Cloud Insights 接続を無効にします

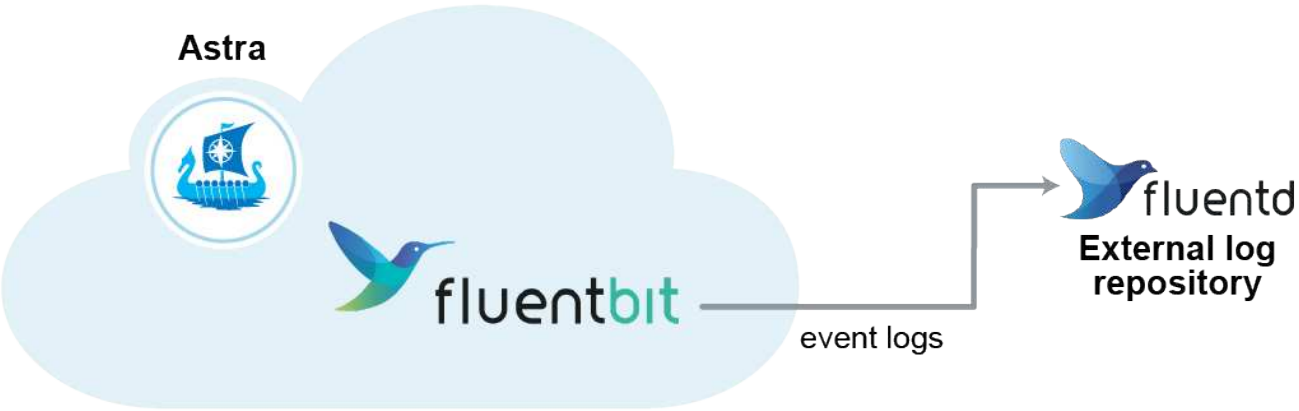
Cloud Insights 接続は、 Astra Control Center で管理されている Kubernetes クラスタに対して無効にすることができます。Cloud Insights 接続を無効にしても、すでに Cloud Insights にアップロードされている計測データは削除されません。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. 接続を無効にするには、ドロップダウンリストから * 切断 * を選択します。
4. 表示されたダイアログボックスで、処理を確認します。操作を確定すると、 [**Account**>*Connections*] ページで、 Cloud Insights のステータスが [*Pending （保留中）] に変わります。ステータスが * 切断された * に変わるまで数分かかります。

Fluentd に接続します

Astra Control Center から Fluentd エンドポイントにログ（ Kubernetes イベント）を送信できます。Fluentd 接続はデフォルトで無効になっています。



管理対象クラスタのイベントログのみが Fluentd に転送されます。

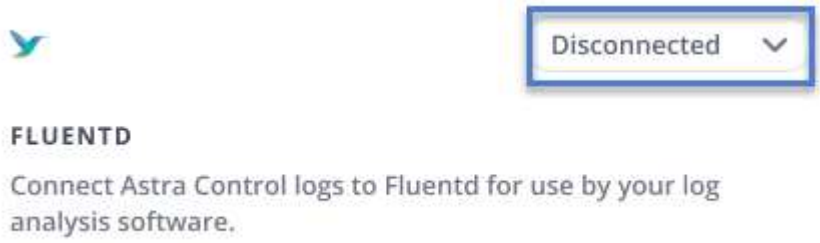
必要なもの

- admin * / * owner * 権限を持つ Astra Control Center アカウント。
- Kubernetes クラスタに Astra Control Center をインストールして実行

Astra Control Center では、 Fluentd サーバーに入力した詳細は検証されません。必ず正しい値を入力してください。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. 接続を追加するには、ドロップダウンリストから [* 接続（ * Connect * ）] を選択します。



4. Fluentd サーバーのホスト IP アドレス、ポート番号、および共有キーを入力します。
5. 「 * 接続」を選択します。

結果

Fluentd サーバーに入力した詳細が保存されている場合は、 * アカウント * > * 接続 * ページの * Fluentd * セクションに接続されていることが示されます。これで、接続した Fluentd サーバーにアクセスし、イベントログを表示できます。

何らかの理由で接続に失敗した場合、ステータスは「 * 失敗 * 」と表示されます。失敗の理由は、 UI の右上にある * Notifications * で確認できます。

同じ情報は、「 * アカウント * > * 通知 * 」にも記載されています。

> ログ収集に問題がある場合は、ワーカーノードにログインして、ログがにあることを確認する必要があります `/var/log/containers/`。

Fluentd 接続を編集します

Fluentd 接続を Astra Control Center インスタンスに編集できます。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. ドロップダウンリストから * Edit * を選択して、接続を編集します。
4. Fluentd エンドポイントの設定を変更します。
5. [保存（ Save ）] を選択します。

Fluentd 接続を無効にします

Astra Control Center インスタンスへの Fluentd 接続を無効にできます。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. 接続を無効にするには、ドロップダウンリストから * 切断 * を選択します。
4. 表示されたダイアログボックスで、処理を確認します。

アプリケーションとクラスタの管理を解除します

管理する必要がなくなったアプリケーションやクラスタを Astra Control Center から削除します。

アプリの管理を解除します

バックアップ、スナップショット、またはクローンを作成する必要がなくなったアプリケーションの管理を Astra Control Center から停止します。

- 既存のバックアップと Snapshot がすべて削除されます。
- アプリケーションとデータは引き続き使用できます。

手順

1. 左側のナビゲーションバーから、「 * アプリケーション * 」を選択します。
2. 管理する必要がなくなったアプリのチェックボックスをオンにします。
3. [アクション * （ Action * ）] メニューから、 [* アンマネージ * （ * Unmanage * ）] を選択し
4. 「 unmanage 」と入力して確定します。

5. アプリの管理を解除することを確認し、 [はい、アプリケーションの管理を解除します *] を選択します。

結果

Astra Control Center がアプリケーションの管理を停止。

クラスタの管理を解除します

管理する必要がなくなったクラスタの管理を Astra Control Center から解除します。

- この処理を実行すると、 Astra Control Center でクラスタが管理されなくなります。クラスタの構成は変更されず、クラスタも削除されません。
- Trident はクラスタからアンインストールされません。 "Trident のアンインストール方法をご確認ください"。

> クラスタの管理を解除する前に、クラスタに関連付けられているアプリケーションの管理を解除する必要があります。

手順

1. 左側のナビゲーションバーから、 * クラスタ * を選択します。
2. Astra Control Center で管理する必要がなくなったクラスタのチェックボックスを選択します。
3. * アクション * 列のオプションメニューから、 * 管理解除 * を選択します。
4. クラスタの管理を解除することを確認し、「 * Yes 、 unmanage cluster * 」を選択します。

結果

クラスタのステータスが「クラスタ」ページから「削除中」に変わり、クラスタが「クラスタ」ページから削除され、 Astra Control Center によって管理されなくなります。

> * Astra Control Center と Cloud Insights が接続されていない場合 * 、クラスタの管理を解除すると、テレメトリ・データの送信用にインストールされたすべてのリソースが削除されます。* Astra Control CenterとCloud Insights が接続されている場合*、クラスタの管理を解除すると、のみが削除されます `fluentbit` および `event-exporter` ポッド

Astra Control Center をアップグレードします

Astra Control Center をアップグレードするには、NetApp Support Siteからインストールバンドルをダウンロードし、以下の手順を実行して、環境内の Astra Control Center コンポーネントをアップグレードします。この手順を使用して、インターネット接続環境またはエアギャップ環境の Astra コントロールセンターをアップグレードできます。

必要なもの

- "アップグレードを開始する前に、環境が Astra Control Center 導入の最小要件を満たしていることを確認します"。
- すべてのクラスタオペレータが正常な状態であり、使用可能であることを確認します。

```
kubectl get clusteroperators
```

- すべての API サービスが正常な状態であり、使用可能であることを確認します。

```
kubectl get apiservices
```

- Astra Control Center からログアウトします。

このタスクについて

Astra Control Center のアップグレードプロセスでは、次の手順を実行できます。

- Astra Control Center バンドルをダウンロードします
- [バンドルを開梱し、ディレクトリを変更します]
- [イメージをローカルレジストリに追加します]
- 更新された Astra Control Center オペレータをインストールします
- Astra Control Center をアップグレードします
- [サードパーティサービスのアップグレード（オプション）]
- [システムステータスを確認します]
- [ロードバランシング用の入力を設定します]

すべてのAstra Control Centerポッドが削除されないように、アップグレードプロセス全体で次のコマンドを実行しないでください。 `kubectl delete -f astra_control_center_operator_deploy.yaml`

スケジュール、バックアップ、 Snapshot が実行されていないときは、メンテナンス時間内にアップグレードを実行します。

Docker Engine の代わりに Red Hat の Podman を使用している場合は、 Docker コマンドの代わりに Podman コマンドを使用できます。

Astra Control Center バンドルをダウンロードします

1. Astra Control Centerアップグレードバンドルをダウンロードします (`astra-control-center-[version].tar.gz`）をダウンロードします。https：mysupport.netapp.com/site/products/all/details/astra-control-center/downloads-tab[NetApp Support Site^]
2. （任意）次のコマンドを使用して、バンドルのシグニチャを確認します。

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

バンドルを開梱し、ディレクトリを変更します

1. 画像を抽出します。

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

イメージをローカルレジストリに追加します

1. コンテナエンジンに応じた手順を実行します。

Docker です

1. Astraディレクトリに移動します。

```
cd acc
```

2. [[[[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1></Z1>_image_local_registry_push]]]]]]]]]</Z2>アストラControl Centerイメージディレクトリ内のパッケージイメージをローカルレジストリにプッシュします。</Z3>コマンドを実行する前に、次の置き換えを行ってください。
   - bundle_fileをAstra Controlバンドルファイルの名前に置き換えます（例： `acc.manifest.yaml` ）。
   - my_registryをDockerリポジトリのURLに置き換えます。
   - my_registry_userをユーザー名に置き換えます。
   - my_registry_tokenをレジストリの認証済みトークンに置き換えます。

```
kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY
-u MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
```

ポドマン

1. レジストリにログインします。

```
podman login [your_registry_path]
```

2. 次のスクリプトを実行して、コメントに記載されているように<your _registry>を置き換えます。

```
# You need to be at the root of the tarball.
# You should see these files to confirm correct location:
#   acc.manifest.yaml
#   acc/

# Replace <YOUR_REGISTRY> with your own registry (e.g
registry.customer.com or registry.customer.com/testing, etc..)
export REGISTRY=<YOUR_REGISTRY>
export PACKAGENAME=acc
export PACKAGEVERSION=22.08.1-26
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
  # Load to local cache
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //')

  # Remove path and keep imageName.
  astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')

  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/netapp/astra/${PACKAGENAME}
/${PACKAGEVERSION}/${astraImageNoPath}

  # Push to the local repo.
  podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

更新された **Astra Control Center** オペレータをインストールします

1. ディレクトリを変更します。

```
cd manifests
```

2. Astra Control Centerオペレータ配置YAMLを編集します(`astra_control_center_operator_deploy.yaml`)を参照して、ローカルレジストリとシークレットを参照してください。

```
vim astra_control_center_operator_deploy.yaml
```

   a. 認証が必要なレジストリを使用する場合は、のデフォルト行を置き換えます `imagePullSecrets: []` 次の条件を満たす場合：

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

b. 変更 `[your_registry_path]` をクリックします `kube-rbac-proxy` でイメージをプッシュしたレジストリパスへのイメージ 前の手順。

c. 変更 `[your_registry_path]` をクリックします `acc-operator-controller-manager` でイメージをプッシュしたレジストリパスへのイメージ 前の手順。

d. に次の値を追加します `env` セクション。

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: [your_registry_path]/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_UPGRADETIMEOUT
          value: 300m
        image: [your_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

3. 更新された Astra Control Center オペレータをインストールします。

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

回答例：

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

4. ポッドが実行中であることを確認します

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

Astra Control Center をアップグレードします

1. Astra Control Centerカスタムリソース（CR）の編集 (`astra_control_center_min.yaml`）をクリックし、Astraバージョンを変更します (`astraVersion` の内部 `Spec`）最新の番号：

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

> ⓘ レジストリパスは、のイメージをプッシュしたレジストリパスと一致する必要があります前の手順。

2. 内に次の行を追加します `additionalValues` の内部 `Spec` Astra Control Center CRで、次の手順を実行します。

```
additionalValues:
    nautilus:
      startupProbe:
        periodSeconds: 30
        failureThreshold: 600
```

3. 次のいずれかを実行します。

   a. 独自のIngressControllerまたは入力がなく、トラフィックゲートウェイをロードバランサタイプサービスとして使用していて、そのセットアップを続行する場合は、別のフィールドを指定します `ingressType` （まだ存在しない場合）を選択し、に設定します `AccTraefik`。

   ```
   ingressType: AccTraefik
   ```

   b. デフォルトのAstra Control Centerの一般的な入力配置に切り替える場合は、独自のIngressController/Ingressセットアップ（TLS終端など）を指定し、Astra Control Centerへのルートを開き、を設定します `ingressType` 終了： `Generic`。

   ```
   ingressType: Generic
   ```

   フィールドを省略すると、プロセスは汎用的な配置になります。汎用的な導入が不要な場合は、必ずフィールドを追加してください。

4. （オプション）ポッドが終了し、再び使用可能になったことを確認します。

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

5. Astra のステータス状態がアップグレードが完了し、準備ができたことを示すまで待ちます。

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

対応：

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

6. ログインし直して、すべての管理対象クラスタとアプリケーションが引き続き存在し、保護されていることを確認します。
7. オペレータが Cert-manager を更新しなかった場合は、次の手順でサードパーティのサービスをアップグレードします。

サードパーティサービスのアップグレード（オプション）

以前のアップグレード手順では、サードパーティサービス Traefik および Cert-manager はアップグレードされません。オプションで、ここで説明する手順を使用してアップグレードしたり、システムに必要な既存のサービスバージョンを保持したりできます。

- * Traefik* ：デフォルトでは、 Astra Control Center が Traefik 導入のライフサイクルを管理します。設定 `externalTraefik` 終了： `false` （デフォルト）システムに外部Traefikが存在せず、Astra Control CenterによってTraefikがインストールおよび管理されていることを示します。この場合、 `externalTraefik` がに設定されます `false`。

  一方、独自のTraefik展開をお持ちの場合は、を設定します `externalTraefik` 終了： `true`。この場合、導入を維持していると、Astra Control CenterはCRDをアップグレードしません。ただし `shouldUpgrade` がに設定されます `true`。

- * Cert-manager]：デフォルトでは、Astra Control Centerは、を設定しない限り、証明書マネージャ(およびCRD)をインストールします `externalCertManager` 終了： `true`。設定 `shouldUpgrade` 終了： `true` Astra Control Centerを使用するには、CRDをアップグレードします。

次のいずれかの条件に該当する場合は、 Traefik がアップグレードされます。

- externalTraefik: false
- externalTraefik: true と shouldUpgrade: true 。

手順

1. を編集します `acc` CR：

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

2. を変更します `externalTraefik` フィールドと `shouldUpgrade` いずれかの値を入力します `true` または `false` 必要に応じて。

```
crds:
    externalTraefik: false
    externalCertManager: false
    shouldUpgrade: false
```

システムステータスを確認します

1. Astra Control Center にログインします。
2. すべての管理対象クラスタとアプリケーションが引き続き存在し、保護されていることを確認します。

ロードバランシング用の入力を設定します

Kubernetes 入力オブジェクトを設定して、クラスタ内でのロードバランシングなどのサービスへの外部アクセスを管理できます。

- デフォルトアップグレードでは、一般的な入力配置が使用されます。この場合は、入力コントローラまたは入力リソースも設定する必要があります。
- 入力コントローラが不要で、すでに持っているものを保持したい場合は、を設定します `ingressType` 終了： `AccTraefik`。

> サービスタイプ「 LoadBalancer 」および入力の詳細については、を参照してください "要件"。

この手順は、使用する入力コントローラのタイプによって異なります。

- nginx 入力コントローラ
- OpenShift 入力コントローラ

必要なもの

- CR 仕様で、
  - 状況 `crd.externalTraefik` が存在し、に設定されている必要があります `false` または
  - 状況 `crd.externalTraefik` はです `true`、 `crd.shouldUpgrade` もです `true`。
- が必要です "入力コントローラ" すでに導入されている必要があります。
- 。 "入力クラス" 入力コントローラに対応するものがすでに作成されている必要があります。
- V1.19 と v1.21 の間で Kubernetes のバージョンを使用している。

Nginx Ingress Controller の手順

1. 既存のシークレットを使用します `secure-testing-cert` または、タイプのシークレットを作成します [`kubernetes.io/tls`]をクリックします `netapp-acc` （またはカスタム名前付き）ネームスペース。を参照してください "TLS シークレット"。

2. 入力リソースをに配置します `netapp-acc` （またはカスタム名前付き）非推奨スキーマまたは新しいスキーマの名前空間：

   a. 廃止されたスキーマについては、次の例を参照してください。

```
apiVersion: extensions/v1beta1
kind: IngressClass
metadata:
  name: ingress-acc
  namespace: [netapp-acc or custom namespace]
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - backend:
        serviceName: traefik
        servicePort: 80
        pathType: ImplementationSpecific
```

   b. 新しいスキーマについては、次の例を参照してください。

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

OpenShift 入力コントローラの手順

1. 証明書を調達し、 OpenShift ルートで使用できるようにキー、証明書、および CA ファイルを取得します。
2. OpenShift ルートを作成します。

```
oc create route edge --service=traefik
--port=web -n [netapp-acc or custom namespace]
--insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem
```

入力セットアップを確認します

入力セットアップを確認してから、続行できます。

1. Traefikがに変更されていることを確認します `clusterIP` ロードバランサから：

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

2. Traefik でルートを確認します。

```
Kubectl get ingressroute ingressroutetls -n [netapp-acc or custom
namespace]
-o yaml | grep "Host("
```

結果は空である必要があります。

Astra Control Center をアンインストールします

試用版からフルバージョンの製品にアップグレードする場合は、 Astra Control Center コンポーネントの削除が必要になることがあります。Astra Control Center と Astra Control Center Operator を削除するには、この手順で説明されているコマンドを順に実行します。

アンインストールに問題がある場合は、を参照してください [アンインストールに関する問題のトラブルシューティング]。

必要なもの

- Astra Control Center UI を使用して、すべての管理を解除します "クラスタ"。

手順

1. Astra Control Center を削除します。次のコマンド例は、デフォルトのインストールに基づいています。カスタム構成を作成した場合は、コマンドを変更します。

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

結果

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

2. を削除するには、次のコマンドを使用します `netapp-acc` ネームスペース：

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

結果

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

3. Astra Control Center オペレータシステムコンポーネントを削除するには、次のコマンドを使用します。

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

結果

```
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
```

アンインストールに関する問題のトラブルシューティング

Astra Control Center のアンインストールで発生した問題に対処するには、次の回避策を実行します。

Astra Control Center をアンインストールしても、管理対象クラスタで監視オペレータポッドがクリーンアップされない

Astra Control Center をアンインストールする前にクラスタの管理を解除していない場合は、次のコマンドを使用して、ネットアップ監視ネームスペースとネームスペース内のポッドを手動で削除できます。

手順

1. 削除 `acc-monitoring` エージェント：

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

結果

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

2. ネームスペースを削除します。

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

結果

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

3. リソースの削除を確認します。

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

結果

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

4. 監視エージェントが削除されたことを確認：

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

サンプル結果：

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

5. カスタムリソース定義（ CRD ）情報の削除：

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

結果

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

Astra Control Center をアンインストールしても、 **Traefik CRD** をクリーンアップできない

Traefik CRD を手動で削除できます。CRD はグローバルリソースであり、削除するとクラスタ上の他のアプリケーションに影響を与える可能性があります。

手順

1. クラスタにインストールされている Traefik CRD を表示します。

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

応答

```
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 を削除します。

```
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
```

詳細については、こちらをご覧ください

- "アンインストールに関する既知の問題"

REST API で自動化

Astra Control REST API による自動化

Astra Control には REST API が搭載されており、 Curl などのプログラミング言語やユーティリティを使用して Astra Control 機能に直接アクセスできます。また、 Ansible などの自動化テクノロジを使用して Astra Control 環境を管理することもできます。

Kubernetes アプリケーションをセットアップおよび管理するには、 Astra UI または Astra Control API を使用できます。

詳細については、を参照してください "Astra 自動ドキュメント"。

知識とサポート

トラブルシューティング

発生する可能性のある一般的な問題を回避する方法について説明します。

"NetApp Knowledge Base for Astra"

詳細については、こちらをご覧ください

- "ネットアップにファイルをアップロードする方法（ログインが必要）"
- "ネットアップにファイルを手動でアップロードする方法（ログインが必要）"

ヘルプを表示します

ネットアップでは、 Astra Control をさまざまな方法でサポートしています。ナレッジベース（KB）記事やDiscordチャネルなど、24時間365日利用可能な無料のセルフサポートオプションをご用意しています。Astra Control アカウントには、 Web チケット発行によるリモートテクニカルサポートが含まれています。

> Astra Control Center の評価用ライセンスをお持ちの場合は、テクニカルサポートを受けることができます。ただし、NetApp Support Site（ NSS ）でケースを作成することはできません。フィードバック・オプションを使用してサポートに連絡するか、Discordチャネルを使用してセルフサービスで連絡できます。

最初に実行する必要があります "ネットアップのシリアル番号のサポートを有効にします" これらの非セルフサービスサポートオプションを使用するには、次の手順に従います。チャットや Web でのチケット発行、ケース管理には、NetApp Support Site（ NSS ）の SSO アカウントが必要です。

セルフサポートオプション

メインメニューから * Support * タブを選択すると、 Astra Control Center UI からサポートオプションにアクセスできます。

次のオプションは、 24 時間 365 日無料で利用できます。

- "* ナレッジベース * （ログインが必要）"： Astra Control に関する記事、 FAQ 、またはトラブルシューティング情報を検索します。
- * ドキュメントセンター * ：現在表示しているドキュメントサイトです。
- "* Discord *で助けを得なさい"：thePubカテゴリのAstraに移動して、他のユーザやエキスパートと交流しましょう。
- * サポートケースの作成 * ：トラブルシューティングのためにネットアップサポートに提供するサポートバンドルを生成
- * Astra Control についてのフィードバック * ： astra.feedback@netapp.com に電子メールを送信して、あなたの考え、アイデア、懸念を知らせてください。

ネットアップサポートへのサポートバンドルの日次アップロードを有効にします

Astra Control Centerのインストール中に、を指定した場合 `enrolled: true` の場合 `autoSupport` Astra Control Center Custom Resource Definition（CRD）ファイル (`astra_control_center_min.yaml`）を指定すると、日次サポートバンドルがに自動的にアップロードされます "NetApp Support Site"。

ネットアップサポートに提供するサポートバンドルを生成する

Astra Control Center を使用すると、管理者ユーザはバンドルを生成できます。バンドルには、ネットアップサポートに役立つログ、 Astra 環境のすべてのコンポーネントに対するイベント、管理対象のクラスタとアプリケーションに関する指標、トポロジ情報などが含まれます。インターネットに接続している場合は、NetApp Support Site（ NSS ）に Astra Control Center の UI から直接サポートバンドルをアップロードすることができます。

> Astra Control Center がバンドルを生成するのにかかる時間は、 Astra Control Center のインストールのサイズと、要求されたサポートバンドルのパラメータによって異なります。サポートバンドルの生成要求時に指定した期間によって、バンドルの生成にかかる時間が決まります（たとえば、期間を短くするとバンドルの生成にかかる時間が短縮されます）。

作業を開始する前に

NSS へのバンドルのアップロードにプロキシ接続が必要かどうかを確認します。プロキシ接続が必要な場合は、 Astra Control Center がプロキシサーバーを使用するように設定されていることを確認します。

1. [* アカウント * > * 接続 *] を選択します。
2. * 接続設定 * でプロキシ設定を確認します。

手順

1. NSS ポータルで、 Astra Control Center UI の * Support * ページに記載されているライセンスシリアル番号を使用してケースを作成します。
2. Astra Control Center UI を使用して、サポートバンドルを生成するには、次の手順を実行します。
   a. [サポート * （ Support * ）] ページの [サポートバンドル（ Support bundle ）] タイルで、 [* 生成（ Generate ）] を選択します。
   b. [* サポートバンドルの生成 * （ Generate a Support Bundle * ）] ウィンドウで、期間を選択します。

      クイックタイムフレームまたはカスタムタイムフレームのいずれかを選択できます。

      > カスタムの日付範囲を選択したり、期間にカスタムの期間を指定したりできます。

   c. 選択したら、 * 確認 * を選択します。
   d. [生成時にNetApp Support Siteにバンドルをアップロードする *] チェックボックスをオンにします。
   e. [* バンドルの生成 *] を選択します。

      サポートバンドルの準備が完了すると、アラート領域の * アカウント * > * 通知 * ページ、 * アクティビティ * ページ、および通知リストに通知が表示されます（ UI の右上にあるアイコンを選択してアクセスできます）。

生成が失敗した場合は、バンドルの生成ページにアイコンが表示されます。アイコンを選択すると、メッ

セージが表示されます。

> UI の右上にある通知アイコンには、バンドルの作成に成功したタイミング、バンドルの作成に失敗したタイミング、バンドルをアップロードできなかったタイミング、バンドルをダウンロードできなかったタイミングなど、サポートバンドルに関連するイベントに関する情報が表示されます。

エアギャップ設置がある場合

エアギャップのある設置の場合は、サポートバンドルが生成されたあとに次の手順を実行します。バンドルがダウンロード可能になると、 [サポート *] ページの [サポートバンドル *] セクションの [生成 *] の横に [ダウンロード] アイコンが表示されます。

手順

1. ダウンロードアイコンを選択して、バンドルをローカルにダウンロードします。
2. 手動で NSS にバンドルをアップロードします。

   これを行うには、次のいずれかの方法を使用します。

   - 使用 "NetApp Authenticated File Upload （ログインが必要）"。
   - NSS でケースにバンドルを直接添付します。
   - NetApp Active IQ を使用する。

詳細については、こちらをご覧ください

- "ネットアップにファイルをアップロードする方法（ログインが必要）"
- "ネットアップにファイルを手動でアップロードする方法（ログインが必要）"

以前のバージョンの Astra Control Center ドキュメント

以前のリリースのドキュメントを参照できます。

- "Astra Control Center 22.04 のドキュメント"
- "Astra Control Center 21.12 ドキュメント"
- "Astra Control Center 21.08 ドキュメント"

法的通知

著作権に関する声明、商標、特許などにアクセスできます。

著作権

"https://www.netapp.com/company/legal/copyright/"

商標

NetApp 、 NetApp のロゴ、および NetApp の商標ページに記載されているマークは、 NetApp, Inc. の商標です。その他の会社名および製品名は、それぞれの所有者の商標である場合があります。

"https://www.netapp.com/company/legal/trademarks/"

特許

ネットアップが所有する特許の最新リストは、次のサイトで入手できます。

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

プライバシーポリシー

"https://www.netapp.com/company/legal/privacy-policy/"

オープンソース

通知ファイルには、ネットアップソフトウェアで使用されるサードパーティの著作権およびライセンスに関する情報が記載されています。

- "Astra Control Center へのお知らせ"
- "Astraデータストアに関する注意事項"

Astra Control API ライセンス

https://docs.netapp.com/us-en/astra-automation/media/astra-api-license.pdf

著作権に関する情報

Copyright © 2023 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、http://www.netapp.com/TMに記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。