Astra を使用する

Astra Control Center

NetApp
November 21, 2023

目次

Astra を使用する 1
アプリの管理 1
アプリを保護します 7
アプリケーションとクラスタの健常性を表示します 30
アカウントを管理します 33
バケットを管理する 44
ストレージバックエンドを管理します 47
インフラを監視、保護 51
アプリケーションとクラスタの管理を解除します 58
Astra Control Center をアップグレードします 59
Astra Control Center をアンインストールします 70

Astra を使用する

アプリの管理

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] 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、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"
  - "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。

> 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。

クラスタにアプリをインストールします

Astra Control にクラスタを追加したので、クラスタにアプリケーションをインストールしたり、既存のアプリケーションを管理したりできます。名前空間にスコープ指定されているアプリケーションはすべて管理でき

ます。ポッドがオンラインになったら、 Astra Control を使用してアプリケーションを管理できます。

Helm チャートから検証済みのアプリケーションを展開する方法については、次を参照してください。

- "Helm チャートから MariaDB を導入します"
- "Helm チャートから MySQL を導入します"
- "Helm チャートから Postgres を導入します"
- "Helm チャートから Jenkins をデプロイします"

アプリの管理

Astra Control を使用すると、アプリケーションをネームスペースレベルまたは Kubernetes ラベルで管理できます。

> Helm 2 でインストールされたアプリケーションはサポートされていません。

次のアクティビティを実行して、アプリケーションを管理できます。

- アプリの管理
  - [ネームスペースでアプリケーションを管理します]
  - Kubernetes ラベルでアプリケーションを管理
- [アプリケーションを無視します]
- [アプリの管理を解除します]

> 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。システムアプリを表示するには、「システムアプリを表示」フィルタを使用します。

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"。

> 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。

ネームスペースでアプリケーションを管理します

アプリページの * 検出された * セクションには、名前空間と Helm がインストールされたアプリ、またはそれらの名前空間内のカスタムラベル付きアプリが表示されます。各アプリケーションを個別に管理することも、ネームスペースレベルで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。

たとえば、毎週同じ頻度で「 Maria 」のバックアップポリシーを設定したいのに、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、単一のネームスペースで管理する必要はありません。

Astra Control を使用すると、階層の両方のレベル（名前空間とその名前空間内のアプリケーション）を個別

に管理できますが、いずれか一方を選択することをお勧めします。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。

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. [* Discovered * （検出されました * ）] フィルタ

3. 検出されたネームスペースのリストを表示します。ネームスペースを展開して、アプリケーションおよび関連するリソースを表示します。

   Astra Control では、 Helm アプリケーションとカスタムラベルの付いたアプリケーションがネームスペースに表示されます。Helm ラベルがある場合は、タグアイコンで指定されます。

4. [**Group**] 列を参照して、アプリケーションが実行している名前空間を確認します (フォルダアイコンで指定されています) 。
5. 各アプリケーションを個別に管理するか、ネームスペースレベルで管理するかを決定します。
6. 階層内の目的のレベルで目的のアプリケーションを検索し、 [アクション *] 列の [オプション] メニューから [* 管理 *] を選択します。
7. アプリを管理しない場合は、 [アクション *] 列の [オプション] メニューから [* 無視 *] を選択します。

   たとえば、「 Maria 」ネームスペースの下にあるすべてのアプリケーションを同じスナップショットポリシーとバックアップポリシーで管理したい場合は、ネームスペースを管理し、ネームスペース内のアプリケーションは無視してください。

8. 管理対象アプリのリストを表示するには、表示フィルターとして「 * 管理対象 * 」を選択します。

> 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。

9. 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。

結果

管理対象として選択したアプリは、 [管理対象 *] タブから利用できるようになりました。無視されたアプリは、 * 無視された * タブに移動します。新しいアプリケーションがインストールされると、検出されたタブにはアプリが表示されないため、見つけやすくなり、管理も簡単になります。

Kubernetes ラベルでアプリケーションを管理

Astra Control の [アプリ] ページの上部には、「 * カスタムアプリの定義 * 」という名前のアクションが含まれています。このアクションを使用して、 Kubernetes ラベルで識別されるアプリケーションを管理できます。 "Kubernetes ラベルでカスタムアプリケーションを定義する方法については、こちらをご覧ください"。

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. [* 定義（ Define ）] を選択します
3. [* カスタムアプリケーションの定義 * （ Define custom application * ）] ダイアログボックスで、アプリケーションを管理するために必要な情報を入力します。
   a. * 新しいアプリ * ：アプリの表示名を入力します。
   b. * クラスタ * ：アプリケーションが存在するクラスタを選択します。
   c. * 名前空間： * アプリケーションの名前空間を選択します。
   d. * ラベル： * ラベルを入力するか、以下のリソースからラベルを選択してください。
   e. * 選択したリソース * ：保護する Kubernetes リソース（ポッド、シークレット、永続ボリュームなど）を表示および管理します。
      - リソースを展開し、ラベル数を選択して、使用可能なラベルを表示します。
      - ラベルを 1 つ選択します。

        ラベルを選択すると、 [**Label**] フィールドにラベルが表示されます。Astra Control は、 [選択されていないリソース *] セクションも更新して、選択したラベルと一致しないリソースを表示します。
   f. * 選択されていないリソース * ：保護する必要がないアプリケーションリソースを確認します。
4. 「 * カスタムアプリケーションの定義 * 」を選択します。

結果

Astra Control を使用すると、アプリケーションを管理できます。これで、 [* 管理対象 * （ * Managed * ）] タブに表示されます。

アプリケーションを無視します

検出されたアプリケーションは、検出されたリストに表示されます。この場合は、新しくインストールされたアプリケーションを簡単に検索できるように、検出されたリストをクリーンアップできます。また、管理しているアプリケーションがあり、後でそれらを管理する必要がなくなる場合もあります。これらのアプリケーションを管理したくない場合は、無視するように指定できます。

また、アプリケーションを 1 つのネームスペースで同時に管理することもできます（ネームスペース管理）。ネームスペースから除外するアプリケーションは無視してかまいません。

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. フィルタとして * Discovered * を選択します。
3. アプリケーションを選択します。
4. [* アクション * （ * Actions * ）] 列の [オプション（ Options ）] メニューから、 [* 無視 * （ * Ignore * ）] を選択
5. 無視を解除するには、 * 無視解除 * を選択します。

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。

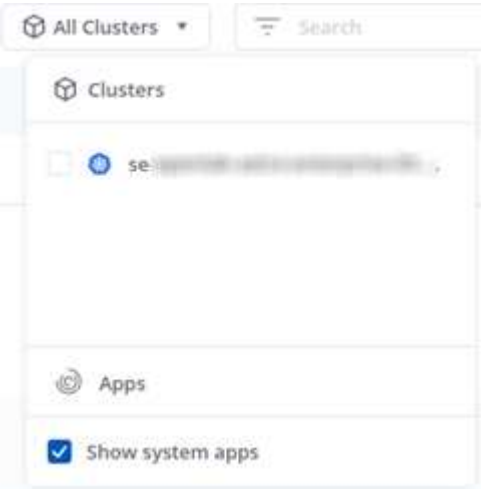
手順

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. フィルタとして [*Managed] を選択します。
3. アプリケーションを選択します。
4. * 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。
5. 情報を確認します。
6. 「 unmanage 」と入力して確定します。
7. [はい、アプリケーションの管理を解除 *] を選択します。

システムアプリケーションについて教えてください。

Astra Control は、 Kubernetes クラスタで実行されているシステムアプリケーションも検出します。これらのシステムアプリは、バックアップが必要になることが稀であるため、デフォルトでは表示されません。

ツールバーのクラスターフィルターの下にあるシステムアプリを表示 * システムアプリを表示 * チェックボックスをオンにすると、アプリケーションページからシステムアプリを表示できます。



> 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。

詳細については、こちらをご覧ください

- "Astra Control API を使用"

カスタムアプリケーションの例を定義します

カスタムアプリケーションを作成すると、 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。Kubernetes リソースのこの収集は、ネームスペースとラベルに基づいています。

カスタムアプリケーションを使用すると、 Astra Control 操作に含めるものをより細かく制御できます。次のものが含まれます。

- クローン
- スナップショット
- バックアップ
- 保護ポリシー

ほとんどの場合、アプリケーション全体で Astra Control の機能を使用します。ただし、これらの機能を使用するカスタムアプリケーションを、名前空間内の Kubernetes オブジェクトに割り当てるラベルで作成することもできます。

> カスタムアプリケーションは、単一クラスタの指定したネームスペース内でのみ作成できます。Astra Control では、カスタムアプリケーションを複数のネームスペースまたはクラスタにまたがって使用することはできません。

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"。

> 同じリソースに対して名前の異なる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。リソースのカスタムアプリケーションを作成する場合は、そのアプリケーションが他のポリシーに基づいてクローニングまたはバックアップされていないことを確認してください。

必要なもの

- Astra Control にクラスタを追加

手順

1. [アプリケーション] ページで、 [**+ 定義] を選択します。

   [カスタムアプリケーション] ウィンドウには、カスタムアプリケーションに含まれるリソースまたはカスタムアプリケーションから除外されるリソースが表示されます。これにより、カスタムアプリケーションを定義するための正しい条件を選択できるようになります。

2. ポップアップウィンドウで、アプリケーション名を入力し、 **Cluster** ドロップダウンでクラスタを選択し、 **Namespace** ドロップダウンからアプリケーションの名前空間を選択します。
3. ドロップダウン * ラベル * リストから、アプリと名前空間のラベルを選択します。
4. 1 つの配置に対してカスタムアプリケーションを定義した後、必要に応じて他の配置についても同じ手順を繰り返します。

2 つのカスタムアプリケーションの作成が完了したら、これらのリソースを他の Astra Control アプリケーシ

ョンとして扱うことができます。Kubernetes ラベルに基づいて、リソースグループごとにデータのクローンを作成し、バックアップと Snapshot を作成し、リソースグループごとにカスタムの保護ポリシーを作成できます。

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。そのクラスタには 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 管理者は、安定したリリースポッドに「展開 = 安定」というラベルを追加します。チームは、カナリアリリースポッドに「展開 = カナリア」というラベルを追加します。

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、より一時的なものです。したがって、「配置」 = 「カナリア」というラベルの付いたすべてのものに対して、より積極的で短期的な保護ポリシーを作成したいと考えています。

データの競合を回避するために、管理者は 2 つのカスタムアプリケーションを作成します。 1 つは「カナリア」リリース用、もう 1 つは「 stable 」リリース用です。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。

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"。

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バックアップ

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] オプション：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.] 災害が発生した場合は、アプリケーションをリストアします

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）。

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、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 ワードプレス ` OC adm policy add -scc to -group anyuid system: serviceaccounts:wordpress `OC adm policy add -scc to -user Privileged -z default-n wordpress

保護ポリシーを設定します

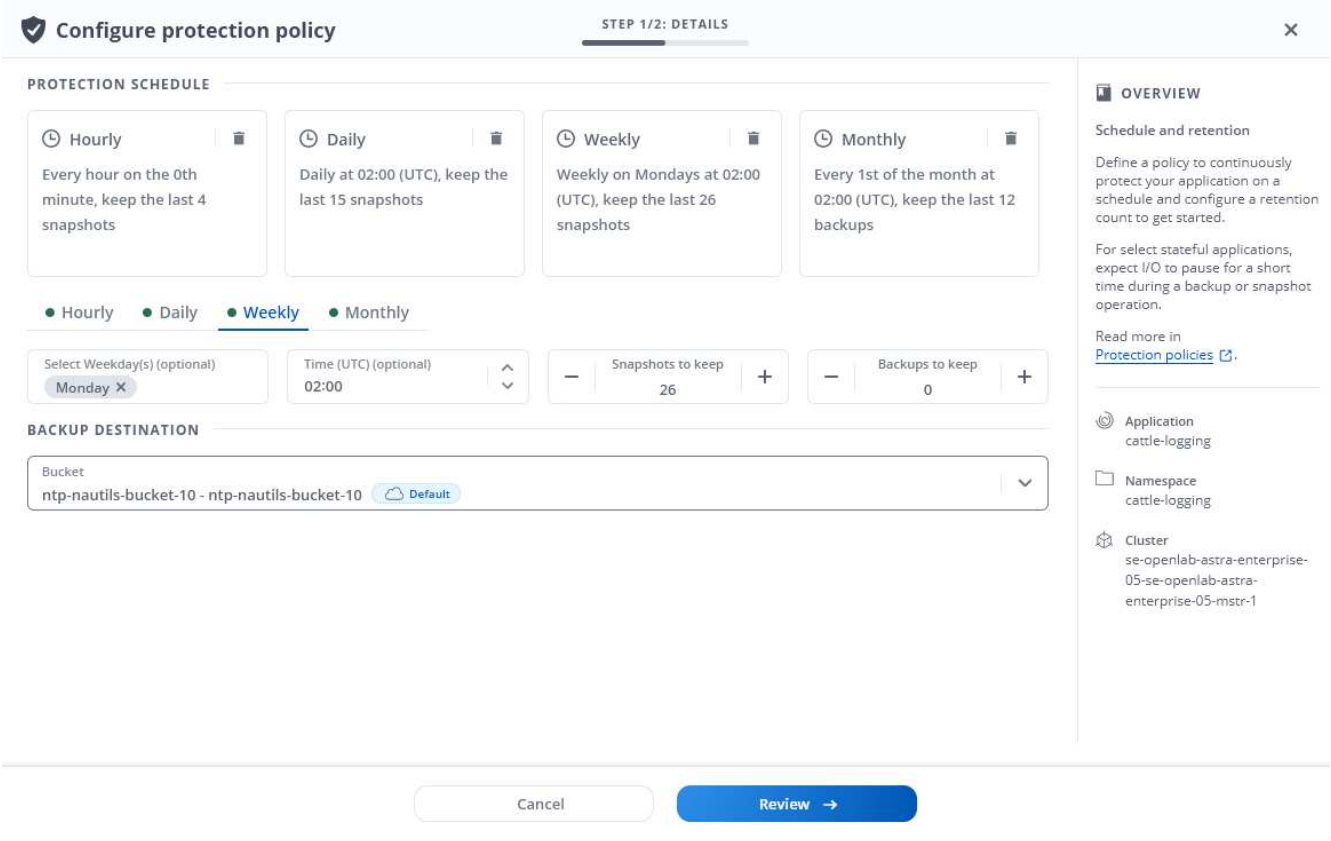
保護ポリシーは、定義されたスケジュールでスナップショット、バックアップ、またはその両方を作成することでアプリケーションを保護します。Snapshot とバックアップを毎時、日次、週次、および月単位で作成し、保持するコピーの数を指定できます。たとえば、保護ポリシーでは、週単位のバックアップと日単位の Snapshot が作成され、 1 カ月間はバックアップと Snapshot が保持されます。スナップショットやバックアップを作成する頻度と、それらを保持する期間は、組織のニーズによって異なります。

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。

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"。

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 ワードプレス ` 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。

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 ワードプレス ` 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・クローンが [*Applications] ページの [Available （使用可能）] 状態になっている場合 ' クローン操作は正常に実行されます

> 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スナップショットの前または後に実行でき

るカスタムスクリプトです。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。

デフォルトの実行フックと正規表現

一部のアプリケーションでは、ネットアップが提供するデフォルトの実行フックが Astra Control に付属しており、スナップショットの前後にフリーズや再開の操作を処理します。Astra Control では、正規表現を使用して、アプリケーションのコンテナイメージを次のアプリケーションに照合します。

- MariaDB
  - 正規表現 \bmariadb\b に一致しています
- MySQL
  - 正規表現 \bmysqql\b に一致しています
- PostgreSQL
  - 正規表現 \bpostgresql\b と一致します

一致した場合は、そのアプリケーションのデフォルトの実行フックがアプリケーションのアクティブな実行フックのリストに表示され、そのアプリケーションのスナップショットが作成されると、それらのフックが自動的に実行されます。カスタムアプリケーションの 1 つに、正規表現の 1 つと一致するように表示されるイメージ名が似ている場合（デフォルトの実行フックを使用しない場合）、イメージ名を変更することができます。 または、そのアプリケーションのデフォルト実行フックを無効にして、代わりにカスタムフックを使用します。

デフォルトの実行フックを削除または変更することはできません。

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。

- Astra Control では、実行フックを実行可能なシェルスクリプトの形式で記述する必要があります。
- スクリプトサイズは 128KB 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他のフックとスナップショットは ' フックが失敗しても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。

> 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。

スナップショットが実行されると、実行フックイベントが次の順序で実行されます。

1. ネットアップが提供するデフォルトの Snapshot 前実行フックは、該当するコンテナで実行されます。
2. 適用可能なカスタムスナップショット前実行フックは、適切なコンテナで実行されます。必要な数のカスタムスナップショット前フックを作成して実行できますが ' スナップショットの実行順序は保証も構成もされていません
3. スナップショットが実行されます。
4. 適用可能なカスタムスナップショット後実行フックは、適切なコンテナで実行されます。必要な数のカスタムスナップショット後フックを作成して実行できますが ' スナップショット後のこれらのフックの実行順序は保証されておらず ' 構成もできません
5. ネットアップが提供するデフォルトのポスト Snapshot 実行フックは、該当するコンテナで実行されます。

> 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、作成されたスナップショットの整合性をテストします。これを行うには、アプリケーションを一時ネームスペースにクローニングし、スナップショットをリストアしてから、アプリケーションをテストします。

既存の実行フックを表示します

既存のカスタム実行フックまたはネットアップが提供するアプリケーションのデフォルト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（スナップショット前またはスナップショット後）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。

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、 Linux コマンドを実行しているか、実行可能ファイルへの完全パスを提供している場合を除き、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. [* 新しいフックを追加 *] を選択します。

4. フックの詳細 * 領域で、フックを実行するタイミングに応じて、「 * Pre-Snapshot * 」または「 * Post-Snapshot * 」を選択します。
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。
   - 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。
   - 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. クリップボードから貼り付け * オプションを選択します。
     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。
9. [* フックを追加 *] を選択します。

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表示します

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

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"。

独自の「 kubeconfig 」ファイルを作成する場合は、その中で * 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" 「 kubeconfig 」ファイルの作成方法については、を参照してください。

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。

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 "ネットアップサポートサイト"。
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、ネットアップサポートサイト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"。
- ネットアップサポートサイト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. ネットアップサポートサイトからソフトウェアパッケージをダウンロードし、「kubectl」ユーティリティがインストールされているマシンに保存します。
2. 圧縮された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 .bundle.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。
   - my_registryをDockerリポジトリのURLに置き換えます。
   - my_registry_userとmy_registry_passwordをリポジトリの資格情報に置き換えます

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

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. アップロードするAstra Controlバンドルファイルを「.yaml」形式で選択します。
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、展開されたパッケージの内容の中で、ディレクトリを「.yaml」バンドルファイルの場所に変更します。
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
- 汎用 S3
- 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。

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]
- [ストレージバックエンドライセンスを更新する]
- [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Astra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

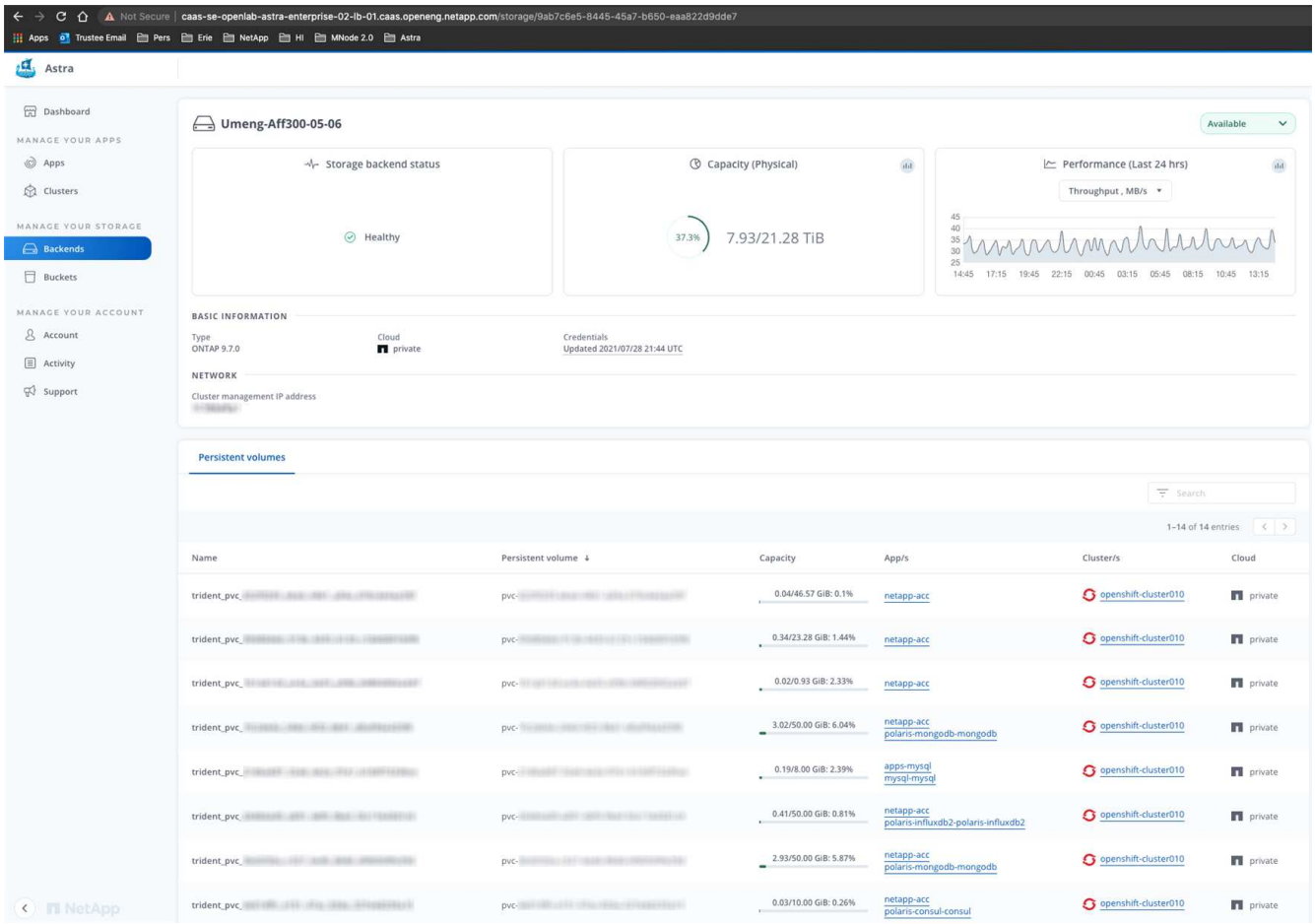
バックエンドの健常性、容量、パフォーマンス（ IOPS スループット、レイテンシ）に関する情報を表示します。

Cloud Insights に接続すると、 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ます。このボリュームは、選択したストレージバックエンドに格納されます。

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。

ストレージバックエンドライセンスを更新する

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。
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。

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。

詳細については、こちらをご覧ください

- "Astra Control API を使用"

インフラを監視、保護

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。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ネットアップサポートサイト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。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。

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ネットアップサポートサイト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。

> 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。"]

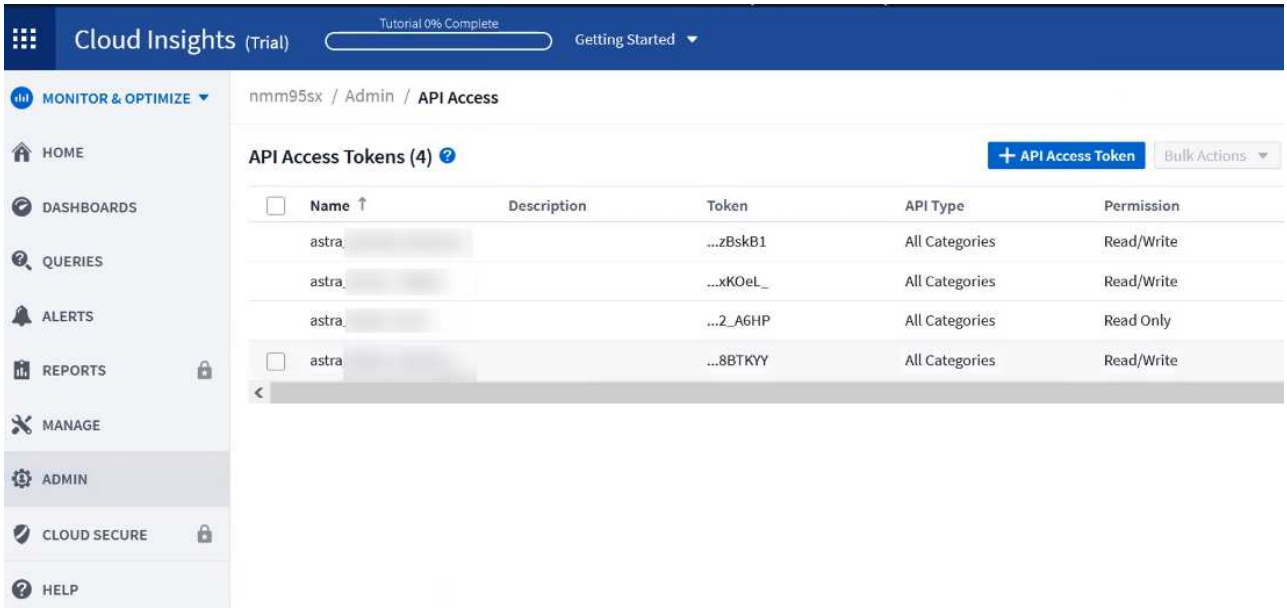
4. Cloud Insights API トークンとテナント URL を入力します。テナント URL の形式は次のようになります。

```
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。

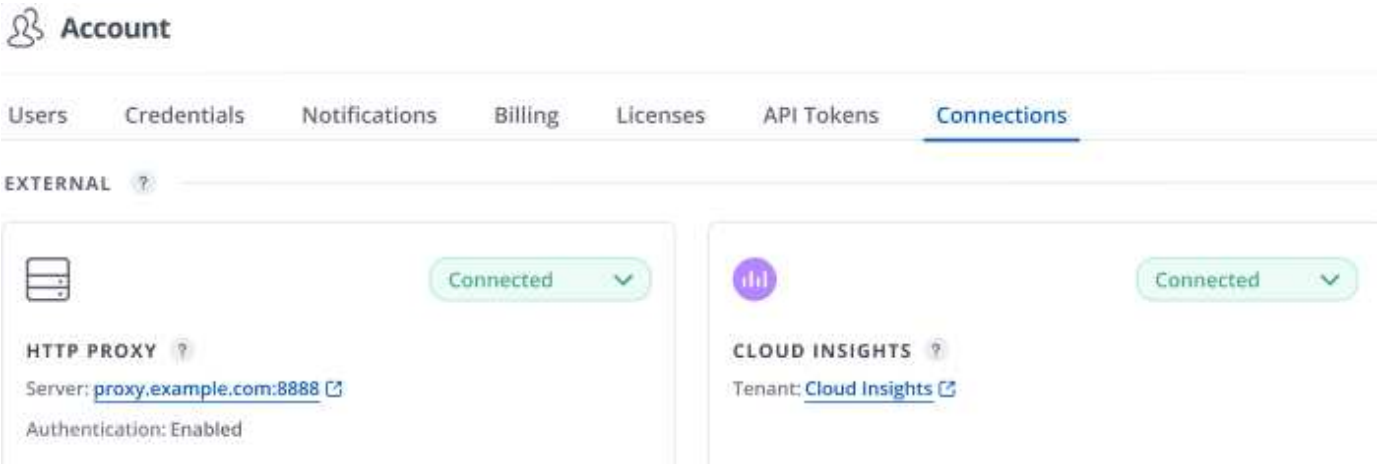
5. 「 * 接続」を選択します。

> [* 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。

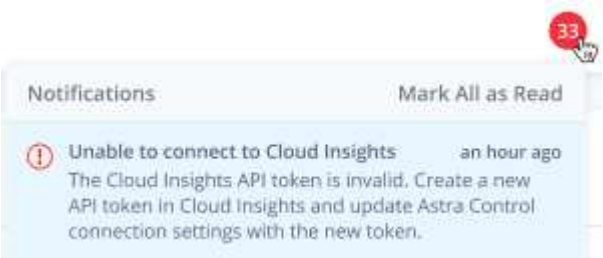
> Astra Control Center と Cloud Insights UI の間を簡単に行き来するには、両方にログインしていることを確認します。

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。

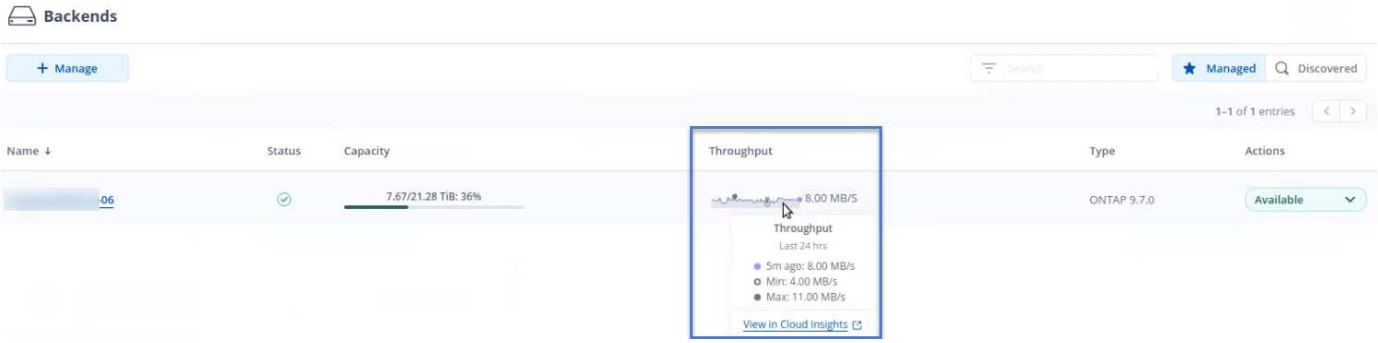


何らかの理由で接続に失敗した場合、ステータスは「 * 失敗 * 」と表示されます。失敗の理由は、 UI の右上にある * Notifications * で確認できます。



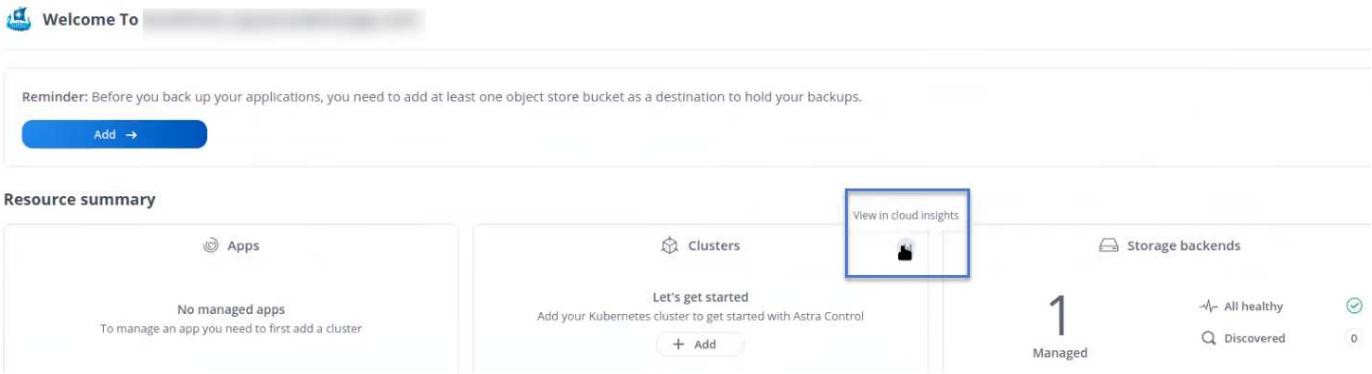
同じ情報は、「 * アカウント * > * 通知 * 」にも記載されています。

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

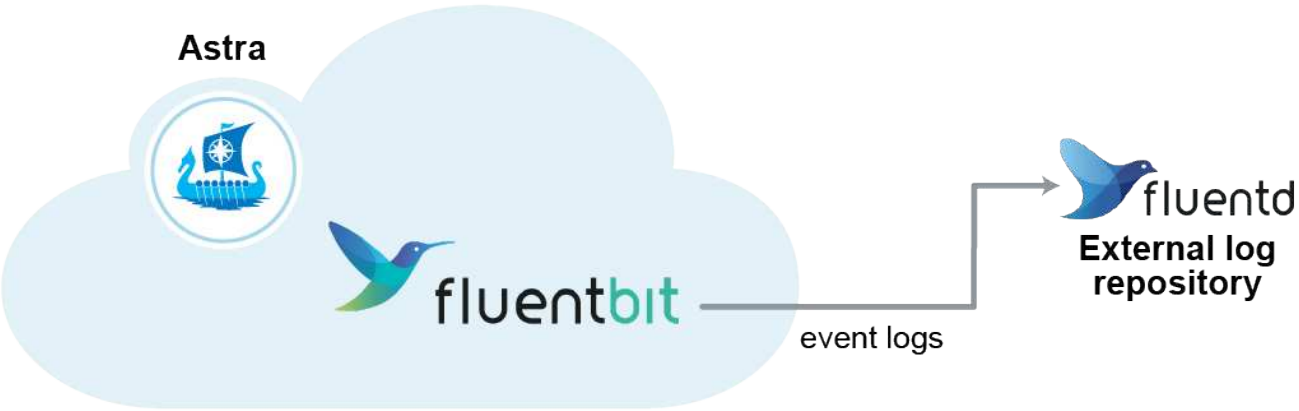
Cloud Insights 接続は、 Astra Control Center で管理されている Kubernetes クラスタに対して無効にすることができます。Cloud Insights 接続を無効にしても、すでに Cloud Insights にアップロードされている計測データは削除されません。

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。

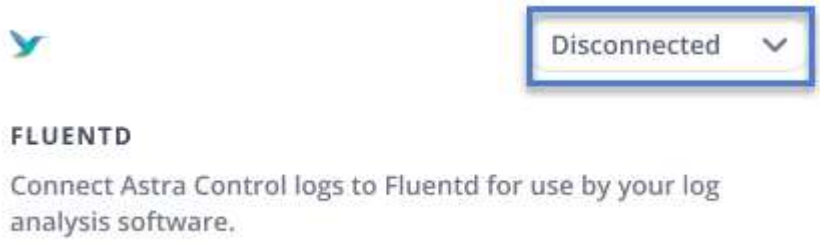
必要なもの

- 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 ' ログが /var/log/container/ で使用可能であることを確認してください

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 'fluentbit' および 'event-exporter ポッドのみが削除されます

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、ネットアップサポートサイト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。

  OpenShift の例：

```
oc 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。

  OpenShift の例：

```
oc 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するため、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_deployment.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. から Astra Control Center アップグレードバンドル (`Astra - control-ccenter-[version].tar.gz') をダウンロードします "ネットアップサポートサイト"。
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 astra-control-center[version].pub
-signature <astra-control-center[version].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
```

2. Astra ディレクトリに移動します。

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

イメージをローカルレジストリに追加します

1. Astra Control Center イメージディレクトリ内のファイルをローカルレジストリに追加します。

   以下の画像の自動ロードについては、サンプルスクリプトを参照してください。

   a. Docker レジストリにログインします。

   ```
   docker login [your_registry_path]
   ```

   b. Docker にイメージをロードする。

   c. 画像にタグを付けます。

   d. [[[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1>[</Z1></Z1></Z1>_image_local_registry_push]]]]]]]]</Z2> ローカルレジストリにイメージをプッシュ

   ```
   export REGISTRY=[your_registry_path]
   for astraImageFile in $(ls images/*.tar)
     # Load to local cache. And store the name of the loaded image
   trimming the 'Loaded images: '
     do astraImage=$(docker load --input ${astraImageFile} | sed
   's/Loaded image: //')
     astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
     # Tag with local image repo.
     docker tag ${astraImage} ${REGISTRY}/${astraImage}
     # Push to the local repo.
     docker push ${REGISTRY}/${astraImage}
   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. Astra Control Center オペレータの配備 YAML (`Astra_control_center_deployment.yaml ') を編集して、ローカルのレジストリと秘密を参照します。

```
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、デフォルト行の「 imagePullSecret:[] 」を次のように置き換えます。

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

   b. 「 kube-rbac プロキシ」イメージの「 [Your_registry_path] 」を、でイメージをプッシュしたレジストリパスに変更します 前の手順。

   c. 「 acc-operator-controller-manager 」イメージの「 [Your_registry_path] 」を、でイメージをプッシュしたレジストリパスに変更します 前の手順。

   d. 「 env 」セクションに次の値を追加します。

```
- 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: []
```

2. 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
```

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 ）（ 'Astra_control_center_min.yaml ）を編集し、 Astra バージョン（ 'Spec' の中の 'astrave) の番号を最新のものに変更します。

```
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. Astra Control Center CR の 'Spec' の中にある 'additionalValues' 内に次の行を追加します

```
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「 ingressType 」を指定し（まだ存在しない場合）、それを「 AccTraefik 」に設定します。

   ```
   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、「 ingressType 」を「 Generic 」に設定します。

   ```
   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。「 externalTraefik 」を「 false 」（デフォルト）に設定すると、外部 Traefik がシステムに存在せず、 Astra Control Center によって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 」は「 false 」に設定されます。

  一方、 Traefik を独自に導入している場合は、「 externalTraefik 」を「 true 」に設定します。この場合 ' 配置を維持して 'Astra Control Center は 'shouldUpgrade' が true' に設定されていない限り 'CRD をアップグレードしません

- **Cert-managor**: デフォルトでは 'externalCertManager` を TRUE に設定しない限り 'Astra Control Center は cert-manager (および CRD) をインストールします’shouldUpgrade' を 'true' に設定すると 'Astra Control Center が 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 または
- externalTraefik: true と shouldUpgrade: true 。

手順

1. 「 acc`cr: 」を編集します。

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

2. 「 externalTraefik 」フィールドと「 shouldUpgrade 」フィールドを必要に応じて「 true 」または「 false 」に変更します。

```
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、「 ingressType 」を「 AccTraefik 」に設定します。

> 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。

- 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、
  - 「 CRD .externalTraefik 」が存在する場合は、「 false 」またはに設定する必要があります
  - 「 CRD.externalTraefik 」が「真」の場合、「 CRD.shoulldUpgrade 」も「真」でなければなりません。
- 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「 secure-testing-cert 」を使用するか、タイプのシークレットを作成します "8a637503539b25b68130b6e8003579d9" に示すように 'NetApp-acc' （またはカスタム名前の）名前空間内の TLS 秘密鍵と証明書の場合 "TLS シークレット"。
2. 非推奨または新しいスキーマのいずれかの入力リソースを NetApp-acc` （またはカスタム名前付き）ネームスペースに配置します。

   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: Ingress
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. Loadbalancer から Traefik が clusterIP に変更されていることを確認します

```
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. 次のコマンドを使用して 'NetApp-acc' ネームスペースを削除します

   ```
   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. 「 acc-monitoring 」エージェントを削除します。

```
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"