



# **Astra Control Center** を使用

## Astra Control Center

NetApp  
August 11, 2025

# 目次

Astra Control Center を使用	1
アプリの管理を開始します	1
アプリケーション管理の要件	1
サポートされているアプリインストール方法	2
クラスタにアプリをインストールします	2
アプリケーションを定義します	2
システムネームスペースについて教えてください。	8
例：リリースごとに保護ポリシーを分ける	9
詳細については、こちらをご覧ください	9
アプリを保護します	9
保護の概要	9
Snapshot とバックアップでアプリケーションを保護	10
[技術プレビュー]クラスタ全体を保護する	22
アプリケーションのリストア	23
SnapMirrorテクノロジーを使用してストレージバックエンド間でアプリケーションをレプリケート	34
アプリケーションのクローン作成と移行	42
アプリケーション実行フックを管理します	45
Astra Control Centerを使用したAstra Control Centerの保護	55
アプリケーションとクラスタの健全性を監視	64
アプリケーションとクラスタの健全性の概要を表示します	64
クラスタの健全性を表示してストレージクラスを管理します	65
アプリの状態と詳細を表示します	66
アカウントを管理します	67
ローカルユーザとロールを管理します	67
リモート認証を管理する	70
リモートユーザとリモートグループを管理します	73
通知を表示および管理します	75
クレデンシャルを追加および削除します	75
アカウントのアクティビティを監視	76
既存のライセンスを更新する	76
バケットを管理する	77
バケットを編集する	78
デフォルトバケットを設定する	78
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	79
バケットを削除する	79
[Tech preview]カスタムリソースを使用したバケットの管理	80
詳細については、こちらをご覧ください	82
ストレージバックエンドを管理します	82
ストレージバックエンドの詳細を表示します	83

ストレージバックエンド認証の詳細を編集します	83
検出されたストレージバックエンドを管理します	84
ストレージバックエンドの管理を解除します	84
ストレージバックエンドを削除します	85
詳細については、こちらをご覧ください	85
実行中のタスクを監視します	85
[技術プレビュー] CRSを使用したAstra Controlアプリケーションの管理	86
Prometheus接続またはFluentd接続でインフラを監視	86
NetApp Support Siteへの接続用プロキシサーバを追加する	86
Prometheusに接続	88
Fluentd に接続します	89
アプリケーションとクラスタの管理を解除します	91
アプリの管理を解除します	91
クラスタの管理を解除します	92
Astra Control Center をアップグレードします	92
Astra Control Centerをダウンロードして展開します	95
ローカルレジストリを使用する場合は、追加の手順を実行します。	96
更新された Astra Control Center オペレータをインストールします	99
Astra Control Center をアップグレードします	101
システムステータスを確認します	103
OpenShift OperatorHubを使用したAstra Control Centerのアップグレード	103
オペレータインストールページへのアクセス	105
既存のオペレータのアンインストール	107
最新のオペレータのインストール	107
Astra Control Center をアップグレードします	108
Astra Control Center をアンインストールします	109
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	111
詳細については、こちらをご覧ください	113

# Astra Control Center を使用

## 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を定義できます。

実行中のポッドを使用してストレージリソースを含むアプリケーション、または実行中のポッドを使用しないストレージリソースを含むアプリケーションを定義および管理できます。ポッドが実行されていないアプリケーションは、データ専用アプリケーションと呼ばれます。

## 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、組み込みのAstra Control Center評価ライセンスまたはフルライセンスのいずれかが必要です。
- **名前空間**：アプリケーションは、Astra Controlを使用して、単一クラスタ上の1つ以上の指定された名前空間内で定義できます。アプリケーションには、同じクラスタ内の複数のネームスペースにまたがるリソースを含めることができます。Astra Controlでは、複数のクラスタ間でアプリケーションを定義する機能はサポートされていません。
- **ストレージクラス**：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インストールし、アプリケーションのクローンを作成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は、元々指定されたストレージクラスが必要です。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。
- **\* 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。

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

## 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。
- オペレータが導入するアプリ : Astra Control は、 ネームスペーススコープの演算子を使用してインストールされたアプリをサポートします。 これらの演算子は、 一般的に「pass-by-reference」アーキテクチャではなく「pass-by-value」アーキテクチャで設計されています。 オペレーターとそれがインストールするアプリは、 同じ名前空間を使用する必要があります。 これを確実にするために、 オペレーターの配置YAMLファイルを変更する必要がある場合があります。

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1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ています。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つ以上の名前空間にスコープされているすべてのアプリケーションを管理できます。

## アプリケーションを定義します

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 [1つ以上のネームスペースにまたがるアプリケーションを管理します](#) または [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](#)。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またはスパンニング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ります。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。

作業を開始する前に

- 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。](#)
- Astra Controlに追加したKubernetesクラスタ上の既存のネームスペース。
- (オプション) すべてののにKubernetesラベルを付けます "[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](#)"。

このタスクについて

- 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のアプリケーションで複数の名前空間を使用する場合は、"[ネームスペースの制約を持つユーザーロールを変更します](#)" 複数の名前空間をサポートするAstra Control Centerバージョンにアップグレードした後。
- 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

- [\[アプリケーションとして管理するリソースを定義します\]](#)
- [\[アプリケーションとして管理するネームスペースを定義します\]](#)
- "[\(テクニカルレビュー\) Kubernetesのカスタムリソースを使用したアプリケーションの定義](#)"

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 [詳細については、例を参照してください](#)。

を展開して、アプリケーション名前空間へのクラスタを対象としたリソースの追加について詳しく説明します。

名前空間リソースに関連付けられているクラスタリソースを、自動的に含まれるアストラコントロールに加えてインポートできます。特定のグループ、種類、バージョンのリソースを含むルールを追加し、必要に応じてラベルを付けることができます。この処理は、Astra Controlに自動的に含まれないリソースがある場合などに実行します。

Astra Controlに自動的に含まれる、クラスタを対象としたリソースを除外することはできません。

以下を追加できます `apiVersions` (APIバージョンと組み合わせたグループ)。

リソースの種類	1回あたりのバージョン (グループ+バージョン)
ClusterRole	rbac.authorization.k8s.io/v1
ClusterRoleBinding	rbac.authorization.k8s.io/v1
CustomResource	apiextensions.k8s.io/v1、apiextensions.k8s.io/v1beta1
CustomResourceDefinition	apiextensions.k8s.io/v1、apiextensions.k8s.io/v1beta1
MutatingWebhookConfiguration	admissionregistration.k8s.io/v1
ValidatingWebhookConfiguration	admissionregistration.k8s.io/v1

## 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、Astra Controlを使用して、単一クラスタ上の1つ以上の指定された名前空間内で定義できます。アプリケーションには、同じクラスタ内の複数の名前空間にまたがるリソースを含めることができます。Astra Controlでは、複数のクラスタ間でアプリケーションを定義する機能はサポートされていません。

5. (オプション) 各名前空間にKubernetesリソース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. (オプション) 「名前空間の追加」を選択し、ドロップダウンリストから名前空間を選択して、アプリケーションの名前空間を追加します。
7. (オプション) 追加するネームスペースのラベルまたはラベルの選択基準を1つ入力します。
8. (オプション) Astra Controlに自動的に含まれるリソースに加えて、クラスタを対象としたリソースを含めるには、\*クラスタを対象とした追加のリソースを含める\*をチェックし、次の手順を実行します。
  - a. 「含めるルールを追加」を選択します。
  - b. グループ：ドロップダウンリストから、リソースのAPIグループを選択します。
  - c. \*kind\*：ドロップダウンリストから'オブジェクトスキーマの名前を選択します
  - d. バージョン：APIのバージョンを入力します。
  - e. ラベルセレクタ：必要に応じて、ルールに追加するラベルを指定します。このラベルは、このラベルに一致するリソースのみを取得するために使用します。ラベルを指定しないと、Astra Controlは、そのクラスタに指定されている種類のリソースのすべてのインスタンスを収集します。
  - f. エントリに基づいて作成されたルールを確認します。
  - g. 「\*追加」を選択します。



クラスタを対象としたリソースルールは必要な数だけ作成できます。[アプリケーションの定義の概要]にルールが表示されます。

9. [\*定義 (Define)] を選択します
10. [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、アプリケーションが表示されます Healthy 「アプリケーション」ページのアプリケーションのリストに表示されます。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。

このアプリに追加されたリソースを表示するには、\*リソース\*タブを選択します。Resource列でリソース名のあとの番号を選択するか、Searchでリソース名を入力して、追加のクラスタを対象としたリソースを確認します。

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。



複数のアプリケーションを定義する場合は、名前空間リストから選択し、左上隅の\*アクション\*ボタンを選択して、\*アプリケーションとして定義\*を選択します。これにより、個々の名前空間に複数のアプリケーションが定義されます。マルチ名前空間アプリケーションについては、を参照してください [\[アプリケーションとして管理するリソースを定義します\]](#)。



[システム名前空間を表示 (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名前空間を表示します。  Show system namespaces ["詳細はこちら"](#)。

このプロセスが完了すると、名前空間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) :

### [テクニカルレビュー] Kubernetesのカスタムリソースを使用したアプリケーションの定義

カスタムリソース (CR) を使用してアプリケーションとして定義することで、Astra Controlで管理するKubernetesリソースを指定できます。たとえば、特定の名前空間内のすべてのリソースを同様の方法で共通の間隔で管理および保護する場合は、それらのリソースを個別に管理するか、または名前空間内のすべてのKubernetesリソースを個別に管理する場合は、クラスター対象のリソースを追加できます。

#### 手順

1. カスタムリソース (CR) ファイルを作成し、という名前を付けます (例: `astra_mysql_app.yaml`)。
2. アプリケーションに名前を付けます。 `metadata.name`。
3. 管理するアプリケーションリソースを定義します。

## spec.includedClusterScopedResources

Astra Controlに自動的に含まれるもののほかに、クラスタを対象としたリソースタイプも含めます。

- \* spec.includedClusterScopedResources\*:\_(オプション)\_含めるクラスタスコープのリソースタイプのリスト。
  - \*groupVersionKind\*:\_(オプション)\_unambiguouslyは種類を識別します。
    - **group**:\_(groupVersionKindが使用されている場合は必須)含めるリソースのAPIグループ。
    - **version**:\_(groupVersionKindが使用されている場合は必須)\_含めるリソースのAPIバージョン。
    - **kind**:\_(groupVersionKindを使用する場合は必須)\_kind含めるリソースの種類。
  - \*labelSelector\*:\_(オプション)\_リソースセットのラベルクエリ。ラベルに一致するリソースのみを取得するために使用されます。ラベルを指定しないと、Astra Controlは、そのクラスタに指定されている種類のリソースのすべてのインスタンスを収集します。matchLabelsとmatchExpressionsの結果はANDで処理されます。
    - **matchLabels**:\_(省略可能)\_{key, value}ペアのマップ。matchLabelsマップ内の1つの{key, value}は、keyフィールドが"key"、演算子が"in"、value"のみを含むvalues配列を持つmatchExpressionsの要素に相当します。要件はANDで処理されます。
    - **matchExpressions**:\_(オプション)\_ラベルセレクタの要件のリスト。要件はANDで処理されます。
      - \*key\*:\_(matchExpressionsを使用する場合は必須)\_ラベルセレクタに関連付けられたラベルキー。
      - 演算子:\_(matchExpressionsが使用されている場合は必須)\_値のセットに対するキーの関係を表します。有効な演算子：In、NotIn、Exists および DoesNotExist。
      - \*values\*:\_(matchExpressionsを使用する場合は必須)\_文字列値の配列。演算子が In または NotIn、values配列must\_not\_be empty。演算子が Exists または `DoesNotExist` values配列は空である必要があります。

## spec.includedNamespaces

アプリケーション内のこれらのリソースに名前空間とリソースを含めます。

- \* spec.includedNamespaces\*:\_(必須)\_リソース選択のための名前空間とオプションのフィルタを定義します。
  - ネームスペース：(必須) Astra Controlで管理するアプリケーションリソースを含むネームスペース。
  - \*labelSelector\*:\_(オプション)\_リソースセットのラベルクエリ。ラベルに一致するリソースのみを取得するために使用されます。ラベルを指定しないと、Astra Controlは、そのクラスタに指定されている種類のリソースのすべてのインスタンスを収集します。matchLabelsとmatchExpressionsの結果はANDで処理されます。
    - **matchLabels**:\_(省略可能)\_{key, value}ペアのマップ。matchLabelsマップ内の1つの{key, value}は、keyフィールドが"key"、演算子が"in"、value"のみを含むvalues配列を持つmatchExpressionsの要素に相当します。要件はANDで処理されます。
    - **matchExpressions**:\_(オプション)\_ラベルセレクタの要件のリスト。key および operator は必須です。要件はANDで処理されます。

- `*key *:_` (matchExpressionsを使用する場合は必須) ラベルセレクタに関連付けられたラベルキー。
- 演算子: `_` (matchExpressionsが使用されている場合は必須) 値のセットに対するキーの関係を表します。有効な演算子: `In`、`NotIn`、`Exists` および `DoesNotExist`。
- `* values *:_` (matchExpressionsを使用する場合は必須) 文字列値の配列。演算子が `In` または `NotIn`、values配列 `must_not_be_empty`。演算子が `Exists` または `DoesNotExist` values配列は空である必要があります。

YAMLの例：

```
apiVersion: astra.netapp.io/v1
kind: Application
metadata:
  name: astra_mysql_app
spec:
  includedNamespaces:
  - namespace: astra_mysql_app
    labelSelector:
      matchLabels:
        app: nginx
        env: production
      matchExpressions:
      - key: tier
        operator: In
        values:
          - frontend
          - backend
```

4. データを入力した後、astra\_mysql\_app.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra_mysql_app.yaml -n astra-connector
```

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。

Show system namespaces



デフォルトでは、管理可能なアプリケーションとしてAstra Control Centerが表示されませんが、別のAstra Control Centerインスタンスを使用してAstra Control Centerインスタンスをバックアップおよびリストアできます。

## 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。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"](#)
- ["アプリの管理を解除します"](#)

## 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"](#)。

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テクノロジー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 Control Center UI またはを使用できます "[Astra Control API](#)" アプリを保護します。

### このタスクについて

- \* Helmでアプリケーションを展開\*：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クラスタのみ）ポリシーの追加：OpenShiftクラスタでアプリケーションをホストするためのプロジェクトを作成すると、プロジェクト（またはKubernetes名前空間）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]
- ONTAP NAS経済性に優れた運用向けのバックアップとリストアを実現
- [変更不可のバックアップの作成]
- 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。保護ポリシーは、Astra Control Web UIまたはカスタムリソース (CR) ファイルを使用して定義できます。

1 時間に 1 回以上の頻度でバックアップや Snapshot を実行する必要がある場合は、次の方法があります "[Astra Control REST API を使用して、スナップショットとバックアップを作成](#)"。



Write Once Read Many (WORM) バケットに書き換え不能なバックアップを作成する保護ポリシーを定義している場合は、バックアップの保持期間がバケットに設定されている保持期間よりも短くないようにしてください。



バックアップとレプリケーションのスケジュールをオフセットして、スケジュールの重複を回避します。たとえば、1時間ごとに1時間の最上部にバックアップを実行し、オフセットを5分、間隔を10分に設定してレプリケーションを開始するようにスケジュールを設定します。

## Web UIを使用して保護ポリシーを設定する

### 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。

バックアップの保持レベルを設定する際に、バックアップを格納するバケットを選択できます。

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。

**Configure protection policy** STEP 1/2: DETAILS

**PROTECTION SCHEDULE**

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

● Hourly ● Daily ● **Weekly** ● Monthly

Select Weekday(s) (optional): Monday X

Time (UTC) (optional): 02:00

Snapshots to keep: 26

Backups to keep: 0

**BACKUP DESTINATION**

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

**OVERVIEW**

Schedule and retention

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

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

Read more in [Protection policies](#)

- Application: cattle-logging
- Namespace: cattle-logging
- Cluster: se-openlab-astra-enterprise-05-se-openlab-astra-enterprise-05-mstr-1

Cancel Review →

5. [技術プレビュー]ストレージバケットのリストから、バックアップまたはスナップショットのデスティネーションバケットを選択します。
6. [\* Review (レビュー) ]を選択します
7. [\* 保護ポリシーの設定 \*] を選択します

## [テクニカルプレビュー] CRを使用した保護ポリシーの設定

### 手順

1. カスタムリソース (CR) ファイルを作成して名前を付けます。astra-control-schedule-

cr.yaml。Astra Control環境、クラスタ構成、データ保護のニーズに合わせて、かっこ<>の値を更新します。

- <CR\_NAME>：このカスタムリソースの名前。環境に適した一意の適切な名前を選択します。
- <APPLICATION\_NAME>：バックアップするアプリケーションのKubernetes名。
- <APPVAULT\_NAME>：バックアップコンテンツを格納するAppVaultの名前。
- <BACKUPS\_RETAINED>：保持するバックアップの数。ゼロは、バックアップを作成しないことを示します。
- <SNAPSHOTS\_RETAINED>：保持するSnapshotの数。ゼロは、スナップショットを作成しないことを示します。
- <GRANULARITY>：スケジュールを実行する頻度。指定可能な値と必須の関連フィールドは次のとおりです。
  - hourly (次を指定する必要があります： spec.minute)
  - daily (次を指定する必要があります： spec.minute および spec.hour)
  - weekly (次を指定する必要があります： spec.minute、spec.hour`および`spec.dayOfWeek)
  - monthly (次を指定する必要があります： spec.minute、spec.hour`および`spec.dayOfMonth)
- <DAY\_OF\_MONTH>：\_ (オプション) \_スケジュールを実行する日にち (1~31)。このフィールドは、粒度が次の値に設定されている場合は必須です。monthly。
- <DAY\_OF\_WEEK>：\_ (オプション) \_スケジュールを実行する曜日 (0~7)。0または7の値は日曜日を示します。このフィールドは、粒度が次の値に設定されている場合は必須です。weekly。
- <HOUR\_OF\_DAY>：\_ (オプション) \_スケジュールを実行する時間 (0~23)。このフィールドは、粒度が次の値に設定されている場合は必須です。daily、weekly`または`monthly。
- <MINUTE\_OF\_HOUR>：\_ (オプション) \_スケジュールを実行する分 (0~59)。このフィールドは、粒度が次の値に設定されている場合は必須です。hourly、daily、weekly`または`monthly。



```
apiVersion: astra.netapp.io/v1
kind: Schedule
metadata:
  namespace: astra-connector
  name: <CR_NAME>
spec:
  applicationRef: <APPLICATION_NAME>
  appVaultRef: <APPVAULT_NAME>
  backupRetention: "<BACKUPS_RETAINED>"
  snapshotRetention: "<SNAPSHOTS_RETAINED>"
  granularity: <GRANULARITY>
  dayOfMonth: "<DAY_OF_MONTH>"
  dayOfWeek: "<DAY_OF_WEEK>"
  hour: "<HOUR_OF_DAY>"
  minute: "<MINUTE_OF_HOUR>"
```

2. データを入力した後、astra-control-schedule-cr.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra-control-schedule-cr.yaml
```

## 結果

Astra Control 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。

このタスクについて

Astra Controlでは、次のドライバでサポートされるストレージクラスを使用したSnapshotの作成がサポートされます。

- ontap-nas
- ontap-san
- ontap-san-economy



アプリケーションがサポートされるストレージクラスを使用している場合 ontap-nas-economy ドライバ、スナップショットを作成できません。スナップショットには代替のストレージクラスを使用します。

## Web UIを使用したSnapshotの作成

### 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、\* 次へ \* を選択します。
4. [技術プレビュー]ストレージバケットのリストからスナップショットのデスティネーションバケットを選択します。
5. Snapshot の概要を確認し、「\* Snapshot \*」を選択します。

## [テクニカルプレビュー] CRを使用したスナップショットの作成

### 手順

1. カスタムリソース (CR) ファイルを作成して名前を付けます。astra-control-snapshot-cr.yaml。カッコ内の値を、Astra Controlの環境とクラスタの構成に合わせて更新します。
  - <CR\_NAME>: このカスタムリソースの名前。環境に適した一意の適切な名前を選択します。
  - <APPLICATION\_NAME>: Snapshotを作成するアプリケーションのKubernetes名。
  - <APPVAULT\_NAME>: スナップショットの内容を格納するAppVaultの名前。
  - <RECLAIM\_POLICY>: \_ (オプション) \_スナップショットCRが削除されたときのスナップショットの処理を定義します。有効なオプション:
    - Retain
    - Delete (デフォルト)

```
apiVersion: astra.netapp.io/v1
kind: Snapshot
metadata:
  namespace: astra-connector
  name: <CR_NAME>
spec:
  applicationRef: <APPLICATION_NAME>
  appVaultRef: <APPVAULT_NAME>
  reclaimPolicy: <RECLAIM_POLICY>
```

2. データを入力した後、astra-control-snapshot-cr.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra-control-snapshot-cr.yaml
```

## 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\* Healthy である場合に成功し

まず (Data protection > Snapshots ページの State \*列)

バックアップを作成します

アプリはいつでもバックアップできます。

このタスクについて

Astra Controlのバケットで使用可能な容量が報告されません。Astra Controlで管理されるアプリケーションをバックアップまたはクローニングする前に、該当するストレージ管理システムでバケット情報を確認してください。

アプリケーションがサポートされるストレージクラスを使用している場合 `ontap-nas-economy` 運転手、あなたがする必要があります [バックアップとリストアの有効化](#) 機能性：次を定義したことを確認してください： `backendType` のパラメータ "[Kubernetesストレージオブジェクト](#)" を使用します `ontap-nas-economy` 保護処理を実行する前に

Astra Controlでは、次のドライバでサポートされるストレージクラスを使用したバックアップの作成がサポートされます。



- `ontap-nas`
- `ontap-nas-economy`
- `ontap-san`
- `ontap-san-economy`

## Web UIを使用したバックアップの作成

### 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. [技術プレビュー]ストレージバケットのリストからバックアップ先のバケットを選択します。
6. 「\* 次へ \*」を選択します。
7. バックアップの概要を確認し、「バックアップ」を選択します。

## [テクニカルプレビュー] CRを使用したバックアップの作成

### 手順

1. カスタムリソース (CR) ファイルを作成して名前を付けます。astra-control-backup-cr.yaml。カッコ内の値を、Astra Controlの環境とクラスタの構成に合わせて更新します。
  - <CR\_NAME>：このカスタムリソースの名前。環境に適した一意の適切な名前を選択します。
  - <APPLICATION\_NAME>：バックアップするアプリケーションのKubernetes名。
  - <APPVAULT\_NAME>：バックアップコンテンツを格納するAppVaultの名前。

```
apiVersion: astra.netapp.io/v1
kind: Backup
metadata:
  namespace: astra-connector
  name: <CR_NAME>
spec:
  applicationRef: <APPLICATION_NAME>
  appVaultRef: <APPVAULT_NAME>
```

2. データを入力した後、astra-control-backup-cr.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra-control-backup-cr.yaml
```

### 結果

Astra Control：アプリケーションのバックアップを作成



- 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キャンセルする必要がある場合は、 の手順に従ってください [\[バックアップをキャンセルします\]](#)。バックアップを削除するには、完了するまで待ってから、 の手順を実行します [\[バックアップを削除します\]](#)。
- 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。

## ONTAP NAS 経済性に優れた運用向けのバックアップとリストアを実現

Astra Control Provisioner は、バックアップとリストアの機能を提供します。この機能は、ontap-nas-economy ストレージクラス。

作業を開始する前に

- これで完了です ["Astra Control Provisioner を有効にしました"](#)。
- Astra Control でアプリケーションを定義しておきます。この手順を完了するまで、このアプリケーションの保護機能は制限されます。
- これで完了です ontap-nas-economy ストレージバックエンドのデフォルトのストレージクラスとして選択されています。

手順

1. ONTAP ストレージバックエンドで次の手順を実行します。

- a. をホストしている SVM を検索します。ontap-nas-economy-アプリケーションのボリュームベース。
- b. ボリュームを作成する ONTAP に接続されている端末にログインします。
- c. SVM の Snapshot ディレクトリを非表示にします。



この変更は SVM 全体に影響します。非表示のディレクトリには引き続きアクセスできません。

```
nfs modify -vserver <svm name> -v3-hide-snapshot enabled
```

+



ONTAP ストレージバックエンドの snapshot ディレクトリが非表示になっていることを確認します。このディレクトリを非表示にしないと、アプリケーション（特に NFSv3 を使用している場合）へのアクセスが失われる可能性があります。

2. Astra Control Provisioner で次の手順を実行します。

- a. 次の PV ごとに Snapshot ディレクトリを有効にします。ontap-nas-economy ベースで、アプリケーションに関連付けられています。

```
tridentctl update volume <pv name> --snapshot-dir=true --pool-level
=true -n trident
```

- b. 関連付けられている各PVに対してSnapshotディレクトリが有効になっていることを確認します。

```
tridentctl get volume <pv name> -n trident -o yaml | grep snapshotDir
```

対応：

```
snapshotDirectory: "true"
```

3. Astra Controlで、関連付けられているSnapshotディレクトリをすべて有効にしたあとにアプリケーションを更新し、Astra Controlが変更された値を認識するようにします。

結果

Astra Controlを使用して、アプリケーションのバックアップとリストアを実行できるようになります。各PVCは、他のアプリケーションでバックアップおよびリストアに使用することもできます。

#### 変更不可のバックアップの作成

変更不可のバックアップは、バックアップを格納するバケットの保持ポリシーで禁止されているかぎり、変更、削除、上書きすることはできません。保持ポリシーが設定されたバケットにアプリケーションをバックアップすることで、変更不可のバックアップを作成できます。を参照してください ["データ保護"](#) を参照してください。

作業を開始する前に

保持ポリシーを使用してデスティネーションバケットを設定する必要があります。その方法は、使用するストレージプロバイダによって異なります。詳細については、ストレージプロバイダのドキュメントを参照してください。

- \* Amazon Web Services \* : "バケットの作成時にS3オブジェクトロックを有効にし、デフォルトの保持モードを「governance」にデフォルトの保持期間を設定する"。
- \* NetApp StorageGRID \* : "バケットの作成時にS3オブジェクトロックを有効にし、デフォルトの保持モードを「compliance」にデフォルトの保持期間を設定する"。



Astra Controlのバケットで使用可能な容量が報告されません。Astra Controlで管理されるアプリケーションをバックアップまたはクローニングする前に、該当するストレージ管理システムでバケット情報を確認してください。



アプリケーションがサポートされるストレージクラスを使用している場合 `ontap-nas-economy` ドライバ。を定義していることを確認してください `backendType` のパラメータ ["Kubernetesストレージオブジェクト"](#) を使用します `ontap-nas-economy` 保護処理を実行する前に

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、バックアップのデスティネーションバケットを選択します。Write Once Read Many (WORM) バケット名の横にステータスが「Locked」と表示されます。



バケットのタイプがサポートされていない場合は、バケットにカーソルを合わせるか選択すると表示されます。

6. 「\* 次へ\*」を選択します。
7. バックアップの概要を確認し、「バックアップ」を選択します。

## 結果

Astra Controlがアプリケーションの変更不可のバックアップを作成



- 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。
- 同じアプリケーションの書き換え不能な2つのバックアップを同じバケットに同時に作成しようとする、Astra Controlによって2つ目のバックアップが開始されなくなります。最初のバックアップが完了してから、別のバックアップを開始してください。
- 実行中の変更不可のバックアップはキャンセルできません。
- 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。



変更不可のバックアップのステータスは、使用しているバケットの横に「Locked」と表示されます。

## 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。



現在レプリケート中のSnapshotは削除できません。

### 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 \*]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 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、バックアップが実行されている必要があります Running 状態。にあるバックアップはキャンセルできません Pending 状態。



実行中の変更不可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. 処理を確認するために「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。バケットの保持ポリシーで変更不可のバケットに作成されたバックアップは削除できません。



保持期間が終了する前に変更不可のバックアップを削除することはできません。



実行中のバックアップをキャンセルする必要がある場合は、の手順に従ってください [\[バックアップをキャンセルします\]](#)。バックアップを削除するには、完了するまで待ってから、次の手順を実行します。

### 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. 「\* 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 がバックアップを削除する。

## [技術レビュー]クラスタ全体を保護する

クラスタ上の管理対象外のネームスペースの一部またはすべてについて、スケジュールされた自動バックアップを作成できます。これらのワークフローは、NetAppによってKubernetesサービスアカウント、ロールバインド、およびcronジョブとして提供され、Pythonスクリプトを使用してオーケストレーションされます。

### 動作の仕組み

フルクラスタバックアップワークフローを設定してインストールすると、cronジョブが定期的に行われ、まだ管理されていないネームスペースが保護され、インストール時に選択したスケジュールに基づいて保護ポリシーが自動的に作成されます。

フルクラスタバックアップワークフローでクラスタ上のすべての管理対象外のネームスペースを保護する必要がある場合は、ラベルベースのバックアップワークフローを使用できます。ラベルベースのバックアップのワークフローでもcronタスクを使用しますが、管理対象外のネームスペースをすべて保護する代わりに、指定したラベルでネームスペースを識別して、Bronze、Silver、またはGoldのバックアップポリシーに基づいてネームスペースを保護することもできます。

選択したワークフローの範囲に含まれる新しい名前空間が作成されると、管理者の操作なしで自動的に保護されます。これらのワークフローはクラスタ単位で実装されるため、クラスタの重要度に応じて、それぞれのクラスタで独自の保護レベルを持つワークフローを使用できます。

### 例：完全なクラスタ保護

たとえば、フルクラスタバックアップワークフローを構成してインストールすると、任意のネームスペース内のすべてのアプリケーションが定期的な管理され、管理者による追加の作業なしに保護されます。ワークフローのインストール時に名前空間が存在している必要はありません。将来追加された名前空間は保護されます。

### 例：ラベルベースの保護

詳細については、ラベルベースのワークフローを使用できます。たとえば、このワークフローをインストールし、必要な保護レベルに応じて、保護する名前空間に複数のラベルのいずれかを適用するようにユーザーに指示できます。これにより、ユーザーはこれらのラベルのいずれかを使用して名前空間を作成でき、管理者に通知する必要はありません。新しいネームスペースとその中のすべてのアプリは自動的に保護されます。

### すべてのネームスペースのスケジュールされたバックアップを作成する

フルクラスタバックアップワークフローを使用して、クラスタ上のすべてのネームスペースのスケジュールされたバックアップを作成できます。

## 手順

1. クラスタにネットワークでアクセスできるマシンに、次のファイルをダウンロードします。
  - ["コンポーネント.yaml CRDファイル"](#)
  - ["protectCluster.py Pythonスクリプト"](#)
2. ツールキットを設定してインストールするには、次の手順に従います。 ["付属の手順に従います。"](#)。

## 特定のネームスペースのスケジュールされたバックアップを作成する

ラベルベースのバックアップワークフローを使用して、ラベル別に特定のネームスペースのスケジュールされたバックアップを作成できます。

## 手順

1. クラスタにネットワークでアクセスできるマシンに、次のファイルをダウンロードします。
  - ["コンポーネント.yaml CRDファイル"](#)
  - ["protectCluster.py Pythonスクリプト"](#)
2. ツールキットを設定してインストールするには、次の手順に従います。 ["付属の手順に従います。"](#)。

## 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 の略"](#) アプリを復元するには、

### 作業を開始する前に

- 最初にアプリケーションを保護する:アプリケーションを復元する前に、アプリケーションのスナップショットまたはバックアップを作成することを強くお勧めします。これにより、リストアに失敗した場合に、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できます。
- デスティネーションボリュームの確認:別のストレージクラスにリストアする場合は、ストレージクラスで同じ永続ボリュームアクセスモード (ReadWriteManyなど) が使用され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。たとえば、ソースの永続ボリュームがRWXアクセスモードを使用している場合は、Azure Managed Disks、AWS EBS、Google Persistent Disk、など、RWXを提供できないデスティネーションストレージクラスを選択します `ontap-san` を指定すると、リストア処理は失敗します。原因は失敗します。永続ボリュームのアクセスモードの詳細については、を参照してください ["Kubernetes"](#) ドキュメント
- 必要なスペースを確保するための計画: NetApp ONTAP ストレージを使用するアプリケーションのインプレースリストアを実行すると、リストアしたアプリケーションで使用されるスペースが2倍になることがあります。In Placeリストアを実行したあとに、リストアしたアプリケーションから不要なSnapshotを削除して、ストレージスペースを解放します。
- (Red Hat OpenShiftクラスタのみ) ポリシーの追加: 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名前空間) 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
```

- サポートされるストレージクラスドライバ：Astra Controlでは、次のドライバに基づくストレージクラスを使用したバックアップのリストアがサポートされます。
  - `ontap-nas`
  - `ontap-nas-economy`
  - `ontap-san`
  - `ontap-san-economy`
- (**ontap-nas-economy**ドライバのみ) バックアップとリストア： `ontap-nas-economy` ドライバを使用して、"**ONTAPストレージバックエンドのSnapshotディレクトリが非表示になっている**"。このディレクトリを非表示にしないと、アプリケーション（特にNFSv3を使用している場合）へのアクセスが失われる可能性があります。
- \* Helmデプロイ済みアプリ\*：Helm 3でデプロイされたアプリ（またはHelm 2からHelm 3にアップグレードされたアプリ）は完全にサポートされます。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。



リソースを共有するアプリケーションでIn Placeリストア処理を実行すると、予期しない結果が生じる可能性があります。アプリケーション間で共有されているリソースは、いずれかのアプリケーションでインプレースリストアが実行されると置き換えられます。詳細については、[を参照してください この例です](#)。

リストアするアーカイブのタイプに応じて、次の手順を実行します。

**Web UI**を使用してバックアップまたは**Snapshot**からデータをリストア

Astra Control Web UIを使用してデータをリストアできます。

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. [オプション]メニューの[操作]列で、\*[リストア]\*を選択します。
3. リストアタイプを選択します。
  - 元のネームスペースにリストア：この手順 を使用して、アプリケーションを元のクラスタにインプレースでリストアします。



アプリケーションがサポートされるストレージクラスを使用している場合 `ontap-nas-economy` ドライバ。元のストレージクラスを使用してアプリケーションをリストアする必要があります。アプリケーションを同じネームスペースにリストアする場合、別のストレージクラスを指定することはできません。

- i. アプリをインプレースで復元するために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。これにより、アプリは以前のバージョンに戻ります。
- ii. 「\* 次へ \*」を選択します。



以前に削除した名前スペースにリストアすると、同じ名前の新しい名前スペースがリストアプロセスで作成されます。以前に削除した名前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、新しく作成した名前スペースに手動で権限を復元する必要があります。

- 新しい名前空間に復元：この手順を使用して、アプリを別のクラスタまたはソースとは異なる名前空間で別のクラスタに復元します。

- i. 復元されたアプリの名前を指定します。
- ii. リストアする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。
- iii. アプリケーションに関連付けられている各ソース名前スペースのデスティネーション名前スペースを入力します。



Astra Controlは、このリストアオプションの一部として新しいデスティネーション名前スペースを作成します。指定するデスティネーション名前スペースがデスティネーションクラスタに存在していないことを確認してください。

- iv. 「\*次へ\*」を選択します。
- v. アプリの復元に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。
- vi. 「\*次へ\*」を選択します。
- vii. 次のいずれかを選択します。
  - 元のストレージクラスを使用してリストア：ターゲットクラスタに存在しない場合を除き、元々関連付けられていたストレージクラスがアプリケーションで使用されます。この場合、クラスタのデフォルトのストレージクラスが使用されます。
  - 別のストレージクラスを使用したリストア：ターゲットクラスタに存在するストレージクラスを選択してください。元々関連付けられていたストレージクラスに関係なく、すべてのアプリケーションボリュームが、リストアの一環としてこの別のストレージクラスに移動されず。
- viii. 「\*次へ\*」を選択します。

#### 4. フィルタするリソースを選択：

- すべてのリソースを復元：元のアプリケーションに関連付けられているすべてのリソースを復元します。
- リソースのフィルタ:元のアプリケーションリソースのサブセットを復元するルールを指定します。
  - i. リストアされたアプリケーションにリソースを含めるか除外するかを選択します。
  - ii. または[除外ルールを追加]\*のいずれかを選択し、アプリケーションのリストア時に正しいリソースをフィルタするようにルールを設定します。設定が正しくなるまで、ルールを編集したり削除したり、ルールを再度作成したりすることができます。



includeルールとexcludeルールの設定については、を参照してください [\[アプリケーションのリストア中にリソースをフィルタリングします\]](#)。

5. 「\*次へ\*」を選択します。
6. リストア処理の詳細をよく確認し、プロンプトが表示されたら「restore」と入力して\*[リストア]\*を選択します。

## 【テクニカルプレビュー】カスタムリソース（CR）を使用したバックアップからのリストア

カスタムリソース（CR）ファイルを使用して、別のネームスペースまたは元のソースネームスペースにバックアップからデータをリストアできます。

## CRを使用したバックアップからのリストア

### 手順

1. カスタムリソース (CR) ファイルを作成して名前を付けます。astra-control-backup-restore-cr.yaml。カッコ内の値を、Astra Controlの環境とクラスタの構成に合わせて更新します。
  - <CR\_NAME>：このCR操作の名前。環境に適した適切な名前を選択します。
  - <APPVAULT\_NAME>：バックアップコンテンツが格納されているAppVaultの名前。
  - <BACKUP\_PATH>：バックアップコンテンツが格納されているAppVault内のパス。例：

```
ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-20231213023800_94347756-9d9b-401d-a0c3
```

- <SOURCE\_NAMESPACE>：リストア処理のソースネームスペース。
- <DESTINATION\_NAMESPACE>：リストア処理のデスティネーションネームスペース。

```
apiVersion: astra.netapp.io/v1
kind: BackupRestore
metadata:
  name: <CR_NAME>
  namespace: astra-connector
spec:
  appVaultRef: <APPVAULT_NAME>
  appArchivePath: <BACKUP_PATH>
  namespaceMapping: [{"source": "<SOURCE_NAMESPACE>",
"destination": "<DESTINATION_NAMESPACE>"}]
```

2. (オプション) リストアするアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルが付いたリソースを含めるか除外するフィルタリングを追加します。
  - 「<INCLUDE-EXCLUDE>」：\_ (フィルタリングに必要) \_使用 include または exclude resourceMatchersで定義されているリソースを含めるか除外します。次のresourceMatchersパラメータを追加して、追加または除外するリソースを定義します。
    - <GROUP>：\_ (オプション) \_フィルタリングするリソースのグループ。
    - <KIND>：\_ (オプション) \_フィルタリングするリソースの種類。
    - <VERSION>：\_ (オプション) \_フィルタリングするリソースのバージョン。
    - <NAMES>：(オプション) namesをフィルタリングするリソースのKubernetes metadata.nameフィールドに入力します。
    - <NAMESPACES>：\_ (オプション) \_NamespacesフィルタリングするリソースのKubernetes metadata.nameフィールド。
    - <SELECTORS>：\_ (オプション) \_で定義されているリソースのKubernetes metadata.nameフィールドのラベルセクタ文字列 "Kubernetes のドキュメント"。例

```
"trident.netapp.io/os=linux".
```

例

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "<INCLUDE-EXCLUDE>"
    resourceMatchers:
      group: <GROUP>
      kind: <KIND>
      version: <VERSION>
      names: <NAMES>
      namespaces: <NAMESPACES>
      labelSelectors: <SELECTORS>
```

3. データを入力した後、astra-control-backup-restore-cr.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra-control-backup-restore-cr.yaml
```

**CR**を使用したバックアップから元のネームスペースへのリストア

手順

1. カスタムリソース (CR) ファイルを作成して名前を付けます。astra-control-backup-ipr-cr.yaml。カッコ内の値を、Astra Controlの環境とクラスタの構成に合わせて更新します。
  - <CR\_NAME>: このCR操作の名前。環境に適した適切な名前を選択します。
  - <APPVAULT\_NAME>: バックアップコンテンツが格納されているAppVaultの名前。
  - <BACKUP\_PATH>: バックアップコンテンツが格納されているAppVault内のパス。例:

```
ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
20231213023800_94347756-9d9b-401d-a0c3
```

```
apiVersion: astra.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  name: <CR_NAME>
  namespace: astra-connector
spec:
  appVaultRef: <APPVAULT_NAME>
  appArchivePath: <BACKUP_PATH>
```

2. (オプション) リストアするアプリケーションの特定のリソースのみを選択する必要がある場合は、

特定のラベルが付いたリソースを含めるか除外するフィルタリングを追加します。

- 「<INCLUDE-EXCLUDE>」：\_ (フィルタリングに必要) \_使用 include または exclude resourceMatchersで定義されているリソースを含めるか除外します。次のresourceMatchersパラメータを追加して、追加または除外するリソースを定義します。
  - <GROUP>：\_ (オプション) \_フィルタリングするリソースのグループ。
  - <KIND>：\_ (オプション) \_フィルタリングするリソースの種類。
  - <VERSION>：\_ (オプション) \_フィルタリングするリソースのバージョン。
  - <NAMES>：(オプション) namesをフィルタリングするリソースのKubernetes metadata.nameフィールドに入力します。
  - <NAMESPACES>：\_ (オプション) \_NamespacesフィルタリングするリソースのKubernetes metadata.nameフィールド。
  - <SELECTORS>：\_ (オプション) \_で定義されているリソースのKubernetes metadata.nameフィールドのラベルセクタ文字列 "[Kubernetes のドキュメント](#)"。例 "trident.netapp.io/os=linux"。

例

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "<INCLUDE-EXCLUDE>"
    resourceMatchers:
      group: <GROUP>
      kind: <KIND>
      version: <VERSION>
      names: <NAMES>
      namespaces: <NAMESPACES>
      labelSelectors: <SELECTORS>
```

3. データを入力した後、astra-control-backup-ipr-cr.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra-control-backup-ipr-cr.yaml
```

### [テクニカルプレビュー]カスタムリソースを使用したSnapshotからのリストア (CR)

カスタムリソース (CR) ファイルを使用して、スナップショットから別の名前スペースまたは元のソース名前スペースにデータをリストアできます。



## CRを使用したSnapshotからのリストア

### 手順

1. カスタムリソース (CR) ファイルを作成して名前を付けます。astra-control-snapshot-restore-cr.yaml。カッコ内の値を、Astra Controlの環境とクラスタの構成に合わせて更新します。
  - <CR\_NAME>：このCR操作の名前。環境に適した適切な名前を選択します。
  - <APPVAULT\_NAME>：バックアップコンテンツが格納されているAppVaultの名前。
  - <BACKUP\_PATH>：バックアップコンテンツが格納されているAppVault内のパス。例：

```
ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-20231213023800_94347756-9d9b-401d-a0c3
```

- <SOURCE\_NAMESPACE>：リストア処理のソースネームスペース。
- <DESTINATION\_NAMESPACE>：リストア処理のデスティネーションネームスペース。

```
apiVersion: astra.netapp.io/v1
kind: SnapshotRestore
metadata:
  name: <CR_NAME>
  namespace: astra-connector
spec:
  appArchivePath: <BACKUP_PATH>
  appVaultRef: <APPVAULT_NAME>
  namespaceMapping: [{"source": "<SOURCE_NAMESPACE>",
"destination": "<DESTINATION_NAMESPACE>"}]
```

2. (オプション) リストアするアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルが付いたリソースを含めるか除外するフィルタリングを追加します。
  - 「<INCLUDE-EXCLUDE>」：\_ (フィルタリングに必要) \_使用 include または exclude resourceMatchersで定義されているリソースを含めるか除外します。次のresourceMatchersパラメータを追加して、追加または除外するリソースを定義します。
    - <GROUP>：\_ (オプション) \_フィルタリングするリソースのグループ。
    - <KIND>：\_ (オプション) \_フィルタリングするリソースの種類。
    - <VERSION>：\_ (オプション) \_フィルタリングするリソースのバージョン。
    - <NAMES>：(オプション) namesをフィルタリングするリソースのKubernetes metadata.nameフィールドに入力します。
    - <NAMESPACES>：\_ (オプション) \_NamespacesフィルタリングするリソースのKubernetes metadata.nameフィールド。
    - <SELECTORS>：\_ (オプション) \_で定義されているリソースのKubernetes metadata.nameフィールドのラベルセクタ文字列 "Kubernetes のドキュメント"。例

```
"trident.netapp.io/os=linux".
```

例

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "<INCLUDE-EXCLUDE>"
    resourceMatchers:
      group: <GROUP>
      kind: <KIND>
      version: <VERSION>
      names: <NAMES>
      namespaces: <NAMESPACES>
      labelSelectors: <SELECTORS>
```

3. データを入力した後、astra-control-snapshot-restore-cr.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra-control-snapshot-restore-cr.yaml
```

## CRを使用したSnapshotから元のネームスペースへのリストア

手順

1. カスタムリソース (CR) ファイルを作成して名前を付けます。astra-control-snapshot-ipr-cr.yaml。カッコ内の値を、Astra Controlの環境とクラスタの構成に合わせて更新します。
  - <CR\_NAME>: このCR操作の名前。環境に適した適切な名前を選択します。
  - <APPVAULT\_NAME>: バックアップコンテンツが格納されているAppVaultの名前。
  - <BACKUP\_PATH>: バックアップコンテンツが格納されているAppVault内のパス。例:

```
ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
20231213023800_94347756-9d9b-401d-a0c3
```

```
apiVersion: astra.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  name: <CR_NAME>
  namespace: astra-connector
spec:
  appArchivePath: <BACKUP_PATH>
  appVaultRef: <APPVAULT_NAME>
```

2. (オプション) リストアするアプリケーションの特定のリソースのみを選択する必要がある場合は、

特定のラベルが付いたリソースを含めるか除外するフィルタリングを追加します。

- 「<INCLUDE-EXCLUDE>」：\_ (フィルタリングに必要) \_使用 include または exclude resourceMatchersで定義されているリソースを含めるか除外します。次のresourceMatchersパラメータを追加して、追加または除外するリソースを定義します。
  - <GROUP>：\_ (オプション) \_フィルタリングするリソースのグループ。
  - <KIND>：\_ (オプション) \_フィルタリングするリソースの種類。
  - <VERSION>：\_ (オプション) \_フィルタリングするリソースのバージョン。
  - <NAMES>：(オプション) namesをフィルタリングするリソースのKubernetes metadata.nameフィールドに入力します。
  - <NAMESPACES>：\_ (オプション) \_NamespacesフィルタリングするリソースのKubernetes metadata.nameフィールド。
  - <SELECTORS>：\_ (オプション) \_で定義されているリソースのKubernetes metadata.nameフィールドのラベルセレクタ文字列 "Kubernetes のドキュメント"。例 "trident.netapp.io/os=linux"。

例

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "<INCLUDE-EXCLUDE>"
    resourceMatchers:
      group: <GROUP>
      kind: <KIND>
      version: <VERSION>
      names: <NAMES>
      namespaces: <NAMESPACES>
      labelSelectors: <SELECTORS>
```

3. データを入力した後、astra-control-snapshot-ipr-cr.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra-control-snapshot-ipr-cr.yaml
```

結果

Astra Control 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コンテンツが、リストアしたアプリケーションの永続ボリュームのコンテンツ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、またはリストア）が完了して永続ボリュームのサイズを変更したあと、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。



ネームスペースの名前/ IDまたはネームスペースのラベルでネームスペースの制約を受けているメンバーユーザは、同じクラスタの新しいネームスペース、または組織のアカウントに含まれる他のクラスタ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。クローン処理またはリストア処理で新しいネームスペースが作成されたあと、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、影響を受けるユーザのロールの制約を更新して、新しいネームスペースへのアクセスを許可できます。

アプリケーションのリストア中にリソースをフィルタリングします

にフィルタルールを追加できます "**リストア**" リストアされたアプリケーションに含める、またはリストアされたアプリケーションから除外する既存のアプリケーションリソースを指定する処理。指定した名前空間、ラベル、またはGVK (GroupVersionKind) に基づいて、リソースを含めたり除外したりできます。

[**Include** (含める)] および [**Exclude** (除外)] のシナリオ

- 元のネームスペースを使用する包含ルールを選択した場合 (インプレースリストア) : ルールで定義した既存のアプリケーションリソースは削除され、リストアに使用する選択したSnapshotまたはバックアップのリソースで置き換えられます。includeルールで指定しないリソースは変更されません。
- 新しい名前空間を持つ**include**ルールを選択した場合: このルールを使用して、リストアされたアプリケーションで使用する特定のリソースを選択します。対象ルールに指定しないリソースは、リストアされたアプリケーションには含まれません。
- 元のネームスペースを含む除外ルールを選択した場合 (インプレースリストア) : 除外するように指定したリソースはリストアされず、変更されません。除外するように指定しないリソースは、スナップショットまたはバックアップからリストアされます。対応するStatefulSetがフィルタリングされたリソースに含まれている場合、永続ボリューム上のすべてのデータが削除されて再作成されます。
- 新しい名前空間を持つ除外ルールを選択した場合: このルールを使用して、リストアされたアプリケーションから削除する特定のリソースを選択します。除外するように指定しないリソースは、スナップショットまたはバックアップからリストアされます。

ルールには、includeまたはexcludeタイプがあります。リソースの包含と除外を組み合わせたルールは使用できません。

手順

1. リソースをフィルタするように選択し、[アプリケーションのリストア]ウィザードで[含める]または[除外するルールを追加する]を選択したら、\*[除外するルールを追加する]\*を選択します。



Astra Controlで自動的に追加されるクラスタ対象のリソースを除外することはできません。

2. フィルタルールを設定します。



ネームスペース、ラベル、またはGVKを少なくとも1つ指定する必要があります。フィルタルールを適用したあとに保持するリソースがあれば、リストアしたアプリケーションを正常な状態に保つのに十分であることを確認してください。

- a. ルールの特定のネームスペースを選択します。選択しない場合は、すべての名前空間がフィルタで使用されます。



アプリケーションに複数の名前空間が含まれていた場合、新しい名前空間にリストアすると、リソースが含まれていなくてもすべての名前空間が作成されます。

- b. (オプション) リソース名を入力します。
- c. (任意) ラベルセクタ：を含めます **"ラベルセクタ"** をクリックしてルールに追加します。ラベルセクタは、選択したラベルに一致するリソースのみをフィルタリングするために使用されます。
- d. (オプション) [Use GVK (GroupVersionKind) set]を選択してリソースをフィルタリング\*し、追加のフィルタリングオプションを指定します。



GVKフィルタを使用する場合は、バージョンと種類を指定する必要があります。

- i. (オプション) \* Group \*：ドロップダウンリストからKubernetes APIグループを選択します。
  - ii. 種類：ドロップダウンリストから、フィルタで使用するKubernetesリソースタイプのオブジェクトスキーマを選択します。
  - iii. バージョン：Kubernetes APIのバージョンを選択します。
3. エントリに基づいて作成されたルールを確認します。
  4. 「\* 追加」を選択します。



ルールを含むリソースと除外するリソースは必要なだけ作成できます。処理を開始する前に、リストアアプリケーションの概要にルールが表示されます。

リソースを別のアプリケーションと共有するアプリケーションでは、インプレースリストアが複雑になります。リソースを別のアプリケーションと共有し、意図しない結果を生成するアプリケーションに対して、インプレースリストア処理を実行できます。アプリケーション間で共有されているリソースは、いずれかのアプリケーションでインプレースリストアが実行されると置き換えられます。

次に、NetApp SnapMirrorレプリケーションを使用してリストアすると望ましくない状況が発生するシナリオの例を示します。

1. アプリケーションを定義します app1 名前空間を使用する ns1。
2. のレプリケーション関係を設定します app1。
3. アプリケーションを定義します app2 (同じクラスタ上) 名前空間を使用します ns1 および ns2。
4. のレプリケーション関係を設定します app2。
5. のレプリケーションを反転した app2。これにより、が起動します app1 非アクティブ化するソースクラスタ上のアプリケーション。

## SnapMirrorテクノロジーを使用してストレージバックエンド間でアプリケーションをレプリケート

Astra Controlを使用すると、NetApp SnapMirrorテクノロジー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、ストレージバックエンド間、同じクラスタ上、または異なるクラスタ間でデータやアプリケーションの変更を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Aへ
- 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でオンプレミスからクラウドへ
- Cloud Volumes ONTAPを使用したクラウドからオンプレミスへの移行
- 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。



別のアプリケーションを逆方向に同時に複製できます。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

Astra Controlによる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。

#### ONTAP クラスタ

- \* Astra Control ProvisionerまたはAstra Trident \* : Astra Control ProvisionerまたはAstra Tridentは、ONTAPをバックエンドとして利用するソースとデスティネーションの両方のKubernetesクラスタに存在する必要があります。Astra Controlは、次のドライバでサポートされるストレージクラスを使用して、NetApp SnapMirrorテクノロジーによるレプリケーションをサポートします。

◦ ontap-nas



◦ ontap-san

- ライセンス：Data Protection Bundleを使用するONTAP SnapMirror非同期ライセンスが、ソースとデスティネーションの両方のONTAPクラスタで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"](#) を参照してください。

## ピアリング

- \*クラスタとSVM\*：ONTAPストレージバックエンドにピア関係が設定されている必要があります。を参照してください ["クラスタとSVMのピアリングの概要"](#) を参照してください。



2つのONTAPクラスタ間のレプリケーション関係で使用されるSVM名が一意であることを確認してください。

- \*Astra Control ProvisionerまたはAstra TridentとSVM\*：ピア関係にあるリモートSVMは、デスティネーションクラスタのAstra Control ProvisionerまたはAstra Tridentで使用できる必要があります。



### Astra Control Center の略

["Astra Control Centerを導入"](#) シームレスなディザスタリカバリのための第3の障害ドメインまたはセカンダリサイト。

- マネージドバックエンド：レプリケーション関係を作成するには、Astra Control CenterでONTAPストレージバックエンドを追加および管理する必要があります。



Astra Control Provisionerを有効にしている場合、Astra Control CenterでONTAPストレージバックエンドの追加と管理はオプションです。

- 管理対象クラスタ：Astra Controlを使用して次のクラスタを追加、管理できます。理想的には、異なる障害ドメインまたはサイトに配置されます。
  - ソースKubernetesクラスタ
  - デスティネーションKubernetesクラスタ
  - 関連付けられているONTAPクラスタ
- ユーザーアカウント：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」ロールのユーザー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`。ONTAP ソースとデスティネーションの両方のクラスタで有効にします。を参照してください ["ONTAP ドキュメントの「ユーザーアカウントの管理」を参照してください"](#) を参照してください。



Astra Control Provisioner機能では、Astra Control Centerでクラスタを管理するために「admin」ロールを明確に定義する必要はありません。これらのクレデンシャルはAstra Control Centerでは必要ありません。



Astra Control Centerでは、NVMe over TCPプロトコルを使用するストレージバックエンドのNetApp SnapMirrorレプリケーションはサポートされません。

## Astra Trident / ONTAP 構成

Astra Control Centerでは、ソースとデスティネーションの両方のクラスタのレプリケーションをサポートするストレージバックエンドを少なくとも1つ設定する必要があります。ソースクラスタとデスティネーションクラスタが同じである場合は、耐障害性を最大限に高めるために、デスティネーションアプリケーションでソ

ースアプリケーションとは別のストレージバックエンドを使用する必要があります。



Astra Controlレプリケーションでは、単一のストレージクラスを使用するアプリケーションがサポートされます。ネームスペースにアプリケーションを追加するときは、そのアプリケーションのストレージクラスがネームスペース内の他のアプリケーションと同じであることを確認してください。レプリケートされたアプリケーションにPVCを追加するときは、新しいPVCのストレージクラスがネームスペース内の他のPVCと同じであることを確認してください。

## レプリケーション関係を設定

レプリケーション関係の設定には、次の作業が含まれます。

- Astra ControlでアプリケーションSnapshotを作成する頻度を選択します（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が含まれます）。
- 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の作成時間の設定

## 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. >[レプリケーション]\*タブを選択します。
3. [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クラスタを入力します（ソースクラスタと同じでもかまいません）。
  - 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クラスタのピア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。ベストプラクティスとして、デスティネーションストレージクラスでソースストレージクラスとは別のストレージバックエンドを指定することを推奨します。
  - レプリケーションタイプ：Asynchronous は、現在使用可能な唯一のレプリケーションタイプです。
  - 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規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。
  - （任意）[Add namespace]を選択し、ドロップダウンリストからネームスペースを選択して、ネームスペースを追加します。
  - レプリケーション頻度：Astra Controlで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設定します。
  - オフセット：Astra ControlでSnapshotを作成する時間（分）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。



バックアップとレプリケーションのスケジュールをオフセットして、スケジュールの重複を回避します。たとえば、1時間ごとに1時間の最上部にバックアップを実行し、オフセットを5分、間隔を10分に設定してレプリケーションを開始するようにスケジュールを設定します。

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が、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確認するには、[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名の形式は次のとおりです。 replication-schedule-<string>。 Astra Controlは、レプリ  
ケーションに使用された最後のSnapshotを保持します。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名前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に対応する宛先名前スペースにPVCを作成します。
- アプリケーションと整合性のある初期スナップショットを作成します。
- 最初の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、このトピックの最後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. >[レプリケーション]\*タブを選択します。
3. [アクション]メニューから\*[フェイルオーバー]\*を選択します。
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

フェイルオーバー手順が発生すると、次の処理が実行されます。

- デスティネーションアプリケーションは、最新のレプリケートされたSnapshotに基づいて起動されます。

- 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。
- ソースアプリの保護ポリシーは、フェイルオーバー時にソースアプリ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にコピーされます。
- ソースアプリで1つ以上のリストア後の実行フックが有効になっている場合、それらの実行フックはデスティネーションアプリに対して実行されます。
- 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ず。



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. >[レプリケーション]\*タブを選択します。
3. [操作]メニューから\*[再同期]\*を選択します。
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

これは、アプリケーションをデスティネーションストレージバックエンドに移動し、元のソースストレージバックエンドに引き続きレプリケートするという計画的な処理です。Astra Controlは、デスティネーションア

リケーションにフェイルオーバーする前に、ソースアプリケーションを停止してデスティネーションにデータをレプリケートします。

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。

#### 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. >[レプリケーション]\*タブを選択します。
3. [操作]メニューから\*[逆レプリケーション]\*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。

- 元のソースアプリのKubernetesリソースのスナップショットが作成されます。
- 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、アプリのボリュームのスナップショットが取得され、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。
- アプリのKubernetesリソースは、元のソースアプリがシャットダウンされた後に複製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スナップショットから復元されます。
- 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、フェイルオーバー処理後に次の一連の処理を使用して「フェイルバック」を実現できます。このワークフローでは、レプリケーションの方向を元に戻すために、Astra Controlがアプリケーションの変更を元のソースアプリケーションにレプリケート（再同期）してからレプリケーションの方向を反転します。

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が完了した関係から開始し、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. >[レプリケーション]\*タブを選択します。
3. [操作]メニューから\*[再同期]\*を選択します。
4. フェイルバック処理の場合は、フェイルオーバーした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されます）。

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、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. >[レプリケーション]\*タブを選択します。
3. [アプリケーションの保護]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全性は、次のステータスで示されます。

- 正常：関係が確立中または確立されており、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（成功したSnapshotがレプリケートされ、フェイルオーバーが可能です）、最新のSnapshotはレプリケートに失敗したか失敗しました。

## 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、必要に応じて名前スペースを作成し、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が使用可能であること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**：Astra Controlは、SnapMirror関係を解除し、最後にレプリケートされたアプリケーションのSnapshotからアプリケーションのKubernetesリソースをリストアします。
- **フェイルオーバー**：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最新の（成功した）レプリケートされ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名前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できます。Astra Controlでアプリケーションをクローニングすると、アプリケーション構成と永続的ストレージのクローンが作成されます。

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名前スペース間でワークロードを移動できます。Astra Control Center UIまたははを使用できます ["Astra Control API の略"](#) アプリケーションのクローン作成と移行を実行します。

## 作業を開始する前に

- デスティネーションボリュームを確認：別のストレージクラスにクローニングする場合は、ストレージクラスで同じ永続ボリュームアクセスモード（ReadWriteManyなど）が使用され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的ボリュームのアクセスモードが異なると、クローニング処理は失敗します。たとえば、ソースの永続ボリュームがRWXアクセスモードを使用している場合は、Azure Managed Disks、AWS EBS、Google Persistent Disk、など、RWXを提供できないデスティネーションストレージクラスを選択します `ontap-san` を指定すると、クローン処理は失敗します。原因は失敗します。永続ボリュームのアクセスモードの詳細については、を参照してください ["Kubernetes" ドキュメント](#)
- 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、ソースクラスタとデスティネーションクラスタを含むクラウドインスタンス（同じでない場合）にデフォルトのバケットを用意する必要があります。クラウドインスタンスごとにデフォルトのバケットを割り当てる必要があります。
- クローン処理中に、IngressClassリソースまたはwebhookを必要とするアプリケーションが正常に機能するためには、これらのリソースがデスティネーションクラスタですでに定義されていない必要があります。

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`

## クローンの制限事項

- 明示的なストレージクラス：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導入し、そのアプリケーションのクローンを作成する必要がある場合、ターゲットクラスタには元々指定されたストレージクラスが必要です。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。
- \* ontap-nas-economy-basedアプリケーション\*：アプリケーションのストレージクラスが `ontap-nas-economy` ドライバ。ただし、["ONTAP NAS経済性に優れた運用向けのバックアップとリストアを実現"](#)。
- クローンとユーザーの制約：名前空間の名前/ IDまたは名前空間のラベルによって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任意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名前スペースからアクセスすることはできません。クローン処理またはリストア処理で新しい名前スペースが作成されたあと、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、影響を受けるユーザのロールの制約を更新して、新しい名前スペースへのアクセスを許可できます。
- クローンはデフォルトバケットを使用：アプリケーションのバックアップまたはアプリケーションのリストア時に、オプションでバケットIDを指定できます。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"](#) その後を押します ["リストア"](#) 個別。
- \* Jenkins CI\*を使用：オペレータがデプロイした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復元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。
- \* S3バケットを使用している場合\*：Astra Control CenterのS3バケットは使用可能容量を報告させ

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。

- \*特定のバージョンのPostgreSQL\*：同じクラスタ内のアプリケーションクローンは、Bitnami PostgreSQL 11.5.0チャートで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。

### OpenShift に関する考慮事項

- クラスタおよびOpenShiftバージョン：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。
- \*プロジェクトおよびUID\*：OpenShiftクラスタでアプリ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名前空間）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. クローンの詳細を指定します。
  - 名前を入力します。
  - 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。
  - クローンのデスティネーション名前スペースを入力してください。アプリケーションに関連付けられた各ソース名前スペースは、定義した宛先名前スペースにマッピングされます。



Astra Controlでは、クローニング処理の一環として新しいデスティネーション名前スペースが作成されます。指定するデスティネーション名前スペースがデスティネーションクラスタに存在していないことを確認してください。

- 「\* 次へ \*」を選択します。
- アプリケーションに関連付けられている元のストレージクラスを保持するか、別のストレージクラスを選択します。





アプリケーションのストレージクラスをネイティブクラウドプロバイダのストレージクラスまたはサポートされている他のストレージクラスに移行したり、ontap-nas-economy をバックアップされたストレージクラスに追加します。ontap-nas を使用するか、から作成されたストレージクラスを含む別のクラスタにアプリケーションをコピーします。ontap-nas-economy ドライバ。



別のストレージクラスを選択し、このストレージクラスがリストア時に存在しない場合は、エラーが返されます。

5. 「\* 次へ \*」を選択します。
6. クローンに関する情報を確認し、\* Clone \*を選択します。

## 結果

Astra Controlは、入力した情報に基づいてアプリケーションをクローニングします。新しいアプリケーションクローンが含まれている場合、クローニング処理は成功します。Healthy 「アプリケーション」 ページで説明します。

クローン処理またはリストア処理で新しい名前スペースが作成されたあと、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、影響を受けるユーザのロールの制約を更新して、新しい名前スペースへのアクセスを許可できます。



データ保護処理（クローン、バックアップ、またはリストア）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 Centerでは、実行可能なタイミングに基づいて、次のタイプの実行フックがサポートされません。

- 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
- フェイルオーバー後



## 実行フックフィルタ

アプリケーションに実行フックを追加または編集するときに、実行フックにフィルタを追加して、フックが一致するコンテナを管理できます。フィルタは、すべてのコンテナで同じコンテナイメージを使用し、各イメージを別の目的（Elasticsearchなど）に使用するアプリケーションに便利です。フィルタを使用すると、一部の同一コンテナで実行フックが実行されるシナリオを作成できます。1つの実行フックに対して複数のフィルタを作成すると、それらは論理AND演算子と結合されます。実行フックごとに最大10個のアクティブフィルタを使用できます。

実行フックに追加する各フィルタは、正規表現を使用してクラスタ内のコンテナを照合します。フックがコンテナと一致すると、そのコンテナに関連付けられたスクリプトがフックによって実行されます。フィルタの正規表現では、正規表現2（RE2）構文を使用します。この構文では、一致リストからコンテナを除外するフィルタの作成はサポートされていません。実行フックフィルタの正規表現でAstra Controlがサポートする構文については、を参照してください ["正規表現2（RE2）構文のサポート"](#)。



リストアまたはクローン処理のあとに実行される実行フックにネームスペースフィルタを追加し、リストアまたはクローンのソースとデスティネーションが異なるネームスペースにある場合、ネームスペースフィルタはデスティネーションネームスペースにのみ適用されます。

## 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、実行中のアプリケーションの機能を低下させたり、完全に無効にしたり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使用されるロジックは、バックアップが完了したとは見なされません。

- 新しいAstra Control環境では、実行フック機能はデフォルトで無効になっています。
  - 実行フックを使用する前に、実行フック機能を有効にする必要があります。
  - 所有者ユーザまたは管理者ユーザは、現在のAstra Controlアカウントで定義されているすべてのユーザの実行フック機能を有効または無効にできます。を参照してください [\[実行フック機能を有効にする\]](#) および [\[実行フック機能を無効にする\]](#) 手順については、を参照し
  - 機能の有効化ステータスは、Astra Controlのアップグレード中も維持されます。
- 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。
- オンデマンドのデータ保護処理では、すべてのフックイベントが生成され、\*アクティビティ\*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）。
- レプリケートされたソースアプリケーションをAstra Control Centerがデスティネーションアプリケーションにフェイルオーバーすると、フェイルオーバーの完了後にソースアプリケーションに対して有効になっているフェイルオーバー後の実行フックがデスティネーションアプリケーションに対して実行されます。



Astra Control Center 23.04でリストア後のフックを実行していて、Astra Control Center を23.07以降にアップグレードした場合、フェイルオーバーレプリケーション後にリストア後の実行フックが実行されなくなります。アプリケーションのフェイルオーバー後の実行フックを新しく作成する必要があります。また、フェイルオーバー用の既存のリストア後フックの処理タイプを「リストア後」から「フェイルオーバー後」に変更することもできます。

#### 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、すべての異なるタイプのフックを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名前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	名前スペース	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	フェールオーバーフックの実行
1.	クローン	N	N	新規	同じ	Y	N	Y	N
2.	クローン	N	N	新規	違う	Y	Y	Y	N
3.	クローン または リストア	Y	N	新規	同じ	N	N	Y	N
4.	クローン または リストア	N	Y	新規	同じ	N	N	Y	N
5.	クローン または リストア	Y	N	新規	違う	N	N	Y	N

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	フェールオーバーフックの実行
6.	クローン またはリストア	N	Y	新規	違う	N	N	Y	N
7.	リストア	Y	N	既存	同じ	N	N	Y	N
8.	リストア	N	Y	既存	同じ	N	N	Y	N
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	N
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	N
11.	バックアップ	Y	該当なし	該当なし	該当なし	N	N	該当なし	N
12	フェールオーバー	Y	該当なし	レプリケーションで作成	違う	N	N	N	Y
13	フェールオーバー	Y	該当なし	レプリケーションで作成	同じ	N	N	N	Y

## 実行フックの例

にアクセスします ["NetApp Verda GitHubプロジェクト"](#) Apache CassandraやElasticsearchなどの一般的なアプリケーションの実行フックをダウンロードします。また、独自のカスタム実行フックを構築するための例やアイデアを得ることもできます。

## 実行フック機能を有効にする

所有者または管理者ユーザーの場合は、実行フック機能を有効にできます。この機能を有効にすると、このAstra Controlアカウントで定義されているすべてのユーザが実行フックを使用して、既存の実行フックとフックスクリプトを表示できます。

## 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。
3. \*実行フックを有効にする\*を選択します。

アカウント>\*機能設定\*タブが表示されます。

4. Execution Hooks\*ペインで、設定メニューを選択します。
5. [有効] を選択します。
6. 表示されるセキュリティ警告を確認します。

7. [はい、実行フックを有効にする]\*を選択します。

### 実行フック機能を無効にする

所有者または管理者ユーザは、このAstra Controlアカウントで定義されているすべてのユーザに対して実行フック機能を無効にすることができます。実行フック機能を無効にする前に、既存の実行フックをすべて削除する必要があります。を参照してください [\[実行フックを削除します\]](#) 既存の実行フックを削除する手順については、を参照してください。

#### 手順

1. に移動し、[機能設定]\*タブを選択します。
2. [実行フック \*] タブを選択します。
3. Execution Hooks\*ペインで、設定メニューを選択します。
4. [Disable] を選択します。
5. 表示される警告を確認します。
6. を入力します disable をクリックして、すべてのユーザに対してこの機能を無効にすることを確認します。
7. [はい、無効にする]\*を選択します。

### 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、一致するコンテナの数、作成時間、および実行時間（プリ/ポストオペレーション）を確認できます。を選択できます + アイコンをクリックして、実行するコンテナのリストを展開します。このアプリケーションの実行フックに関連するイベントログを表示するには、\*アクティビティ\*タブに移動します。

### 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

各実行フックは、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。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つのスクリプトを変更するだけで多くの実行フックを更新できます。

### 手順

1. 実行フック機能が **有効**。
2. 「アカウント」に移動します。
3. [スクリプト]タブを選択します。
4. 「\* 追加」を選択します。
5. 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。
6. 「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してAstra Controlに追加できます。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. 実行フック機能が **有効**。
  2. 「\* 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。
  3. [実行フック \*] タブを選択します。
  4. 「\* 追加」を選択します。
  5. [フックの詳細\* (Hook Details \*)] 領域で、次の
    - a. \*操作\*ドロップダウンメニューから操作タイプを選択して、フックをいつ実行するかを決定します。
    - b. フックの一意の名前を入力します。
    - c. (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。
  6. (オプション) フックフィルタの詳細 (\* Hook Filter Details \*) 領域で、実行フックが実行されるコンテナを制御するフィルタを追加できます。
    - a. [フィルタの追加]を選択します。
    - b. [フックフィルタータイプ\*]列で、フィルターを適用する属性をドロップダウンメニューから選択します。
    - c. **[Regex]**列に、フィルタとして使用する正規表現を入力します。Astra Controlでは、を使用します **"正規表現2 (RE2) 正規表現の正規表現構文"**。
      - a. フィルタをさらに追加するには、\*フィルタを追加\*を選択します。
- 
- 正規表現フィールドに他のテキストが含まれていない属性 (ポッド名など) の正確な名前前でフィルタリングすると、サブストリングの一致が実行されます。正確な名前とその名前だけを照合するには、完全に一致する文字列の一致構文を使用します (例: ^exact\_podname\$)。
- a. フィルタをさらに追加するには、\*フィルタを追加\*を選択します。
- 
- 実行フックの複数のフィルタは、論理AND演算子と結合されます。実行フックごとに最大10個のアクティブフィルタを使用できます。

- 7. 完了したら、「次へ」を選択します。
- 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。
- 52

- 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. 「\* 次へ \*」を選択します。
10. 実行フックの設定を確認します。
11. 「\* 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。

### 実行フックを編集します

実行フックを編集して、その属性、フィルタ、または使用するスクリプトを変更できます。実行フックを編集するには、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. 編集するフックの\*アクション\*列のオプションメニューを選択します。
4. 「\* 編集 \*」を選択します。
5. 各セクションを完了したら、「次へ」を選択して、必要な変更を行います。
6. [保存 (Save) ]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。

5. 表示されたダイアログで、「delete」と入力して確定します。

6. [はい]を選択し、実行フックを削除します。\*

を参照してください。

- ["NetApp Verda GitHubプロジェクト"](#)

## Astra Control Centerを使用したAstra Control Centerの保護

Astra Control Centerが実行されているKubernetesクラスタで致命的なエラーに対する耐障害性を高めるには、Astra Control Centerアプリケーション自体を保護します。セカンダリのAstra Control Centerインスタンスを使用してAstra Control Centerをバックアップおよびリストアするか、基盤となるストレージでONTAPを使用している場合はAstraレプリケーションを使用できます。

これらのシナリオでは、Astra Control Centerの2つ目のインスタンスを別のフォールトドメインに導入して設定し、プライマリAstra Control Centerインスタンスとは別の2つ目のKubernetesクラスタで実行します。2つ目のAstra Control Centerインスタンスは、プライマリのAstra Control Centerインスタンスのバックアップに使用され、場合によってはリストアにも使用されます。リストアまたはレプリケートされたAstra Control Centerインスタンスは、引き続きアプリケーションクラスタアプリケーションのアプリケーションデータ管理機能を提供し、それらのアプリケーションのバックアップやSnapshotへのアクセスをリストアします。

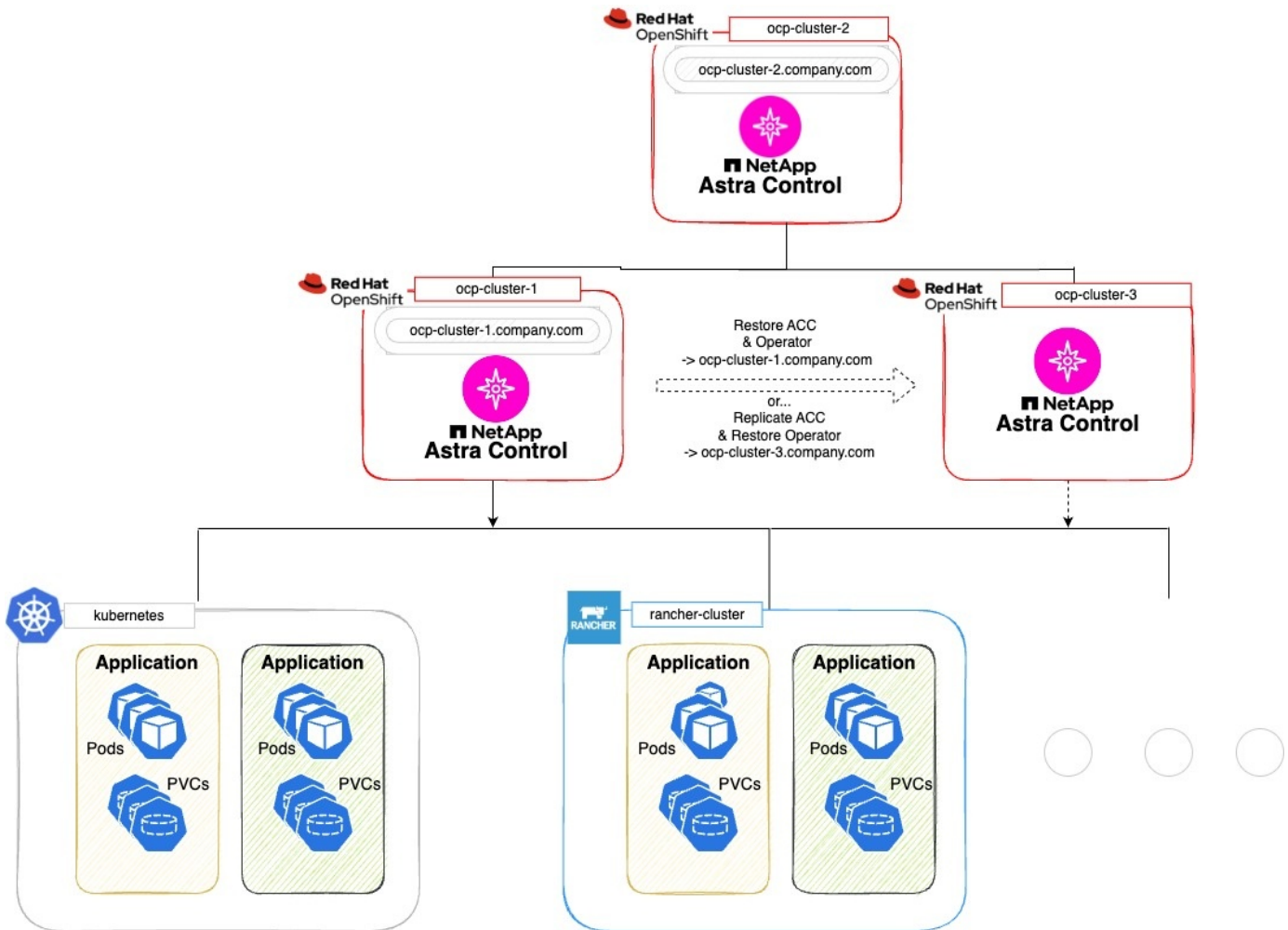
作業を開始する前に

Astra Control Centerの保護シナリオを設定する前に、次の情報を確認してください。

- **プライマリAstra Control Center**インスタンスを実行する**Kubernetes**クラスタ：このクラスタは、アプリケーションクラスタを管理するプライマリAstra Control Centerインスタンスをホストします。
- **セカンダリAstra Control Center**インスタンスを実行しているプライマリと同じ**Kubernetes**ディストリビューションタイプの**2つ目のKubernetes**クラスタ：このクラスタは、プライマリAstra Control Centerインスタンスを管理するAstra Control Centerインスタンスをホストします。
- **プライマリと同じKubernetes**ディストリビューションタイプの**3つ目のKubernetes**クラスタ：このクラスタは、Astra Control Centerのリストアまたはレプリケートされたインスタンスをホストします。現在プライマリに導入されているものと同じAstra Control Center名前空間を使用する必要があります。たとえば、Astra Control Centerが名前空間に導入されている場合 netapp-acc ソースクラスタで、名前空間 netapp-acc デスティネーションKubernetesクラスタ上のどのアプリケーションでも使用できない状態である必要があります。
- **\* S3互換バケット\***：各Astra Control Centerインスタンスには、アクセス可能なS3互換オブジェクトストレージバケットがあります。
- **設定されたロードバランサ**：ロードバランサはAstraのIPアドレスを提供し、アプリケーションクラスタと両方のS3バケットへのネットワーク接続を確立する必要があります。
- **クラスタはAstra Control Centerの要件に準拠**：Astra Control Center保護で使用される各クラスタは、["Astra Control Centerの一般的な要件"](#)。

このタスクについて

以下の手順では、以下のコマンドを使用してAstra Control Centerを新しいクラスタにリストアするために必要な手順について説明します。 [バックアップとリストア](#) または [レプリケーション](#)。手順は、ここに示す構成例に基づいています。



この設定例では、次の情報が表示されています。

- プライマリ **Astra Control Center** インスタンスを実行する **Kubernetes** クラスタ：
  - OpenShift クラスタ： ocp-cluster-1
  - Astra Control Center プライマリ インスタンス： ocp-cluster-1.company.com
  - このクラスタは、アプリケーションクラスタを管理します。
- セカンダリ **Astra Control Center** インスタンスを実行しているプライマリと同じ **Kubernetes** ディストリビューションタイプの2つ目の **Kubernetes** クラスタ：
  - OpenShift クラスタ： ocp-cluster-2
  - Astra Control Center のセカンダリ インスタンス： ocp-cluster-2.company.com
  - このクラスタを使用して、プライマリの Astra Control Center インスタンスをバックアップするか、別のクラスタへのレプリケーションを設定します（この例では、ocp-cluster-3 クラスタ）。
- リストア処理に使用されるプライマリと同じ **Kubernetes** ディストリビューションタイプの3つ目の **Kubernetes** クラスタ：
  - OpenShift クラスタ： ocp-cluster-3
  - Astra Control Center 3つ目のインスタンス： ocp-cluster-3.company.com
  - このクラスタは、Astra Control Center のリストアまたはレプリケーションのフェイルオーバーに使用

されます。



アプリケーションクラスタは、上図のKubernetesクラスタとRancherクラスタからわかるように、3つのAstra Control Centerクラスタの外部に配置するのが理想的です。

図には示されていません。

- すべてのクラスタに、Astra TridentまたはAstra Control ProvisionerがインストールされたONTAPバックエンドがあります。
- この構成では、OpenShiftクラスタがMetalLBをロードバランサとして使用しています。
- SnapshotコントローラとVolumeSnapshotClassもすべてのクラスタにインストールされています（を参照）。"[前提条件](#)"。

### ステップ1オプション：Astra Control Centerのバックアップとリストア

この手順では、バックアップとリストアを使用して新しいクラスタにAstra Control Centerをリストアするために必要な手順について説明します。

この例では、Astra Control Centerは常に `netapp-acc` 名前空間と演算子は、`netapp-acc-operator` ネームスペース：



ここでは説明しませんが、Astra Control Centerのオペレータは、Astra CRと同じネームスペースに導入することもできます。

作業を開始する前に

- プライマリのAstra Control Centerをクラスタにインストールしておきます。
- セカンダリのAstra Control Centerを別のクラスタにインストールしておきます。

手順

1. プライマリAstra Control Centerアプリケーションとデスティネーションクラスタを、（実行中の）セカンダリAstra Control Centerインスタンスから管理（`ocp-cluster-2` クラスタ）：
  - a. セカンダリAstra Control Centerインスタンスにログインします。
  - b. "[プライマリAstra Control Centerクラスタを追加](#)"（`ocp-cluster-1`）。
  - c. "[デスティネーションの3つ目のクラスタを追加](#)"（`ocp-cluster-3`）をクリックします。
2. セカンダリのAstra Control CenterでAstra Control CenterとAstra Control Centerオペレータを管理します。
  - a. [アプリケーション（Applications）]ページで、[定義（Define）]を選択します
  - b. [アプリケーションの定義]ウィンドウで、新しいアプリケーション名を入力します。（`netapp-acc`）。
  - c. プライマリAstra Control Centerを実行しているクラスタを選択（`ocp-cluster-1`）をクリックします。
  - d. を選択します `netapp-acc` Astra Control Centerのネームスペース（\*[ネームスペース]\*ドロップダウンリスト）。
  - e. [クラスタリソース]ページで、\*[クラスタを対象とした追加のリソースを含める]\*をオンにします。
  - f. 「含めるルールを追加」を選択します。

g. 次のエントリを選択し、\*[追加]\*を選択します。

- ラベルセレクタ:<label name>
- グループ：apiextensions.k8s.io
- バージョン：v1
- 種類: CustomResourceDefinition

h. アプリケーション情報を確認します。

i. [\* 定義 ( Define ) ]を選択します

「\* define \*」を選択した後、演算子に対して「アプリケーションの定義」プロセスを繰り返します。netapp-acc-operator) をクリックし、netapp-acc-operator [アプリケーションの定義]ウィザードの名前空間。

3. Astra Control Centerとオペレータのバックアップ：

- a. セカンダリAstra Control Centerで、[Applications]タブを選択して[Applications]ページに移動します。
- b. "バックアップ" Astra Control Centerアプリケーション (netapp-acc) 。
- c. "バックアップ" 演算子 (netapp-acc-operator) 。

4. Astra Control Centerとオペレータをバックアップしたら、次のツールでディザスタリカバリ (DR) シナリオをシミュレートします。"Astra Control Centerのアンインストール" プライマリクラスタから削除します。



新しいクラスタ（この手順で説明する3つ目のKubernetesクラスタ）にAstra Control Centerをリストアし、新しくインストールしたAstra Control Centerのプライマリクラスタと同じDNSを使用します。

5. セカンダリAstra Control Centerを使用 "リストア" バックアップから作成したAstra Control Centerアプリケーションのプライマリインスタンス：

- a. [Applications]\*を選択し、Astra Control Centerアプリケーションの名前を選択します。
- b. [オプション]メニューの[操作]列で、\*[リストア]\*を選択します。
- c. リストアタイプとして\*[新しいネームスペースにリストアする]\*を選択します。
- d. リストア名を入力 (netapp-acc) 。
- e. デスティネーションの3番目のクラスタを選択 (ocp-cluster-3) 。
- f. 元のネームスペースと同じネームスペースになるようにデスティネーションネームスペースを更新します。
- g. [Restore Source]ページで、リストア・ソースとして使用するアプリケーション・バックアップを選択します。
- h. [元のストレージクラスを使用してリストアする]\*を選択します。
- i. [すべてのリソースをリストア]\*を選択します。
- j. リストア情報を確認し、\*[Restore]\*を選択して、Astra Control Centerをデスティネーションクラスタにリストアするリストアプロセスを開始します。(ocp-cluster-3) 。アプリケーションが起動すると、リストアが完了します。 available 状態。

6. デスティネーションクラスタでAstra Control Centerを設定します。

- a. ターミナルを開き、kubecfgを使用してデスティネーションクラスタに接続します。(ocp-cluster-3) をクリックします。
- b. を確認します ADDRESS Astra Control Center構成の列で参照されるプライマリシステムのDNS名は次のとおりです。

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

対応：

NAME	UUID	VERSION	ADDRESS
READY			
astra	89f4fd47-0cf0-4c7a-a44e-43353dc96ba8	24.02.0-69	ocp-cluster-1.company.com
		True	

- a. 状況に応じて ADDRESS 上記の応答のフィールドにプライマリAstra Control CenterインスタンスのFQDNがない場合は、Astra Control CenterのDNSを参照するように設定を更新します。

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

- i. を変更します astraAddress の下 spec: FQDNへ (ocp-cluster-1.company.com (この例では) のプライマリAstra Control Centerインスタンス。
- ii. 設定を保存します。
- iii. アドレスが更新されたことを確認します。

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

- b. にアクセスします [Astra Control Centerのオペレータのリストア](#) セクションを参照して、リストアプロセスを完了してください。

ステップ1オプション：レプリケーションを使用して**Astra Control Center**を保護

この手順では、設定に必要な手順について説明します。"[Astra Control Centerのレプリケーション](#)" を使用して、プライマリAstra Control Centerインスタンスを保護します。

この例では、Astra Control Centerは常に netapp-acc 名前空間と演算子は、netapp-acc-operator ネームスペース：

作業を開始する前に

- プライマリのAstra Control Centerをクラスタにインストールしておきます。
- セカンダリのAstra Control Centerを別のクラスタにインストールしておきます。

手順



1. セカンダリAstra Control CenterインスタンスからプライマリAstra Control Centerアプリケーションとデスティネーションクラスタを管理します。
  - a. セカンダリAstra Control Centerインスタンスにログインします。
  - b. "プライマリAstra Control Centerクラスタを追加" (ocp-cluster-1)。
  - c. "デスティネーションの3つ目のクラスタを追加" (ocp-cluster-3) をクリックします。
2. セカンダリのAstra Control CenterでAstra Control CenterとAstra Control Centerオペレータを管理します。
  - a. [Clusters]\*を選択し、プライマリAstra Control Centerが含まれるクラスタを選択します。(ocp-cluster-1)。
  - b. [名前空間]タブを選択します。
  - c. 選択するオプション netapp-acc および netapp-acc-operator 名前空間。
  - d. [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。
  - e. 定義されたアプリケーションを表示するには、\*[アプリケーションで表示]\*を選択します。
3. レプリケーションのバックエンドを構成します。



レプリケーションには、プライマリのAstra Control Centerクラスタとデスティネーションクラスタが必要 (ocp-cluster-3) 別のピアONTAPストレージバックエンドを使用します。各バックエンドがピアリングされてAstra Controlに追加されると、[Backends]ページの\*[Discovered]\*タブにバックエンドが表示されます。

- a. "ピアバックエンドの追加" をプライマリクラスタのAstra Control Centerに接続します。
  - b. "ピアバックエンドの追加" デスティネーションクラスタのAstra Control Centerに接続します。
4. レプリケーションを設定します。
    - a. [Applications]画面で、netapp-acc アプリケーション：
    - b. [レプリケーションポリシーの設定]\*を選択します。
    - c. 選択するオプション ocp-cluster-3 をデスティネーションクラスタとして指定します。
    - d. ストレージクラスを選択します。
    - e. 入力するコマンド netapp-acc をデスティネーションネームスペースとして指定します。
    - f. 必要に応じてレプリケーション頻度を変更します。
    - g. 「\*次へ\*」を選択します。
    - h. 設定が正しいことを確認し、\*[保存]\*を選択します。

レプリケーション関係の移行元 Establishing 終了：Established。アクティブな場合、このレプリケーションは、レプリケーション設定が削除されるまで5分おきに実行されます。

5. プライマリシステムが破損しているかアクセスできなくなった場合は、レプリケーションをもう一方のクラスタにフェイルオーバーします。



フェイルオーバーが正常に実行されるように、デスティネーションクラスタにAstra Control Centerがインストールされていないことを確認してください。

- a. 縦の楕円アイコンを選択し、\*フェイルオーバー\*を選択します。

The screenshot shows the NetApp Cloud Manager interface. At the top, there are navigation tabs: Data protection, Storage, Resources, Execution hooks, Activity, and Tasks. Below these is a 'Configure' dropdown and buttons for Snapshots, Backups, and Replication. The main area displays a replication relationship between two 'netapp-acc' instances. The source instance is 'Available' and the destination is also 'Available'. A context menu is open over the source instance, with 'Fail over' selected. The right sidebar shows the 'Replication relationship' details: STATUS is 'Healthy | Established', SCHEDULE is 'Replicate snapshot every 5 minutes to ocp-cluster-3', and LAST SYNC is '2023/08/01 17:18 UTC' with a duration of 32 seconds.

- b. 詳細を確認し、\*[フェイルオーバー]\*を選択してフェイルオーバープロセスを開始します。

レプリケーション関係のステータスがに変わります。Failing over 次に Failed over 完了したら、

6. フェイルオーバーの設定を完了します。

- a. ターミナルを開き、3番目のクラスタのkubeconfigを使用して接続します。(ocp-cluster-3)。このクラスタにはAstra Control Centerがインストールされています。
- b. 3つ目のクラスタのAstra Control Center FQDNを確認 (ocp-cluster-3)。
- c. Astra Control CenterのDNSを参照するように設定を更新します。

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

- i. を変更します astraAddress の下 spec: FQDNを使用 (ocp-cluster-3.company.com) をクリックします。
- ii. 設定を保存します。
- iii. アドレスが更新されたことを確認します。

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

- d. 必要なすべてのtraefik CRDが存在することを確認します。

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

必要なtraefik CRD :



```
ingressroutes.traefik.containo.us
ingressroutes.traefik.io
ingressroutetcps.traefik.containo.us
ingressroutetcps.traefik.io
ingressrouteudps.traefik.containo.us
ingressrouteudps.traefik.io
middlewares.traefik.containo.us
middlewares.traefik.io
middlewareetcps.traefik.containo.us
middlewareetcps.traefik.io
serverstransports.traefik.containo.us
serverstransports.traefik.io
tloptions.traefik.containo.us
tloptions.traefik.io
tIsstores.traefik.containo.us
tIsstores.traefik.io
traefikservices.traefik.containo.us
traefikservices.traefik.io
```

a. 上記のCRDの一部がない場合は、次の手順を実行します。

- i. に進みます ["traefikドキュメント"](#)。
- ii. 「定義」領域をファイルにコピーします。
- iii. 変更を適用：

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

iv. traefikを再起動します。

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

b. にアクセスします [Astra Control Centerのオペレータのリストア](#) セクションを参照して、リストアプロセスを完了してください。

## ステップ2：Astra Control Centerのオペレータをリストア

セカンダリのAstra Control Centerを使用して、プライマリのAstra Control Centerオペレータをバックアップからリストアデスティネーション名前スペースは、ソース名前スペースと同じである必要があります。Astra Control Centerをプライマリソースクラスタから削除しても、同じリストア手順を実行するためのバックアップは引き続き存在します。

### 手順

1. \*アプリケーション\*を選択し、オペレータアプリの名前を選択します。(netapp-acc-operator)。

2. [操作]列の[オプション]メニューから\*[リストア]\*を選択します。
3. リストアタイプとして\*[新しい名前スペースにリストアする]\*を選択します。
4. デスティネーションの3番目のクラスタを選択 (ocp-cluster-3)。
5. 名前スペースをプライマリソースクラスタに関連付けられている名前スペースと同じに変更する (netapp-acc-operator)。
6. リストア・ソースとして以前に作成されたバックアップを選択します。
7. [元のストレージクラスを使用してリストアする]\*を選択します。
8. [すべてのリソースをリストア]\*を選択します。
9. 詳細を確認し、\*[リストア]\*をクリックしてリストアプロセスを開始します。

[Applications]ページには、Astra Control Centerオペレータがデスティネーションの第3のクラスタにリストアされていることが表示される (ocp-cluster-3)。プロセスが完了すると、状態はとして表示されます。Available。10分以内に、ページでDNSアドレスが解決されます。

## 結果

Astra Control Centerとその登録済みクラスタ、Snapshotとバックアップを使用した管理対象アプリケーションを、デスティネーションの第3のクラスタで利用できるようになりました。(ocp-cluster-3)。元のインスタンスに対して使用していた保護ポリシーは、新しいインスタンスにも適用されます。スケジュールされたバックアップやオンデマンドのバックアップやスナップショットを引き続き作成できます。

## トラブルシューティング

システムの健全性と保護プロセスが成功したかどうかを確認します。

- ポッドが実行されていません：すべてのポッドが実行中であることを確認します。

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

一部のポッドが CrashLoopBackOff 状態、再起動し、次の状態に移行する必要があります。Running 状態。

- システムステータスの確認：Astra Control Centerシステムが ready 都道府県：

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

対応：

```
NAME      UUID                                VERSION  ADDRESS
READY
astra 89f4fd47-0cf0-4c7a-a44e-43353dc96ba8 24.02.0-69 ocp-cluster-
1.company.com                True
```

- 導入ステータスの確認：Astra Control Centerの導入情報を表示して Deployment State はです Deployed。

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

- リストアしたAstra Control Center UIで404エラーが返される：AccTraefik 入力オプションとして、[traefik CRD](#) すべてインストールされていることを確認します。

## アプリケーションとクラスタの健全性を監視

アプリケーションとクラスタの健全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トル

「\* アプリケーション \*」タイトルは、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。

### クラスタタイトル

クラスタタイトルには、Astra Control Center を使用して管理しているクラスタの健全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トル張りです

「ストレージバックエンド \*」タイトルは、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。

## クラスタの健全性を表示してストレージクラスを管理します

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。管理対象クラスタのデフォルトのストレージクラスを変更することもできます。

クラスタの健全性と詳細を表示します

クラスタの場所、ワーカーノード、永続ボリューム、ストレージクラスなどの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成功する場合) は、Astra Controlに指定したkubefconfig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、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。

## 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。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。



kubectlコマンドを使用してストレージクラスを変更しないでください。代わりに、この手順を使用してください。kubectlを使用して変更を行った場合、Astra Controlはその変更を元に戻します。

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。

## 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、そのアプリケーションの詳細がAstra Controlに表示されます。これにより、アプリケーションの通信ステータス（Astra Controlがアプリケーションと通信できるかどうか）、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特定できます。

### 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. 情報を確認します。

### アプリステータス

Astra Controlがアプリケーションと通信できるかどうかを示すステータスが表示されます。

- アプリ保護ステータス：アプリの保護状態を表示します。
  - \* 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、既存のスナップショット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的ボリュームを表示します。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されている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

## ローカルユーザとロールを管理します

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 の略"](#) 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 Control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 ControlユーザおよびLDAP定義に対応するグループを定義することです。Astra Control APIまたはWeb UIを使用して、LDAP認証とLDAPユーザおよびグループを設定できます。詳細については、次のドキュメントを参照してください。

- ["リモート認証とユーザーの管理には、Astra Control APIを使用します"](#)
- ["リモートユーザとリモートグループの管理には、Astra Control UIを使用します"](#)
- ["リモート認証を管理するには、Astra Control UIを使用します"](#)

### 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。

拘束の追加の詳細については、を参照してください "[ローカルユーザとロールを管理します](#)".

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 の略"](#) 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

管理者または所有者ユーザーは、メンバーまたはビューアーの役割に名前空間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"](#)

## リモート認証を管理する

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 Control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 ControlユーザおよびLDAP定義に対応するグループを定義するこ  
とです。Astra Control APIまたはWeb UIを使用して、LDAP認証とLDAPユーザおよびグループを設定できま  
す。



Astra Control Centerでは、リモート認証を有効にしたときに設定されるユーザログイン属性を  
使用して、リモートユーザを検索して追跡します。Astra Control Centerに表示するリモートユ  
ーザのこのフィールドには、Eメールアドレス（「mail」）またはユーザプリンシパル名  
（「userPrincipalName」）の属性が存在している必要があります。この属性は、Astra Control  
Centerで認証およびリモートユーザの検索に使用されるユーザ名です。

## LDAPS認証用の証明書を追加します

LDAPサーバのプライベートTLS証明書を追加して、LDAPS接続を使用する際にAstra Control CenterがLDAP  
サーバで認証できるようにします。この処理は、1回だけ、またはインストールした証明書の有効期限が切れ  
たときにのみ実行してください。

手順

1. 「アカウント」に移動します。

2. [証明書] タブを選択します。
3. 「\* 追加」を選択します。
4. をアップロードします .pem クリップボードからファイルの内容をファイルまたは貼り付けます。
5. [Trusted]チェックボックスをオンにします。
6. [証明書の追加]を選択します。

リモート認証を有効にします

LDAP認証を有効にして、Astra ControlとリモートLDAPサーバ間の接続を設定できます。

作業を開始する前に

LDAPSを使用する場合は、Astra Control CenterがLDAPサーバに対して認証できるように、Astra Control CenterにLDAPサーバのプライベートTLS証明書がインストールされていることを確認してください。を参照してください [LDAPS認証用の証明書を追加します](#) 手順については、を参照し

手順

1. 「\*アカウント」 > 「接続」に移動します。
2. [\* Remote Authentication (リモート認証)] ペインで、設定メニューを選択します。
3. 「\* 接続」を選択します。
4. サーバのIPアドレス、ポート、および優先接続プロトコル (LDAPまたはLDAPS) を入力します。



ベストプラクティスとして、LDAPサーバに接続するときはLDAPSを使用してください。LDAPSに接続する前に、LDAPサーバのプライベートTLS証明書をAstra Control Centerにインストールする必要があります。

5. サービスアカウントのクレデンシャルをEメール形式で入力します ([administrator@example.com](mailto:administrator@example.com))。Astra Controlは、LDAPサーバとの接続時にこれらのクレデンシャルを使用します。
6. [\* User Match]セクションで、次の手順を実行します。
  - a. LDAPサーバからユーザ情報を取得するときに使用するベースDNと適切なユーザ検索フィルタを入力します。
  - b. (オプション) ディレクトリでuser login属性が使用されている場合 userPrincipalName ではなく mail` と入力します `userPrincipalName [ユーザーログイン属性 (User login attribute)] フィールドの正しい属性に入力します。
7. [グループ一致] セクションで、グループ検索ベースDNと適切なカスタムグループ検索フィルタを入力します。



正しいベース識別名 (DN) と、\* User Match および Group Match \*の適切な検索フィルタを使用してください。ベースDNは、検索を開始するディレクトリツリーのレベルをAstra Controlに指示し、検索フィルタは、Astra Controlが検索するディレクトリツリーの部分を制限します。

8. [送信] を選択します。

結果

[リモート認証]ペインのステータスは、LDAPサーバへの接続が確立されると、[保留中]になり、次に[接続済

み]になります。

リモート認証を無効にします

LDAPサーバへのアクティブな接続を一時的に無効にすることができます。



LDAPサーバへの接続を無効にすると、すべての設定が保存され、Astra Controlに追加されたすべてのリモートユーザとリモートグループがそのLDAPサーバから保持されます。このLDAPサーバにいつでも再接続できます。

手順

1. 「\*アカウント」 > 「接続」に移動します。
2. [\* Remote Authentication (リモート認証)] ペインで、設定メニューを選択します。
3. [Disable] を選択します。

結果

[\* Remote Authentication (リモート認証)] ペインのステータスが[\* Disabled (無効)] に変わります。すべてのリモート認証設定、リモートユーザ、およびリモートグループが維持され、いつでも接続を再度有効にすることができます。

リモート認証の設定を編集します

LDAPサーバへの接続を無効にした場合、または\*リモート認証\*ペインが「接続エラー」状態にある場合は、設定を編集できます。



「リモート認証」ペインが「無効」状態の場合、LDAPサーバのURLまたはIPアドレスを編集することはできません。必要です [\[リモート認証を切断します\]](#) 最初に。

手順

1. 「\*アカウント」 > 「接続」に移動します。
2. [\* Remote Authentication (リモート認証)] ペインで、設定メニューを選択します。
3. 「\*編集\*」を選択します。
4. 必要な変更を行い、\* Edit \*を選択します。

リモート認証を切断します

LDAPサーバから切断して、Astra Controlから構成設定を削除できます。



LDAPユーザが切断した場合、セッションはすぐに終了します。LDAPサーバから切断すると、そのLDAPサーバのすべての構成設定がAstra Controlから削除されるだけでなく、そのLDAPサーバから追加されたすべてのリモートユーザとリモートグループも削除されます。

手順

1. 「\*アカウント」 > 「接続」に移動します。
2. [\* Remote Authentication (リモート認証)] ペインで、設定メニューを選択します。
3. 「切断」を選択します。

## 結果

「リモート認証」パネルのステータスが「切断済み」に変わります。リモート認証設定、リモートユーザ、およびリモートグループがAstra Controlから削除される。

## リモートユーザとリモートグループを管理します

Astra ControlシステムでLDAP認証を有効にしている場合は、LDAPユーザおよびグループを検索して、承認されたシステムのユーザに含めることができます。

### リモートユーザを追加します

アカウント所有者と管理者は、リモートユーザをAstra Controlに追加できます。Astra Control Centerは最大10,000人のLDAPリモートユーザをサポートします。



Astra Control Centerでは、リモート認証を有効にしたときに設定されるユーザログイン属性を使用して、リモートユーザを検索して追跡します。Astra Control Centerに表示するリモートユーザのこのフィールドには、Eメールアドレス（「mail」）またはユーザプリンシパル名（「userPrincipalName」）の属性が存在している必要があります。この属性は、Astra Control Centerで認証およびリモートユーザの検索に使用されるユーザ名です。



同じEメールアドレス（「mail」または「user principal name」属性に基づく）を持つローカルユーザがシステムにすでに存在する場合は、リモートユーザを追加できません。ユーザをリモートユーザとして追加するには、最初にローカルユーザをシステムから削除してください。

## 手順

1. **[Account (アカウント\*)]**領域に移動します。
2. **[\*Users & groups]**タブを選択します。
3. ページの右端で、**\*リモートユーザー\***を選択します。
4. 「**\* 追加**」を選択します。
5. 必要に応じて、ユーザのEメールアドレスを**\* Filter by email \***フィールドに入力して、LDAPユーザを検索します。
6. リストから1人以上のユーザを選択します。
7. ユーザにロールを割り当てます。



ユーザとユーザのグループに異なるロールを割り当てると、より権限の高いロールが優先されます。

8. 必要に応じて、このユーザに1つ以上のネームスペースの制約を割り当て、**\*ロールを制約に制限\***を選択して適用します。新しい名前空間制約を追加するには、**\*制約の追加\***を選択します。



ユーザにLDAPグループメンバーシップを使用して複数のロールを割り当てると、最も権限の高いロールの制約だけが有効になります。たとえば、ローカルビューアロールを持つユーザがメンバーロールにバインドされた3つのグループを結合すると、メンバーロールからの制約の合計が有効になり、ビューアロールからの制約はすべて無視されます。

9. 「**\* 追加**」を選択します。

## 結果

新しいユーザがリモートユーザのリストに表示されます。このリストでは、ユーザーに対するアクティブな拘束を表示したり、\*アクション\*メニューからユーザーを管理したりできます。

## リモートグループを追加します

複数のリモートユーザを一度に追加するには、アカウント所有者と管理者がリモートグループをAstra Controlに追加します。リモートグループを追加すると、そのグループ内のすべてのリモートユーザがAstra Controlにログインできるようになり、グループと同じロールが継承されます。

Astra Control Centerでは、最大5,000のLDAPリモートグループがサポートされます。

## 手順

1. [Account (アカウント\*)]領域に移動します。
2. [\*Users & groups]タブを選択します。
3. ページの右端で、\*リモートグループ\*を選択します。
4. 「\*追加」を選択します。

このウィンドウには、Astra Controlがディレクトリから取得したLDAPグループの共通名と識別名のリストが表示されます。

5. 必要に応じて、「共通名でフィルタ」フィールドにグループの共通名を入力してLDAPグループを検索します。
6. リストから1つ以上のグループを選択します。
7. グループにロールを割り当てます。



選択したロールは、このグループのすべてのユーザに割り当てられます。ユーザとユーザのグループに異なるロールを割り当てると、より権限の高いロールが優先されます。

8. 必要に応じて、このグループに1つ以上の名前空間制約を割り当て、\*制約にロールを制限\*を選択して適用します。新しい名前空間制約を追加するには、\*制約の追加\*を選択します。



- アクセス対象のリソースが最新の**Astra Connector**がインストールされているクラスタに属している場合：LDAPグループメンバーシップを通じてユーザに複数のロールが割り当てられると、ロールの制約が結合されます。たとえば、ローカルのビューアロールを持つユーザが、メンバーロールにバインドされている3つのグループに参加した場合、ユーザは元のリソースへのビューアロールアクセス権と、グループメンバーシップによって取得されたリソースへのメンバーロールアクセス権を持つようになります。
- アクセス対象のリソースが**Astra Connector**がインストールされていないクラスタに属している場合：ユーザにLDAPグループメンバーシップを通じて複数のロールが割り当てられている場合、最も権限の厳しいロールの制約のみが有効になります。

9. 「\*追加」を選択します。

## 結果

新しいグループがリモートグループのリストに表示されます。このグループのリモートユーザは、各リモートユーザがログインするまで、リモートユーザのリストに表示されません。このリストでは、\*アクション\*メニューからグループの詳細を表示したり、グループを管理したりできます。



## 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方法については、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、その中に\* 1つの\*コンテキスト要素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 Control Center UIまたはを使用できます ["Astra Control API の略"](#) 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"](#)

## 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を汎用の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をサポートしていると主張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](#)"
- [\[バケットを編集する\]](#)
- [\[デフォルトバケットを設定する\]](#)
- [\[バケットのクレデンシャルをローテーションするか、削除する\]](#)
- [\[バケットを削除する\]](#)
- "[\[Tech preview\]カスタムリソースを使用したバケットの管理](#)"



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 ) ]列のメニューから、[\*編集 ( Edit) ]を選択します。
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。

## デフォルトバケットを設定する

クラスタ間でクローニングを実行する場合、Astra Controlにはデフォルトバケットが必要です。すべてのクラ

スタにデフォルトバケットを設定するには、次の手順を実行します。

#### 手順

1. 「\* Cloud Instances \*」に移動します。
2. リスト内のクラウドインスタンスの\*アクション\*列でメニューを選択します。
3. 「\* 編集 \*」を選択します。
4. [\* Bucket\*]リストで、デフォルトにするバケットを選択します。
5. [保存 ( Save ) ]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。
- バケットのクラウドプロバイダの保持期間が終了する前にWrite Once Read Many (WORM) バケットを削除することはできません。WORMバケットは、バケット名の横に「Locked」と表示されます。

- 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。

作業を開始する前に

- 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。

あれば続けることはできません。

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。

## [Tech preview]カスタムリソースを使用したバケットの管理

アプリケーションクラスターでAstra Controlのカスタムリソース (CR) を使用してバケットを追加できます。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、クラスター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。カスタムリソースメソッドを使用している場合、アプリケーションのスナップショット機能にはバケットが必要です。

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ーにクローニングする場合、Astra Controlにバケットを作成する必要はありません。

Astra ControlのバケットカスタムリソースはAppVaultと呼ばれます。このCRには、保護処理でバケットを使用するために必要な設定が含まれています。

作業を開始する前に

- Astra Control Centerで管理されているクラスターから到達できるバケットを用意します。
- バケットのクレデンシャルがあることを確認します。
- バケットが次のいずれかのタイプであることを確認します。
  - NetApp ONTAP S3
  - NetApp StorageGRID S3 の略

- Microsoft Azure
- 汎用 S3



Amazon Web Services (AWS) では、Generic S3バケットタイプが使用されます。



Astra Control CenterはAmazon S3を汎用の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をサポートしていると主張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。

## 手順

1. カスタムリソース (CR) ファイルを作成し、という名前を付けます (例: `astra-appvault.yaml`) 。
2. 次の属性を設定します。
  - `* metadata.name*:` `_` (必須) `_` AppVaultカスタムリソースの名前。
  - `* spec.prefix *:` (オプション) AppVaultに保存されているすべてのエンティティの名前のプレフィックスが付いたパス。
  - `* spec.providerConfig*:` `_` (必須) `_` 指定されたプロバイダを使用してAppVaultにアクセスするために必要な設定を保存します。
  - `* spec.providerCredentials*:` `_` (必須) `_` 指定されたプロバイダを使用してAppVaultにアクセスするために必要なすべての資格情報への参照を保存します。
    - `* spec.providerCredentials.valueFromSecret*:` `_` (オプション) `_` は、クレデンシャル値がシークレットから取得される必要があることを示します。
      - `* key *:` `_` (valueFromSecretを使用する場合は必須) `_` 選択するシークレットの有効なキー。
      - `* name *:` `_` (valueFromSecretを使用する場合は必須) `_` このフィールドの値を含むシークレットの名前。同じネームスペースになければなりません。
  - `* spec.providerType*:` `_` (必須) `_` バックアップの提供元を決定します (例: NetApp ONTAP S3、Microsoft Azure) 。

YAMLの例:

```
apiVersion: astra.netapp.io/v1
kind: AppVault
metadata:
  name: astra-appvault
spec:
  providerType: generic-s3
  providerConfig:
    path: testpath
    endpoint: 192.168.1.100:80
    bucketName: bucket1
    secure: "false"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        name: s3-creds
        key: accessKeyID
    secretAccessKey:
      valueFromSecret:
        name: s3-creds
        key: secretAccessKey
```

3. データを入力した後、astra-appvault.yaml 正しい値を持つファイルを作成し、CRを適用します。

```
kubectl apply -f astra-appvault.yaml -n astra-connector
```



バケットを追加すると、デフォルトのバケットインジケータで1つのバケットがAstra Controlによってマークされます。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。バケットを追加する際、あとでを選択できます ["別のデフォルトバケットを設定する"](#)。

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。

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認証の詳細を編集します]
- [検出された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の管理を解除します]
- [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。

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. ダッシュボードのストレージバックエンドパネルで状態を確認します。
  - \* 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

バックエンドの健全性、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。

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。

## ストレージバックエンド認証の詳細を編集します

Astra Control Centerには、ONTAP バックエンドの認証に2つのモードがあります。

- クレデンシャルベースの認証：必要な権限を持つONTAP ユーザのユーザ名とパスワード。ONTAP のバージョンとの互換性を最大限に高めるには、adminなどの事前定義されたセキュリティログインロールを使用する必要があります。
- 証明書ベースの認証：Astra Control Centerは、バックエンドにインストールされている証明書を使用してONTAP クラスタと通信することもできます。クライアント証明書、キー、および信頼されたCA証明書を使用する（推奨）。

既存のバックエンドを更新して、あるタイプの認証から別の方法に移行できます。一度にサポートされる認証

方式は1つだけです。

証明書ベースの認証の有効化の詳細については、を参照してください ["ONTAP ストレージバックエンドで認証を有効にします"](#)。

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. [Credentials]フィールドで、\*[Edit]\*アイコンを選択します。
4. [Edit]ページで、次のいずれかを選択します。
  - 管理者のクレデンシャルを使用：ONTAP クラスタ管理IPアドレスと管理者のクレデンシャルを入力します。クレデンシャルはクラスタ全体のクレデンシャルである必要があります。



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、アクセス方法が指定された「admin」ロールのユーザクレデンシャルを適用します ontapi および http、ソースとデスティネーションの両方のONTAP クラスタ。を参照してください ["ONTAP ドキュメントの「ユーザーアカウントの管理」を参照してください"](#) を参照してください。

- 証明書を使用：証明書をアップロードします .pem ファイル、証明書キー .key ファイルを指定し、必要に応じて認証局ファイルを指定します。
5. [保存（ Save ） ]を選択します。

## 検出されたストレージバックエンドを管理します

管理対象外で検出されたストレージバックエンドを管理するように選択できます。ストレージバックエンドを管理する際に、Astra Controlに認証用の証明書の有効期限が切れているかどうかが表示されます。

手順

1. 左のナビゲーションから、 \* Backends \* を選択します。
2. [Discovered]\*オプションを選択します。
3. ストレージバックエンドを選択します。
4. [オプション]メニューの\*列から、[管理]\*を選択します。
5. 変更を行います。
6. [保存（ Save ） ]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。

作業を開始する前に

- 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定します。
  - d. 「\* Yes、remove storage backend \*」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"](#)

## 実行中のタスクを監視します

Astra Controlの過去24時間に完了、失敗、またはキャンセルされた実行中のタスクとタスクの詳細を表示できます。たとえば、実行中のバックアップ、リストア、またはクローン処理のステータスを表示して、完了した割合や推定残り時間などの詳細を確認できます。実行済みのスケジュール済み処理または手動で開始した処理のステータスを表示できます。

実行中または完了済みのタスクを表示している間に、タスクの詳細を展開して、各サブタスクのステータスを確認できます。進行中のタスクまたは完了したタスクの場合はタスクの進捗状況バーが緑色で表示され、キャンセルされたタスクの場合は青色で表示され、エラーのために失敗したタスクの場合は赤色で表示されます。



クローン処理の場合、タスクサブタスクはSnapshotとSnapshotのリストア処理で構成されま  
す。

失敗したタスクの詳細については、を参照してください "[アカウントのアクティビティを監視](#)".

手順

1. タスクの実行中に、「アプリケーション」に移動します。
2. リストからアプリケーションの名前を選択します。
3. アプリケーションの詳細で、[タスク]タブを選択します。

現在または過去のタスクの詳細を表示したり、タスクの状態でフィルタリングしたりできます。



タスクは最大24時間、\*タスク\*リストに保存されます。を使用して、この制限とその他のタスク  
モニタの設定を行うことができます "[Astra Control API の略](#)".

## [技術プレビュー] CRSを使用したAstra Controlアプリケーションの管理

Kubernetesのカスタムリソース (CR) を使用してAstra Controlアプリケーションを管理  
次のオプションを使用できます。

- "[Kubernetesのカスタムリソースを使用してアプリケーションを定義する](#)"
- "[カスタムリソースを使用したバケットの管理](#)"

## Prometheus接続またはFluentd接続でインフラを監視

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インサイトを取得するには、Prometheusを設定する  
か、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 (サポートバンドル  
をNetApp Support Siteにアップロードするための) プロキシが必要な場合は、Astra Control Centerでプロキ  
シサーバを設定する必要があります。

- [Prometheusに接続](#)
- [Fluentd に接続します](#)

## NetApp Support Siteへの接続用プロキシサーバを追加する

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 (サポートバンドル  
をNetApp Support Siteにアップロードするための) プロキシが必要な場合は、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。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\*[編集]\*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。無効にする前に、他の接続が中断される可能性があることを警告するメッセージが表示されます。

## 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。

## Prometheusに接続

Prometheusを使用して、Astra Control Centerのデータを監視できます。Kubernetesクラスタの指標エンドポイントから指標を収集するようにPrometheusを設定したり、Prometheusを使用して指標データを表示したりすることもできます。

Prometheusの使用の詳細については、それぞれのドキュメントを参照してください "[Prometheusでの作業の開始](#)"。

### 必要なもの

PrometheusパッケージがAstra Control Centerクラスタ、またはAstra Control Centerクラスタと通信可能な別のクラスタにダウンロードしてインストールされていることを確認します。

の公式ドキュメントに記載されている手順に従ってください "[Prometheus をインストールする](#)"。

Prometheusは、Astra Control Center Kubernetesクラスタと通信する必要があります。PrometheusがAstra Control Centerクラスタにインストールされていない場合は、Astra Control Centerクラスタで実行されている指標サービスと通信できることを確認する必要があります。

### Prometheus を設定する

Astra Control Centerは、KubernetesクラスタのTCPポート9090で指標サービスを公開します。このサービスから指標を収集するには、Prometheus を設定する必要があります。

## 手順

1. Prometheusサーバにログインします。
2. クラスタエントリを追加します prometheus.yml ファイル。を参照してください yml ファイルで、クラスタに関する次のようなエントリをに追加します scrape\_configs section:

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



を設定した場合は `tls_config insecure_skip_verify` 終了: ``true``では、TLS暗号化プロトコルは必要ありません。

3. Prometheusサービスを再起動します。

```
sudo systemctl restart prometheus
```

## Prometheusにアクセスする

PrometheusのURLにアクセスします。

手順

1. ブラウザで、Prometheus URLをポート9090と入力します。
2. \* Status > Targets \*を選択して、接続を確認します。

## Prometheusでデータを表示する

Prometheusを使用してAstra Control Centerのデータを表示できます。

手順

1. ブラウザで、PrometheusのURLを入力します。
2. Prometheusメニューで\* Graph \*を選択します。
3. メトリクスエクスプローラを使用するには、[Execute]の横にあるアイコンを選択します。
4. 選択するオプション `scrape_samples_scraped` をクリックし、\* Execute \*を選択します。
5. 時間の経過に伴うサンプルのスクレイピングを確認するには、\* Graph \*を選択します。



複数のクラスターデータが収集された場合、各クラスターの指標は異なる色で表示されます。

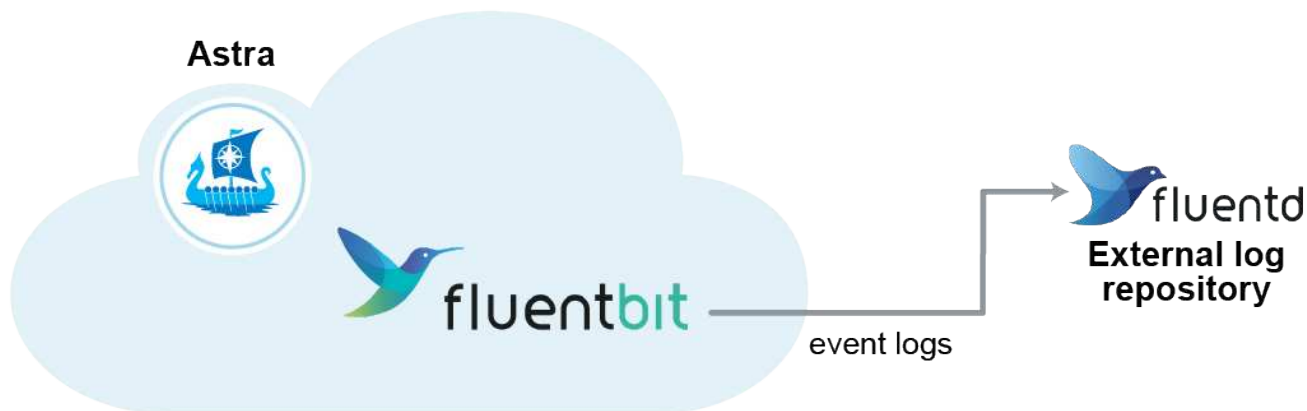
## Fluentd に接続します

Astra Control Centerで監視されているシステムから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。



Fluentd接続は、宣言型Kubernetesワークフローで管理されるクラスターではサポートされません。Fluentdは、非Kubernetesネイティブワークフローで管理されるクラスターにのみ接続できません。






 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。

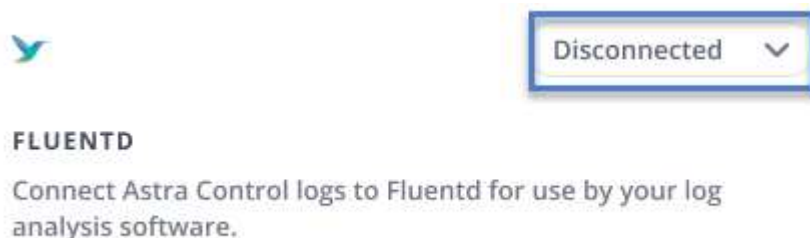
作業を開始する前に

- 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\*[編集]\*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。

アプリケーションの管理を解除すると、次のようになります。

- 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. アプリケーションを選択します。

3. [アクション]列の[オプション]メニューから、\*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。

#### 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をAstra Control Centerから管理しないようにします。



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。

クラスタの管理を解除する場合：

- 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。
- Astra Control ProvisionerまたはAstra Tridentはクラスタからアンインストールされません。"[Astra Trident のアンインストール方法をご確認ください](#)"。

#### 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. 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「Removing \*」に変わります。その後、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。



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ていたすべてのリソースが削除されます。

## 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、インストールイメージをダウンロードし、以下の手順を実行します。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。

以下の手順では、2番目に新しいリリースから今回の最新リリースへのAstra Control Centerのアップグレードプロセスについて説明します。現在のリリースより2つ以上遅れているバージョンから直接アップグレードすることはできません。インストールされているAstra Control Centerのバージョンが最新リリースの背後に多数のバージョンがある場合は、インストールされているAstra Control Centerのバージョンが最新リリースの

背後にあるまで、より新しいバージョンへのチェンアップグレードが必要になることがあります。リリースされたバージョンの完全なリストについては、"[リリースノート](#)"。

作業を開始する前に

アップグレードする前に、環境が "[Astra Control Center環境の最小要件](#)"。環境に次の要素が必要です。

- 有効 "[Astra Controlプロビジョニングツール](#)" **Astra Trident**を使用

a. 実行しているAstra Tridentのバージョンを確認します。

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



Astra Trident 23.01以前を実行している場合は、以下を使用 "[手順](#)" Astra Control Provisionerにアップグレードする前に、Astra Tridentの最新バージョンにアップグレードすること。Astra Tridentがバージョン24.02の4リリース期間内にある場合は、Astra Control Provisioner 24.02への直接アップグレードを実行できます。たとえば、Astra Trident 23.04からAstra Control Provisioner 24.02に直接アップグレードできます。

b. Astra Control Provisionerが "**有効**"。Astra Control Provisionerは、23.10より前のリリースのAstra Control Centerでは機能しません。最新の機能にアクセスするには、アップグレードするAstra Control Centerと同じバージョンのAstra Control Provisionerをアップグレードしてください。

- サポートされている**Kubernetes**ディストリビューション

実行しているKubernetesのバージョンを確認します。

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

- 十分なクラスタリソース

使用可能なクラスタリソースを確認します。

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

- デフォルトのストレージクラス

デフォルトのストレージクラスを確認します。

```
kubectl get storageclass
```

- 正常で利用可能な**API**サービス

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
```

- (ローカルレジストリのみ) **Astra Control Center**イメージのプッシュとアップロードに使用できるローカルレジストリ
- (**OpenShift**のみ) 正常で利用可能なクラスタオペレータ

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
```

また、次の点も考慮する必要があります。



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。

- \* NetApp Astra Controlイメージレジストリへのアクセス\*：  
Astra Control Provisionerなど、Astra Controlのインストールイメージや機能強化された機能をNetAppイメージレジストリから取得することができます。

- a. レジストリへのログインに必要なAstra ControlアカウントIDを記録します。

アカウントIDはAstra Control Service Web UIで確認できます。ページ右上の☒アイコンを選択し、\* APIアクセス\*を選択して、アカウントIDを書き留めます。

- b. 同じページから\* APIトークンの生成\*を選択し、APIトークン文字列をクリップボードにコピーしてエディターに保存します。
- c. Astra Controlレジストリにログインします。

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

- \* Istioサービスマッシュの導入\*  
Astra Control Centerのインストール時にIstioサービスマッシュをインストールした場合、このAstra Control CenterのアップグレードにはIstioサービスマッシュが含まれます。サービスマッシュをまだ持っていない場合は、"**初期導入**" Astra Control Centerの略。

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。



アップグレードを開始する前に、Astra Control Center UIからログアウトします。

- [Astra Control Centerをダウンロードして展開します](#)
- [\[ローカルレジストリを使用する場合は、追加の手順を実行します。\]](#)
- [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](#)
- [\[システムステータスを確認します\]](#)



Astra Control Centerオペレータ (たとえば、`kubectl delete -f astra_control_center_operator_deploy.yaml`) Astra Control Centerのアップグレードまたは操作中はいつでもポッドを削除しないようにします。

## Astra Control Centerをダウンロードして展開します

次のいずれかの場所からAstra Control Centerのイメージをダウンロードします。

- \* Astra Controlサービスのイメージレジストリ\* : Astra Control Centerのイメージでローカルレジストリを使用しない場合や、NetApp Support Siteからバンドルをダウンロードするよりもこの方法を使用する場合は、このオプションを使用します。
- \* NetApp Support Site \* : このオプションは、Astra Control Centerのイメージでローカルレジストリを使用する場合に使用します。

### Astra Controlイメージレジストリ

1. Astra Control Serviceにログインします。
2. ダッシュボードで、\*[Deploy a self-managed instance of Astra Control]\*を選択します。
3. 手順に従ってAstra Controlイメージのレジストリにログインし、Astra Control Centerのインストールイメージを取得してイメージを展開します。

### NetApp Support Site

1. Astra Control Centerを含む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 "[Astra Control Centerのダウンロードページ](#)"。
2. (推奨ですがオプション) Astra Control Centerの証明書と署名のバンドルをダウンロードします (`astra-control-center-certs-[version].tar.gz`) をクリックして、バンドルのシグネチャを確認します。

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

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

出力にはと表示されます `verified OK` 検証が成功したあとに、

3. Astra Control Centerバンドルからイメージを抽出します。

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

ローカルレジストリを使用する場合は、追加の手順を実行します。

Astra Control Centerバンドルをローカルのレジストリにプッシュする場合は、NetApp Astra kubectlコマンドラインプラグインを使用する必要があります。

**NetApp Astra kubectl**プラグインを削除して、再度インストールします

イメージをローカルのDockerリポジトリにプッシュするには、最新バージョンのNetApp Astra kubectlコマンドラインプラグインを使用する必要があります。

1. プラグインがインストールされているかどうかを確認します。

```
kubectl astra
```

2. 次のいずれかを実行します。

- プラグインがインストールされている場合は、コマンドによってkubectlプラグインのヘルプが返され、既存のバージョンのkubectl-Astraを削除できます。 `delete /usr/local/bin/kubectl-astra`。
- コマンドからエラーが返された場合は、プラグインがインストールされていません。次の手順に進んでインストールしてください。

3. プラグインをインストールします。

- a. 使用可能なNetApp Astra kubectlプラグインのバイナリを表示し、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。



kubectlプラグインライブラリはtarバンドルの一部であり、フォルダに解凍されます  
kubectl-astra。

```
ls kubectl-astra/
```

- a. 正しいバイナリを現在のパスに移動し、名前をに変更します kubectl-astra :

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

イメージをレジストリに追加する

1. Astra Control Centerバンドルをローカルのレジストリにプッシュする場合は、コンテナエンジンに応じた手順を実行します。



## Docker です

- a. tarballのルートディレクトリに移動します。次のように表示されます。  
acc.manifest.bundle.yaml ファイルと次のディレクトリ：

```
acc/  
kubect1-astra/  
acc.manifest.bundle.yaml
```

- b. Astra Control Centerのイメージディレクトリにある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。を実行する前に、次の置換を行ってください push-images コマンドを実行します

- `<BUNDLE_FILE>` をAstra Controlバンドルファイルの名前に置き換えます  
(acc.manifest.bundle.yaml)。
- `&lt;MY_FULL_REGISTRY_PATH&gt;` をDockerリポジトリのURLに置き換えます。次に例を示します。 "`<a href="https://&lt;docker-registry&gt;" class="bare">https://&lt;docker-registry&gt;"</a>`。
- `<MY_REGISTRY_USER>` をユーザ名に置き換えます。
- `<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。

```
kubect1 astra packages push-images -m <BUNDLE_FILE> -r  
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p  
<MY_REGISTRY_TOKEN>
```

## ポドマン

- a. tarballのルートディレクトリに移動します。次のファイルとディレクトリが表示されます。

```
acc/  
kubect1-astra/  
acc.manifest.bundle.yaml
```

- b. レジストリにログインします。

```
podman login <YOUR_REGISTRY>
```

- c. 使用するPodmanのバージョンに合わせてカスタマイズされた次のいずれかのスクリプトを準備して実行します。 `<MY_FULL_REGISTRY_PATH>` を'サブディレクトリを含むリポジトリのURLに置き換えます

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

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::~:')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

**Podman 3**

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::~:')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```



レジストリ設定に応じて、スクリプトが作成するイメージパスは次のようになります。

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

## 2. ディレクトリを変更します。

```
cd manifests
```

## 更新された **Astra Control Center** オペレータをインストールします

1. (ローカルレジストリのみ) ローカルレジストリを使用している場合は、次の手順を実行します。
  - a. Astra Control Centerオペレータによる導入YAMLを開きます。

```
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。

- b. 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換または編集します  
imagePullSecrets: [] 次の条件を満たす場合：

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

- c. 変更 `ASTRA_IMAGE_REGISTRY` 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](#)。
- d. 変更 `ASTRA_IMAGE_REGISTRY` をクリックします `acc-operator` 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](#)。
- e. に次の値を追加します `env` セクション。

```
- name: ACCOP_HELM_UPGRADE_TIMEOUT  
  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  
  strategy:  
    type: Recreate  
  template:  
    metadata:  
      labels:  
        control-plane: controller-manager
```

```
spec:
  containers:
  - args:
    - --secure-listen-address=0.0.0.0:8443
    - --upstream=http://127.0.0.1:8080/
    - --logtostderr=true
    - --v=10
    image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
    name: kube-rbac-proxy
    ports:
    - containerPort: 8443
      name: https
  - args:
    - --health-probe-bind-address=:8081
    - --metrics-bind-address=127.0.0.1:8080
    - --leader-elect
    env:
    - name: ACCOP_LOG_LEVEL
      value: "2"
    - name: ACCOP_HELM_UPGRADE_TIMEOUT
      value: 300m
    image: ASTRA_IMAGE_REGISTRY/acc-operator:24.02.68
    imagePullPolicy: IfNotPresent
    livenessProbe:
      httpGet:
        path: /healthz
        port: 8081
        initialDelaySeconds: 15
        periodSeconds: 20
    name: manager
    readinessProbe:
      httpGet:
        path: /readyz
        port: 8081
        initialDelaySeconds: 5
        periodSeconds: 10
    resources:
      limits:
        cpu: 300m
        memory: 750Mi
      requests:
        cpu: 100m
        memory: 75Mi
    securityContext:
      allowPrivilegeEscalation: false
    imagePullSecrets: []
```

```
securityContext:
  runAsUser: 65532
  terminationGracePeriodSeconds: 10
```

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

3. 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) を編集します。

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



注釈付きサンプルYAMLは以下の手順に従います。

2. Astraのバージョン番号を変更します (astraVersion の内部 spec) から 23.10.0 終了： 24.02.0 :



現在のリリースより2つ以上遅れているバージョンから直接アップグレードすることはできません。リリースされたバージョンの完全なリストについては、"[リリースノート](#)"。

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

3. イメージレジストリを変更します。

- (ローカルレジストリのみ) ローカルレジストリを使用している場合は、イメージのレジストリパスがイメージをプッシュしたレジストリパスと一致していることを確認します。 [前の手順](#)。更新 imageRegistry の内部 spec 前回のインストール後にローカルレジストリが変更された場合。
- (Astra Controlイメージレジストリ) Astra Controlイメージレジストリを使用 (cr.astra.netapp.io) を使用して、更新されたAstra Controlバンドルをダウンロードしました。

```
imageRegistry:
  name: "[cr.astra.netapp.io or your_registry_path]"
```

4. に次の項目を追加します crds の内部の設定 spec :

```
crds:
  shouldUpgrade: true
```

5. 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
  keycloak-operator:
    livenessProbe:
      initialDelaySeconds: 180
    readinessProbe:
      initialDelaySeconds: 180
```

6. ファイルエディタを保存して終了します。変更が適用され、アップグレードが開始されます。

7. (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 pods -n [netapp-acc or custom namespace]
```

8. アップグレードが完了して準備ができたことを示すため、Astra Controlのステータス状態が表示されるまで待ちます (True) :

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

対応:

NAME	UUID	VERSION	ADDRESS
READY			
astra	9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f	24.02.0-69	
10.111.111.111	True		



処理中のアップグレードステータスを監視するには、次のコマンドを実行します。  
kubectl get AstraControlCenter -o yaml -n [netapp-acc or custom namespace]



Astra Control Centerのオペレータログを調べるには、次のコマンドを実行します。  
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-operator -c manager -f

## 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. バージョンがアップグレードされたことを確認します。UIの\* Support \*ページを参照してください。
3. 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。

## OpenShift OperatorHubを使用したAstra Control Centerのアップグレード

Red Hat認定オペレータを使用してAstra Control Centerをインストールした場合は、OperatorHubから更新されたオペレータを使用してAstra Control Centerをアップグレードできます。この手順を使用して、["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。

作業を開始する前に

- 環境の前提条件を満たしている: アップグレードする前に、環境が ["Astra Control Center環境の最小要件"](#)。



- 有効になっていることを確認します。 **"Astra Controlプロビジョニングツール" Astra Trident**を使用

- a. 実行しているAstra Tridentのバージョンを確認します。

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



Astra Trident 23.01以前を実行している場合は、以下を使用 **"手順"** Astra Control Provisionerにアップグレードする前に、Astra Tridentの最新バージョンにアップグレードすること。Astra Tridentがバージョン24.02の4リリース期間内にある場合は、Astra Control Provisioner 24.02への直接アップグレードを実行できます。たとえば、Astra Trident 23.04からAstra Control Provisioner 24.02に直接アップグレードできます。

- b. Astra Control Provisionerが **"有効"**。Astra Control Provisionerは、23.10より前のリリースのAstra Control Centerでは機能しません。最新の機能にアクセスするには、アップグレードするAstra Control Centerと同じバージョンのAstra Control Provisionerをアップグレードしてください。
- 正常なクラスタオペレータとAPIサービスを確保：
    - 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。

```
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。

```
oc get apiservices
```

- \* OpenShiftの権限\*：ここに記載されているアップグレード手順を実行するために必要なすべての権限とRed Hat OpenShift Container Platformへのアクセス権が必要です。
- (ONTAP SANドライバのみ) マルチパスの有効化：ONTAP SANドライバを使用している場合は、すべてのKubernetesクラスタでマルチパスが有効になっていることを確認してください。

また、次の点も考慮する必要があります。

- \* NetApp Astra Controlイメージレジストリへのアクセス\*：

Astra Control Provisionerなど、Astra Controlのインストールイメージや機能強化された機能をNetAppイメージレジストリから取得することができます。

- a. レジストリへのログインに必要なAstra ControlアカウントIDを記録します。

アカウントIDはAstra Control Service Web UIで確認できます。ページ右上の☒アイコンを選択し、\* APIアクセス\*を選択して、アカウントIDを書き留めます。

- b. 同じページから\* APIトークンの生成\*を選択し、APIトークン文字列をクリップボードにコピーしてエディターに保存します。
- c. Astra Controlレジストリにログインします。

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

## 手順

- [\[オペレータインストールページへのアクセス\]](#)
- [\[既存のオペレータのアンインストール\]](#)
- [\[最新のオペレータのインストール\]](#)
- [Astra Control Center をアップグレードします](#)

## オペレータインストールページへのアクセス

1. OpenShift Container Platformまたはエコシステムカタログに対応する手順を完成させます。

## Red Hat OpenShift Webコンソール

- OpenShift Container Platform UI にログインします。
- サイドメニューから、\* 演算子 > OperatorHub \* を選択します。



このオペレータを使用している場合は、Astra Control Centerの最新バージョンにのみアップグレードできます。

- を検索します netapp-acc にアクセスし、NetApp Astra Control Centerオペレータを選択します。

The screenshot shows the Red Hat OpenShift console interface. On the left is a navigation sidebar with categories like Administrator, Home, Operators, Workloads, Networking, Storage, Builds, Observe, Compute, User Management, and Administration. The main content area is titled 'OperatorHub' and shows a search for 'netapp'. A card for 'netapp-acc-operator' is displayed, indicating it is 'Installed'. On the right, a detailed view of the operator is shown, including the 'Uninstall' button, 'Latest version' (24.2.0), 'Capability level' (Basic Install), 'Source' (Certified), 'Provider' (NetApp), 'Infrastructure features' (Disconnected), 'Repository' (N/A), and 'Container image' (registry.connect.redhat.co). A blue box highlights the 'Installed Operator' section, stating 'Version 23.10.0 of this Operator has been installed on the cluster. View it here.'

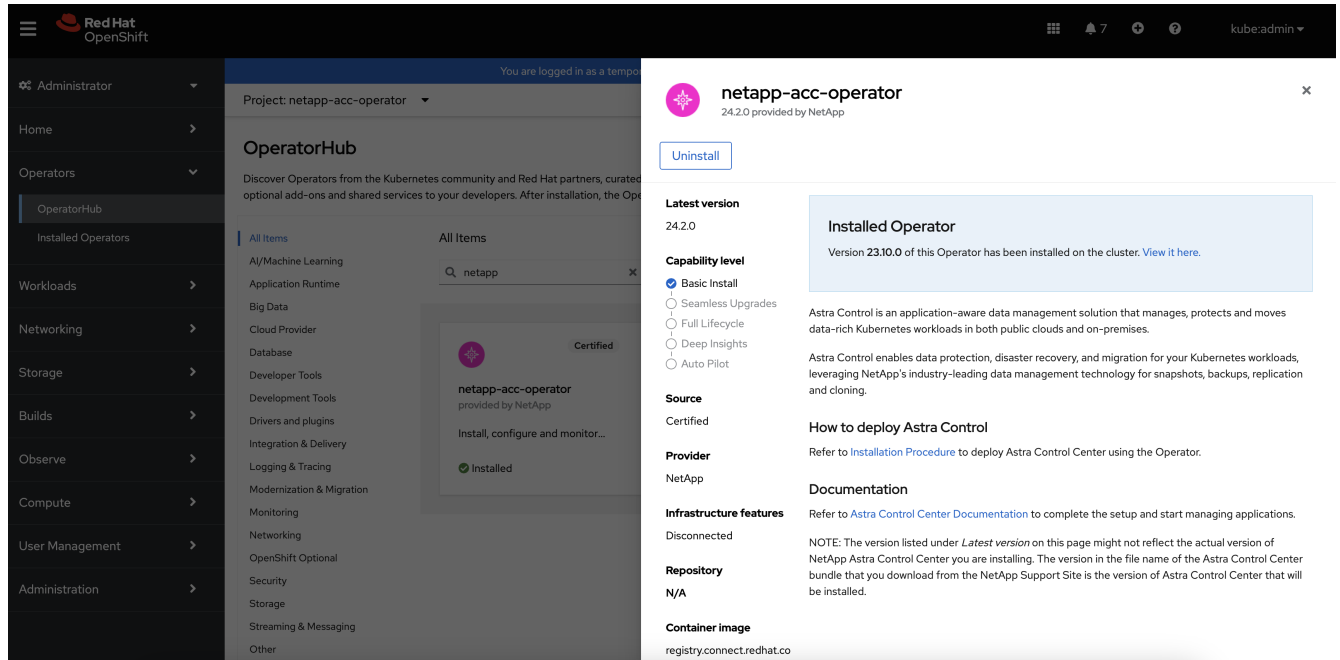
## Red Hat エコシステムカタログ

- NetApp Astra Control Center を選択します "演算子".
- [Deploy and Use]\*を選択します。

The screenshot shows the Red Hat Ecosystem Catalog page for Astra Control Center. The page header includes the Red Hat logo and 'Ecosystem Catalog' with navigation links for Hardware, Software, and Cloud & service providers. The breadcrumb trail is 'Home > Software > OpenShift operators > Astra Control Center'. The main heading is 'Astra Control Center' with the provider 'NetApp'. Below this is the description 'Application-aware data management built for OpenShift' and a prominent red 'Deploy and use' button. At the bottom, there are navigation links for Overview, Features & benefits, Documentation, Deploy & use, FAQs, and Get support. A 'Have feedback?' button is also visible in the bottom right corner.

## 既存のオペレータのアンインストール

1. [netapp-acc-operator]ページで、\*[アンインストール]\*を選択して既存のオペレータを削除します。



2. 操作を確定します。



この処理では、netapp-acc-operatorが削除されますが、関連付けられている元のネームスペースとリソース（シークレットなど）は保持されます。

## 最新のオペレータのインストール

1. に移動します netapp-acc オペレータページを再度表示します。
2. [Install Operator]ページに入力し、最新のオペレータをインストールします。

## Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

### Update channel \* ⓘ

stable

### Installation mode \*

- All namespaces on the cluster (default)  
Operator will be available in all Namespaces.
- A specific namespace on the cluster  
This mode is not supported by this Operator

### Installed Namespace \*

**⚠ Namespace already exists**  
Namespace `netapp-acc-operator` already exists and will be used. Other users can already have access to this namespace.

### Update approval \* ⓘ

- Automatic
- Manual

**netapp-acc-operator**  
provided by NetApp

Provided APIs

**ACC Astra Control Center**  
AstraControlCenter is the Schema for the astracontrolcenters API.



オペレータはすべてのクラスタ名前空間で使用できます。

- オペレータの `netapp-acc-operator` 削除されたオペレータの以前のインストールから残っている名前空間(またはカスタム名前空間)。
- 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。

- 「 \* Install \* 」を選択します。

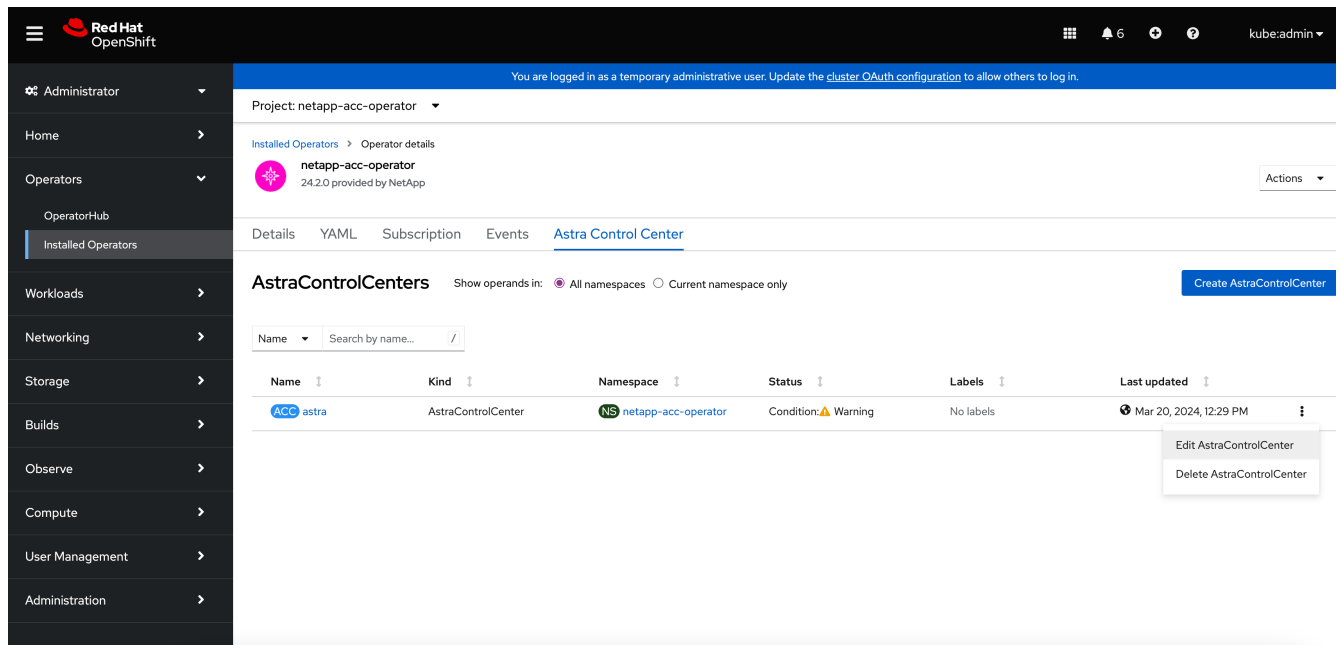


手動承認ストラテジーを選択した場合は、このオペレータの手動インストール計画を承認するように求められます。

- 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 をアップグレードします

- [Astra Control Center]の[Operator]タブで、前のインストールで使用していたAstra Control Centerを選択し、\*[Edit AstraControlCenter]\*を選択します。



## 2. を更新します AstraControlCenter YAML :

- Astra Control Centerの最新バージョン（24.02.0-69など）を入力します。
- インチ `imageRegistry.name` 必要に応じて、イメージレジストリパスを更新します。
  - Astra Controlレジストリオプションを使用している場合は、パスをに変更します。  
cr.astra.netapp.io。
  - ローカルレジストリを設定した場合は、前の手順でイメージをプッシュしたローカルイメージレジストリパスを変更または保持します。



入らないでください http:// または https:// をクリックします。

- を更新します imageRegistry.secret 必要に応じて。



オペレータによるアンインストールプロセスでは、既存のシークレットは削除されません。このフィールドを更新する必要があるのは、既存のシークレットとは異なる名前での新しいシークレットを作成する場合だけです。

- に次の項目を追加します crds 構成 :

```
crds:  
  shouldUpgrade: true
```

- 変更を保存します。
- アップグレードが正常に完了したことを示す画面が表示されます。

## 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 [\[アンインストールに関する問題のトラブルシューティング\]](#)。

作業を開始する前に

1. "すべてのアプリケーションの管理を解除します" クラスタ。
2. "すべてのクラスタの管理を解除します"。

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.yaml -n netapp-acc
```

結果

```
astracenter.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（またはカスタム名）名前空間：

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

結果の例：

```
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
```

- 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.
```

- 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
```

- 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

- 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
middlewareetcps.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
middlewareetcps.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"](#)

## 著作権に関する情報

Copyright © 2025 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权を有し、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权については、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。