



Astra Control Center 23.04のドキュメント

Astra Control Center

NetApp
November 21, 2023

目次

Astra Control Center 23.04のドキュメント	1
リリースノート	2
このリリースの Astra Control Center の新機能	2
既知の問題	6
既知の制限	9
はじめに	15
Astra Controlの詳細をご確認ください	15
Astra Control Center の要件	18
Astra Control Center のクイックスタート	23
インストールの概要	24
Astra Control Center をセットアップします	93
Astra Control Center に関するよくある質問	114
概念	117
アーキテクチャとコンポーネント	117
データ保護	118
ライセンス	121
アプリケーション管理	123
ストレージクラスと永続的ボリュームサイズ	125
ユーザロールとネームスペース	125
ポッドセキュリティ	126
Astra Control Center を使用	130
アプリの管理を開始します	130
アプリを保護します	136
アプリケーションとクラスタの健全性を監視	165
アカウントを管理します	168
バケットを管理する	178
ストレージバックエンドを管理します	181
実行中のタスクを監視します	185
Cloud Insights、Prometheus、Fluentd接続でインフラを監視します	186
アプリケーションとクラスタの管理を解除します	195
Astra Control Center をアップグレードします	196
Astra Control Center をアンインストールします	206
Astra Control REST APIで自動化	211
Astra Control REST API による自動化	211
知識とサポート	212
トラブルシューティング	212
ヘルプを表示します	212
以前のバージョンの Astra Control Center ドキュメント	215
法的通知	216

著作権	216
商標	216
特許	216
プライバシーポリシー	216
オープンソース	216
Astra Control API ライセンス	216

Astra Control Center 23.04のドキュメント

リリースノート

最新リリースのAstra Control Centerを発表しました。

- ["このリリースの Astra Control Center の内容"](#)
- ["既知の問題"](#)
- ["既知の制限"](#)

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

このリリースの Astra Control Center の新機能

最新リリースのAstra Control Centerを発表しました。

2023年5月18日 (23.04.2)

Astra Control Center (23.04.0) 向けのこのパッチリリース (23.04.2) では、がサポートされます ["Kubernetes CSI外部Snapshotコピーv6.1.0"](#) およびは、次の項目を修正します。

- 実行フックを使用する場合のインプレースアプリケーションリストアのバグ
- バケットサービスとの接続に問題があります

2023年4月25日 (23.04.0)

新機能とサポート

- ["Astra Control Centerの新規インストールでは、90日間の評価用ライセンスがデフォルトで有効になります"](#)
- ["強化された実行フック機能と追加のフィルタオプション"](#)
- ["Astra Control Centerでレプリケーションのフェイルオーバー後に実行フックを実行できるようになりました"](#)
- ["「ontap-nas-economy storage」クラスから「ontap-nas」ストレージクラスへのボリュームの移行がサポートされます"](#)
- ["リストア処理中のアプリケーションリソースの追加または除外がサポートされます"](#)
- ["データ専用アプリケーションの管理がサポートされます"](#)

既知の問題および制限事項

- ["このリリースの既知の問題"](#)
- ["このリリースの既知の制限事項は以下のとおりです"](#)

2022年11月22日 (22.11.0)

詳細

新機能とサポート

- "複数のネームスペースにまたがるアプリケーションのサポート"
- "アプリケーション定義にクラスタリソースを含めることができます"
- "ロールベースアクセス制御 (RBAC) を統合してLDAP認証を強化"
- "Kubernetes 1.25およびポッドセキュリティアドミSSION (PSA) のサポートを追加"
- "バックアップ、リストア、クローニングの各処理の進捗状況レポートが強化されました"

既知の問題および制限事項

- "このリリースの既知の問題"
- "このリリースの既知の制限事項は以下のとおりです"

2022年9月8日 (22.08.1)

詳細

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

2022年8月10日 (22.08.0)

詳細

新機能とサポート

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



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

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

既知の問題および制限事項

- ["このリリースの既知の問題"](#)
- ["このリリースの既知の制限事項は以下のとおりです"](#)

2022 年 4 月 26 日 (22.04.0)

詳細

新機能とサポート

- ["ネームスペースのロールベースアクセス制御 \(RBAC\) "](#)
- ["Cloud Volumes ONTAP のサポート"](#)
- ["Astra Control Center の一般的な入カインーブルメント"](#)
- ["Astra Control からバケットを取り外す"](#)
- ["VMware Tanzu ポートフォリオのサポート"](#)

既知の問題および制限事項

- ["このリリースの既知の問題"](#)
- ["このリリースの既知の制限事項は以下のとおりです"](#)

2021 年 12 月 14 日 (21.12)

詳細

新機能とサポート

- ["アプリケーションのリストア"](#)
- ["実行フック"](#)
- ["ネームスペースを対象とした演算子を使用して展開されたアプリケーションのサポート"](#)
- ["アップストリーム Kubernetes と Rancher もサポートしています"](#)
- ["Astra Control Center のアップグレード"](#)
- ["インストール用の Red Hat OperatorHub オプションです"](#)

解決済みの問題

- ["このリリースの解決済みの問題"](#)

既知の問題および制限事項

- ["このリリースの既知の問題"](#)
- ["このリリースの既知の制限事項は以下のとおりです"](#)

2021 年 8 月 5 日（ 21.08 ）

詳細

Astra Control Center の初回リリース。

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

詳細については、こちらをご覧ください

- ["このリリースの既知の問題"](#)
- ["このリリースの既知の制限事項は以下のとおりです"](#)
- ["以前のバージョンの Astra Control Center ドキュメント"](#)

既知の問題

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

現在のリリースに影響する既知の問題は次のとおりです。

アプリケーション

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

クラスター

- デフォルトの kubeconfig ファイルに複数のコンテキストが含まれている場合、Astra Control Center を使用したクラスターの管理が失敗します
- Astra Control Center 23.04へのアップグレード後に一部のポッドが起動しない
- [一部のポッドでは、23.04から23.04.2へのアップグレードのパージステージ後にエラー状態が表示されません]
- Istio環境で監視ポッドがクラッシュする可能性があります

その他の問題

- プロキシ経由で接続している場合、管理対象クラスターはNetApp Cloud Insights に表示されません
- Astra Trident がオフラインの場合、Internal Service Error（500）によりアプリケーションデータ管理処理が失敗する

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

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

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

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

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

OpenShift Container Platform クラスターのネームスペース内のサービスアカウントレベルで元のセキュリティコンテキストの制約が設定されていると、アプリケーションクローンが失敗する場合があります。アプリケーションクローンが失敗すると、Astra Control Centerの管理対象アプリケーション領域にステータスとともに

表示されます Removed。 を参照してください ["技術情報アーティクル"](#) を参照してください。

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

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

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

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

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

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

Astra Control Center 23.04へのアップグレード後に一部のポッドが起動しない

Astra Control Center 23.04にアップグレードすると、一部のポッドが起動しないことがあります。回避策として、次の手順に従って該当するポッドを手動で再起動します。

1. 影響を受けるポッドを特定し、<namespace> を現在のネームスペースに置き換えます。

```
kubectl get pods -n <namespace> | grep au-pod
```

影響を受けるポッドの結果は次のようになります。

```
pcloud-astra-control-center-au-pod-0 0/1 CreateContainerConfigError 0  
13s
```

2. 影響を受ける各ポッドを再起動し、<namespace> を現在のネームスペースに置き換えます。

```
kubectl delete pod pcloud-astra-control-center-au-pod-0 -n <namespace>
```

一部のポッドでは、**23.04**から**23.04.2**へのアップグレードのパージステージ後にエラー状態が表示されます

Astra Control Center 23.04.2にアップグレードすると、一部のポッドでエラーが表示されることがある

カンレンノログ task-service-task-purge :

```
kubectl get all -n netapp-acc -o wide|grep purge

pod/task-service-task-purge-28282828-ab1cd      0/1      Error      0
48m      10.111.0.111      openshift-clstr-ol-07-zwlj8-worker-jhp2b      <none>
<none>
```

このエラー状態は、クリーンアップステップが正しく実行されなかったことを意味します。23.04.2への全体的なアップグレードが成功しました。次のコマンドを実行してタスクをクリーンアップし、エラー状態を解消します。

```
kubectl delete job task-service-task-purge-[system-generated task ID] -n
<netapp-acc or custom namespace>
```

Istio環境で監視ポッドがクラッシュする可能性があります

Istio環境でAstra Control CenterをCloud Insights とペアリングする場合は telegraf-rs ポッドがクラッシュすることがあります。回避策として、次の手順を実行します。

1. クラッシュしたポッドを検索します。

```
kubectl -n netapp-monitoring get pod | grep Error
```

次のような出力が表示されます。

```
NAME READY STATUS RESTARTS AGE
telegraf-rs-fhhrh 1/2 Error 2 (26s ago) 32s
```

2. クラッシュしたポッドを再起動し、交換します <pod_name_from_output> 影響を受けるポッドの名前を入力します。

```
kubectl -n netapp-monitoring delete pod <pod_name_from_output>
```

次のような出力が表示されます。

```
pod "telegraf-rs-fhhrh" deleted
```

3. PODが再起動し、Error状態でないことを確認します。

```
kubectl -n netapp-monitoring get pod
```

次のような出力が表示されます。

```
NAME READY STATUS RESTARTS AGE
telegraf-rs-rrnsb 2/2 Running 0 11s
```

プロキシ経由で接続している場合、管理対象クラスタは**NetApp Cloud Insights** に表示されません

アストラコントロールセンターがプロキシ経由でネットアップCloud Insights に接続している場合、管理対象クラスタがCloud Insights に表示されないことがあります。回避策として、管理対象の各クラスタで次のコマンドを実行します。

```
kubectl get cm telegraf-conf -o yaml -n netapp-monitoring | sed
'\/[\[outputs.http\]]\/c\ [[outputs.http]]\n use_system_proxy =
true' | kubectl replace -f -
```

```
kubectl get cm telegraf-conf-rs -o yaml -n netapp-monitoring | sed
'\/[\[outputs.http\]]\/c\ [[outputs.http]]\n use_system_proxy =
true' | kubectl replace -f -
```

```
kubectl get pods -n netapp-monitoring --no-headers=true | grep 'telegraf-
ds\|telegraf-rs' | awk '{print $1}' | xargs kubectl delete -n netapp-
monitoring pod
```

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

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

詳細については、こちらをご覧ください

- ["既知の制限"](#)

既知の制限

既知の制限事項は、このリリースの製品でサポートされていないプラットフォーム、デ

バイス、機能、または製品と正しく相互運用できない機能を特定します。これらの制限事項を慎重に確認してください

クラスタ管理の制限事項

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

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

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

アプリケーション管理の制限

- [1つのネームスペース内の複数のアプリケーションをまとめて別のネームスペースにリストアすることはできません]
- Astra Controlでは、ネームスペースごとに複数のストレージクラスを使用するアプリケーションはサポートされていません
- Astra Controlでは、クラウドインスタンスにデフォルトのバケットは自動的に割り当てられません
- [パスバイリファレンス演算子を使用してインストールされたアプリケーションのクローンが失敗することがあります]
- 証明書マネージャを使用するアプリケーションの In Place リストア処理はサポートされていません
- OLM 対応およびクラスタ対象のオペレータ展開アプリケーションはサポートされていません
- Helm 2 で展開されたアプリケーションはサポートされていません

一般的な制限事項

- Astra Control Center の S3 バケットは、使用可能容量を報告しません
- Astra Control Center は、プロキシサーバー用に入力した詳細を検証しません
- Postgres ポッドへの既存の接続が原因で障害が発生します
- Astra Control Center インスタンスの削除中にバックアップとスナップショットが保持されない場合があります
- LDAPユーザおよびグループの制限事項
- <<[Activity]ページには、最大10万件のイベントが表示されます>>
- 特定のバージョンのSnapshotコントローラを含むKubernetes 1.25以降のクラスタでは、Snapshotが失敗することがあります

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

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

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

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

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

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

回避策として、次の手順を実行します。

1. を編集します `kubeadm-config` 構成マップ：

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

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

```
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data:
  ExAmPLERb2tCcJZ5K3E2Njk4eQotLExAMpLEORCBDRVJUSUZJQ0FURS0txxxxXX==
  server: https://x.x.x.x:6443
  name: xyz-cluster
contexts:
- context:
  cluster: xyz-cluster
  namespace: default
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
```

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

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

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

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

1つのネームスペース内の複数のアプリケーションをまとめて別のネームスペースにリストアすることはできません

複数のアプリケーションを1つのネームスペースで管理する場合（Astra Controlで複数のアプリケーション定義を作成する）、すべてのアプリケーションを別の1つのネームスペースにリストアすることはできません。各アプリケーションを専用のネームスペースにリストアする必要があります。

Astra Controlでは、ネームスペースごとに複数のストレージクラスを使用するアプリケーションはサポートされていません

Astra Controlは、ネームスペースごとに単一のストレージクラスを使用するアプリケーションをサポートします。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。

Astra Controlでは、クラウドインスタンスにデフォルトのバケットは自動的に割り当てられません

Astra Controlでは、どのクラウドインスタンスに対してもデフォルトのバケットが自動的に割り当てられることはありません。クラウドインスタンスのデフォルトバケットは手動で設定する必要があります。デフォルトのバケットが設定されていないと、2つのクラスタ間でアプリケーションのクローニング処理を実行できません。

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

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

- ["Apache K8ssandra"](#)



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

- ["Jenkins CI"](#)
- ["Percona XtraDB クラスタ"](#)

Astra Controlでは、「パスバイリファレンス」アーキテクチャ（CockroachDBオペレータなど）で設計されたオペレータをクローニングできない場合があります。クローニング処理では、クローニング処理の一環として

独自の新しいシークレットが存在する場合でも、クローニングされたオペレータがソースオペレータから Kubernetes シークレットを参照しようとして、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。

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

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

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

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

Helm 2 で展開されたアプリケーションはサポートされていません

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

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

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

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

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

Postgres ポッドへの既存の接続が原因で障害が発生します

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

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

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

LDAPユーザおよびグループの制限事項

Astra Control Centerは、最大5,000のリモートグループと10,000のリモートユーザをサポートします。

[Activity]ページには、最大10万件のイベントが表示されます

[Astra Control Activity]ページには、最大10,000件のイベントを表示できます。ログに記録されたすべてのイベントを表示するには、を使用してイベントを取得します ["Astra Control REST API"](#)。

特定のバージョンのSnapshotコントローラを含むKubernetes 1.25以降のクラスターでは、Snapshotが失敗することがあります

バージョン1.25以降を実行しているKubernetesクラスターのSnapshotは、クラスターにSnapshotコントローラAPIのバージョンv1beta1がインストールされている場合に失敗することがあります。

既存のKubernetes 1.25以降のインストールをアップグレードする場合は、回避策として次の手順を実行します。

1. 既存のSnapshot CRDと既存のSnapshotコントローラをすべて削除します。
2. ["Astra Trident をアンインストール"](#)。
3. ["スナップショットCRDとスナップショットコントローラをインストールします"](#)。
4. ["最新バージョンのAstra Tridentをインストール"](#)。
5. ["VolumeSnapshotClassを作成します"](#)。

詳細については、[こちら](#)をご覧ください

- ["既知の問題"](#)

はじめに

Astra Controlの詳細をご確認ください

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

機能

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

- 永続的ストレージを自動的に管理
- アプリケーション対応のオンデマンドの Snapshot とバックアップを作成
- ポリシーベースのスナップショットおよびバックアップ操作を自動化します
- Kubernetes クラスタ間でアプリケーションとデータを移行
- NetApp SnapMirrorテクノロジーを使用してアプリケーションをリモートシステムにレプリケート (Astra Control Center)
- ステージング環境から本番環境へのアプリケーションのクローニング
- アプリケーションの稼働状態と保護状態を視覚化します
- Web UIまたはAPIを使用して、バックアップと移行のワークフローを実装します

導入モデル

Astra Control には、次の 2 つの導入モデルがあります。

- *** Astra Control Service ***：ネットアップが管理するサービス。複数のクラウドプロバイダ環境や自己管理型のKubernetesクラスタで、アプリケーションに対応したデータ管理を提供します。
- *** Astra Control Center ***：オンプレミス環境で実行される Kubernetes クラスタのアプリケーション対応データ管理を提供する、自己管理ソフトウェアです。また、NetApp Cloud Volumes ONTAPストレージバックエンドを使用する複数のクラウドプロバイダ環境にAstra Control Centerをインストールすることもできます。

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

	Astra 制御サービス	Astra Control Center の略
サポートされているストレージバックエンドは何ですか。	<ul style="list-style-type: none"> • Amazon Web Servicesの特長 <ul style="list-style-type: none"> ◦ Amazon EBSのことです ◦ NetApp ONTAP 対応の Amazon FSX ◦ "Cloud Volumes ONTAP" • Google Cloud <ul style="list-style-type: none"> ◦ Google Persistent Disk のことす ◦ NetApp Cloud Volumes Service の略 ◦ "Cloud Volumes ONTAP" • Microsoft Azure <ul style="list-style-type: none"> ◦ Azure Managed Disksの略 ◦ Azure NetApp Files の特長 ◦ "Cloud Volumes ONTAP" • 自己管理クラスタ： <ul style="list-style-type: none"> ◦ Amazon EBSのことです ◦ Google Persistent Disk のことす ◦ Azure Managed Disksの略 ◦ "Cloud Volumes ONTAP" 	<ul style="list-style-type: none"> • NetApp ONTAP AFF および FAS システム • "Cloud Volumes ONTAP"

Astra Control Service の仕組み

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

Astra Control Service の概要は次のように機能します。

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

す。

Azure では、Astra Control Service によって、BLOB コンテナ用のリソースグループ、ストレージアカウント、およびキーも作成されます。

- クラスタに新しい admin ロールと Kubernetes サービスアカウントを作成します。
- 新しい admin ロールを使用してインストールします ["Astra Trident"](#) をクリックして、1 つ以上のストレージクラスを作成します。
- ネットアップのクラウドサービスストレージサービスをストレージバックエンドとして使用している場合、Astra Control ServiceはAstra Tridentを使用して、アプリケーション用の永続的ボリュームをプロビジョニングします。Amazon EBSまたはAzureで管理されているディスクをストレージバックエンドとして使用している場合は、プロバイダ固有のCSIドライバをインストールする必要があります。インストール手順については、[を参照してください "Amazon Web Servicesをセットアップする"](#) および ["Azure で管理されているディスクを使用して Microsoft Azure をセットアップする"](#)。
- この時点で、アプリケーションをクラスタに追加できます。永続ボリュームは、新しいデフォルトのストレージクラスでプロビジョニングされます。
- 次に、Astra Control Service を使用してこれらのアプリケーションを管理し、スナップショット、バックアップ、クローンの作成を開始します。

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

Astra Control Center の仕組み

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

Astra Control Centerは、ONTAP 9.5以上のストレージバックエンドを備えたAstra TridentベースのストレージクラスでKubernetesクラスタをサポートします。

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

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

90日間の組み込み評価用ライセンスを使用して、Astra Control Centerを試用できます。Astra Control Center の評価中は、Eメールとコミュニティのオプションでサポートを受けることができます。また、製品内サポートダッシュボードから技術情報アークティクルやドキュメントにアクセスすることもできます。

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

Astra Control Center の概要は次のように機能します。

- Astra Control Center は、ローカル環境にインストールします。方法の詳細については、[こちらをご覧ください "Astra Control Center をインストールします"](#)。
- 次のようなセットアップタスクを実行したとします。
 - ライセンスをセットアップする

- 最初のクラスタを追加します。
- クラスタを追加したときに検出されたストレージバックエンドを追加します。
- アプリケーションバックアップを格納するオブジェクトストアバケットを追加します。

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

クラスタにアプリケーションを追加できます。また、管理対象のクラスタにすでにアプリケーションがある場合は、Astra Control Centerを使用してそれらを管理できます。次に、Astra Control Centerを使用して、スナップショット、バックアップ、クローン、およびレプリケーション関係を作成します。

を参照してください。

- ["Astra Control Service のマニュアル"](#)
- ["Astra Control Center のドキュメント"](#)
- ["Astra Trident のドキュメント"](#)
- ["Astra Control API を使用"](#)
- ["Cloud Insights のドキュメント"](#)
- ["ONTAP のドキュメント"](#)

Astra Control Center の要件

運用環境、アプリケーションクラスタ、アプリケーション、ライセンス、Web ブラウザの準備ができているかどうかを検証します。Astra Control Centerを導入して運用するために、お客様の環境が上記の要件を満たしていることを確認してください。

- [サポート対象のホストクラスタKubernetes環境](#)
- [\[ホストクラスタリソースの要件\]](#)
- [Astra Trident の要件](#)
- [\[ストレージバックエンド\]](#)
- [\[イメージレジストリ\]](#)
- [Astra Control Centerのライセンス](#)
- [ONTAP ライセンス](#)
- [\[ネットワーク要件\]](#)
- [オンプレミス Kubernetes クラスタへの入力](#)
- [サポートされている Web ブラウザ](#)
- [\[アプリケーションクラスタのその他の要件\]](#)

サポート対象のホストクラスタKubernetes環境

Astra Control Centerは、次のKubernetesホスト環境で検証済みです。



Astra Control CenterをホストするKubernetes環境が、環境の公式ドキュメントに記載されている基本的なリソース要件を満たしていることを確認します。

ホストクラスタ上のKubernetesディストリビューション	サポートされるバージョン
Azure Stack HCIで実行されるAzure Kubernetes Service	Azure Stack HCI 21H2および22H2 (AKS 1.23および1.24を使用)
Google Anthos	1.12~1.14 (を参照してください Google Anthos Ingressの要件)
Kubernetes (アップストリーム)	1.24~1.26 (Kubernetes 1.25以降にはAstra Trident 22.10以降が必要)
Rancher Kubernetes Engine (RKE)	RKE 1.3とRancher 2.6 RKE 1.4とRancher 2.7 RKE 2 (v1.23.x) とRancher 2.6 RKE 2 (v1.24.x) とRancher 2.7
Red Hat OpenShift Container Platform	4.10から4.12まで
VMware Tanzu Kubernetesグリッド	1.6 ([ホストクラスタリソースの要件])
VMware Tanzu Kubernetes Grid統合エディション	1.14および1.15 ([ホストクラスタリソースの要件])

ホストクラスタリソースの要件

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



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

- * CPU拡張機能* : ホスティング環境のすべてのノードのCPUでAVX拡張機能が有効になっている必要があります。
- ワーカーノード: 少なくとも3つのワーカーノードで、それぞれ4つのCPUコアと12GBのRAMを備えています
- * VMware Tanzu Kubernetes Gridクラスタの要件* : VMware Tanzu Kubernetes Grid (TKG) またはTanzu Kubernetes Grid Integrated Edition (TKGi) クラスタでAstra Control Centerをホストする場合は、次の考慮事項に注意してください。
 - デフォルトの VMware TKG および TKGi 設定ファイルトークンの有効期限は、展開後 10 時間です。Tanzu ポートフォリオ製品を使用する場合は、Astra Control Center と管理対象アプリケーションクラスタ間の接続の問題を回避するために、期限切れにならないトークンを含む Tanzu Kubernetes Cluster 構成ファイルを生成する必要があります。手順については、[を参照してください "VMware NSX-T Data Center 製品ドキュメント"](#)
 - を使用します `kubectl get nsxlbmonitors -A` 入力トラフィックを受け入れるように設定されたサービスモニタがすでにあるかどうかを確認するコマンド。MetalLB が存在する場合は、既存のサービスモニタが新しいロードバランサ設定を上書きするため、MetalLB をインストールしないでください。
 - TKG または TKGi のデフォルト・ストレージ・クラス・エンフォースメントは、Astra Control によって管理されるすべてのアプリケーション・クラスタで無効にします。これを行うには、[を編集します](#)

TanzuKubernetesCluster ネームスペースクラスタ上のリソース。

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

Astra Trident の要件

お客様の環境のニーズに固有のAstra Tridentの次の要件を満たしていることを確認します。

- * Astra Control Centerで使用する最小バージョン*：Astra Trident 22.04以降のインストールと設定。
- * SnapMirrorレプリケーション*：SnapMirrorベースのアプリケーションレプリケーション用にAstra Trident 22.07以降がインストールされています。
- * Kubernetes 1.25以降のサポート*：Kubernetes 1.25以降のクラスタ用にインストールされたAstra Trident 22.10以降（Kubernetes 1.25以降にアップグレードする前にAstra Trident 22.10にアップグレードする必要があります）
- * Astra Tridentを使用したONTAP 構成*：
 - ストレージクラス：クラスタに少なくとも1つのAstra Tridentストレージクラスを設定します。デフォルトのストレージクラスが設定されている場合は、そのストレージクラスがデフォルトで指定された唯一のストレージクラスであることを確認します。
 - ストレージドライバとワーカーノード:ポッドがバックエンドストレージと対話できるように、クラスタ内のワーカーノードに適切なストレージドライバが設定されていることを確認します。Astra Control Center は、Astra Trident が提供する次の ONTAP ドライバをサポートしています。
 - `ontap-nas`
 - `ontap-san`
 - `ontap-san-economy`（このストレージクラスタイプではアプリケーションレプリケーションは使用できません）
 - `ontap-nas-economy`（Snapshot、レプリケーションポリシー、保護ポリシーは、このストレージクラスタイプでは使用できません）。

ストレージバックエンド

十分な容量を備えたサポート対象のバックエンドがあることを確認してください。

- サポートされるバックエンド：Astra Control Centerは次のストレージバックエンドをサポートします。
 - NetApp ONTAP 9.8以降のAFF、FAS、ASAシステム
 - NetApp ONTAP Select 9.8以降
 - NetApp Cloud Volumes ONTAP 9.8以降
- 必要なストレージバックエンド容量：500GB以上の空き容量

ONTAP ライセンス

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

- FlexClone

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

ONTAP システムに必要なライセンスがあるかどうかを確認するには、を参照してください "[ONTAPライセンスを管理します](#)"。

イメージレジストリ

Astra Control Centerのビルドイメージをプッシュできる既存のプライベートDockerイメージレジストリが必要です。イメージをアップロードするイメージレジストリの URL を指定する必要があります。

Astra Control Centerのライセンス

Astra Control CenterにはAstra Control Centerライセンスが必要です。Astra Control Centerをインストールすると、4、800 CPUユニットの90日間の評価用ライセンスがすでにアクティブ化されています。容量の追加や評価期間の変更が必要な場合や、フルライセンスにアップグレードする場合は、ネットアップから別の評価用ライセンスまたはフルライセンスを取得できます。アプリケーションとデータを保護するにはライセンスが必要です。

Astra Control Centerは無償トライアルにサインアップして試すことができます。登録することでサインアップできます "[こちらをご覧ください](#)"。

ライセンスをセットアップするには、を参照してください "[90 日間の評価版ライセンスを使用する](#)"。

ライセンスの機能の詳細については、を参照してください "[ライセンス](#)"。

ネットワーク要件

Astra Control Centerが適切に通信できるように運用環境を設定します。次のネットワーク設定が必要です。

- * FQDNアドレス*：Astra Control CenterのFQDNアドレスが必要です。
- インターネットへのアクセス：インターネットに外部からアクセスできるかどうかを判断する必要があります。この処理を行わないと、NetApp Cloud Insights からの監視データや指標データの受信や、へのサポートバンドルの送信など、一部の機能が制限される可能性があります "[NetApp Support Site](#)"。
- ポートアクセス：Astra Control Centerをホストする運用環境は、次のTCPポートを使用して通信します。これらのポートがファイアウォールを通過できることを確認し、Astra ネットワークからのHTTPS 出力トラフィックを許可するようにファイアウォールを設定する必要があります。一部のポートでは、Astra Control Center をホストする環境と各管理対象クラスター（該当する場合はメモ）の両方の接続方法が必要です。



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

ソース	宛先	ポート	プロトコル	目的
クライアントPC	Astra Control Center の略	443年	HTTPS	UI / API アクセス - Astra Control Center をホストしているクラスタと各管理対象クラスタの間で、このポートが双方向に開いていることを確認します
指標利用者	Astra Control Center ワーカーノード	9 -90だ	HTTPS	メトリックデータ通信 - 各管理対象クラスタが、アストラコントロールセンターをホストしているクラスタ上のこのポートにアクセスできることを確認します（双方向通信が必要）
Astra Control Center の略	Hosted Cloud Insights サービスの略 (https://www.netapp.com/cloud-services/cloud-insights/)	443年	HTTPS	Cloud Insights 通信
Astra Control Center の略	Amazon S3 ストレージバケットプロバイダ	443年	HTTPS	Amazon S3 ストレージ通信
Astra Control Center の略	NetApp AutoSupport (https://support.netapp.com)	443年	HTTPS	NetApp AutoSupport 通信

オンプレミス Kubernetes クラスタへの入力

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



ロードバランサは、Astra Control Center ワーカーノードのIPアドレスと同じサブネットにあるIPアドレスを使用する必要があります。

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

Google Anthos Ingressの要件

Google AnthosクラスタでAstra Control Centerをホストする場合、Google AnthosにはMetalLBロードバランサ

とIstio Ingressサービスがデフォルトで含まれているため、インストール時にAstra Control Centerの一般的な入力機能を簡単に使用できます。を参照してください ["Astra Control Center を設定します"](#) を参照してください。

サポートされている **Web** ブラウザ

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

アプリケーションクラスタのその他の要件

次のAstra Control Center機能を使用する場合は、次の要件に注意してください。

- アプリケーションクラスタの要件：["クラスタ管理の要件"](#)
 - アプリケーション要件の管理：["アプリケーション管理の要件"](#)
 - アプリケーション・レプリケーションの追加要件：["レプリケーションの前提条件"](#)

次のステップ

を表示します ["クイックスタート"](#) 概要（Overview）：

Astra Control Center のクイックスタート

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

1

Kubernetes クラスタの要件を確認

環境が次の要件を満たしていることを確認します。

- Kubernetesクラスタ*
- ["ホストクラスタが運用環境の要件を満たしていることを確認します"](#)
- ["オンプレミスKubernetesクラスタでロードバランシングを行うための入力を設定する"](#)

ストレージ統合

- ["サポートされているAstra Tridentバージョンが環境に含まれていることを確認します"](#)
- ["ワーカーノードを準備します"](#)
- ["Astra Tridentストレージバックエンドを設定"](#)
- ["Astra Tridentストレージクラスを設定する"](#)
- ["Astra Tridentボリュームスナップショットコントローラをインストール"](#)
- ["ボリュームSnapshotクラスを作成します"](#)
- ONTAP クレデンシャル*
- ["ONTAP クレデンシャルを設定する"](#)

2

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

次のインストールタスクを実行します。

- ["NetApp Support Site のダウンロードページからAstra Control Centerをダウンロードします"](#)
- ネットアップライセンスファイル入手します。
 - Astra Control Centerを評価する場合は、組み込みの評価用ライセンスがすでに付属しています
 - ["Astra Control Centerをすでに購入している場合は、ライセンスファイルを生成します"](#)
- ["Astra Control Center をインストールします"](#)
- ["追加のオプション設定手順を実行します"](#)

3

いくつかの初期セットアップ作業を完了します

開始するには、いくつかの基本的なタスクを実行します。

- ["ライセンスを追加します"](#)
- ["クラスタ管理のための環境を準備します"](#)
- ["クラスタを追加"](#)
- ["ストレージバックエンドを追加します"](#)
- ["バケットを追加します"](#)

4

Astra Control Center を使用

Astra Control Centerのセットアップが完了したら、Astra Control UIまたはを使用します ["Astra Control API の略"](#) アプリの管理と保護を開始するには：

- ["アプリの管理"](#):管理するリソースを定義します。
- ["アプリを保護します"](#)：保護ポリシーを構成し、アプリケーションのレプリケーション、クローニング、移行を行います。
- ["アカウントを管理"](#)：ユーザ、ロール、LDAP、クレデンシャルなど。
- ["必要に応じて、Cloud Insights に接続します"](#)：システムの健全性に関する指標を表示します。

を参照してください。

- ["Astra Control API の略"](#)
- ["Astra Control Center をアップグレードします"](#)
- ["Astra Controlのヘルプ"](#)

インストールの概要

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

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

環境によっては、Astra Control Centerのインストール後に追加の設定が必要になる場合があります。

- "インストール後にAstra Control Centerを設定します"

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

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

その他のインストール手順

- * RedHat OpenShift OperatorHub *でのインストール：これを使用してください ["代替手順"](#) OperatorHubを使用してOpenShiftにAstra Control Centerをインストールするには、次の手順を実行します。
- * Cloud Volumes ONTAP バックエンドを使用してパブリッククラウドにインストール*：ユース ["これらの手順に従います"](#) Amazon Web Services (AWS)、Google Cloud Platform (GCP)、またはCloud Volumes ONTAP ストレージバックエンドを使用するMicrosoft AzureにAstra Control Centerをインストールするには、次の手順を実行します。

Astra Control Centerのインストールプロセスのデモについては、を参照してください ["このビデオでは"](#)。

作業を開始する前に

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

```
kubectl get apiservices
```

- 使用するネットアップFQDNがこのクラスタにルーティング可能であることを確認します。つまり、内部DNSサーバにDNSエントリがあるか、すでに登録されているコアURLルートを使用しています。
- クラスタに証明書マネージャがすでに存在する場合は、いくつかの手順を実行する必要があります ["事前に必要な手順"](#) そのため、Astra Control Centerは独自の証明書マネージャのインストールを試みません。デフォルトでは、Astra Control Centerはインストール時に独自の証明書マネージャをインストールします。



3つ目の障害ドメインまたはセカンダリサイトにAstra Control Centerを導入これは、アプリケーションのレプリケーションとシームレスなディザスタリカバリに推奨されます。

このタスクについて

Astra Control Centerのインストールプロセスでは、次の作業を行うことができます。

- にAstraコンポーネントを取り付けます netapp-acc (またはカスタム名) ネームスペース。
- デフォルトのAstra Control Owner管理者アカウントを作成します。
- 管理ユーザのEメールアドレスとデフォルトの初期セットアップパスワードを設定します。このユーザには、UIへの初回ログインに必要なオーナーロールが割り当てられます。
- Astra Control Centerのすべてのポッドが稼働していることを確認します。
- Astra Control Center UIをインストールします。



Astra Control Centerオペレータ (たとえば、`kubectl delete -f astra_control_center_operator_deploy.yaml`) Astra Control Centerのインストール中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。

手順

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

- [Astra Control Centerをダウンロードして展開します](#)
- [ネットアップAstra kubectlプラグインをインストール](#)
- [\[イメージをローカルレジストリに追加します\]](#)
- [\[認証要件を持つレジストリのネームスペースとシークレットを設定します\]](#)
- [Astra Control Center オペレータを設置します](#)
- [Astra Control Center を設定します](#)
- [Astra Control Center とオペレータのインストールを完了します](#)
- [\[システムステータスを確認します\]](#)
- [\[ロードバランシング用の入力を設定します\]](#)
- [Astra Control Center UI にログインします](#)

Astra Control Centerをダウンロードして展開します

1. にアクセスします ["Astra Control Centerのダウンロードページ"](#) をクリックしますNetApp Support Site 。
2. 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`) 。
3. (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`) バンドルの署名を確認するには、次の手順を実行します。

```
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、

4. 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
```

ネットアップAstra kubectlプラグインをインストール

NetApp Astra kubectlコマンドラインプラグインを使用して、ローカルのDockerリポジトリにイメージをプッシュできます。

作業を開始する前に

ネットアップでは、CPUアーキテクチャやオペレーティングシステム別にプラグインのバイナリを提供しています。このタスクを実行する前に、使用しているCPUとオペレーティングシステムを把握しておく必要があります。

以前のインストールからプラグインがインストールされている場合は、["最新バージョンがインストールされていることを確認してください"](#) これらの手順を実行する前に。

手順

1. 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/
```

2. 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
```

イメージをローカルレジストリに追加します

1. コンテナエンジンに応じた手順を実行します。

Docker です

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

2. 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) 。
 - <MY_FULL_REGISTRY_PATH> 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://<docker-registry>" class="bare">https://<docker-registry>"。
 - <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。

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

ポドマン

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

2. レジストリにログインします。

```
podman login <YOUR_REGISTRY>
```

3. 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=23.04.2-7
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=23.04.2-7
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://netappdownloads.jfrog.io/docker-astra-control-
prod/netapp/astra/acc/23.04.2-7/image:version
```

認証要件を持つレジストリのネームスペースとシークレットを設定します

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


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



インストールを完了する前に、KUBECONFIGがAstra Control Centerをインストールするクラスターを指していることを確認してください。KUBECONFIGには、1つのコンテキストのみを含めることができます。

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

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

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

対応：

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

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



プレースホルダ `your_registry_path` 以前にアップロードした画像の場所と一致する必要があります（例： `[Registry_URL]/netapp/astra/astracc/23.04.2-7`）。

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

回答例：

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



シークレットの生成後にネームスペースを削除した場合は、ネームスペースを再作成し、ネームスペースのシークレットを再生成します。

c. を作成します netapp-acc（またはカスタム名）ネームスペース。

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

回答例：

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

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

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

応答

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

Astra Control Center オペレータを設置します

1. ディレクトリを変更します。

```
cd manifests
```

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

```
vim astra_control_center_operator_deploy.yaml
```



注釈付きサンプルYAMLは以下の手順に従います。

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

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

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

```
<strong>astra_control_center_operator_deploy.yaml</strong>
```

```

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: [your_registry_path]/kube-rbac-proxy:v4.8.0
          name: kube-rbac-proxy
          ports:
            - containerPort: 8443
              name: https
        - args:
            - --health-probe-bind-address=:8081
            - --metrics-bind-address=127.0.0.1:8080
            - --leader-elect
          env:
            - name: ACCOP_LOG_LEVEL
              value: "2"
            - name: ACCOP_HELM_INSTALLTIMEOUT
              value: 5m
            image: [your_registry_path]/acc-operator:23.04.36
          imagePullPolicy: IfNotPresent
          livenessProbe:
            httpGet:
              path: /healthz
              port: 8081
            initialDelaySeconds: 15

```

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

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

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

回答例：

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

4. ポッドが実行中であることを確認します

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

Astra Control Center を設定します

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

```
vim astra_control_center.yaml
```



注釈付きサンプルYAMLは以下の手順に従います。

2. 次の設定を変更または確認します。

`<code>accountName</code>`

設定	ガイダンス (Guidance)	を入力します	例
accountName	を変更します accountName stringには、Astra Control Centerアカウントに関連付ける名前を指定します。アカウント名は1つだけです。	文字列	Example

`<code>astraVersion</code>`

設定	ガイダンス (Guidance)	を入力します	例
astraVersion	導入するAstra Control Centerのバージョン。この設定には値があらかじめ入力されているため、対処は不要です。	文字列	23.04.2-7

<code>astraAddress</code>

設定	ガイダンス (Guidance)	を入力します	例
astraAddress	<p>を変更します</p> <p>astraAddress ブラウザで使用するFQDN (推奨) またはIPアドレスを指定して、Astra Control Centerにアクセスします。このアドレスは、データセンターでAstra Control Centerがどのように検出されるかを定義します。このアドレスは、完了時にロードバランサからプロビジョニングしたFQDNまたはIPアドレスと同じです "Astra Control Center の要件"。</p> <p>注：は使用しないでください http:// または https:// をクリックします。この FQDN をコピーしてで使 用しま す 後の手順。</p>	文字列	astra.example.com

`<code>autoSupport</code>`

このセクションで選択することで、ネットアップのプロアクティブサポートアプリケーション、NetApp Active IQ、およびデータの送信先のどちらに参加するかが決まります。インターネット接続が必要です（ポート442）。サポートデータはすべて匿名化されます。

設定	使用	ガイダンス (Guid ance)	を入力します	例
autoSupport.en rolled	または enrolled または url フィー ルドを選択する必 要があります	変更 enrolled を 選択しま すAutoSupport false インターネ ットに接続されて いないか、または 保持されているサ イト true 接続さ れているサイト 用。の設定 true サポート目的で匿 名データをNetApp に送信できるよう にします。デフォ ルトの選択はです false およびは、 サポートデータが ネットアップに送 信されないことを 示します。	ブール値	false (デフォル ト値)
autoSupport.ur l	または enrolled または url フィー ルドを選択する必 要があります	このURLは匿名デ ータの送信先を決 定します。	文字列	https://support.netapp.com/asupprod/post/1.0/postAsup

<code>email</code>

設定	ガイダンス (Guidance)	を入力します	例
email	を変更します email デフォルトの初期管理者アドレスを表す文字列。この E メールアドレスをコピーしてで使用します 後の手順 。この E メールアドレスは、最初のアカウントが UI にログインする際のユーザ名として使用され、Astra Control のイベントが通知されます。	文字列	admin@example.com

<code>firstName</code>

設定	ガイダンス (Guidance)	を入力します	例
firstName	アストラアカウントに関連付けられている初期管理者の名前。ここで使用した名前は、初回ログイン後に UI の見出しに表示されます。	文字列	SRE

<code>LastName</code>

設定	ガイダンス (Guidance)	を入力します	例
lastName	アストラアカウントに関連付けられている初期管理者の姓です。ここで使用した名前は、初回ログイン後に UI の見出しに表示されます。	文字列	Admin

<code>imageRegistry</code>

このセクションで選択すると、Astraアプリケーションイメージ、Astra Control Center Operator、Astra Control Center Helmリポジトリをホストするコンテナイメージレジストリが定義されます。

設定	使用	ガイダンス (Guidance)	を入力します	例
<code>imageRegistry.name</code>	必須	でイメージをプッシュしたイメージレジストリの名前の手順。使用しないでください http:// または https:// をレジストリ名に追加します。	文字列	<code>example.registry.com/astra</code>
<code>imageRegistry.secret</code>	に入力した文字列の場合は必須です <code>imageRegistry.name</code> requires a secret. IMPORTANT: If you are using a registry that does not require authorization, you must delete this <code>secret</code> ラインの内側 <code>imageRegistry</code> または、インストールが失敗します。	イメージレジストリでの認証に使用するKubernetesシークレットの名前。	文字列	<code>astra-registry-cred</code>

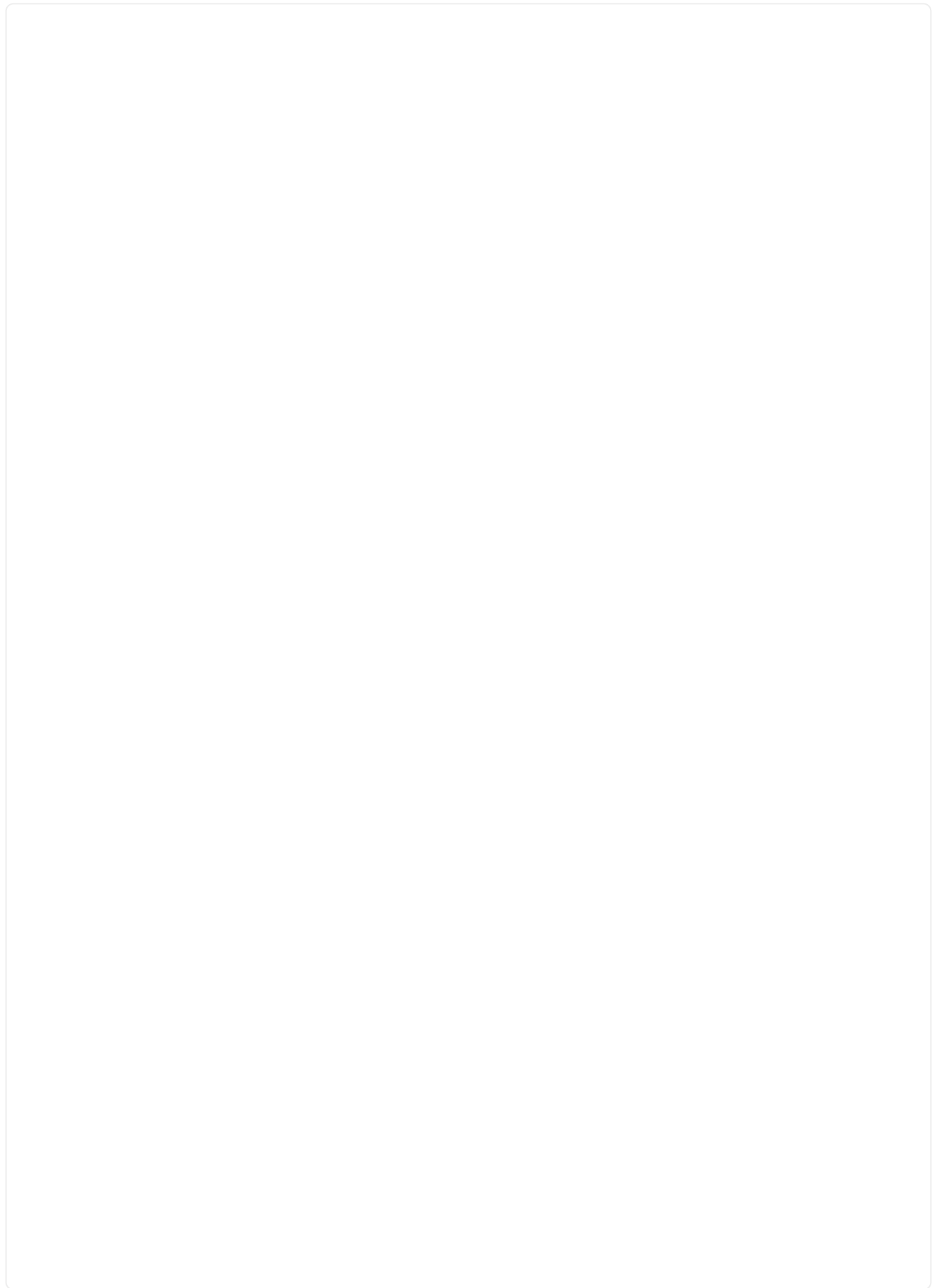
<code>storageClass</code>

設定	ガイダンス (Guidance)	を入力します	例
storageClass	<p>を変更します</p> <p>storageClass からの値 <code>ontap-gold</code> インストール環境に必要な別の Astra Trident storageClass リソースに移動します。コマンドを実行します</p> <p><code>kubectl get sc</code> をクリックして、設定済みの既存のストレージクラスを確認します。Astra Trident ベースのストレージクラスのいずれかをマニフェストファイルに入力する必要があります</p> <p>(astra-control-center-<code><version></code>.manifest) とを Astra PVS に使用します。設定されていない場合は、デフォルトのストレージクラスが使用されます。</p> <p>メモ：デフォルトのストレージクラスが設定されている場合は、デフォルトのアノテーションが設定されている唯一のストレージクラスであることを確認してください。</p>	文字列	ontap-gold

<code>volumeReclaimPolicy</code>

設定	ガイダンス (Guidance)	を入力します	オプション (Options)
volumeReclaimPolicy	これにより、AstraのPVSの再利用ポリシーが設定されます。このポリシーをに設定しています Retain Astraが削除されたあとに永続的なボリュームを保持このポリシーをに設定しています Delete Astraが削除されたあとに永続的ボリュームを削除する。この値が設定されていない場合、PVSは保持されま	文字列	<ul style="list-style-type: none">• Retain (デフォルト値)• Delete

`<code>ingressType</code>`





設定	ガイダンス (Guidance)	を入力します	オプション (Options)
ingressType	<p>次の入力タイプのいずれかを使用します。</p> <p>Generic (ingressType: "Generic") (デフォルト) このオプションは、別の入力コントローラを使用している場合、または独自の入力コントローラを使用する場合に使用します。Astra Control Centerを導入したら、を設定する必要があります "入力コントローラ" URLを使用してAstra Control Centerを公開します。</p> <p>AccTraefik (ingressType: "AccTraefik") 入力コントローラを設定しない場合は、このオプションを使用します。これにより、Astra Control Centerが導入されます traefik Gateway as a Kubernetes LoadBalancer type serviceの略。</p> <p>Astra Control Centerは、タイプ「LoadBalancer」のサービスを使用します。(svc/traefik Astra Control Centerの名前空間) で、アクセス可能な外部IPアドレスが割り当てられている必要があります。お使いの環境でロードバランサが許可されていて、設定されていない場合は、MetalLBまたは別の外部サービスロードバランサを使用して外部IPアドレスをサービスに割り当てることができます。内部 DNS サーバ構成では、Astra</p>	文字列	<ul style="list-style-type: none"> • Generic (デフォルト値) • AccTraefik

<code>scaleSize</code>

設定	ガイダンス (Guidance)	を入力します	オプション (Options)
scaleSize	<p>デフォルトでは、Astraで高可用性 (HA) が使用されます。</p> <p>scaleSize の Medium`ほとんどのサービスをHAに導入し、冗長性を確保するために複数のレプリカを導入します。を使用`scaleSizeとして Small`Astraは、消費量を削減するための必須サービスを除き、すべてのサービスのレプリカ数を削減します。</p> <p>ヒント： `Medium`環境は約100個のポッドで構成されています (一時的なワークロードは含まれません)。100個のポッドは、3つのマスターノードと3つのワーカーノード構成に基づいています)。特にディザスタリカバリのシナリオを検討する場合は、環境で問題となる可能性があるポッド単位のネットワーク制限に注意してください。</p>	文字列	<ul style="list-style-type: none">• Small• Medium (デフォルト値)

<code>astraResourcesScaler</code>

設定	ガイダンス (Guidance)	を入力します	オプション (Options)
<code>astraResourcesScaler</code>	<p>AstraeControlCenterリソース制限のスケールリングオプションデフォルトでは、Astra Control CenterはAstra内のほとんどのコンポーネントに対してリソース要求を設定して展開します。この構成により、アプリケーションの負荷と拡張性が高い環境では、Astra Control Centerソフトウェアスタックのパフォーマンスが向上します。</p> <p>ただし、小規模な開発またはテストクラスタを使用するシナリオでは、CRフィールドを使用します</p> <p><code>astraResourcesScaler</code> に設定できます</p> <p>Off。これにより、リソース要求が無効になり、小規模なクラスタへの導入が可能になります。</p>	文字列	<ul style="list-style-type: none">• Default (デフォルト値)• Off

`<code>additionalValues</code>`

- アストラルコントロールセンターおよびCloud Insights 通信では、TLS証明書の検証はデフォルトで無効になっています。の次のセクションを追加して、Cloud Insights とAstra Control Center のホストクラスタと管理対象クラスタの両方の間の通信に対してTLS証明書の検証を有効にすることができます additionalValues。

```
additionalValues:
  netapp-monitoring-operator:
    config:
      ciSkipTlsVerify: false
  cloud-insights-service:
    config:
      ciSkipTlsVerify: false
  telemetry-service:
    config:
      ciSkipTlsVerify: false
```

<code>crds</code>

このセクションで選択した内容によって、Astra Control CenterでのCRDの処理方法が決まります。

設定	ガイダンス (Guidance)	を入力します	例
<code>crds.externalCertManager</code>	<p>外部証明書マネージャを使用する場合は、変更します</p> <p><code>externalCertManager</code> 終了: <code>true</code>。デフォルト <code>false</code> Astra Control Centerが、インストール時に独自の証明書マネージャCRDをインストールするようにします。</p> <p>SSDはクラスタ全体のオブジェクトであり、クラスタの他の部分に影響を及ぼす可能性があります。このフラグを使用すると、これらのCRDがAstra Control Centerの外部にあるクラスタ管理者によってインストールおよび管理されることをAstra Control Centerに伝えることができます。</p>	ブール値	False (デフォルト値)
<code>crds.externalTraefik</code>	<p>デフォルトでは、Astra Control Centerは必要なTraefik CRDをインストールします。SSDはクラスタ全体のオブジェクトであり、クラスタの他の部分に影響を及ぼす可能性があります。このフラグを使用すると、これらのCRDがAstra Control Centerの外部にあるクラスタ管理者によってインストールおよび管理されることをAstra Control Centerに伝えることができます。</p>	ブール値	False (デフォルト値)



インストールを完了する前に、構成に適したストレージクラスと入力タイプを選択していることを確認してください。

```
<strong>astra_control_center.yaml</strong>
```

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues: {}
  crds:
    externalTraefik: false
    externalCertManager: false
```

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

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

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

回答例 :

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

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

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

回答例:

```
astracenter.astra.netapp.io/astra created
```



Astra Control Centerのオペレータが環境要件の自動チェックを実行ありません "要件" 原因でインストールが失敗するか、Astra Control Centerが正常に動作しない可能性があります。を参照してください [次のセクション](#) 自動システムチェックに関連する警告メッセージをチェックします。

システムステータスを確認します

kubectlコマンドを使用すると、システムステータスを確認できます。OpenShift を使用する場合は、同等のOC コマンドを検証手順に使用できます。

手順

1. インストールプロセスで検証チェックに関連する警告メッセージが生成されなかったことを確認します。

```
kubectl get acc [astra or custom Astra Control Center CR name] -n [netapp-acc or custom namespace] -o yaml
```



その他の警告メッセージは、Astra Control Centerのオペレータログでも報告されます。

2. 自動化された要件チェックによって報告された環境の問題を修正します。



問題を解決するには、環境が満たしていることを確認します "要件" (Astra Control Center向け)。

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

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

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

回答例

NAME	READY	STATUS	
RESTARTS AGE			
acc-helm-repo-6cc7696d8f-pmhm8 9h	1/1	Running	0
activity-597fb656dc-5rd4l 9h	1/1	Running	0
activity-597fb656dc-mqmcw 9h	1/1	Running	0
api-token-authentication-62f84 9h	1/1	Running	0
api-token-authentication-68nlf 9h	1/1	Running	0
api-token-authentication-ztgrm 9h	1/1	Running	0
asup-669d4ddbc4-fnmwp (9h ago) 9h	1/1	Running	1
authentication-78789d7549-1k686 9h	1/1	Running	0
bucket-service-65c7d95496-24x7l (9h ago) 9h	1/1	Running	3
cert-manager-c9f9fbf9f-k8zq2 9h	1/1	Running	0
cert-manager-c9f9fbf9f-qj1zm 9h	1/1	Running	0
cert-manager-cainjector-dbbbd8447-b5q1l 9h	1/1	Running	0
cert-manager-cainjector-dbbbd8447-p5whs 9h	1/1	Running	0
cert-manager-webhook-6f97bb7d84-4722b 9h	1/1	Running	0
cert-manager-webhook-6f97bb7d84-86kv5 9h	1/1	Running	0
certificates-59d9f6f4bd-2j899 9h	1/1	Running	0
certificates-59d9f6f4bd-9d9k6 9h	1/1	Running	0
certificates-expiry-check-28011180--1-8lkxz 9h	0/1	Completed	0
cloud-extension-5c9c9958f8-jdhrp 9h	1/1	Running	0
cloud-insights-service-5cdd5f7f-pp8r5 9h	1/1	Running	0
composite-compute-66585789f4-hxn5w	1/1	Running	0

9h			
composite-volume-68649f68fd-tb7p4	1/1	Running	0
9h			
credentials-dfc844c57-jsx92	1/1	Running	0
9h			
credentials-dfc844c57-xw26s	1/1	Running	0
9h			
entitlement-7b47769b87-4jb6c	1/1	Running	0
9h			
features-854d8444cc-c24b7	1/1	Running	0
9h			
features-854d8444cc-dv6sm	1/1	Running	0
9h			
fluent-bit-ds-9tlv4	1/1	Running	0
9h			
fluent-bit-ds-bpkcb	1/1	Running	0
9h			
fluent-bit-ds-cxmwx	1/1	Running	0
9h			
fluent-bit-ds-jgnhc	1/1	Running	0
9h			
fluent-bit-ds-vtr6k	1/1	Running	0
9h			
fluent-bit-ds-vxqd5	1/1	Running	0
9h			
graphql-server-7d4b9d44d5-zdbf5	1/1	Running	0
9h			
identity-6655c48769-4pwk8	1/1	Running	0
9h			
influxdb2-0	1/1	Running	0
9h			
keycloak-operator-55479d6fc6-slvmt	1/1	Running	0
9h			
krakend-f487cb465-78679	1/1	Running	0
9h			
krakend-f487cb465-rjsxx	1/1	Running	0
9h			
license-64cbc7cd9c-qxsr8	1/1	Running	0
9h			
login-ui-5db89b5589-ndb96	1/1	Running	0
9h			
loki-0	1/1	Running	0
9h			
metrics-facade-8446f64c94-x8h7b	1/1	Running	0
9h			
monitoring-operator-6b44586965-pvcl4	2/2	Running	0

9h			
nats-0	1/1	Running	0
9h			
nats-1	1/1	Running	0
9h			
nats-2	1/1	Running	0
9h			
nautilus-85754d87d7-756qb	1/1	Running	0
9h			
nautilus-85754d87d7-q8j7d	1/1	Running	0
9h			
openapi-5f9cc76544-7fnjm	1/1	Running	0
9h			
openapi-5f9cc76544-vzr7b	1/1	Running	0
9h			
packages-5db49f8b5-lrzhd	1/1	Running	0
9h			
polaris-consul-consul-server-0	1/1	Running	0
9h			
polaris-consul-consul-server-1	1/1	Running	0
9h			
polaris-consul-consul-server-2	1/1	Running	0
9h			
polaris-keycloak-0	1/1	Running	2
(9h ago) 9h			
polaris-keycloak-1	1/1	Running	0
9h			
polaris-keycloak-2	1/1	Running	0
9h			
polaris-keycloak-db-0	1/1	Running	0
9h			
polaris-keycloak-db-1	1/1	Running	0
9h			
polaris-keycloak-db-2	1/1	Running	0
9h			
polaris-mongodb-0	1/1	Running	0
9h			
polaris-mongodb-1	1/1	Running	0
9h			
polaris-mongodb-2	1/1	Running	0
9h			
polaris-ui-66fb99479-qp9gq	1/1	Running	0
9h			
polaris-vault-0	1/1	Running	0
9h			
polaris-vault-1	1/1	Running	0

9h	polaris-vault-2	1/1	Running	0
9h	public-metrics-76fbf9594d-zmxzw	1/1	Running	0
9h	storage-backend-metrics-7d7fbc9cb9-lmd25	1/1	Running	0
9h	storage-provider-5bdd456c4b-2fftc	1/1	Running	0
9h	task-service-87575df85-dnn2q	1/1	Running	3
(9h ago) 9h	task-service-task-purge-28011720--1-q6w4r	0/1	Completed	0
28m	task-service-task-purge-28011735--1-vk6pd	1/1	Running	0
13m	telegraf-ds-2r2kw	1/1	Running	0
9h	telegraf-ds-6s9d5	1/1	Running	0
9h	telegraf-ds-96jl7	1/1	Running	0
9h	telegraf-ds-hbp84	1/1	Running	0
9h	telegraf-ds-plwzv	1/1	Running	0
9h	telegraf-ds-sr22c	1/1	Running	0
9h	telegraf-rs-4sbg8	1/1	Running	0
9h	telemetry-service-fb9559f7b-mk917	1/1	Running	3
(9h ago) 9h	tenancy-559bbc6b48-5msgg	1/1	Running	0
9h	traefik-d997b8877-7xpf4	1/1	Running	0
9h	traefik-d997b8877-9xv96	1/1	Running	0
9h	trident-svc-585c97548c-d25z5	1/1	Running	0
9h	vault-controller-88484b454-2d6sr	1/1	Running	0
9h	vault-controller-88484b454-fc5cz	1/1	Running	0
9h	vault-controller-88484b454-jktld	1/1	Running	0
9h				

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

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



`accHost` クラスタの登録は最後の処理の1つです。登録に失敗しても原因の導入は失敗しません。ログにクラスタ登録エラーが記録されている場合は、を使用して再度登録を試行できます ["UIでクラスタワークフローを追加します"](#) または API。

5. すべてのポッドが実行中の場合は、インストールが正常に完了したことを確認します (`READY` は `True`) を使用して、Astra Control Centerにログインするときに使用する初期セットアップパスワードを取得します。

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

対応：

NAME	UUID	VERSION	ADDRESS
READY			
astra	9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f	23.04.2-7	10.111.111.111
True			



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

ロードバランシング用の入力を設定します

サービスへの外部アクセスを管理するKubernetes入力コントローラを設定できます。これらの手順では、デフォルトのを使用した場合の入力コントローラの設定例を示します `ingressType: "Generic" Astra Control Centerのカスタムリソース (astra_control_center.yaml)`。を指定した場合、この手順を使用する必要はありません `ingressType: "AccTraefik" Astra Control Centerのカスタムリソース (astra_control_center.yaml)`。

Astra Control Center を展開したら、Astra Control Center を URL で公開するように入力コントローラを設定する必要があります。

セットアップ手順は、使用する入力コントローラのタイプによって異なります。Astra Control Centerは、多くの入力コントローラタイプをサポートしています。これらのセットアップ手順では、次の入力コントローラタイプの手順の例を示します。

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

作業を開始する前に

- が必要です **"入力コントローラ"** すでに導入されている必要があります。
- **"入力クラス"** 入力コントローラに対応するものがすでに作成されている必要があります。

Istio Ingressの手順

1. Istio Ingressを設定します。



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

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

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

コマンド例：

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

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

コマンド例：

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



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

4. に入力リソースを配置します `netapp-acc` (またはカスタムネームスペース)。スキーマにはv1リソースタイプを使用します (`istio-ingress.yaml` は次の例で使用されています)。

```

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

```

5. 変更を適用します。

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

6. 入力ステータスを確認します。

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

対応:

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

7. Astra Control Centerのインストールを完了します。

Ngix Ingress Controller の手順

1. タイプのシークレットを作成します `kubernetes.io/tls` でTLSの秘密鍵と証明書を使用する場合 `netapp-acc` (またはカスタム名前付き) ネームスペース。を参照してください "[TLS シークレット](#)"。
2. 入力リソースをに配置します `netapp-acc` (またはカスタムネームスペース)。スキーマにはv1リソースタイプを使用します (`nginx-Ingress.yaml` は次の例で使用されています)。

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

3. 変更を適用します。

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



ネットアップでは、nginxコントローラをではなく導入環境としてインストールすることを推奨します `daemonSet`。

OpenShift 入力コントローラの手順

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

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

Astra Control Center UI にログインします

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

手順

1. ブラウザで、（を含む）FQDNを入力します `https://` プレフィックス）を使用します `astraAddress` を参照してください `astra_control_center.yaml` CR When（時間） [Astra Control Center をインストールした](#)。
2. プロンプトが表示されたら、自己署名証明書を承認します。



カスタム証明書はログイン後に作成できます。

3. Astra Control Centerのログインページで、に使用した値を入力します `email` インチ `astra_control_center.yaml` CR When（時間） [Astra Control Center をインストールした](#) をクリックし、次に初期セットアップパスワードを入力します (`ACC-[UUID]`) 。



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

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



初めてログインしたときにパスワードを忘れ、他の管理ユーザアカウントがまだ作成されていない場合は、にお問い合わせください ["ネットアップサポート"](#) パスワード回復のサポートを受けるには、

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

インストールのトラブルシューティングを行います

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

オプション（Options）

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

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

- Astra Control Center CRの出力を確認するには、次の手順を実行します。

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

次のステップ

- (オプション) お使いの環境に応じて、インストール後に実行します "設定手順".
- を実行して導入を完了します "セットアップのタスク".

外部証明書マネージャを設定します

Kubernetesクラスタに証明書マネージャがすでに存在する場合は、Astra Control Centerで独自の証明書マネージャがインストールされないように、いくつかの前提条件となる手順を実行する必要があります。

手順

1. 証明書マネージャがインストールされていることを確認します。

```
kubectl get pods -A | grep 'cert-manager'
```

回答例：

```
cert-manager    essential-cert-manager-84446f49d5-sf2zd    1/1
Running        0      6d5h
cert-manager    essential-cert-manager-cainjector-66dc99cc56-9ldmt    1/1
Running        0      6d5h
cert-manager    essential-cert-manager-webhook-56b76db9cc-fjqrq    1/1
Running        0      6d5h
```

2. の証明書とキーのペアを作成します astraAddress FQDN：

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

回答例：

```
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'tls.key'
```

3. 以前に生成したファイルを使用してシークレットを作成します。

```
kubectl create secret tls selfsigned-tls --key tls.key --cert tls.crt -n
<cert-manager-namespace>
```

回答例：

```
secret/selfsigned-tls created
```

4. を作成します ClusterIssuer *とまったく同じ*のファイル。ただし、の名前空間の場所が含まれます cert-manager ポッドがインストールされます。

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: astra-ca-clusterissuer
  namespace: <cert-manager-namespace>
spec:
  ca:
    secretName: selfsigned-tls
```

```
kubectl apply -f ClusterIssuer.yaml
```

回答例：

```
clusterissuer.cert-manager.io/astra-ca-clusterissuer created
```

5. を確認します ClusterIssuer が正常に起動しました。Ready はである必要があります True 次の手順に進む前に、次の手順

```
kubectl get ClusterIssuer
```

回答例：

NAME	READY	AGE
astra-ca-clusterissuer	True	9s

6. を実行します ["Astra Control Center のインストールプロセス"](#)。があります ["Astra Control Center クラスターYAMLの必須の設定手順"](#) CRD値を変更して、証明書マネージャが外部にインストールされていることを示します。Astra Control Centerが外部証明書マネージャを認識するように、インストール時にこの手順を完了する必要があります。

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

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

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

作業を開始する前に

- 環境前提条件を満たしている： ["インストールを開始する前に、Astra Control Center の導入環境を準備します"](#)。
- 健全なクラスタオペレータとAPIサービス：
 - 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
```

- * FQDN address * : データセンターのAstra Control CenterのFQDNアドレスを取得します。
- * OpenShift Permissions * : 説明されているインストール手順を実行するために必要な権限を取得し、Red Hat OpenShift Container Platformにアクセスします。
- **cert manager configure** 済み: クラスタにcertマネージャがすでに存在する場合はいくつかを実行する必要があります ["事前に必要な手順"](#)。そのため、Astra Control Centerは独自の証明書管理ツールをインストールしません。デフォルトでは、Astra Control Centerはインストール時に独自の証明書マネージャをインストールします。
- * Kubernetes入力コントローラ* : クラスタ内のロードバランシングなどのサービスへの外部アクセスを管理するKubernetes入力コントローラがある場合は、Astra Control Centerで使用するようにセットアップする必要があります。
 - a. operatorネームスペースを作成します。

```
oc create namespace netapp-acc-operator
```

- b. ["セットアップを完了"](#) 入力コントローラのタイプ。

手順

- [Astra Control Centerをダウンロードして展開します](#)
- [ネットアップAstra kubectiプラグインをインストール](#)
- [\[イメージをローカルレジストリに追加します\]](#)

- [\[オペレータインストールページを検索します\]](#)
- [\[オペレータをインストールします\]](#)
- [Astra Control Center をインストールします](#)

Astra Control Centerをダウンロードして展開します

1. にアクセスします "[Astra Control Centerのダウンロードページ](#)" をクリックしますNetApp Support Site。
2. 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)。
3. (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) バンドルの署名を確認するには、次の手順を実行します。

```
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、

4. 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
```

ネットアップAstra kubectlプラグインをインストール

NetApp Astra kubectlコマンドラインプラグインを使用して、ローカルのDockerリポジトリにイメージをプッシュできます。

作業を開始する前に

ネットアップでは、CPUアーキテクチャやオペレーティングシステム別にプラグインのバイナリを提供しています。このタスクを実行する前に、使用しているCPUとオペレーティングシステムを把握しておく必要があります。

手順

1. 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 kubect1-astra/
```

2. 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 kubect1-astra :

```
cp kubect1-astra/<binary-name> /usr/local/bin/kubect1-astra
```

イメージをローカルレジストリに追加します

1. コンテナエンジンに応じた手順を実行します。

Docker です

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

2. 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)。
 - <MY_FULL_REGISTRY_PATH> をDockerリポジトリのURLに置き換えます。次に例を示します。 "https://<docker-registry>"。
 - <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。

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

ポドマン

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

2. レジストリにログインします。

```
podman login <YOUR_REGISTRY>
```

3. 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=23.04.2-7
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=23.04.2-7
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
```



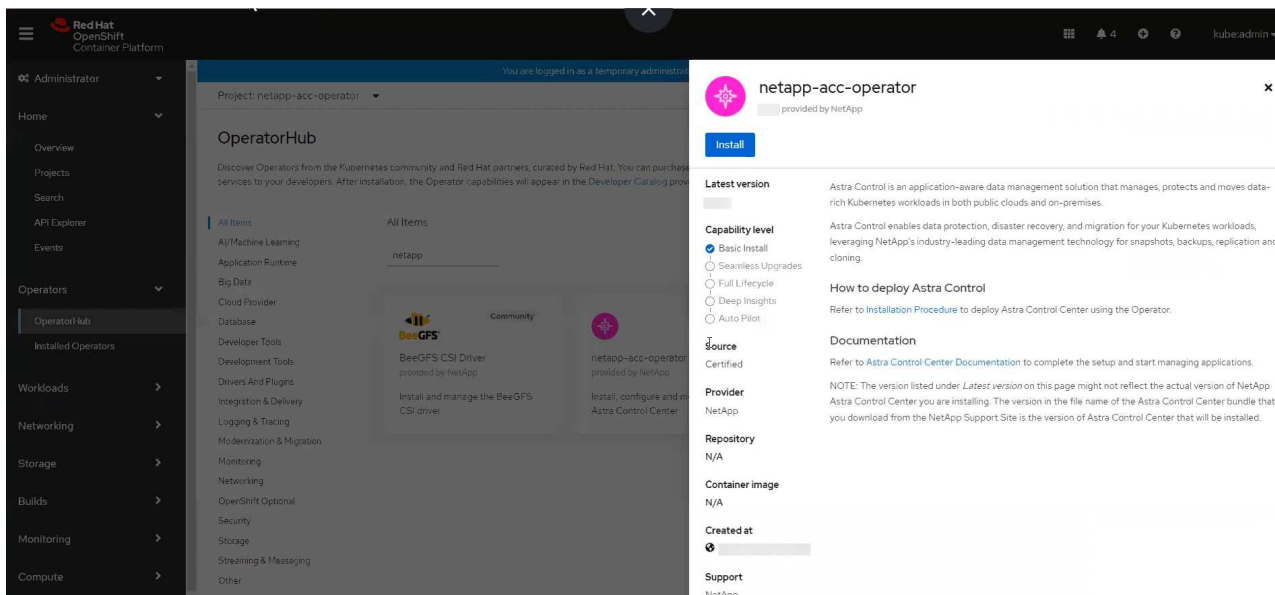
レジストリ設定に応じて、スクリプトが作成するイメージパスは次のようになります。

```
https://netappdownloads.jfrog.io/docker-astra-control-
prod/netapp/astra/acc/23.04.2-7/image:version
```

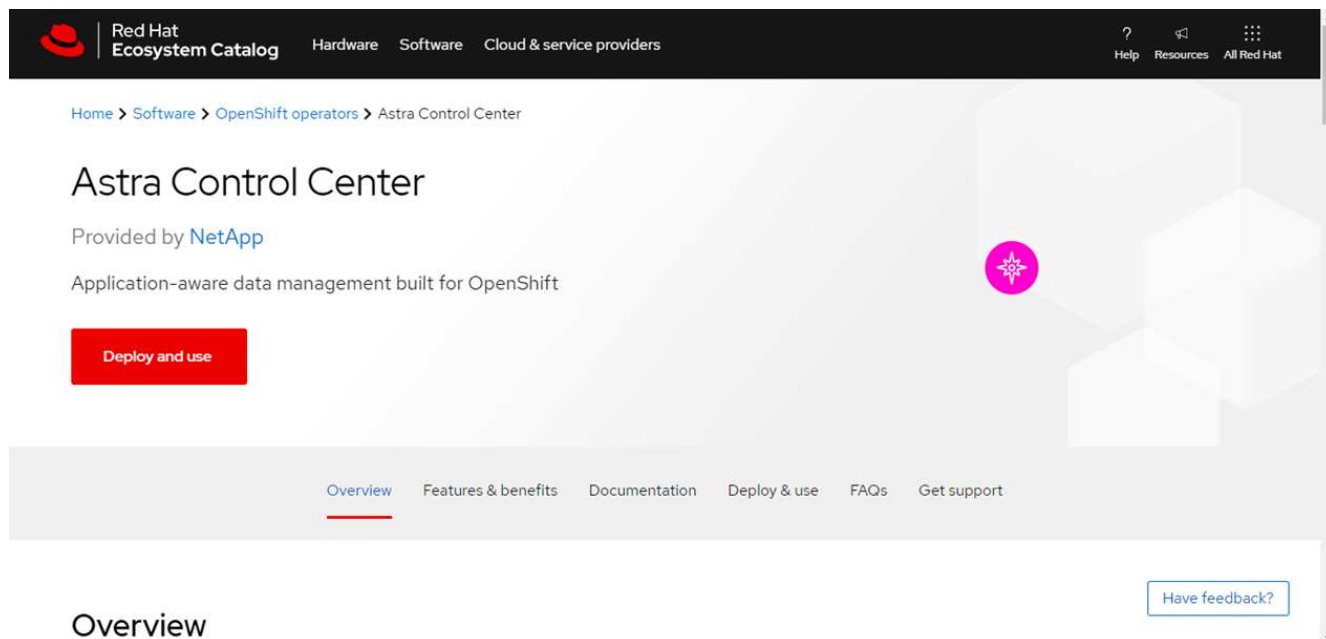
オペレータインストールページを検索します

1. 次のいずれかの手順を実行して、オペレータインストールページにアクセスします。

- Red Hat OpenShift の Web コンソールから：
 - i. OpenShift Container Platform UI にログインします。
 - ii. サイドメニューから、* 演算子 > OperatorHub * を選択します。
 - iii. NetApp Astra Control Centerオペレータを検索して選択します。



- Red Hat エコシステムカタログから：
 - i. NetApp Astra Control Center を選択します "演算子"。
 - ii. [Deploy and Use] を選択します。



オペレータをインストールします

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



オペレータはすべてのクラスタ名前空間で使用できます。

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



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

- c. 「 * Install * 」を選択します。



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

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

Astra Control Center をインストールします

1. Astra Control Centerオペレータの[Astra Control Center]タブ内のコンソールから[*Create AstraControlCenter *]を選択します

Project: netapp-acc-operator

Installed Operators > Operator details

netapp-acc-operator
23.4.0 provided by NetApp

Actions

Details YAML Subscription Events Astra Control Center

AstraControlCenters Show operands in: All namespaces Current namespace only [Create AstraControlCenter](#)

No operands found

Operands are declarative components used to define the behavior of the application.

2. を実行します Create AstraControlCenter フォームフィールド：
 - a. Astra Control Center の名前を保持または調整します。
 - b. Astra Control Centerのラベルを追加します。
 - c. AutoSupportを有効または無効にします。Auto Support 機能の保持を推奨します。
 - d. Astra Control CenterのFQDNまたはIPアドレスを入力します。入らないでください http:// または https:// をクリックします。
 - e. Astra Control Centerのバージョンを入力します（例：23.04.2-7）。
 - f. アカウント名、E メールアドレス、および管理者の姓を入力します。
 - g. ボリューム再利用ポリシーを選択してください Retain、Recycle`または `Delete。デフォルト値はです Retain。

h. インストールのscaleSizeを選択します。



デフォルトでは、Astraで高可用性 (HA) が使用されます。scaleSize の Medium`ほとんどのサービスをHAに導入し、冗長性を確保するために複数のレプリカを導入します。を使用 `scaleSize` として `Small` Astraは、消費量を削減するための必須サービスを除き、すべてのサービスのレプリカ数を削減します。

i. 入力タイプを選択します。

▪ **Generic** (ingressType: "Generic") (デフォルト)

このオプションは、別の入力コントローラを使用している場合、または独自の入力コントローラを使用する場合に使用します。Astra Control Centerを導入したら、を設定する必要があります "[入力コントローラ](#)" URLを使用してAstra Control Centerを公開します。

▪ **AccTraefik** (ingressType: "AccTraefik")

入力コントローラを設定しない場合は、このオプションを使用します。これにより、Astra Control Centerが導入されます traefik ゲートウェイをKubernetesの「LoadBalancer」タイプのサービスとして使用します。

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



「LoadBalancer」およびIngressのサービスタイプの詳細については、を参照してください "[要件](#)"。

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

b. 認証が必要なイメージレジストリを使用する場合は、イメージシークレットを入力します。



認証が必要なレジストリを使用する場合は、[クラスタでシークレットを作成](#)します。

c. 管理者の名を入力します。

d. リソースの拡張を構成する。

e. デフォルトのストレージクラスを指定します。



デフォルトのストレージクラスが設定されている場合は、そのストレージクラスがデフォルトのアノテーションを持つ唯一のストレージクラスであることを確認します。

f. CRD 処理の環境設定を定義します。

3. YAMLビューを選択して、選択した設定を確認します。

4. 選択するオプション Create。

レジストリシークレットを作成します

認証が必要なレジストリを使用する場合は、OpenShiftクラスタでシークレットを作成し、にシークレット名を入力します `Create AstraControlCenter` フォームフィールド。

1. Astra Control Centerオペレータの名前空間を作成します。

```
oc create ns [netapp-acc-operator or custom namespace]
```

2. この名前空間にシークレットを作成します。

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



Astra Controlは、Dockerレジストリシークレットのみをサポートします。

3. の残りのフィールドに値を入力します [Create AstraControlCenterフォーム・フィールド](#)。

次のステップ

を実行します ["残りのステップ"](#) Astra Control Centerが正常にインストールされたことを確認するには、入力コントローラ（オプション）をセットアップし、UIにログインします。また、を実行する必要があります ["セットアップのタスク"](#) インストールが完了したら、

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

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

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

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

- [Amazon Web Services に Astra Control Center を導入](#)
- [Astra Control CenterをGoogle Cloud Platformに導入](#)
- [Microsoft Azure に Astra Control Center を導入](#)

OpenShift Container Platform (OCP) などの自己管理型Kubernetesクラスタを使用して、ディストリビュー

ション内のアプリケーションを管理できます。Astra Control Centerを導入するために検証されるのは、自己管理型のOCPクラスタのみです。

Amazon Web Services に Astra Control Center を導入

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

AWSに必要なもの

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

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

AWS の運用環境の要件

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

- Red Hat OpenShift Container Platform 4.8 の場合



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

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

コンポーネント	要件
バックエンドの NetApp Cloud Volumes ONTAP ストレージ容量	300GB 以上のデータがあります
ワーカーノード (AWS EC2 の要件)	少なくとも 3 つのワーカーノードが必要です。vCPU コア 4 基、RAM はそれぞれ 12GB です
ロードバランサ	動作環境クラスタ内のサービスに送信される入力トラフィックに使用できるサービスタイプ「LoadBalancer」
FQDN	Astra Control Center の FQDN をロードバランシング IP アドレスに指定する方法

コンポーネント	要件
Astra Trident （以前の Cloud Manager で、 Kubernetes クラスタ検出の一部として NetApp BlueXP にインストール）	Trident 21.04 以降がインストールおよび設定され、NetApp ONTAP バージョン 9.5 以降がストレージバックエンドとしてインストールされている必要があります
イメージレジストリ	<p>Astra Control Center のビルドイメージをプッシュできる、AWS Elastic Container Registry などの既存のプライベートレジストリが必要です。イメージをアップロードするイメージレジストリの URL を指定する必要があります。</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> Restic ベースのイメージを使用してアプリケーションをバックアップおよび復元するには、Astra Control Center ホストクラスタと管理対象クラスタが同じイメージレジストリにアクセスできる必要があります。</p> </div>
Astra Trident / ONTAP 構成	<p>Astra Control Center を使用するには、ストレージクラスを作成してデフォルトのストレージクラスとして設定する必要があります。Astra Control Center は、Kubernetes クラスタを NetApp BlueXP（旧 Cloud Manager）にインポートするときに作成される次の ONTAP Kubernetes ストレージクラスをサポートします。Astra Trident によって提供される機能は次のとおりです。</p> <ul style="list-style-type: none"> • vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io • vsaworkingenvironment-<>-ha-san csi.trident.netapp.io • vsaworkingenvironment-<>-single-nas csi.trident.netapp.io • vsaworkingenvironment-<>-single-san csi.trident.netapp.io



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



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

AWS の導入の概要を参照してください

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

これらの各手順については、以下で詳しく説明します。

1. [十分な IAM 権限があることを確認します。](#)

2. [AWS に Red Hat OpenShift クラスタをインストールします。](#)
3. [AWSを設定。](#)
4. [NetApp BlueXP for AWSを構成します。](#)
5. [Astra Control Center for AWSをインストール。](#)

十分な IAM 権限があることを確認します

Red Hat OpenShiftクラスタとNetApp BlueXP（旧Cloud Manager）コネクタをインストールできる十分なIAMロールと権限があることを確認します。

を参照してください "[AWS の初期クレデンシャル](#)".

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

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

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

AWSを設定

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

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

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



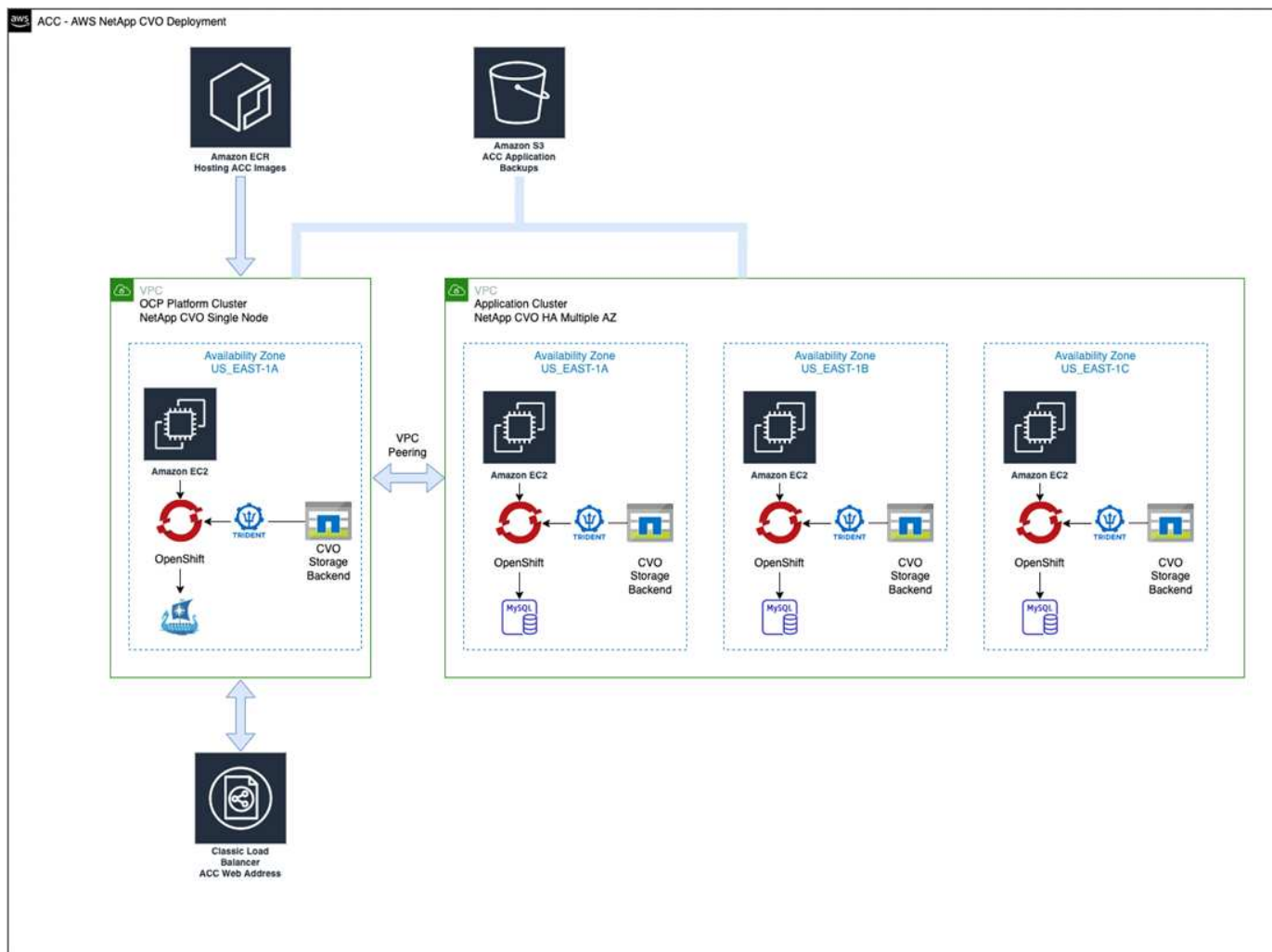
ECRを作成しないと、Astra Control Centerは、AWSバックエンドを持つCloud Volumes ONTAP を含むクラスタからモニタリングデータにアクセスできません。問題は、Astra Control Center を使用して検出および管理しようとしたクラスタに AWS ECR アクセスがない場合に発生します。

6. ACC イメージを定義済みのレジストリにプッシュします。



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

AWS 環境の例を次に示します。



NetApp BlueXP for AWSを構成します

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

BlueXPのマニュアルに従って'次の手順を実行します以下を参照してください。

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

手順

1. 資格情報をBlueXPに追加します。
2. ワークスペースを作成します。

3. AWS 用のコネクタを追加します。プロバイダとして AWS を選択します。
4. クラウド環境の作業環境を構築
 - a. 場所: 「Amazon Web Services (AWS)」
 - b. 「Cloud Volumes ONTAP HA」と入力します。
5. OpenShift クラスタをインポートします。作成した作業環境にクラスタが接続されます。
 - a. ネットアップクラスタの詳細を表示するには、* K8s * > * Cluster list * > * Cluster Details * を選択します。
 - b. 右上にある Astra Trident のバージョンを確認します。
 - c. Cloud Volumes ONTAP クラスタのストレージクラスは、プロビジョニングツールとして ネットアップ を使用していることに注目してください。

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



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

Astra Control Center for AWS をインストール

標準に従ってください "[Astra Control Center のインストール手順](#)".



AWS では汎用の S3 バケットタイプが使用されます。

Astra Control Center を Google Cloud Platform に導入

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

GCP に必要なもの

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

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



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

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

コンポーネント	要件
バックエンドの NetApp Cloud Volumes ONTAP ストレージ容量	300GB 以上のデータがあります
ワーカーノード (GCP コンピューティング要件)	少なくとも 3 つのワーカーノードが必要です。vCPU コア 4 基、RAM はそれぞれ 12GB です
ロードバランサ	動作環境クラスタ内のサービスに送信される入力トラフィックに使用できるサービスタイプ「LoadBalancer」
FQDN (GCP DNS ゾーン)	Astra Control Center の FQDN をロードバランシング IP アドレスに指定する方法
Astra Trident (以前の Cloud Manager で、 Kubernetes クラスタ検出の一部として NetApp BlueXP にインストール)	Trident 21.04 以降がインストールおよび設定され、NetApp ONTAP バージョン 9.5 以降がストレージバックエンドとしてインストールされている必要があります
イメージレジストリ	<p>Astra Control Centerビルドイメージをプッシュできる、Google Container Registryなどの既存のプライベートレジストリが必要です。イメージをアップロードするイメージレジストリの URL を指定する必要があります。</p> <div style="border-left: 1px solid #ccc; padding-left: 10px; margin-top: 10px;">  バックアップ用にリストイメージを取得するには、匿名アクセスを有効にする必要があります。 </div>
Astra Trident / ONTAP 構成	<p>Astra Control Center を使用するには、ストレージクラスを作成してデフォルトのストレージクラスとして設定する必要があります。Astra Control Centerは、KubernetesクラスタをNetApp BlueXPにインポートするときに作成される次のONTAP Kubernetesストレージクラスをサポートします。Astra Trident によって提供される機能は次のとおりです。</p> <ul style="list-style-type: none"> • vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io • vsaworkingenvironment-<>-ha-san csi.trident.netapp.io • vsaworkingenvironment-<>-single-nas csi.trident.netapp.io • vsaworkingenvironment-<>-single-san csi.trident.netapp.io



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

GCPの導入の概要

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

これらの各手順については、以下で詳しく説明します。

1. [GCPにRed Hat OpenShiftクラスターをインストールします。](#)
2. [GCPプロジェクトとVirtual Private Cloudを作成します。](#)
3. [十分な IAM 権限があることを確認します。](#)
4. [GCPを設定します。](#)
5. [GCP向けNetApp BlueXPの設定。](#)
6. [Astra Control Center for GCPをインストールします。](#)

GCPにRed Hat OpenShiftクラスターをインストールします

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

インストール手順については、次を参照してください。

- ["GCPにOpenShiftクラスターをインストールする"](#)
- ["GCPサービスアカウントの作成"](#)

GCPプロジェクトとVirtual Private Cloudを作成します

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



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

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

十分な IAM 権限があることを確認します

Red Hat OpenShiftクラスターとNetApp BlueXP (旧Cloud Manager) コネクタをインストールできる十分なIAMロールと権限があることを確認します。

を参照してください ["GCPの初期資格情報と権限"](#)。

GCPを設定します

次に、VPCの作成、コンピューティングインスタンスのセットアップ、Google Cloud Object Storageの作成、Astra Control CenterイメージのホストにGoogle Container Registerの作成、このレジストリへのイメージ

のプッシュを行うようにGCPを設定します。

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

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

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

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

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

例

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

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

8. DNS ゾーンを設定します。

GCP向けNetApp BlueXPの設定

NetApp BlueXP（旧Cloud Manager）を使用して、ワークスペースを作成し、GCPにコネクタを追加し、作業

環境を作成して、クラスタをインポートします。

BlueXPのマニュアルに従って'次の手順を実行しますを参照してください "[GCPでCloud Volumes ONTAP の使用を開始する](#)".

作業を開始する前に

- 必要なIAM権限と役割を持つGCPサービスアカウントにアクセスします

手順

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

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



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

Astra Control Center for GCPをインストールします

標準に従ってください "[Astra Control Center のインストール手順](#)".



GCPでは汎用S3バケットタイプが使用されます。

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

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

Microsoft Azure に Astra Control Center を導入

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

Azureに必要なもの

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

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

Azure の運用環境の要件

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

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

を参照してください "[Astra Control Center の運用環境要件](#)"。

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

コンポーネント	要件
イメージレジストリ	<p>Astra Control Center ビルドイメージをプッシュできる、Azure Container Registry (ACR) などの既存のプライベートレジストリが必要です。イメージをアップロードするイメージレジストリの URL を指定する必要があります。</p> <p> バックアップ用にリストイメージを取得するには、匿名アクセスを有効にする必要があります。</p>
Astra Trident / ONTAP 構成	<p>Astra Control Center を使用するには、ストレージクラスを作成してデフォルトのストレージクラスとして設定する必要があります。Astra Control Centerは、KubernetesクラスタをNetApp BlueXPにインポートするときに作成される次のONTAP Kubernetesストレージクラスをサポートします。Astra Trident によって提供される機能は次のとおりです。</p> <ul style="list-style-type: none"> • vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io • vsaworkingenvironment-<>-ha-san csi.trident.netapp.io • vsaworkingenvironment-<>-single-nas csi.trident.netapp.io • vsaworkingenvironment-<>-single-san csi.trident.netapp.io



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

Azure の導入の概要

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

これらの各手順については、以下で詳しく説明します。

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

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

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

インストール手順については、次を参照してください。

- ["Azure への OpenShift クラスターのインストール"](#)。
- ["Azure アカウントをインストールする"](#)。

Azure リソースグループを作成する

Azure リソースグループを少なくとも 1 つ作成します。



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

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

十分な **IAM** 権限があることを確認します

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

を参照してください ["Azure のクレデンシャルと権限"](#)。

Azure を設定

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

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

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

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

◦ 例 * :

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

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

10. DNS ゾーンを設定します。

NetApp BlueXP (旧Cloud Manager) を Azure 向けに設定します

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

BlueXP のマニュアルに従って、次の手順を実行しますを参照してください ["BlueXP の使用を開始しました"](#)。

作業を開始する前に

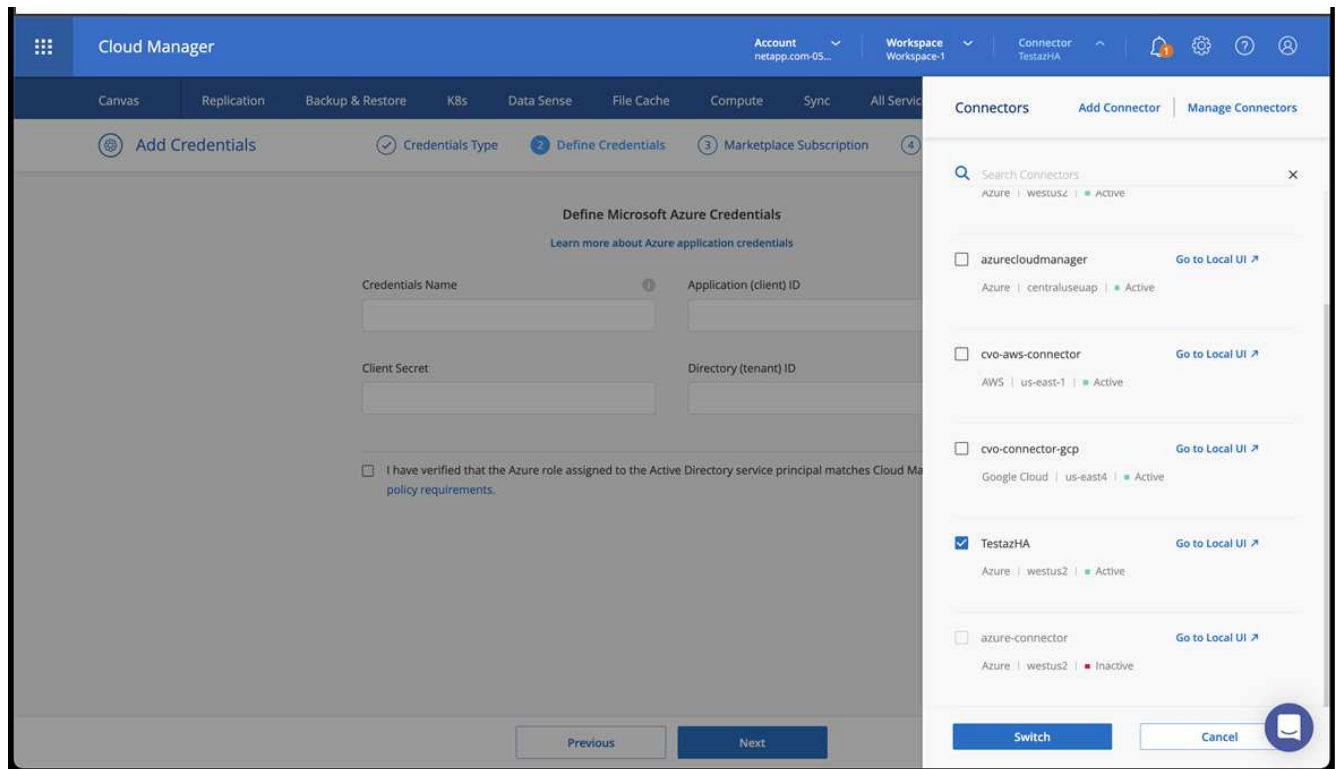
必要な IAM 権限とロールを持つ Azure アカウントにアクセスします

手順

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

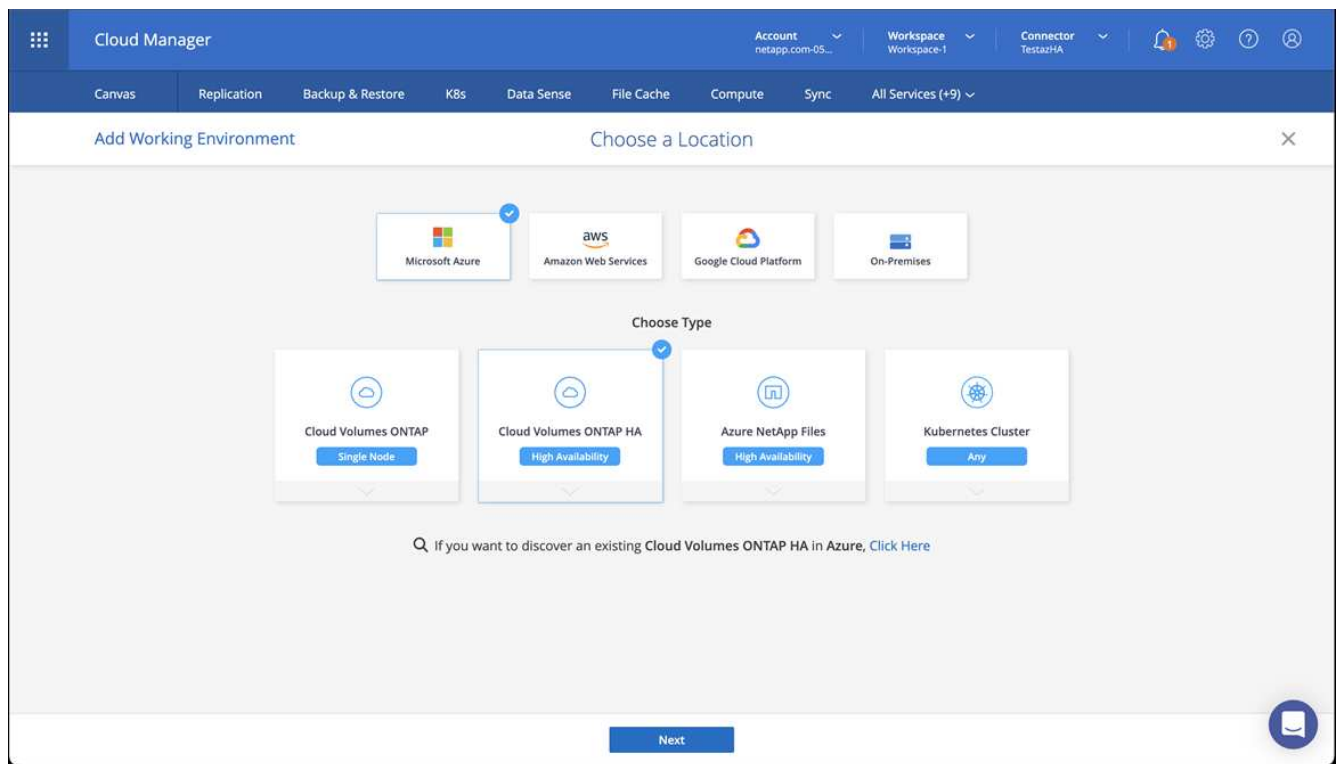
を参照してください ["BlueXP から Azure でコネクタを作成しています"](#)。

3. コネクタが動作していることを確認し、コネクタに切り替えます。



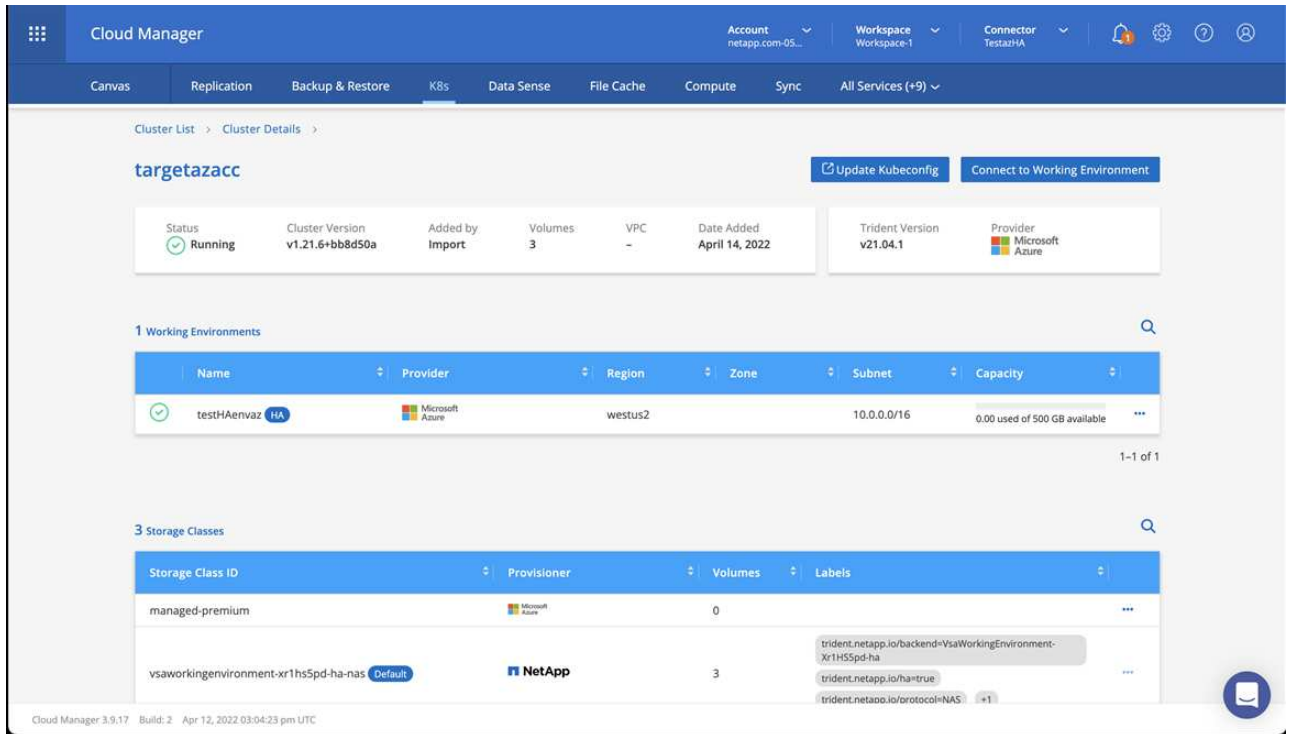
4. クラウド環境の作業環境を構築

- a. 場所：「Microsoft Azure」。
- b. 「Cloud Volumes ONTAP HA」と入力します。



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

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



- b. 右上にあるAstra Tridentのバージョンを確認します。

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

これにより、Red Hat OpenShift クラスタがインポートされ、デフォルトのストレージクラスが割り当てられます。ストレージクラスを選択します。

Astra Tridentは、インポートと検出のプロセスで自動的にインストールされます。

6. このCloud Volumes ONTAP 環境内のすべての永続ボリュームとボリュームをメモします。
7. Cloud Volumes ONTAP は、シングルノードまたはハイアベイラビリティとして動作できます。HA が有効になっている場合は、Azure で実行されている HA ステータスとノード導入ステータスを確認します。

Azure向けAstra Control Centerのインストールと設定

Astra Control Center を標準でインストールします ["インストール手順"](#)。

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

インストール後にAstra Control Centerを設定します

環境によっては、Astra Control Centerのインストール後に追加の設定が必要になる場合があります。

リソースの制限を解除します

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

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

手順

1. でリソースクォータを取得します netapp-acc（またはカスタム名）ネームスペース：

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

対応：

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

2. 名前別にすべてのリソースクォータを削除します。

```
kubectl delete resourcequota pods-high -n [netapp-acc or custom namespace]
```

```
kubectl delete resourcequota pods-low -n [netapp-acc or custom namespace]
```

```
kubectl delete resourcequota pods-medium -n [netapp-acc or custom namespace]
```

3. で制限範囲を取得します netapp-acc（またはカスタム名）ネームスペース：

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

対応：

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

4. 制限範囲を名前で削除します。

```
kubectl delete limitrange cpu-limit-range -n [netapp-acc or custom namespace]
```

ネームスペース間のネットワーク通信を有効にします

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

手順

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

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

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

```
kubectl delete networkpolicy [OBJECT_NAME] -n [netapp-acc or custom namespace]
```

3. 次のリソースファイルを適用してを設定します acc-avp-network-policy オブジェクトを使用して、AstraプラグインサービスからAstra Control Centerサービスへの要求を許可します。角かっこ内の情報を環境内の情報に置き換えます。

```

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

```

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

```

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

```

カスタム TLS 証明書を追加します

Astra Control Centerは、入力コントローラトラフィック（一部の設定のみ）およびWebブラウザでのWeb UI 認証に、デフォルトで自己署名TLS証明書を使用します。既存の自己署名 TLS 証明書を削除して、認証局（CA）が署名した TLS 証明書に置き換えることができます。

デフォルトの自己署名証明書は、次の2種類の接続に使用されます。



- Astra Control Center Web UIへのHTTPS接続
- 入力コントローラトラフィック (がの場合のみ) `ingressType: "AccTraefik"` プロパティはで設定されました `astra_control_center.yaml` Astra Control Centerのインストール中にファイルを作成)

これらの接続の認証に使用される証明書は、デフォルトのTLS証明書に置き換えられます。

作業を開始する前に

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

自己署名証明書を削除します

既存の自己署名 TLS 証明書を削除します。

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

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

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

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

コマンドラインを使用して新しい証明書を追加します

CAによって署名された新しい TLS 証明書を追加します。

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

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

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

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

#spec:
#  selfSigned: {}

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

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

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

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

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

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

5. 次のコマンドを使用して証明書を作成します。

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

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

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

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

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

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

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

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

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

Astra Control Center をセットアップします

Astra Control Centerをインストールし、UIにログインしてパスワードを変更したら、ライセンスのセットアップ、クラスタの追加、認証の有効化、ストレージの管理、バケットの追加を行うことができます。

タスク

- [Astra Control Center のライセンスを追加します](#)
- [Astra Controlを使用して、クラスタ管理のための環境を準備する](#)
- [\[クラスタを追加\]](#)
- [ONTAP ストレージバックエンドで認証を有効にします](#)
- [\[ストレージバックエンドを追加します\]](#)
- [\[バケットを追加します\]](#)

Astra Control Center のライセンスを追加します

Astra Control Centerをインストールすると、組み込みの評価用ライセンスがすでにインストールされています。Astra Control Centerを評価する場合は、この手順を省略できます。

新しいライセンスは、Astra Control UIまたはを使用して追加できます ["API"](#)。

Astra Control Centerライセンスは、Kubernetes CPUユニットを使用してCPUリソースを測定し、すべての管理対象Kubernetesクラスターのワーカーノードに割り当てられたCPUリソースを考慮します。ライセンスはvCPUの使用量に基づいています。ライセンスの計算方法の詳細については、を参照してください ["ライセンス"](#)。



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



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

作業を開始する前に

- 新しくインストールしたAstra Control Centerインスタンスへのアクセス。
- 管理者ロールの権限。
- A ["ネットアップライセンスファイル"](#) (NLF) 。

手順

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

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



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

Astra Controlを使用して、クラスタ管理のための環境を準備する

クラスタを追加する前に、次の前提条件を満たしていることを確認する必要があります。また、資格チェックを実行して、クラスタをAstra Control Centerに追加し、クラスタ管理の役割を作成する準備ができていることを確認する必要があります。

作業を開始する前に

- クラスタ内のワーカーノードで適切なストレージドライバが設定されていることを確認します。これにより、ポッドがバックエンドストレージと通信できるようになります。
- が環境に合っている ["運用環境の要件"](#) Astra TridentとAstra Control Centerに最適。
- Astra Tridentの一バージョン ["Astra Control Centerによってサポートされます"](#) がインストールされている
:



可能です **"Astra Tridentを導入"** Astra Tridentオペレータ（手動またはHelmチャートを使用）またはを使用 `tridentctl`。Astra Tridentのインストールまたはアップグレードを行う前に、を参照してください **"サポートされるフロントエンド、バックエンド、およびホスト構成"**。

- * Astra Tridentストレージバックエンドの設定* : Astra Tridentストレージバックエンドが少なくとも1つ必要です **"を設定します"** クラスタのポリシーを確認してください。
- * Astra Tridentストレージクラスを設定* : Astra Tridentストレージクラスを少なくとも1つ設定する必要があります **"を設定します"** クラスタのポリシーを確認してください。デフォルトのストレージクラスが設定されている場合は、そのストレージクラスがデフォルトのアノテーションを持つ唯一のストレージクラスであることを確認します。
- * Astra Tridentボリュームスナップショットコントローラとボリュームスナップショットクラスがインストールおよび設定されている* : ボリュームスナップショットコントローラが必要があります **"インストール済み"** Astra Controlでスナップショットを作成できるようにします。Astra Tridentが少なくとも1つ `VolumeSnapshotClass` はい **"セットアップ"** 管理者による。
- **Kubeconfig**にアクセス可能:にアクセスできます **"クラスタkubeconfig"** コンテキスト要素が1つだけ含まれます。
- * ONTAP クレデンシャル* : Astra Control Centerを使用してアプリケーションをバックアップおよびリストアするには、バックアップONTAP システムでONTAP クレデンシャルとスーパーユーザーIDを設定する必要があります。

ONTAP コマンドラインで次のコマンドを実行します。

```
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -superuser sys
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -anon 65534
```

- **rancherのみ**: Rancher環境でアプリケーションクラスタを管理する場合、rancherから提供されたkubeconfigファイルでアプリケーションクラスタのデフォルトコンテキストを変更して、rancher APIサーバコンテキストではなくコントロールプレーンコンテキストを使用します。これにより、Rancher APIサーバの負荷が軽減され、パフォーマンスが向上します。

資格チェックを実行します

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

手順

1. Astra Tridentのバージョンを確認

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

Astra Tridentが存在する場合は、次のような出力が表示されます。

```
NAME          VERSION
trident       22.10.0
```

Astra Tridentが存在しない場合は、次のような出力が表示されます。

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



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

2. ポッドが実行されていることを確認します。

```
kubectl get pods -n trident
```

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

```
kubectl get sc
```

回答例：

```
NAME          PROVISIONER          RECLAIMPOLICY
VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
ontap-gold (default)  csi.trident.netapp.io  Delete          Immediate
true                5d23h
```

制限されたクラスターロール**kubeconfig**を作成します

必要に応じて、Astra Control Centerの限定管理者ロールを作成できます。これは、Astra Control Centerのセットアップに必要な手順ではありません。この手順を使用すると、管理対象のクラスターのAstra Control権限を制限する別の**kubeconfig**を作成できます。

作業を開始する前に

手順 の手順を実行する前に、管理するクラスターに次の情報があることを確認してください。

- kubectl v1.23以降がインストールされている
- Astra Control Centerを使用して追加および管理するクラスターへのアクセス



この手順 では、Astra Control Centerを実行しているクラスターにkubectlでアクセスする必要はありません。

- アクティブなコンテキストのクラスタ管理者の権限で管理するクラスタのアクティブなkubeconfigです

1. サービスアカウントを作成します。

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

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

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

+

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

- a. サービスアカウントを適用します。

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

2. Astra Controlでクラスタを管理するために必要な最小限の権限を持つ、制限付きのクラスターロールを作成します。

- a. を作成します `ClusterRole` という名前のファイルです `astra-admin-account.yaml`。

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

```
<strong>astra-admin-account.yaml</strong>
```

+

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:

# Get, List, Create, and Update all resources
```

```
# Necessary to backup and restore all resources in an app
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - create
  - patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
  - deployments
  - horizontalpodautoscalers
  - ingresses
  - jobs
  - namespaces
  - networkpolicies
  - persistentvolumeclaims
  - poddisruptionbudgets
  - pods
  - podtemplates
  - podsecuritypolicies
  - replicaset
  - replicationcontrollers
  - replicationcontrollers/scale
  - rolebindings
  - roles
  - secrets
```

```

- serviceaccounts
- services
- statefulsets
- tridentmirrorrelationships
- tridentnapshotinfos
- volumesnapshots
- volumesnapshotcontents
verbs:
- delete

# Watch resources
# Necessary to monitor progress
- apiGroups:
  - ""
  resources:
  - pods
  - replicationcontrollers
  - replicationcontrollers/scale
  verbs:
  - watch

# Update resources
- apiGroups:
  - ""
  - build.openshift.io
  - image.openshift.io
  resources:
  - builds/details
  - replicationcontrollers
  - replicationcontrollers/scale
  - imagestreams/layers
  - imagestreamtags
  - imagetags
  verbs:
  - update

# Use PodSecurityPolicies
- apiGroups:
  - extensions
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use

```

a. クラスターロールを適用します。

```
kubectl apply -f astra-admin-account.yaml
```

3. サービスアカウントへのクラスターロールバインド用に、クラスターロールを作成します。

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

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

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

+

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

- a. クラスターロールバインドを適用します。

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

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

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

出力の末尾は次のようになります。


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

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

5. 次のように `kubeconfig` を生成します。

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

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

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

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

CONTEXT=$(kubectl config current-context)

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

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

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

```

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

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

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

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

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

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

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

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

```

- b. コマンドをソースにし、Kubernetes クラスタに適用します。

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

6. (オプション) クラスタにわかりやすい名前にコバーベキューの名前を変更します。

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

次の手順

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

クラスタを追加

アプリケーションの管理を開始するには、Kubernetes クラスタを追加し、コンピューティングリソースとして管理します。Kubernetes アプリケーションを検出するには、Astra Control Center のクラスタを追加する必要があります。



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

作業を開始する前に

- クラスタを追加する前に、必要な確認し、実行しておきます [前提条件となるタスク](#)。

手順

1. ダッシュボードまたはクラスタメニューのいずれかから移動します。
 - リソースサマリの*ダッシュボード*で、クラスタペインから*追加*を選択します。
 - 左側のナビゲーション領域で、*クラスタ*を選択し、クラスタページから*クラスタの追加*を選択します。
2. 表示された*クラスタの追加*ウィンドウで、をアップロードします kubeconfig.yaml の内容をファイルまたは貼り付けます kubeconfig.yaml ファイル。



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



自分で作成する場合は kubeconfig ファイルには、* 1つの*コンテキストエレメントのみを定義する必要があります。を参照してください "[Kubernetes のドキュメント](#)" を参照してください kubeconfig ファイル。を使用して、制限されたクラスタロールのkubeconfigを作成した場合 [上記のプロセス](#)この手順では、kubeconfigをアップロードまたは貼り付けてください。

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



ONTAP ストレージをベースとするAstra Tridentストレージクラスを選択する必要があります。

6. 情報を確認し、すべてが良好な場合は、「*追加」を選択します。

結果

クラスタが「* discovering *」状態になり、「Healthy *」に変わります。これで、Astra Control Centerを使用

してクラスタを管理できるようになりました。



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

ONTAP ストレージバックエンドで認証を有効にします

Astra Control Centerには、ONTAP バックエンドの認証に次の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やvsadmin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）。

後で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することができます。一度にサポートされる認証方式は1つだけです。

クレデンシャルベースの認証を有効にします

Astra Control Centerには、クラスタを対象としたクレデンシャルが必要です admin ONTAP バックエンドと通信するため。事前定義された標準のロール（など）を使用する必要があります admin。これにより、Astra Control Centerの今後のリリースで使用する機能APIが公開される可能性がある、将来のONTAP リリースとの前方互換性が確保されます。



カスタムのセキュリティログインロールはAstra Control Centerで作成して使用できますが、推奨されません。

バックエンド定義の例を次に示します。

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "admin",
  "password": "secret"
}
```

クレデンシャルがプレーンテキストで保存されるのは、バックエンド定義のみです。クレデンシャルの知識が

必要なのは、バックエンドの作成または更新だけです。そのため、Kubernetes管理者またはストレージ管理者が実行するのは管理者専用の操作です。

証明書ベースの認証を有効にします

Astra Control Centerでは、証明書を使用して新規および既存のONTAP バックエンドと通信できます。バックエンド定義には、次の情報を入力する必要があります。

- `clientCertificate`:クライアント証明書。
- `clientPrivateKey`:関連付けられた秘密鍵。
- `trustedCACertificate`:信頼されたCA証明書。信頼された CA を使用する場合は、このパラメータを指定する必要があります。信頼された CA が使用されていない場合は無視してかまいません。

次のいずれかのタイプの証明書を使用できます。

- 自己署名証明書
- サードパーティの証明書

自己署名証明書による認証を有効にします

一般的なワークフローは次の手順で構成されます。

手順

1. クライアント証明書とキーを生成します。生成時に、認証に使用するONTAP ユーザに共通名 (CN) を設定します。

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key  
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=<common-name>"
```

2. タイプがのクライアント証明書をインストールします `client-ca` とキーをONTAP 入力します。

```
security certificate install -type client-ca -cert-name <certificate-name> -vserver <vserver-name>  
security ssl modify -vserver <vserver-name> -client-enabled true
```

3. ONTAP のセキュリティログインロールが証明書認証方式をサポートしていることを確認します。

```
security login create -user-or-group-name vsadmin -application ontapi  
-authentication-method cert -vserver <vserver-name>  
security login create -user-or-group-name vsadmin -application http  
-authentication-method cert -vserver <vserver-name>
```

4. 生成した証明書を使用して認証をテストします。ONTAP 管理LIF>と<vserver name> を管理のIPと名前に置き換えてください。LIFのサービスポリシーがに設定されていることを確認する必要があります `default-data-management`。

```
curl -X POST -Lk https://<ONTAP-Management-LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key --cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp xmlns=http://www.netapp.com/filer/admin version="1.21" vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>
```

5. 前の手順で得た値を使用して、Astra Control CenterのUIでストレージバックエンドを追加します。

サードパーティの証明書による認証を有効にします

サードパーティの証明書がある場合は、次の手順で証明書ベースの認証を設定できます。

手順

1. 秘密鍵とCSRを生成します。

```
openssl req -new -newkey rsa:4096 -nodes -sha256 -subj "/" -outform pem -out ontap_cert_request.csr -keyout ontap_cert_request.key -addext "subjectAltName = DNS:<ONTAP_CLUSTER_FQDN_NAME>,IP:<ONTAP_MGMT_IP>"
```

2. CSRをWindows CA（サードパーティCA）に渡し、署名済み証明書を問題 します。
3. 署名済み証明書をダウンロードし、「ontap_signed_cert.crt」という名前を付けます。
4. Windows CA（サードパーティCA）からルート証明書をエクスポートします。
5. このファイルに名前を付けます ca_root.crt

これで、次の3つのファイルが作成されました。

- 秘密鍵：ontap_signed_request.key（これは、ONTAP のサーバ証明書に対応するキーです。サーバ証明書のインストール時に必要です）。
 - 署名済み証明書：ontap_signed_cert.crt（これは、ONTAP の_server certificate_inとも呼ばれます）。
 - ルート**CA**証明書：ca_root.crt（これは、ONTAP の_server-ca certificate_inとも呼ばれます）。
6. これらの証明書をONTAP にインストールします。生成してインストールします server および server-ca ONTAP の証明書。

詳細はsample.yamlを参照してください

```
# Copy the contents of ca_root.crt and use it here.
```

```
security certificate install -type server-ca
```

```
Please enter Certificate: Press <Enter> when done
```

```
-----BEGIN CERTIFICATE-----
```

```
<certificate details>
```

```
-----END CERTIFICATE-----
```

You should keep a copy of the CA-signed digital certificate for future reference.

The installed certificate's CA and serial number for reference:

CA:

serial:

The certificate's generated name for reference:

===

```
# Copy the contents of ontap_signed_cert.crt and use it here. For key, use the contents of ontap_cert_request.key file.
```

```
security certificate install -type server
```

```
Please enter Certificate: Press <Enter> when done
```

```
-----BEGIN CERTIFICATE-----
```

```
<certificate details>
```

```
-----END CERTIFICATE-----
```

```
Please enter Private Key: Press <Enter> when done
```

```
-----BEGIN PRIVATE KEY-----
```

```
<private key details>
```

```
-----END PRIVATE KEY-----
```

Enter certificates of certification authorities (CA) which form the certificate chain of the server certificate. This starts with the issuing CA certificate of the server certificate and can range up to the root CA certificate.

Do you want to continue entering root and/or intermediate certificates {y|n}: n

The provided certificate does not have a common name in the subject field.

Enter a valid common name to continue installation of the certificate: <ONTAP_CLUSTER_FQDN_NAME>

You should keep a copy of the private key and the CA-signed digital certificate for future reference.

The installed certificate's CA and serial number for reference:

CA:

serial:

The certificate's generated name for reference:

==

```
# Modify the vsserver settings to enable SSL for the installed certificate
```

```
ssl modify -vsserver <vsserver_name> -ca <CA> -server-enabled true  
-serial <serial number> (security ssl modify)
```

==

```
# Verify if the certificate works fine:
```

```
openssl s_client -CAfile ca_root.crt -showcerts -servername server  
-connect <ONTAP_CLUSTER_FQDN_NAME>:443  
CONNECTED(00000005)
```

```
depth=1 DC = local, DC = umca, CN = <CA>
```

```
verify return:1
```

```
depth=0
```

```
verify return:1
```

```
write W BLOCK
```

```
---
```

```
Certificate chain
```

```
0 s:
```

```
  i:/DC=local/DC=umca/<CA>
```

```
-----BEGIN CERTIFICATE-----
```

```
<Certificate details>
```

7. パスワードを使用しない通信用に同じホストのクライアント証明書を作成します。Astra Control Center は、このプロセスを使用してONTAP と通信します。

8. クライアント証明書を生成してONTAP にインストールします。

詳細はsample.yamlを参照してください


```
# Use /CN=admin or use some other account which has privileges.
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout
ontap_test_client.key -out ontap_test_client.pem -subj "/CN=admin"
```

Copy the content of ontap_test_client.pem file and use it in the below command:

```
security certificate install -type client-ca -vserver <vserver_name>
```

Please enter Certificate: Press <Enter> when done

```
-----BEGIN CERTIFICATE-----
```

```
<Certificate details>
```

```
-----END CERTIFICATE-----
```

You should keep a copy of the CA-signed digital certificate for future reference.

The installed certificate's CA and serial number for reference:

CA:

serial:

The certificate's generated name for reference:

==

```
ssl modify -vserver <vserver_name> -client-enabled true
(security ssl modify)
```

```
# Setting permissions for certificates
```

```
security login create -user-or-group-name admin -application ontapi
-authentication-method cert -role admin -vserver <vserver_name>
```

```
security login create -user-or-group-name admin -application http
-authentication-method cert -role admin -vserver <vserver_name>
```

==

```
#Verify passwordless communication works fine with the use of only
certificates:
```

```
curl --cacert ontap_signed_cert.crt --key ontap_test_client.key
--cert ontap_test_client.pem
```

```
https://<ONTAP_CLUSTER_FQDN_NAME>/api/storage/aggregates
```

```
{
```

```
"records": [
```

```
{
  "uuid": "f84e0a9b-e72f-4431-88c4-4bf5378b41bd",
  "name": "<aggr_name>",
  "node": {
    "uuid": "7835876c-3484-11ed-97bb-d039ea50375c",
    "name": "<node_name>",
    "_links": {
      "self": {
        "href": "/api/cluster/nodes/7835876c-3484-11ed-97bb-d039ea50375c"
      }
    }
  },
  "_links": {
    "self": {
      "href": "/api/storage/aggregates/f84e0a9b-e72f-4431-88c4-4bf5378b41bd"
    }
  }
},
]
,
"num_records": 1,
"_links": {
  "self": {
    "href": "/api/storage/aggregates"
  }
}
}
}%
```

9. Astra Control CenterのUIでストレージバックエンドを追加し、次の値を指定します。

- クライアント証明書：ontap_test_client.pem
- 秘密鍵：ontap_test_client.key
- 信頼されたCA証明書：ontap_signed_cert.crt

ストレージバックエンドを追加します

既存のONTAP ストレージバックエンドをAstra Control Centerに追加して、そのリソースを管理できます。

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

クレデンシャルまたは証明書認証情報を設定したら、Astra Control Centerに既存のONTAP ストレージバックエンドを追加してリソースを管理できます。

手順

1. 左側のナビゲーション領域のダッシュボードで、* Backends *を選択します。
2. 「* 追加」を選択します。
3. [Add storage backend]ページの[Use existing]セクションで、* ONTAP *を選択します。
4. 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. 「* 次へ *」を選択します。
 6. バックエンドの詳細を確認し、* Manage * を選択します。

結果

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



バックエンドが表示されるようにページを更新する必要がある場合があります。

バケットを追加します

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

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、Astra Controlにバケットを作成する必要はありません。アプリケーションのSnapshot機能にはバケットは必要ありません。

作業を開始する前に

- Astra Control Centerで管理しているクラスタから到達できるバケット。
- バケットのクレデンシャル。
- 次のタイプのバケット
 - NetApp ONTAP S3の略
 - 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。

手順

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



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

4. 既存のバケット名とオプションの概要を入力します。



バケット名と概要はバックアップ先として表示されるため、あとでバックアップを作成する際に選択できます。この名前は、保護ポリシーの設定時にも表示されます。

5. S3 エンドポイントの名前または IP アドレスを入力します。
6. [資格情報の選択*]で、[追加]または[*既存の*を使用]タブのいずれかを選択します。
 - 「*追加」を選択した場合：
 - i. Astra Control の他のクレデンシャルと区別するクレデンシャルの名前を入力します。
 - ii. クリップボードからコンテンツを貼り付けて、アクセス ID とシークレットキーを入力します。
 - [既存の使用*]を選択した場合：
 - i. バケットで使用する既存のクレデンシャルを選択します。
7. 選択するオプション Add。



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"](#)。

次の手順

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

- ["ローカルユーザとロールを管理します"](#)
- ["アプリの管理を開始します"](#)

- "アプリを保護します"
- "通知を管理します"
- "Cloud Insights に接続します"
- "カスタム TLS 証明書を追加します"
- "デフォルトのストレージクラスを変更する"

詳細については、こちらをご覧ください

- "Astra Control API を使用"
- "既知の問題"

Astra Control Center に関するよくある質問

この FAQ は、質問に対する簡単な回答を探している場合に役立ちます。

概要

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

Astra Control Center へのアクセス

- Astra Control の URL は何であるか。 *

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

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

ライセンス

評価ライセンスを使用しています。フルライセンスに変更するにはどうすればよいですか？

フルライセンスに変更するには、ネットアップからネットアップライセンスファイル (NLF) を入手します。

- 手順 *
- 1. 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。
- 2. ライセンスの概要で、ライセンス情報の右側にある[Options]メニューを選択します。
- 3. [置換]*を選択します。
- 4. ダウンロードしたライセンスファイルを参照し、* 追加 * を選択します。

*評価ライセンスを使用しています。アプリを管理できますか？ *

はい。評価ライセンス (デフォルトでインストールされている組み込み評価ライセンスを含む) を使用して、

アプリケーションの管理機能をテストできます。評価用ライセンスとフルライセンスでは、機能や機能に違いはありません。評価用ライセンスは、単純に寿命が短くなります。を参照してください ["ライセンス"](#) を参照してください。

Kubernetes クラスタを登録しています

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

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

- クラスタの管理を適切に解除するにはどうすればよいですか *
 1. ["Astra Control からアプリケーションの管理を解除"](#)。
 2. ["Astra Control からクラスタの管理を解除"](#)。
- Kubernetes クラスタを Astra Control から削除した後、アプリケーションとデータはどうなりますか。 *

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



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

管理を解除すると、**NetApp Astra Trident**はクラスタから自動的にアンインストールされますか？ Astra Control Centerでクラスタの管理を解除しても、Astra Tridentはクラスタから自動的にアンインストールされません。Astra Trident をアンインストールするには、が必要です ["Astra Trident のドキュメントで次の手順を実行します"](#)。

アプリケーションの管理

- Astra Control はアプリケーションを導入できますか。 *

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

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

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

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

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

- Astra Control自体を管理する必要がありますか？*

いいえ。Astra Control自体は「システムアプリケーション」であるため、管理しないでください。

不健全なポッドはアプリ管理に影響しますか？

いいえ、ポッドの健全性はアプリ管理には影響しません。

データ管理の操作

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

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

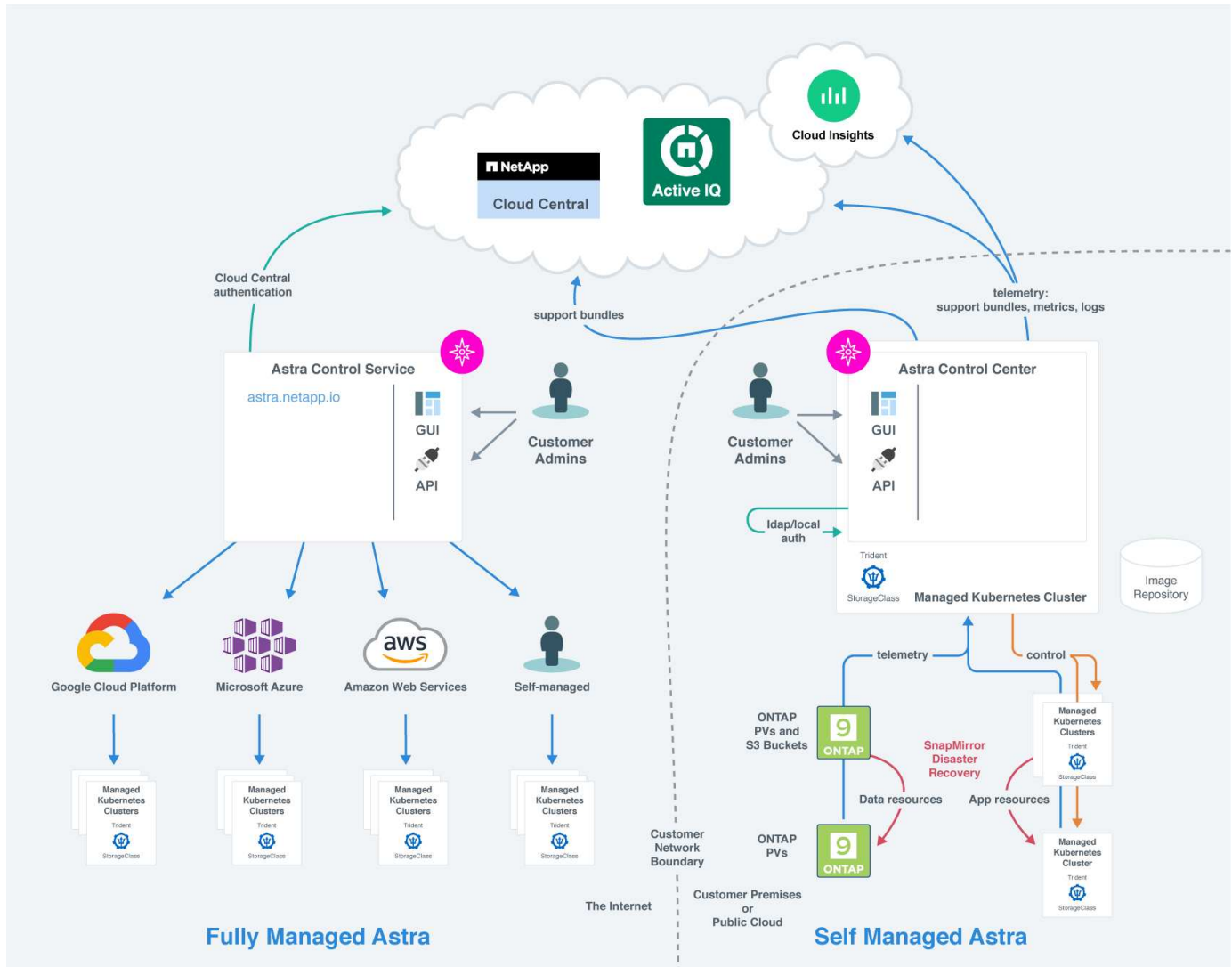
- Astra Control で取得したスナップショットを、別のインターフェイスやオブジェクトストレージから直接管理できますか。 *

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

概念

アーキテクチャとコンポーネント

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



Astra Control コンポーネント

- * Kubernetes クラスタ * : Kubernetes は、コンテナ化されたワークロードとサービスを管理するための、ポータブルで拡張性に優れたオープンソースプラットフォームであり、宣言型の設定と自動化の両方を促進します。Astra は、Kubernetes クラスタでホストされているアプリケーションに管理サービスを提供します。
- * Trident * : ネットアップが管理する、完全にサポートされているオープンソースのストレージプロビジョニングおよびオーケストレーションツールである Astra Trident を使用すると、Docker と Kubernetes で管理されるコンテナ化アプリケーション用のストレージボリュームを作成できます。Astra Control Center とともに導入した場合、Astra Trident には ONTAP ストレージバックエンドが設定されています。
- * ストレージバックエンド * :

- Astra Control Serviceは、次のストレージバックエンドを使用します。
 - ["NetApp Cloud Volumes Service for Google Cloud"](#) または、GKEクラスタのストレージバックエンドとしてGoogle Persistent Diskを使用します
 - ["Azure NetApp Files の特長"](#) またはAzure Managed DisksをAKSクラスタのストレージバックエンドとして使用します。
 - ["Amazon Elastic Block Store \(EBS\) "](#) または ["NetApp ONTAP 対応の Amazon FSX"](#) EKSクラスタのバックエンドストレージオプションとして使用できます。
- Astra Control Center は、次のストレージバックエンドを使用します。
 - ONTAP AFF、FAS、およびASA。ONTAPは、ストレージソフトウェアおよびハードウェアプラットフォームとして、コアストレージサービス、複数のストレージアクセスプロトコルのサポート、Snapshotやミラーリングなどのストレージ管理機能を提供します。
 - Cloud Volumes ONTAP
- * Cloud Insights * : NetAppクラウドインフラ監視ツールであるCloud Insightsを使用すると、Astra Control Centerで管理されるKubernetesクラスタのパフォーマンスと利用率を監視できます。Cloud Insights : ストレージ使用率とワークロードの相関関係を示します。Cloud Insights 接続を Astra コントロールセンターで有効にすると、テレメータの情報が Astra コントロールセンターの UI ページに表示されます。

Astra Control インターフェイス

さまざまなインターフェイスを使用してタスクを完了できます。

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

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

を参照してください。

- ["Astra Control Service のマニュアル"](#)
- ["Astra Control Center のドキュメント"](#)
- ["Astra Trident のドキュメント"](#)
- ["Astra Control API を使用"](#)
- ["Cloud Insights のドキュメント"](#)
- ["ONTAP のドキュメント"](#)

データ保護

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

Snapshot、バックアップ、保護のポリシー

Snapshotとバックアップのどちらも、次のタイプのデータを保護します。

- アプリケーション自体
- アプリケーションに関連付けられている永続的データボリューム
- アプリケーションに属するリソースアーティファクト

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

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

`a_protection policy_`は、アプリケーション用に定義したスケジュールに従って、スナップショット、バックアップ、またはその両方を自動的に作成することで、アプリケーションを保護する方法です。また、保護ポリシーでは、スケジュールで保持するSnapshotとバックアップの数を選択したり、さまざまなスケジュールレベルを設定したりすることもできます。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は、組織のニーズやSLA (Service Level Agreement) の要件に応じて各アプリケーションを確実に保護するための最良の方法です。



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

クローン

`a_clone_`は、アプリケーション、その設定、および永続データボリュームの完全な複製です。クローンは、同じ Kubernetes クラスターまたは別のクラスターに手動で作成できます。アプリケーションとストレージを Kubernetes クラスター間で移動する必要がある場合は、アプリケーションをクローニングすると便利です。

リモートクラスターへのレプリケーション

Astra Controlを使用すると、NetApp SnapMirrorテクノロジ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データやアプリケーションの変更をクラスター間でレプリケートできるようになります。

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

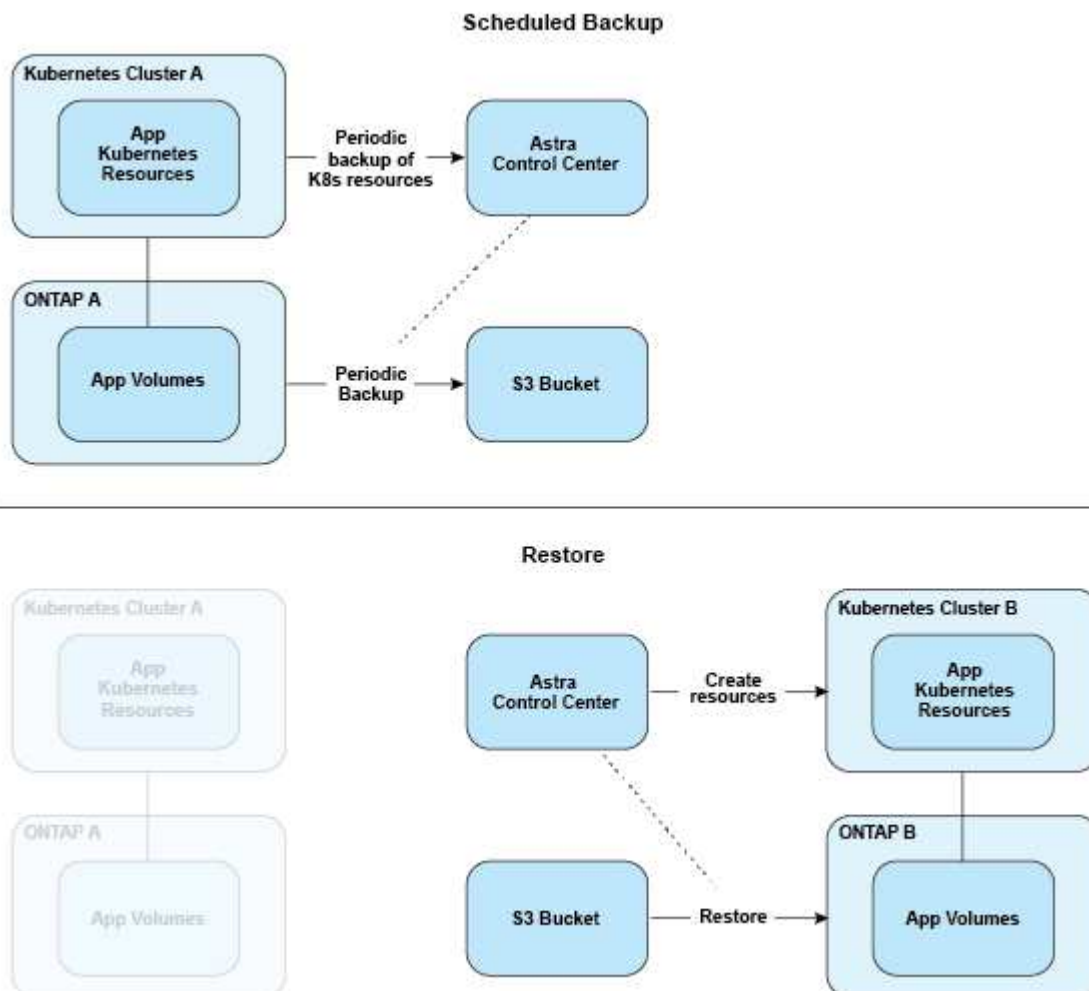
アプリケーションのレプリケーションは、次のようにアプリケーションのバックアップとリストアとは異なります。

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

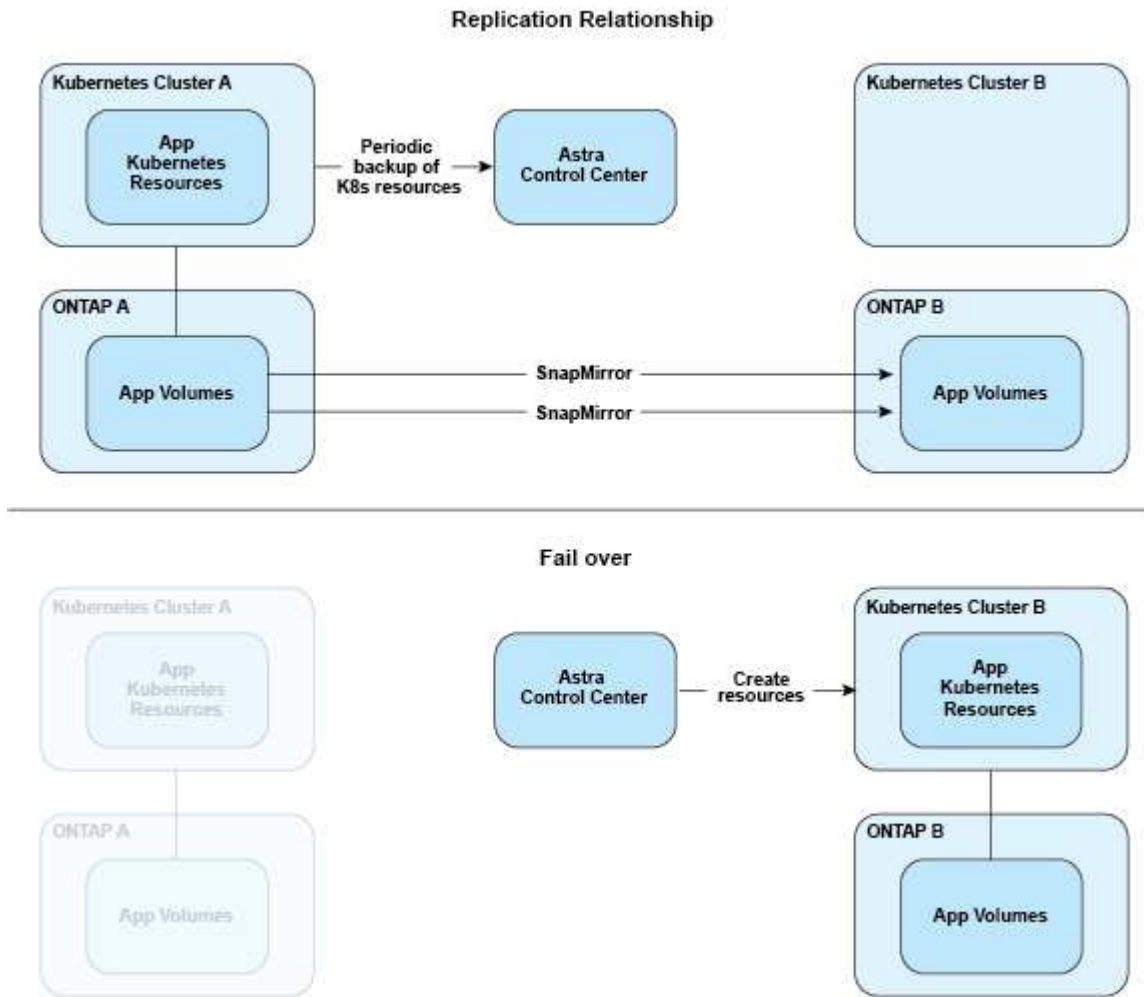
アプリケーションを複製する方法については、を参照してください "[SnapMirrorテクノロジーを使用してアプリケーションをリモートシステムにレプリケート](#)"。

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

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



一方、レプリケーションはONTAPにレプリケートすることで実行され、フェイルオーバーによってKubernetesリソースが作成されます。



ライセンスの有効期限が切れたバックアップ、スナップショット、クローン

ライセンスの有効期限が切れた場合、追加または保護するアプリケーションが別のAstra Control Centerインスタンスである場合にのみ、新しいアプリケーションの追加やアプリケーションの保護処理（Snapshot、バックアップ、クローン、リストア処理など）を実行できます。

ライセンス

Astra Control Centerを導入すると、4、800 CPUユニットの90日間の評価ライセンスが組み込まれてインストールされます。容量の追加や評価期間の延長が必要な場合、またはフルライセンスにアップグレードする場合は、ネットアップから別の評価用ライセンスまたはフルライセンスを取得できます。

次のいずれかの方法でライセンスを取得します。

- Astra Control Centerを評価する際に、組み込みの評価用ライセンスと異なる評価条件が必要な場合は、ネットアップに連絡して別の評価用ライセンスファイルをリクエストしてください。

- "Astra Control Centerを購入済みの場合は、ネットアップライセンスファイル (NLF) を生成する" NetApp Support Site にログインし、[Systems]メニューからソフトウェアライセンスに移動します。

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



ライセンスで有効になっているCPUユニットが必要な数以上であることを確認してください。Astra Control Centerで現在管理しているCPUユニット数が、適用する新しいライセンスで使用可能なCPUユニット数を超えると、新しいライセンスを適用できなくなります。

評価用ライセンスとフルライセンス

Astra Control Centerの新しいインストールには、評価用ライセンスが組み込まれています。評価用ライセンスでは、フルライセンスと同じ機能を制限付き (90日間) で使用できます。評価期間が終了したら、フル機能を使用するにはフルライセンスが必要です。

ライセンスの有効期限

アクティブなAstra Control Centerのライセンスが期限切れになると、次の機能のUIおよびAPI機能は使用できなくなります。

- ローカルのスナップショットとバックアップを手動で実行します
- スケジュールされたローカルSnapshotおよびバックアップ
- Snapshot またはバックアップからのリストア
- Snapshot または現在の状態からクローニングしています
- 新しいアプリケーションの管理
- レプリケーションポリシーを設定しています

ライセンス消費量の計算方法

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

クラスタでアプリケーションの管理を開始すると、そのクラスタのすべてのCPUユニットがAstra Control Centerのライセンス消費に含まれます。ただし、でラベルを使用して報告されるRed Hat OpenShiftクラスタノードのCPUユニットは除きます `node-role.kubernetes.io/infra: ""`。



Red Hat OpenShiftインフラノードでは、Astra Control Centerのライセンスは使用されません。ノードをインフラストラクチャノードとしてマークするには、ラベルを適用します `node-role.kubernetes.io/infra: ""` ノードに追加します。

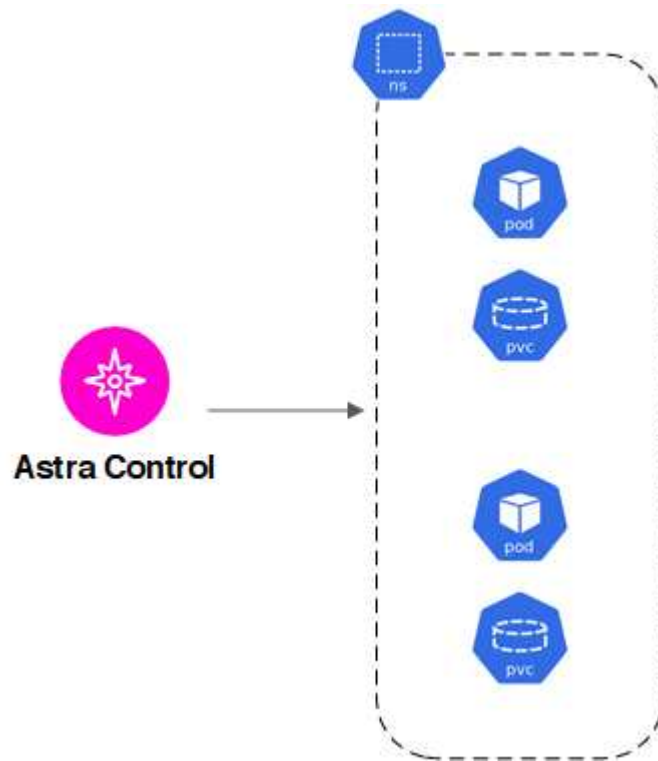
詳細については、こちらをご覧ください

- "[Astra Control Centerの初回セットアップ時にライセンスを追加します](#)"
- "[既存のライセンスを更新する](#)"

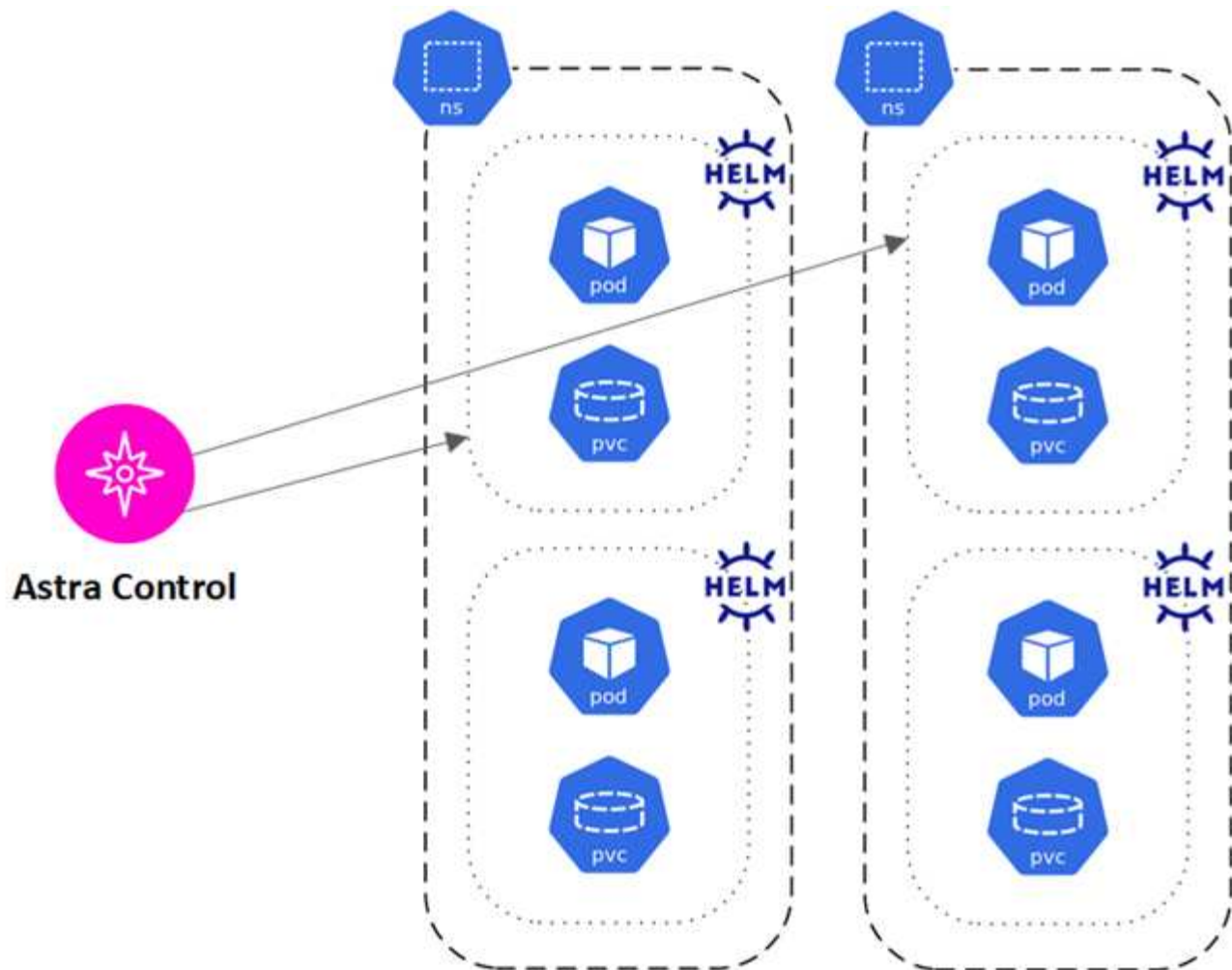
アプリケーション管理

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

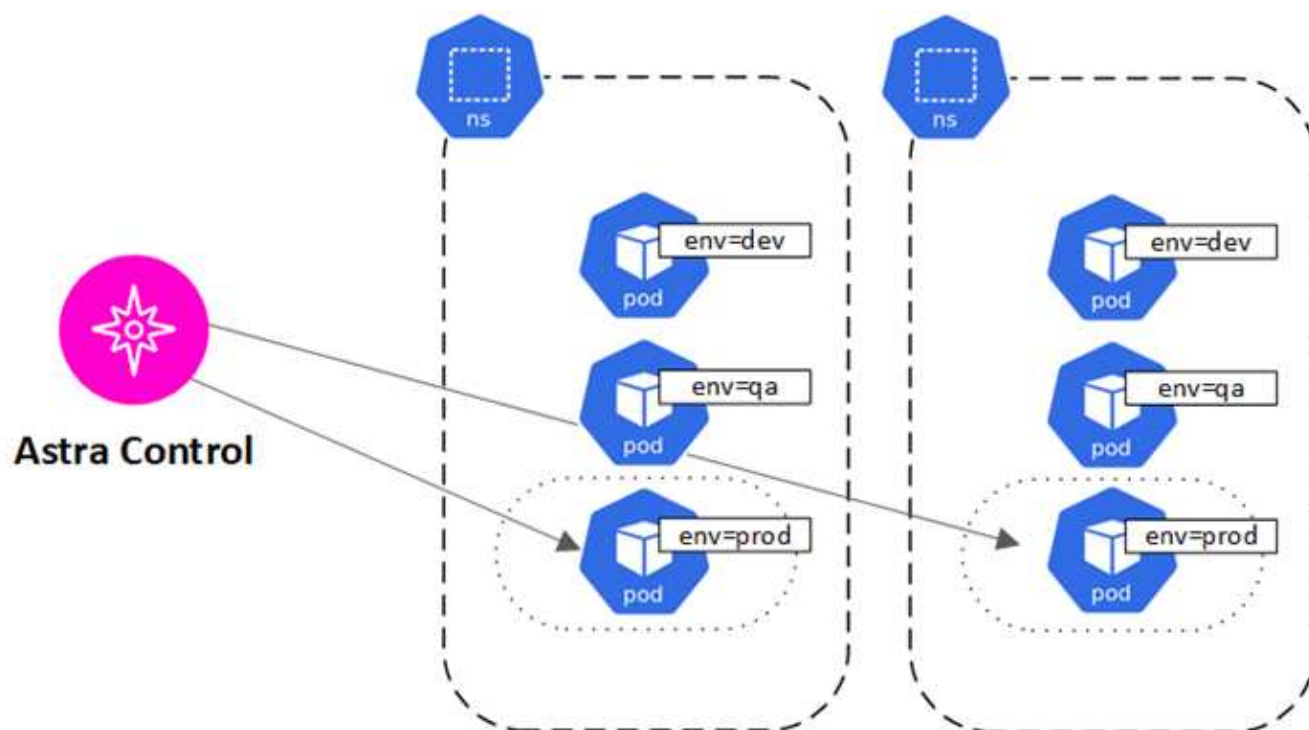
- ネームスペース。ネームスペース内のすべてのリソースを含みます



- 1つ以上のネームスペース内に導入された個々のアプリケーション（この例では、helm3を使用）



- 1つ以上の名前空間内のKubernetesラベルで識別されるリソースのグループ



ストレージクラスと永続的ボリュームサイズ

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

概要

Astra Control Center は、次の機能をサポートします。

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



Astra Tridentのストレージクラスは、Astra Control Center以外で事前に設定する必要があります。

ストレージクラス

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

を参照してください。

- ["Astra Trident のドキュメント"](#)

ユーザロールとネームスペース

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

ユーザロール

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

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

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

ネームスペース

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

詳細については、こちらをご覧ください

["ローカルユーザとロールを管理します"](#)

ポッドセキュリティ

Astra Control Centerは、PoDセキュリティポリシー（PSP）およびPoDセキュリティアドミッション（PSA）による特権制限をサポートします。これらのフレームワークを使用すると、ユーザまたはグループがコンテナを実行できる対象や、コンテナに付与できる権限を制限できます。

Kubernetesディストリビューションの中には、デフォルトのポッドセキュリティ構成が用意されているものがあり、制限が厳しく、Astra Control Centerのインストール時に問題が発生する場合があります。

ここに記載されている情報と例を使用して、Astra Control Centerが行うポッドのセキュリティの変更を理解し、Astra Control Centerの機能を妨げずに必要な保護を提供するポッドのセキュリティアプローチを使用できます。

Astra Control Centerによって強制されるPSAS

Astra Control Centerのインストール中に、次のラベルをに追加することで、ポッドのセキュリティアドミッションを強制できます `netapp-acc` またはカスタムネームスペース：

```
pod-security.kubernetes.io/enforce: privileged
```

Astra Control CenterによってインストールされたPSP

Kubernetes 1.23または1.24にAstra Control Centerをインストールすると、インストール時にいくつかのポッドセキュリティポリシーが作成されます。これらの中には永続的なものもあれば、一部のものは特定の処理中に作成され、処理が完了すると削除されます。ホストクラスタでKubernetes 1.25以降が実行されている場合、これらのバージョンではサポートされていないため、Astra Control CenterはPSPのインストールを試みません。

インストール中に作成されたPSP

Astra Control Centerのインストール中、Astra Control CenterオペレータはカスタムポッドセキュリティポリシーAをインストールします `Role` オブジェクト、および `RoleBinding` Astra Control Centerネームスペース

へのAstra Control Centerサービスの導入をサポートするオブジェクト。

新しいポリシーとオブジェクトには次の属性があります。

```
kubectl get psp
```

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

```
kubectl get role -n <namespace_name>
```

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

```
kubectl get rolebinding -n <namespace_name>
```

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

バックアップ処理中に作成されたPSP

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

新しいポリシーとオブジェクトには次の属性があります。

```
kubectl get psp
```

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

```
kubectl get role -n <namespace_name>
```

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

```
kubectl get rolebinding -n <namespace_name>
```

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

クラスタ管理中に作成されたPSP

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

新しいポリシーとオブジェクトには次の属性があります。

```
kubectl get psp
```

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

```
kubectl get role -n <namespace_name>
```

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

```
kubectl get rolebinding -n <namespace_name>
```

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

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。

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、kubecti 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環境として独自の新しいシークレットが存在する場合でも、クローニングされたオペレータがソースオペレータから Kubernetes シークレットを参照しようとします。Astra Control がソースオペレータの Kubernetes シークレットを認識しないため、クローニング処理が失敗する場合があります。

クラスタにアプリをインストールします

お先にどうぞ ["クラスタが追加されました"](#) Astra Control を使用すると、アプリケーションをインストールしたり、クラスタ上の既存のアプリケーションを管理したりできます。1 つ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、同じ名前空間にある「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、シングル名前空間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名前空間。
- (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 "[標準名前空間とシステム名前空間の管理](#)"。
- 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、"[名前空間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

- [\[アプリケーションとして管理するリソースを定義します\]](#)
- [\[アプリケーションとして管理する名前空間を定義します\]](#)

アプリケーションとして管理するリソースを定義します

を指定できます "[アプリケーションを構成するKubernetesリソース](#)" Astra Controlで管理したい。アプリケーションを定義すると、Kubernetesクラスタの要素を1つのアプリケーションにグループ化できます。このKubernetesリソースの集まりは、名前空間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](#)。

アプリケーションネームスペースへのクラスタを対象としたリソースの追加の詳細については、こちらをご覧ください。

ネームスペース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。

5. (オプション) 各ネームスペースに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に表示されます 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) :

システム名前空間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名前空間も検出します。これらのシステム名前空間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[名前空間]タブからシステム名前空間を表示するには、[システム名前空間を表示]チェックボックスをオンにします。

Show system namespaces



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

例：リリースごとに保護ポリシーを分ける

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

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

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

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

詳細については、こちらをご覧ください

- ["Astra Control API を使用"](#)
- ["アプリ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接続します。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](#)
- [\[バックアップを作成します\]](#)
- [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。

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"](#)。



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。



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、保護ポリシーは使用できません。バックアップとSnapshotのスケジュールを設定する場合は、Astra Controlでサポートされるストレージクラスに移行します。

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. 「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。

The screenshot shows the 'Configure protection policy' interface with the following details:

- PROTECTION SCHEDULE:** Four options are shown:
 - 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.
- Weekly Selection:** Radio buttons for Hourly, Daily, Weekly (selected), and Monthly.
- Configuration Fields:**
 - 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
- Buttons:** Cancel and Review →

5. 「* Review (レビュー) 」を選択します

6. [* 保護ポリシーの設定 *] を選択します

結果

Astra Control 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。



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。

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. Snapshot の概要を確認し、「 * Snapshot * 」を選択します。

結果

スナップショットプロセスが開始されます。スナップショットはステータスが * Healthy である場合に成功します (Data protection > Snapshots ページの State *列)

バックアップを作成します

アプリケーションはいつでもバックアップできます。



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



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によってバックアップされたアプリケーションのバックアップ `ontap-nas-economy` システムの停止を伴うため、バックアップ処理が完了するまでアプリケーションを使用できなくなります。

手順

1. 「 * アプリケーション * 」を選択します。
2. 目的のアプリケーションの * アクション * 列のオプションメニューから、 * バックアップ * を選択します。
3. バックアップ名をカスタマイズする。
4. 既存のスナップショットからアプリケーションをバックアップするかどうかを選択します。このオプションを選択すると、既存の Snapshot のリストから選択できます。

5. ストレージバケットのリストから、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。
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 : 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、完了するまで待つから、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。

Snapshot とバックアップを表示します

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

手順

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

デフォルトでは、Snapshot が表示されます。

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

Snapshot を削除します

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



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。

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。



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。

アプリケーションのリストア

Astra Control を使用すると、スナップショットまたはバックアップからアプリケーションをリストアできます。同じクラスタにアプリケーションをリストアする場合、既存の Snapshot からのリストアは高速です。Astra Control UI またはを使用できます ["Astra Control API"](#) アプリを復元するには、



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。

このタスクについて

- 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。リストアに失敗した場合に、Snapshotまたはバックアップからクローニングできます。
- 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。
- (OpenShiftクラスタのみ) ポリシーの追加: OpenShiftクラスタでアプリ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ネームスペース) にSecurityContext UIDが割り当てられます。Astra Control Center でアプリケーションを保護し、OpenShift でそのアプリケーションを別のクラスタまたはプロジェクトに移動できるようにするには、アプリケーションを任意の UID として実行できるようにポリシーを追加する必要があります。たとえば、次の OpenShift CLI コマンドは、WordPress アプリケーションに適切なポリシーを付与します。

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

- * 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](#)。

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名前空間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空間にリストアすると、同じ名前の新しい名前空間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空間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空間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。



この手順は、どちらにも使用できます 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` 同じクラスタ*または*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。

- 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空間のデスティネーション名前空間を入力します。



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空間を作成します。指定するデスティネーション名前空間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ます。
- 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。

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。クローンまたはリストア処理によって新しいネームスペースが作成されると、アカウントの管理者 / 所有者はメンバーユーザアカウントを編集し、該当するユーザに新しいネームスペースへのアクセスを許可するロールの制限を更新できます。

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。

対象と除外のシナリオについて詳しくは、こちらをご覧ください

- 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ネームスペースが含まれていた場合、新しいネームスペースにリストアすると、リソースが含まれていなくてもすべてのネームスペースが作成されます。

- b. (オプション) リソース名を入力します。
- c. (任意) ラベルセレクタ：を含めます **"ラベルセレクタ"** をクリックしてルールに追加します。ラベルセレクタは、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。

経済性に優れたONTAP-NASストレージからONTAP-NASストレージへの移行

Astra Controlを使用できます "アプリケーションのリストア" または "アプリケーションのクローン" に対応するストレージクラスからアプリケーションボリュームを移行する処理 `ontap-nas-economy`` では、でサポートされるストレージクラスに制限されたアプリケーション保護オプションが許可されます `ontap-nas Astra Controlのあらゆる保護オプションを利用できます。クローンまたはリストア処理では、を使用するqtreeベースのボリュームが移行されます `ontap-nas-economy` でサポートされる標準ボリュームへのバックエンド `ontap-nas`。ボリューム (ボリュームが存在するかどうかに関係なく) `ontap-nas-economy BACKED ONLY` または `MIXED` は、ターゲットストレージクラスに移行されます。移行が完了すると、保護オプションの制限がなくなります。

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` ネームスペースを使用する `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) ネームスペースを使用します `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アプリケーションをリモートシステムにレプリケート

Astra Controlを使用すると、NetApp SnapMirrorテクノロジ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データやアプリケーションの変更をクラスタ間でレプリケートできるようになります。

バックアップ/リストアとレプリケーションの比較については、を参照してください "[データ保護の概念](#)"。

アプリケーションは、オンプレミスのみ、ハイブリッド、マルチクラウドなど、さまざまなシナリオでレプリケートできます。

- オンプレミスサイトAからオンプレミスサイトBへ
- Cloud Volumes ONTAP を使用してオンプレミスからクラウドに移行できます
- Cloud Volumes ONTAP を使用したクラウドをオンプレミスに移行
- Cloud Volumes ONTAP を使用したクラウドからクラウドへ（同じクラウドプロバイダ内の異なるリージョン間または異なるクラウドプロバイダ間）

Astra Controlを使用すれば、オンプレミスのクラスタからクラウドへ（Cloud Volumes ONTAP を使用）、またはクラウド間（Cloud Volumes ONTAP からCloud Volumes ONTAP へ）にアプリケーションをレプリケートできます。



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

Astra Controlを使用すると、アプリケーションのレプリケーションに関連する次のタスクを実行できます。

- [レプリケーション関係を設定]
- [デスティネーションクラスタでレプリケートされたアプリケーションをオンラインにする（フェイルオーバー）]
- [フェイルオーバーしたレプリケーションを再同期します]
- [アプリケーションのレプリケーションを反転する]
- [アプリケーションを元のソースクラスタにフェイルバックします]
- [アプリケーションレプリケーション関係を削除します]

レプリケーションの前提条件

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* :
 - * Trident * : 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、Astra Tridentバージョン22.07以降が存在している必要があります。
 - 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。
- ペアリング :
 - * クラスタとSVM * : ONTAPクラスタとホストSVMがペアリング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。
 - * TridentとSVM * : ペアリングされたりモートSVMが、デスティネーションクラスタのAstra Tridentで使用可能である必要があります。
- * Astra Control Center * :



"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。

- 管理対象クラスタ：次のクラスタをAstra Controlに追加して管理する必要があります（理想的には障害ドメインやサイトが異なる場合）。
 - 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 Trident / ONTAPの設定*：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クラスを少なくとも1つ設定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でのアプリケーションSnapshotの作成頻度の選択（アプリケーションのKubernetesリソースと、アプリケーションの各ボリュームのボリュームSnapshotを含む）
- レプリケーションスケジュールの選択（Kubernetesリソースと永続ボリュームデータを含む）
- Snapshotを作成する時刻を設定します

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. [アプリケーション] ページで、[データ保護] > [レプリケーション] タブを選択します。
3. [データ保護] > [レプリケーション] タブで、[レプリケーションポリシーの設定] を選択します。または、[アプリケーション保護]ボックスから[アクション]オプションを選択し、[レプリケーションポリシーの構成]を選択します。
4. 次の情報を入力または選択します。
 - デスティネーションクラスタ：ソースとは異なるデスティネーションクラスタを入力してください。
 - デスティネーションストレージクラス：デスティネーションONTAP クラスタでペアリングされているSVMを使用するストレージクラスを選択または入力します。
 - レプリケーションタイプ：現在使用できるレプリケーションタイプは「非同期」のみです。
 - デスティネーションネームスペース：デスティネーションクラスタの新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名前空間を選択して、名前空間を追加します。
- レプリケーション頻度：Snapshotを作成してデスティネーションにレプリケートする頻度を指定します。
- オフセット：Astra Controlでスナップショットを作成する時間の上部から分数を設定します。オフセットを使用すると、他のスケジュールされた処理と競合しないようにすることができます。



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」と表示されません。

Astra Control：レプリケーションに使用するアプリケーションSnapshotを作成

6. アプリケーションのスナップショットステータスを表示するには、アプリケーション>*スナップショット*タブを選択します。

Snapshot名はこの形式を使用します 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空間を作成します（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空間にPVCを作成します。
- アプリケーションと整合性のある最初のSnapshotを作成します。
- 初期Snapshotを使用して、永続ボリュームのSnapMirror関係を確立します。

[Data Protection]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>

例：
正常|確立

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. [アプリケーション] ページで、[データ保護] > [レプリケーション] タブを選択します。
3. [データ保護 (Data Protection)] > [複製 (Replication)] タブの[アクション (Actions)] メニューから、[フェイルオーバー* (フェイルオーバー*)] を選択し
4. フェイルオーバーページで、情報を確認し、*フェイルオーバー*を選択します。

結果

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クラスタで、レプリケートされた最新のSnapshotに基づいてアプリケーションが起動されます。
- 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ません。



再同期プロセスの間、ライフサイクルの状態は「Establishing」と表示されます。

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. [アプリケーション] ページで、[データ保護] > [レプリケーション] タブを選択します。
3. [データ保護 (Data Protection)] > [レプリケーション (Replication)] タブの[アクション (Actions)] メニューから、[*再同期 (Resync *)] を
4. 再同期 (Resync) ページで、保持するデータを含むソースまたはデスティネーションのアプリケーションインスタンスを選択します。



デスティネーションのデータが上書きされるため、再同期元は慎重に選択してください。

5. 続行するには、* Resync *を選択します。
6. 「resync」と入力して確定します。
7. 「* Yes、resync *」を選択して終了します。

結果

- Replication（レプリケーション）ページに、レプリケーションステータスとしてEstablishing（確立）が表示されます。
- Astra Controlは、新しいデスティネーションクラスタのアプリケーションを停止します。
- SnapMirror resyncを使用して、指定した方向に永続的ボリュームのレプリケーションを再確立します。
- [レプリケーション]ページに、更新された関係が表示されます。

アプリケーションのレプリケーションを反転する

元のソースクラスタへのレプリケートを続行したまま、アプリケーションをデスティネーションクラスタに移動する計画的処理です。Astra Controlは、ソースクラスタ上のアプリケーションを停止し、デスティネーションにデータをレプリケートしてから、デスティネーションクラスタにアプリケーションをフェイルオーバーします。

この状況では、ソースとデスティネーションを交換しようとしています。元のソースクラスタが新しいデスティネーションクラスタになり、元のデスティネーションクラスタが新しいソースクラスタになります。

手順

1. Astra Controlの左ナビゲーションから、「アプリケーション」を選択します。
2. [アプリケーション] ページで、[データ保護] > [レプリケーション] タブを選択します。
3. [データ保護 (Data Protection)] > [レプリケーション (Replication)] タブの[アクション (Actions)] メニューから、[レプリケーションを反転 (Reverse replication)] を選択します
4. リバース・レプリケーションのページで情報を確認し、「リバース・レプリケーション」を選択して続行します。

結果

リバースレプリケーションの結果、次の処理が実行されます。

- Snapshotは、元のソースアプリケーションのKubernetesリソースから作成されます。
- 元のソースアプリケーションのポッドは、アプリケーションのKubernetesリソースを削除することで正常に停止されます（PVCとPVはそのまま維持されます）。
- ポッドがシャットダウンされると、アプリケーションのボリュームのSnapshotが作成されてレプリケートされます。
- SnapMirror関係が解除され、デスティネーションボリュームが読み取り/書き込み可能な状態になります。
- アプリケーションのKubernetesリソースは、元のソースアプリケーションのシャットダウン後にレプリケートされたボリュームデータを使用して、シャットダウン前のSnapshotからリストアされます。
- 逆方向にレプリケーションが再確立されます。

アプリケーションを元のソースクラスタにフェイルバックします

Astra Controlを使用すると、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、元のレプリケーション方向を復元するために、レプリケーションの方向を反転する前に、Astra Controlによってアプリケーションの変更が元のソースクラスタにレプリケート（再同期）されます。

このプロセスは、デスティネーションへのフェイルオーバー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. [アプリケーション] ページで、[データ保護] > [レプリケーション] タブを選択します。
3. [データ保護 (Data Protection)] > [レプリケーション (Replication)] タブの[アクション (Actions)] メニューから、[*再同期 (Resync *)] を
4. フェイルバック処理の場合は、フェイルオーバーした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. [アプリケーション] ページで、[データ保護] > [レプリケーション] タブを選択します。
3. [データ保護] > [レプリケーション] タブの[アプリケーション保護] ボックスまたは関係図で、[レプリケーション関係の削除*] を選択します。

結果

レプリケーション関係を削除すると、次の処理が実行されます。

- 関係が確立されていても、アプリケーションがデスティネーションクラスタでオンラインになっていない

(フェイルオーバーした) 場合、Astra Controlは、初期化中に作成されたPVCを保持し、「空」の管理対象アプリケーションをデスティネーションクラスタに残します。また、作成されたバックアップを保持するためにデスティネーションアプリケーションを保持します。

- アプリケーションがデスティネーションクラスタでオンラインになった (フェイルオーバーした) 場合、Astra ControlはPVCと宛先アプリケーションを保持します。ソースとデスティネーションのアプリケーションは、独立したアプリケーションとして扱われるようになりました。バックアップスケジュールは、両方のアプリケーションで維持されますが、相互に関連付けられていません。

レプリケーション関係のヘルスステータスと関係のライフサイクル状態

Astra Controlには、関係の健全性と、レプリケーション関係のライフサイクルの状態が表示されます。

レプリケーション関係のヘルスステータス

レプリケーション関係の健全性は、次のステータスで示されます。

- 正常：関係が確立されているか確立されており、最新のSnapshotが転送されました。
- 警告：関係がフェイルオーバーされているかフェイルオーバーされています (そのためソースアプリは保護されなくなりました)。
- * 重要 *
 - 関係が確立されているか、フェイルオーバーされていて、前回の調整が失敗しました。
 - 関係が確立され、新しいPVCの追加を最後に調整しようとしても失敗しています。
 - 関係は確立されていますが (Snapshotが正常にレプリケートされ、フェイルオーバーが可能になります)、最新のSnapshotはレプリケートに失敗したか、レプリケートに失敗しています。

レプリケーションのライフサイクル状態

次の状態は、レプリケーションのライフサイクルの各段階を表しています。

- * Establishing * : 新しいレプリケーション関係を作成中です。Astra Controlは、必要に応じてネームスペースを作成し、デスティネーションクラスタの新しいボリュームにPersistent Volumeクレーム (PVC ; 永続ボリューム要求) を作成し、SnapMirror関係を作成します。このステータスは、レプリケーションが再同期中であること、またはレプリケーションを反転中であることを示している可能性もあり
- * established * : レプリケーション関係が存在します。Astra Controlは、PVCが使用可能かどうかを定期的にチェックし、レプリケーション関係をチェックし、アプリケーションのSnapshotを定期的に作成し、アプリケーション内の新しいソースPVCを特定します。その場合は、レプリケーションに含めるリソースがAstra Controlによって作成されます。
- フェイルオーバー : SnapMirror関係が解除され、アプリケーションのKubernetesリソースが最後にレプリケートされたアプリケーションのSnapshotからリストアされます。
- *フェイルオーバーした場合 : Astra Controlは、ソースクラスタからのレプリケーションを停止し、デスティネーションでレプリケートされた最新の (成功した) アプリケーションSnapshotを使用して、Kubernetesリソースをリストアします。
- * resyncing * : Astra Controlは、SnapMirror resyncを使用して、再同期元の新しいデータを再同期先に再同期します。この処理では、同期の方向に基づいて、デスティネーション上の一部のデータが上書きされる可能性があります。Astra Controlは、デスティネーションネームスペースで実行されているアプリケーションを停止し、Kubernetesアプリケーションを削除します。再同期処理の実行中、ステータスは「Establishing」と表示されます。

- リバース：は、元のソースクラスタへのレプリケーションを続行しながらアプリケーションをデスティネーションクラスタに移動する予定の処理です。Astra Controlは、ソースクラスタ上のアプリケーションを停止し、デスティネーションにデータをレプリケートしてから、デスティネーションクラスタにアプリケーションをフェイルオーバーします。リバースレプリケーションの間、ステータスは「Establishing」と表示されます。
- 削除中：
 - レプリケーション関係が確立されたものの、まだフェイルオーバーされていない場合は、レプリケーション中に作成されたPVCがAstra Controlによって削除され、デスティネーションの管理対象アプリケーションが削除されます。
 - レプリケーションがすでにフェイルオーバーされている場合、Astra ControlはPVCと宛先アプリケーションを保持します。

アプリケーションのクローン作成と移行

既存のアプリケーションをクローニングして、同じKubernetesクラスタまたは別のクラスタに重複するアプリケーションを作成できます。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名前空間間でワークロードを移動できます。Astra Control Center UIまたはを使用できます ["Astra Control API"](#) アプリケーションのクローン作成と移行を実行します。



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名前空間フィルタを追加し、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名前空間にある場合、名前空間フィルタはデスティネーション名前空間にのみ適用されます。

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にクローニングすると、失敗します。
- * `ontap-nas-economy-backed storage class` *：アプリケーションがに基づくストレージクラスを使用している場合 `ontap-nas-economy` ドライバ：クローン処理のバックアップ部分はシステムの停止を伴います。バックアップが完了するまで、ソースアプリケーションは使用できません。クローン処理のリストア部分は無停止で実行されます。
- 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に対して、アプリケーションのクローンまたはリストアを実行できます。ただし、同じユーザが、クローニングまたはリストアされたアプリケーションに新しいネームスペースからアクセスすることはできません。クローンまたはリストア処理によって新しいネームスペースが作成されると、アカウントの管理者 / 所有者はメンバーユーザアカウントを編集し、該当するユーザに新しいネームスペースへのアクセスを許可するロールの制限を更新できます。
- 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。

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。
- 既存の Snapshot からクローンを作成するかバックアップを作成するかを選択します。このオプションを選択しない場合、Astra Control Center はアプリケーションの現在の状態からクローンを作成します。
 - 既存のSnapshotまたはバックアップからクローニングする場合は、使用するSnapshotまたはバックアップを選択します。
- 「* 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やサポートされているその他のストレージクラスに移行できます。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ップ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。

クローンまたはリストア処理によって新しいネームスペースが作成されると、アカウントの管理者 / 所有者はメンバーユーザアカウントを編集し、該当するユーザに新しいネームスペースへのアクセスを許可するロールの制限を更新できます。



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は、実行可能なタイミングに基づいて、次の種類の実行フックをサポートします。

- Snapshot前
- Snapshot後
- バックアップ前
- バックアップ後
- リストア後のPOSTコマンドです

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、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、同じコンテナの一部で実行フックが実行されるシナリオを作成できますが、すべてのコンテナで実行フックが実行されるわけではありません。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

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るか、完全に無効にすることが多いため、カスタム実行フックの実行時間を最小限に抑えるようにしてください。実行フックが関連付けられている状態でバックアップまたはスナップショット操作を開始した後、キャンセルした場合でも、バックアップまたはスナップショット操作がすでに開始されていればフックは実行できます。つまり、バックアップ後の実行フックで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。

- 実行フックは、スクリプトを使用してアクションを実行する必要があります。多くの実行フックは、同じスクリプトを参照できます。
- Astra Controlでは、実行フックが実行可能なシェルスクリプトの形式で記述されるようにするスクリプトが必要です。
- スクリプトのサイズは96KBに制限されています。
- Astra Controlは、実行フックの設定と一致条件を使用して、スナップショット、バックアップ、または復元操作に適用できるフックを決定します。
- 実行フックの障害はすべて'ソフトな障害'です。フックが失敗しても、他のフックとデータ保護操作は試行されます。ただし、フックが失敗すると、* アクティビティ * ページイベントログに警告イベントが記録されます。
- 実行フックを作成、編集、または削除するには、Owner、Admin、または Member 権限を持つユーザーである必要があります。
- 実行フックの実行に 25 分以上かかる場合、フックは失敗し、戻りコードが N/A のイベント・ログ・エントリが作成されます。該当する Snapshot はタイムアウトして失敗とマークされ、タイムアウトを通知するイベントログエントリが生成されます。
- アドホックデータ保護操作の場合、すべてのフックイベントが生成され、[Activity]ページのイベントログに保存されます。ただし、スケジュールされたデータ保護処理については、フック障害イベントだけがイベントログに記録されます（スケジュールされたデータ保護処理自体によって生成されたイベントは記録されません）。
- 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リストア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。
- 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。

実行順序

データ保護操作を実行すると、実行フックイベントが次の順序で実行されます。

1. 適用可能なカスタムプリオペレーション実行フックは、適切なコンテナで実行されます。カスタムのプリオペレーションフックは必要なだけ作成して実行できますが、操作前のこれらのフックの実行順序は保証も構成もされていません。
2. データ保護処理が実行されます。

3. 適用可能なカスタムポストオペレーション実行フックは、適切なコンテナで実行されます。必要な数のカスタムポストオペレーションフックを作成して実行できますが、操作後のこれらのフックの実行順序は保証されず、設定もできません。

同じ種類の実行フック（スナップショット前など）を複数作成する場合、これらのフックの実行順序は保証されません。ただし、異なるタイプのフックの実行順序は保証されています。たとえば、5つの異なるタイプのフックをすべて持つ構成の実行順序は、次のようになります。

1. 予備フックが実行されます
2. スナップショット前フックが実行されます
3. スナップショット後フックが実行されます
4. バックアップ後のフックが実行されます
5. 復元後のフックが実行されます

シナリオ番号2のこの設定の例は、の表を参照してください [\[フックが実行されるかどうかを確認します\]](#)。



本番環境で実行スクリプトを有効にする前に、必ず実行フックスクリプトをテストしてください。'kubectl exec' コマンドを使用すると、スクリプトを簡単にテストできます。本番環境で実行フックを有効にしたら、作成されたSnapshotとバックアップをテストして整合性があることを確認します。これを行うには、アプリケーションを一時的な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	ネームスペース	クラスタ	スナップショットフックが実行されます	バックアップフックが実行されます	フックを元に戻します
1.	クローン	N	N	新規	同じ	Y	N	Y
2.	クローン	N	N	新規	違う	Y	Y	Y
3.	クローン またはリストア	Y	N	新規	同じ	N	N	Y
4.	クローン またはリストア	N	Y	新規	同じ	N	N	Y
5.	クローン またはリストア	Y	N	新規	違う	N	N	Y
6.	クローン またはリストア	N	Y	新規	違う	N	N	Y
7.	リストア	Y	N	既存	同じ	N	N	Y
8.	リストア	N	Y	既存	同じ	N	N	Y
9.	スナップショット	該当なし	該当なし	該当なし	該当なし	Y	該当なし	該当なし
10.	バックアップ	N	該当なし	該当なし	該当なし	Y	Y	該当なし
11.	バックアップ	Y	該当なし	該当なし	該当なし	N	N	該当なし

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。

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つのスクリプトのみを変更することで、多数の実行フックを更新できます。

手順

1. 「アカウント」に移動します。
2. [スクリプト]タブを選択します。
3. 「* 追加」を選択します。
4. 次のいずれかを実行します。
 - カスタムスクリプトをアップロードする。
 - i. [ファイルのアップロード (Upload file)] オプションを選択します。
 - ii. ファイルを参照してアップロードします。
 - iii. スクリプトに一意の名前を付けます。
 - iv. (オプション) 他の管理者がスクリプトについて知っておく必要があるメモを入力します。
 - v. 「スクリプトを保存」を選択します。
 - クリップボードからカスタムスクリプトを貼り付けます。
 - i. [貼り付け (Paste)] または [タイプ (* type)] オプションを選択する
 - ii. テキストフィールドを選択し、スクリプトテキストをフィールドに貼り付けます。
 - iii. スクリプトに一意の名前を付けます。
 - iv. (オプション) 他の管理者がスクリプトについて知っておく必要があるメモを入力します。
5. 「スクリプトを保存」を選択します。

結果

新しいスクリプトが、[スクリプト]タブのリストに表示されます。

スクリプトを削除します

不要になって実行フックで使用されなくなったスクリプトは、システムから削除できます。

手順

1. 「アカウント」に移動します。
2. [スクリプト]タブを選択します。
3. 削除するスクリプトを選択し、「アクション」列のメニューを選択します。
4. 「* 削除」を選択します。



スクリプトが1つまたは複数の実行フックに関連付けられている場合、*Delete*アクションは使用できません。スクリプトを削除するには、まず関連する実行フックを編集し、別のスクリプトに関連付けます。

カスタム実行フックを作成します

アプリケーションのカスタム実行フックを作成できます。を参照してください [\[実行フックの例\]](#) フックの例を参照してください。実行フックを作成するには、Owner、Admin、または Member のいずれかの権限が必要です。



実行フックとして使用するカスタムシェルスクリプトを作成する場合は、特定のコマンドを実行するか、実行可能ファイルへの完全パスを指定する場合を除き、ファイルの先頭に適切なシェルを指定するようにしてください。

手順

1. 「* アプリケーション」を選択し、管理アプリの名前を選択します。
2. [実行フック*]タブを選択します。
3. 「* 追加」を選択します。
4. [フックの詳細* (Hook Details *)]領域で、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。
5. (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"](#)。



正規表現フィールドに他のテキストを含まない属性 (ポッド名など) の正確な名前です。フィルタリングすると、部分文字列の照合が実行されます。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\$)。

- d. 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。

6. 完了したら、「次へ」を選択します。
7. [* スクリプト * (* Script *)]領域で、次のいずれかを実行します。
 - 新しいスクリプトを追加します。
 - i. 「* 追加」を選択します。
 - ii. 次のいずれかを実行します。
 - カスタムスクリプトをアップロードする。
 - I. [ファイルのアップロード (Upload file)]オプションを選択します。
 - II. ファイルを参照してアップロードします。
 - III. スクリプトに一意の名前を付けます。
 - IV. (オプション) 他の管理者がスクリプトについて知っておく必要があるメモを入力します。
 - V. 「スクリプトを保存」を選択します。
 - クリップボードからカスタムスクリプトを貼り付けます。
 - I. [貼り付け (Paste)]または[タイプ (* type)]オプションを選択する
 - II. テキストフィールドを選択し、スクリプトテキストをフィールドに貼り付けます。
 - III. スクリプトに一意の名前を付けます。
 - IV. (オプション) 他の管理者がスクリプトについて知っておく必要があるメモを入力します。
 - リストから既存のスクリプトを選択します。

このスクリプトを使用するように実行フックに指示します。

8. 「* 次へ *」を選択します。
9. 実行フックの設定を確認します。
10. 「* 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"](#)

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ル

「* アプリケーション*」タイルは、次の項目を識別するのに役立ちます。

- Astra で現在管理しているアプリケーションの数。
- それらの管理アプリが正常であるかどうか。
- アプリケーションが完全に保護されているかどうか（最新のバックアップがある場合は保護されます）。
- 検出されたものの、まだ管理されていないアプリケーションの数。

アプリケーションが検出された後で管理または無視するため、この数はゼロになるのが理想的です。さらに、ダッシュボードで検出されたアプリケーションの数を監視して、開発者がクラスタに新しいアプリケーションを追加するタイミングを特定します。

クラスタタイル

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

「ストレージバックエンド *」 タイルは、ストレージバックエンドの健全性を特定するための情報を提供します。これには次のものが含まれます。

- 管理対象のストレージバックエンドの数
- これらの管理バックエンドが正常であるかどうか
- バックエンドが完全に保護されているかどうか
- 検出されたがまだ管理されていないバックエンドの数。

クラスタの健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場合と成功する) 場合は、Astra Controlに指定したkubeconfigが無効になる可能性があります。クラスタでの証明書のローテーションまたは有効期限が原因の可能性があります。この問題を修正するには、を使用して、Astra Control のクラスタに関連付けられたクレデンシャルを更新します "[Astra Control API の略](#)"。

3. [Overview (概要)]、[* Storage (* ストレージ)]、[* Activity * (アクティビティ *)] タブの情報を表示して、必要な情報を検索します。
 - * 概要 * : 状態を含むワーカーノードの詳細。
 - * ストレージ * : ストレージクラスと状態を含む、コンピューティングに関連付けられた永続的ボリューム。
 - * アクティビティ * : クラスタに関連するアクティビティを表示します。



Astra Control Center * Dashboard * から始まるクラスタ情報を表示することもできます。[* クラスタ *] タブの [* リソースサマリ *] で、管理対象クラスタを選択して [* クラスタ *] ページに移動できます。[* Clusters] ページが表示されたら、上記の手順を実行します。

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、アプリケーションのステータス（正常かどうか）、保護ステータス（障害発生時に完全に保護されているかどうか）、ポッド、永続的ストレージなどを識別できる詳細がAstra Controlに表示されます。

手順

1. Astra Control Center UI で、* アプリケーション* を選択し、アプリの名前を選択します。
2. 情報を確認します。
 - アプリステータス：Kubernetesでのアプリの状態を反映するステータスを提供します。たとえば、ポッドと永続ボリュームはオンラインか？アプリケーションが正常な状態でない場合は、Kubernetesのログでクラスタの問題を調べてトラブルシューティングする必要があります。Astraは、壊れたアプリケーションの修正に役立つ情報を提供していません。
 - アプリ保護ステータス：アプリ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](#)" ユーザを管理するには、[こちら](#)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せん。

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"](#) 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は、LDAPの「メール」属性でメールアドレスを使用して、リモートユーザーを検索し、追跡します。この属性は、ディレクトリ内のオプションのフィールドまたは空のフィールドです。Astra Control Centerに表示するリモートユーザの場合は、このフィールドにEメールアドレスが存在している必要があります。この電子メールアドレスは、Astra Control Centerのユーザー名として認証に使用されます。

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)。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. [ユーザー一致]セクションで、LDAPサーバからユーザー情報を取得するときに使用するベースDNと適切なユーザー検索フィルタを入力します。
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サーバー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サーバー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。



同じEメールアドレスの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。



Astra Control Centerは、LDAPの「メール」属性でメールアドレスを使用して、リモートユーザを検索し、追跡します。この属性は、ディレクトリ内のオプションのフィールドまたは空のフィールドです。Astra Control Centerに表示するリモートユーザの場合は、このフィールドにEメールアドレスが存在する必要があります。この電子メールアドレスは、Astra Control Centerのユーザー名として認証に使用されます。

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に追加され、同じロールを継承します。

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。



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

新しいグループがリモートグループのリストに表示され、このグループ内のすべてのリモート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、*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"](#)。



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

クレデンシャルを削除する

アカウントからのクレデンシャルの削除はいつでも実行できます。クレデンシャルは、のあとに削除してください ["関連するすべてのクラスタの管理を解除します"](#)。



Astra Control Center は、Astra Control Center の認証情報を使用してバックアップバケットに認証するため、Astra Control Center に追加する最初の資格情報セットは常に使用されています。これらのクレデンシャルは削除しないことを推奨します。

手順

1. 「* アカウント *」を選択します。
2. [*Credentials] タブを選択します。
3. 削除するクレデンシャルの [状態 *] 列で [オプション] メニューを選択します。
4. 「* 削除」を選択します。
5. 削除を確認するために「削除」と入力し、「はい」、「認証情報を削除」を選択します。

結果

Astra Control Center は、アカウントから資格情報を削除します。

アカウントのアクティビティを監視

Astra Control アカウントのアクティビティの詳細を表示できます。たとえば、新しいユーザーを招待したとき、クラスタが追加されたとき、Snapshot が作成されたときなどです。アカウントアクティビティを CSV ファイルにエクスポートすることもできます。



KubernetesクラスタをAstra Controlから管理し、Astra ControlをCloud Insights に接続した場合、Astra ControlはCloud Insights にイベントログを送信する。ポッドの導入やPVCの添付ファイルに関する情報などのログ情報が、Astra Control Activityログに記録されます。この情報を使用して、管理しているKubernetesクラスタの問題を特定します。

Astra Control のアカウントアクティビティをすべて表示

1. 「* Activity *」を選択します。
2. フィルタを使用してアクティビティのリストを絞り込むか、検索ボックスを使用して探しているものを正確に検索します。
3. アカウントアクティビティを CSV ファイルにダウンロードするには、「* CSV にエクスポート」を選択します。

特定のアプリケーションのアカウントアクティビティを表示します

1. 「* アプリケーション」を選択し、アプリケーションの名前を選択します。
2. 「* Activity *」を選択します。

クラスタのアカウントアクティビティを表示します

1. 「* クラスタ」を選択し、クラスタの名前を選択します。
2. 「* Activity *」を選択します。

対応が必要なイベントを解決するための操作を実行します

1. 「* Activity *」を選択します。
2. 注意が必要なイベントを選択してください。
3. [Take action] ドロップダウンオプションを選択します。

このリストから、実行できる対処方法のほか、問題に関するドキュメントを参照したり、問題の解決に役立つサポートを受けたりできます。

既存のライセンスを更新する

評価用ライセンスをフルライセンスに変換したり、既存の評価用ライセンスまたはフルライセンスを新しいライセンスで更新したりできます。フルライセンスがない場合は、ネットアップの営業担当者に連絡して、ライセンスとシリアル番号の全文を入手してください。Astra Control Center UIまたはを使用できます ["Astra Control API"](#) 既存のライセンスを更新します。

手順

1. にログインします ["NetApp Support Site"](#)。
2. Astra Control Center のダウンロードページにアクセスし、シリアル番号を入力して、ネットアップライセンスファイル（NLF）をダウンロードする。
3. Astra Control Center UI にログインします。
4. 左側のナビゲーションから、* アカウント * > * ライセンス * を選択します。
5. **[Account>*License*]** ページで、既存のライセンスのステータスドロップダウンメニューを選択し、**[Replace]** を選択します。
6. ダウンロードしたライセンスファイルを参照します。
7. 「* 追加」を選択します。

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

を参照してください。

- ["Astra Control Center のライセンス"](#)

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の略
- 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"](#)
- [\[バケットを編集する\]](#)
- [\[デフォルトバケットを設定する\]](#)
- [\[バケットのクレデンシャルをローテーションするか、削除する\]](#)
- [\[バケットを削除する\]](#)



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。

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。

ある場合は、続行できません。

手順

1. 左ナビゲーションから、*バケット* を選択します。
2. [アクション* (Actions*)] メニューから、[*削除 (Remove)] を選択します。



Astra Control を使用すると、最初にバケットを使用してバックアップを実行するスケジュールポリシーが存在せず、削除しようとしているバケットにアクティブなバックアップが存在しないようにすることができます。

3. 「remove」と入力して操作を確認します。
4. 「*Yes、remove bucket*」を選択します。

詳細については、こちらをご覧ください

- ["Astra Control API を使用"](#)

ストレージバックエンドを管理します

ストレージバックエンドとして Astra Control のストレージクラスを管理することで、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クを取得できるだけでなく、追加のストレージ指標も取得できます。ストレージ容量と健全性の詳細を監視できます。Astra Control Center が Cloud Insights に接続されている場合のパフォーマンスも監視できます。

Astra Control API を使用してストレージバックエンドを管理する方法については、を参照してください ["Astra の自動化と API に関する情報"](#)。

ストレージバックエンドの管理に関連して、次のタスクを実行できます。

- ["ストレージバックエンドを追加します"](#)
- [\[ストレージバックエンドの詳細を表示します\]](#)
- [\[ストレージバックエンド認証の詳細を編集します\]](#)

- [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。Cloud Insights を使用すると、追加情報が表示されます。を参照してください "[Cloud Insights のドキュメント](#)"。

手順

1. 左側のナビゲーション領域で、* Backends * を選択します。
2. ストレージバックエンドを選択します。



NetApp Cloud Insights に接続した場合、Cloud Insights からの抜粋がバックエンドのページに表示されます。

Name	Persistent volume	Capacity	App/s	Cluster/s	Cloud
trident_pvc_...	pvc_...	0.04/46.57 GiB: 0.1%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc_...	0.34/23.28 GiB: 1.44%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc_...	0.02/0.93 GiB: 2.33%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc_...	3.02/50.00 GiB: 6.04%	netapp-acc polaris-mongodb-mongodb	openshift-cluster010	private
trident_pvc_...	pvc_...	0.19/8.00 GiB: 2.39%	apps-mysql mysql-mysql	openshift-cluster010	private
trident_pvc_...	pvc_...	0.41/50.00 GiB: 0.81%	netapp-acc polaris-influxdb2-polaris-influxdb2	openshift-cluster010	private
trident_pvc_...	pvc_...	2.93/50.00 GiB: 5.87%	netapp-acc polaris-mongodb-mongodb	openshift-cluster010	private
trident_pvc_...	pvc_...	0.03/10.00 GiB: 0.26%	netapp-acc polaris-consul-consul	openshift-cluster010	private

3. Cloud Insights に直接移動するには、指標画像の横にある * Cloud Insights * アイコンを選択します。

ストレージバックエンド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法は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. オプションから[管理解除]*を選択します。
 - 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. オプションから[削除]*を選択します。
 - c. 「remove」と入力して操作を確認します。
 - d. 「* Yes、 remove storage backend *」を選択します。

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せん。

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"](#)。

Cloud Insights、Prometheus、Fluentd接続でインフラを監視します

複数のオプション設定を構成して、Astra Control Center の操作性を高めることができます。インフラ全体を監視して詳細を把握するには、NetApp Cloud Insights への接続を作成し、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プロキシが必要な場合（サポートバンドルをNetApp Support Site にアップロードする場合、またはCloud Insights への接続を確立する場合は、Astra Control Centerでプロキシサーバを設定する必要があります。

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

Cloud Insights またはNetApp Support Site への接続に使用するプロキシサーバを追加します

Astra Control Centerを実行しているネットワークで、インターネットに接続するためのプロキシが必要な場合（サポートバンドルをNetApp Support Site にアップロードする場合、またはCloud Insights への接続を確立する場合は、Astra Control Centerでプロキシサーバを設定する必要があります。



Astra Control Center は、プロキシサーバ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。



HTTP PROXY

Configure Astra Control to send traffic through a proxy server.

Disconnected

Connect

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。



Connected



HTTP PROXY ?

Server: proxy.example.com:8888

Authentication: Enabled

プロキシサーバーの設定を編集します

プロキシサーバー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。

プロキシサーバー接続を無効にします

プロキシサーバー接続を無効にすることができます。他の接続が中断される可能性があることを無効にする前に警告が表示されます。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account>*Connections***] を選択します。
3. 接続を無効にするには、ドロップダウンリストから * 切断 * を選択します。
4. 表示されたダイアログボックスで、処理を確認します。

Cloud Insights に接続します

NetApp Cloud Insights を Astra Control Center インスタンスに接続すると、インフラ全体を監視して詳細に把握できます。Cloud Insights は、Astra Control Center ライセンスに含まれています。

Cloud Insights には、Astra Control Center が使用するネットワークから、またはプロキシサーバー経由で間接的にアクセスできる必要があります。

Cloud Insights にアストラコントロールセンターを接続すると、Acquisition Unit ポッドが作成されます。この

ポッドは、Astra Control Center で管理されているストレージバックエンドからデータを収集し、Cloud Insights にプッシュします。このポッドには、8GB の RAM と 2 つの CPU コアが必要です。



Astra Control CenterをCloud Insights とペアリングしている場合は、Cloud Insights の*[Modify Deployment]*オプションは使用しないでください。



Cloud Insights 接続を有効にすると、* backends ページでスループット情報を確認できるようになります。また、ストレージバックエンドを選択したあとに**Cloud Insights** に接続することもできます。[クラスタ]セクションの[ダッシュボード]*で情報を確認し、そこからCloud Insights に接続することもできます。

作業を開始する前に

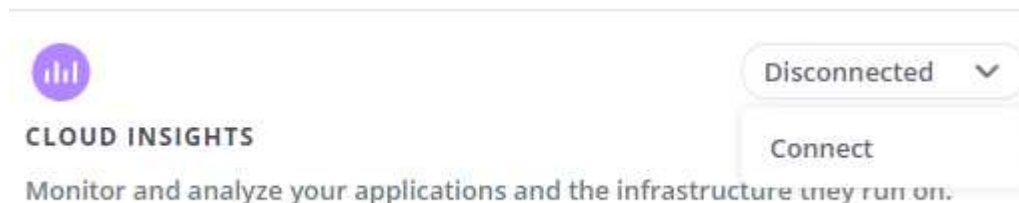
- admin * / * owner * 権限を持つ Astra Control Center アカウント。
- 有効な Astra Control Center ライセンス。
- Astra Control Center を実行しているネットワークで、インターネットに接続するためにプロキシが必要な場合は、プロキシサーバーです。



Cloud Insights を初めて使用する場合は、の機能について理解しておいてください。を参照してください "[Cloud Insights のドキュメント](#)"。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. 接続を追加するには、ドロップダウンリストで * 切断されている * と表示されている * 接続 * を選択します。



オプションを表示し

て、Cloud Insights 接続を有効にします。"]

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

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

テナント URL は、Cloud Insights ライセンスを取得すると取得されます。テナント URL がない場合は、を参照してください "[Cloud Insights のドキュメント](#)"。

- a. をダウンロードしてください "[API トークン](#)"をクリックし、Cloud Insights テナントの URL にログインします。
- b. Cloud Insights で、* Admin*>* API Access* をクリックして、* Read/Write * と * Read Only* API Access トークンの両方を生成します。

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

- c. 「 * Read Only * 」キーをコピーします。Cloud Insights 接続を有効にするには、[Astra Control Center] ウィンドウに貼り付ける必要があります。Read API Access Token Key 権限で、Assets、Alerts、Acquisition Unit、and Data Collection を選択します。
- d. 「 * Read/Write 」キーをコピーします。Astra Control Center * Connect Cloud Insights * ウィンドウに貼り付ける必要があります。Read/Write API Access Tokenキーの権限で、Data Ingestion、Log Ingestion、Acquisition Unit、およびData Collectionを選択します。



* 読み取り専用 * キーと * 読み取り / 書き込み * キーを生成することを推奨します。両方の目的で同じキーを使用することは推奨しません。デフォルトでは、トークンの有効期限は 1 年に設定されています。トークンが期限切れになるまでの最大期間を指定するために、デフォルトの選択を維持することをお勧めします。トークンの有効期限が切れると、テレメトリが停止します。

e. Cloud Insights からコピーしたキーを Astra コントロールセンターに貼り付けます。

5. 「 * 接続 」を選択します。



[* 接続] を選択すると、[* アカウント * > * 接続 *] ページの [* Cloud Insights *] セクションで、接続の状態が [* 保留中] に変わります。接続が有効になり、ステータスが * 接続済み * に変わるまで数分かかることがあります。



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

Cloud Insights でデータを表示します

接続に成功した場合は、「 * アカウント * > * 接続 * 」ページの「 * Cloud Insights * 」セクションに接続されていることが示され、テナントの URL が表示されます。Cloud Insights にアクセスして、データが正常に受信されて表示されることを確認できます。

EXTERNAL ?

The screenshot shows two connection cards. The first is for 'HTTP PROXY' with a server address of 'proxy.example.com:8888' and authentication enabled. The second is for 'CLOUD INSIGHTS' with a tenant of 'Cloud Insights'. Both cards show a 'Connected' status with a dropdown arrow.

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。

The notification message states: 'Unable to connect to Cloud Insights an hour ago. The Cloud Insights API token is invalid. Create a new API token in Cloud Insights and update Astra Control connection settings with the new token.'

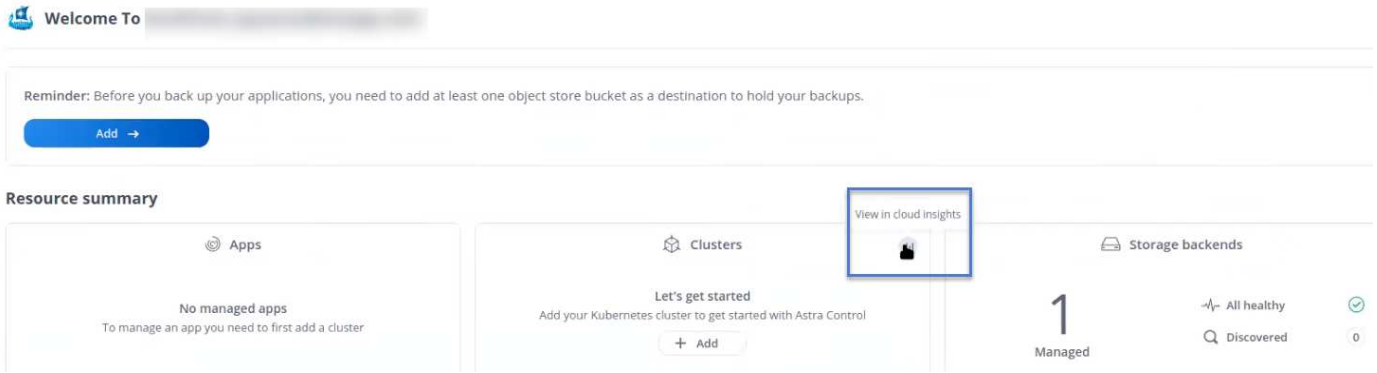
同じ情報は、「* アカウント * > * 通知 *」にも記載されています。

Astra Control Center では、スループット情報をバックエンド * ページで表示したり、ストレージバックエンドを選択した後にここから Cloud Insights に接続したりできます。

The screenshot shows the 'Backends' page with a table of storage backends. One backend is highlighted with a 'Throughput' graph overlay. The graph shows a throughput of 8.00 MB/s, with a 5m ago value of 8.00 MB/s, a minimum of 4.00 MB/s, and a maximum of 11.00 MB/s. A 'View in Cloud Insights' link is provided.

Cloud Insights に直接移動するには、指標画像の横にある * Cloud Insights * アイコンを選択します。

また、情報は * ダッシュボード * でも確認できます。



Cloud Insights 接続を有効にした後、Astra Control Center に追加したバックエンドを削除すると、バックエンドは Cloud Insights へのレポートを停止します。

Cloud Insights 接続を編集します

Cloud Insights 接続を編集できます。



編集できるのは API キーのみです。Cloud Insights テナント URL を変更するには、Cloud Insights 接続を切断して新しい URL に接続することを推奨します。

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. ドロップダウンリストから*[編集]*を選択して、接続を編集します。
4. Cloud Insights 接続設定を編集します。
5. [保存 (Save)] を選択します。

Cloud Insights 接続を無効にします

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

手順

1. * admin * / * owner * 権限を持つアカウントを使用して Astra Control Center にログインします。
2. [**Account**>*Connections*] を選択します。
3. 接続を無効にするには、ドロップダウンリストから * 切断 * を選択します。
4. 表示されたダイアログボックスで、処理を確認します。
操作を確定すると、[**Account**>*Connections*] ページで、Cloud Insights のステータスが [*Pending (保留中)] に変わります。ステータスが * 切断された * に変わるまで数分かかります。

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. に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

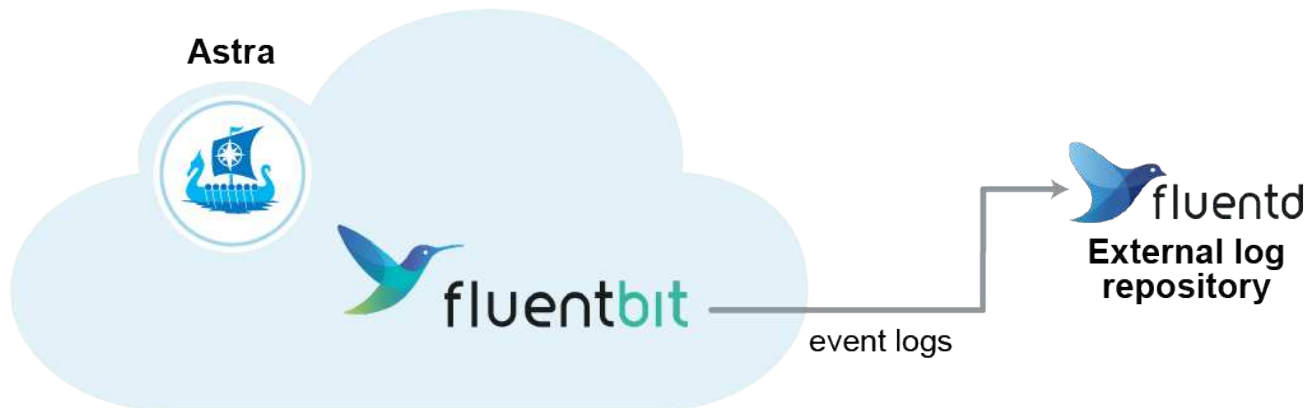
1. ブラウザで、PrometheusのURLを入力します。
2. Prometheusメニューで* Graph *を選択します。
3. メトリクスエクスプローラを使用するには、[Execute]の横にあるアイコンを選択します。
4. 選択するオプション `scrape_samples_scraped` をクリックし、* Execute *を選択します。
5. 時間の経過に伴うサンプルのスクレイピングを確認するには、* Graph *を選択します。



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。



管理対象クラスター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

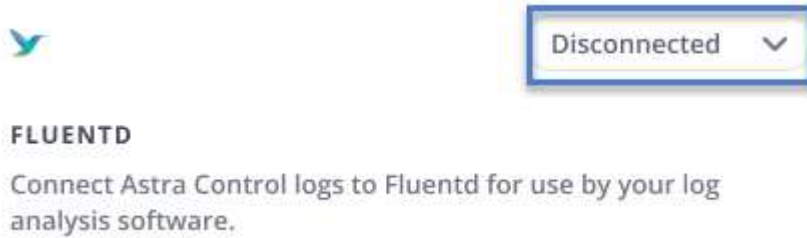
- admin * / * owner * 権限を持つ Astra Control Center アカウント。
- Kubernetes クラスターに Astra Control Center をインストールして実行



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 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。



* Astra Control Center と Cloud Insights が接続されていない場合 *、クラスタの管理を解除すると、テレメトリ・データの送信用にインストールされたすべてのリソースが削除されます。* Astra Control CenterとCloud Insights が接続されている場合*、クラスタの管理を解除すると、のみが削除されます fluentbit および event-exporter ポッド

Astra Control Center をアップグレードします

Astra Control Centerをアップグレードするには、NetApp Support Site からインストールバンドルをダウンロードし、以下の手順を実行します。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。

作業を開始する前に

- アップグレードする前に、を参照してください ["運用環境の要件"](#) 環境がAstra Control Center導入の最小要件を満たしていることを確認する。環境に次の要素が必要です。
 - サポートされるAstra Tridentバージョン
実行しているバージョンを確認するには、既存のAstra Control Centerに対して次のコマンドを実行します。

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

を参照してください ["Astra Trident のドキュメント"](#) 古いバージョンからアップグレードするには、次の手順に従います。



Kubernetes 1.25にアップグレードするには、Astra Trident 22.10 *にアップグレードする必要があります。

- サポートされているKubernetesディストリビューション
実行しているバージョンを確認するには、既存のAstra Control Centerに対して次のコマンドを実行します。 `kubectl get nodes -o wide`
- 十分なクラスタリソース
クラスタのリソースを確認するには、既存のAstra Control Centerクラスタで次のコマンドを実行します。 `kubectl describe node <node name>`
- Astra Control Centerイメージのプッシュおよびアップロードに使用できるレジストリ

- 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するには、既存のAstra Control Centerに対して次のコマンドを実行します。 `kubectl get storageclass`
- (OpenShiftのみ) すべてのクラスタオペレータが正常な状態であり、使用可能であることを確認します。

```
kubectl get clusteroperators
```

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

```
kubectl get apiservices
```

- 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。

このタスクについて

Astra Control Center のアップグレードプロセスでは、次の手順を実行できます。

- [Astra Control Center](#)をダウンロードして展開します
- [NetApp Astra kubectlプラグイン](#)を削除して、再度インストールします
- [\[イメージをローカルレジストリに追加します\]](#)
- [更新された Astra Control Center オペレータ](#)をインストールします
- [Astra Control Center](#) をアップグレードします
- [\[システムステータスを確認します\]](#)



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。



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。

Astra Control Centerをダウンロードして展開します

1. にアクセスします ["Astra Control Centerの製品ダウンロードページ"](#) をクリックしますNetApp Support Site。ドロップダウンメニューから最新バージョンまたは別のバージョンを選択できます。
2. 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`)。
3. (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`) バンドルの署名を確認するには、次の手順を実行します。

```
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、

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

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

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

NetApp Astra kubectlコマンドラインプラグインを使用して、ローカルのDockerリポジトリにイメージをプッシュできます。

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、コマンドはkubectlプラグインのヘルプを返す必要があります。既存のバージョンのkubectl-mstraを削除するには、次のコマンドを実行します。 `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
```

イメージをローカルレジストリに追加します

1. コンテナエンジンに応じた手順を実行します。

Docker です

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

2. 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)。
 - <MY_FULL_REGISTRY_PATH> 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://<docker-registry>" class="bare">https://<docker-registry>"。
 - <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。

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

ポドマン

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

2. レジストリにログインします。

```
podman login <YOUR_REGISTRY>
```

3. 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=23.04.2-7
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=23.04.2-7
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://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/23.04.2-7/image:version>

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

1. ディレクトリを変更します。

```
cd manifests
```

2. Astra Control Centerオペレータ配置YAMLを編集します

(astra_control_center_operator_deploy.yaml)を参照して、ローカルレジストリとシークレットを参照してください。

```
vim astra_control_center_operator_deploy.yaml
```

- a. 認証が必要なレジストリを使用する場合は、のデフォルト行を置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}]
```

- b. 変更 [your_registry_path] をクリックします kube-rbac-proxy でイメージをプッシュしたレジストリパスへのイメージ [前の手順](#)。
- c. 変更 [your_registry_path] をクリックします acc-operator でイメージをプッシュしたレジストリパスへのイメージ [前の手順](#)。
- d. に次の値を追加します env セクション。

```
- name: ACCOP_HELM_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: [your_registry_path]/kube-rbac-proxy:v4.8.0
  name: kube-rbac-proxy
  ports:
  - containerPort: 8443
    name: https
- args:
  - --health-probe-bind-address=:8081
  - --metrics-bind-address=127.0.0.1:8080
  - --leader-elect
  env:
  - name: ACCOP_LOG_LEVEL
    value: "2"
  - name: ACCOP_HELM_UPGRADE_TIMEOUT
    value: 300m
  image: [your_registry_path]/acc-operator:23.04.36
  imagePullPolicy: IfNotPresent
  livenessProbe:
    httpGet:
      path: /healthz
      port: 8081
      initialDelaySeconds: 15
      periodSeconds: 20
  name: manager
  readinessProbe:
    httpGet:
      path: /readyz
      port: 8081
      initialDelaySeconds: 5
      periodSeconds: 10
  resources:
    limits:
      cpu: 300m
      memory: 750Mi
    requests:
      cpu: 100m
      memory: 75Mi
  securityContext:
    allowPrivilegeEscalation: false
  imagePullSecrets: []
  securityContext:

```

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

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

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

回答例：

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

4. ポッドが実行中であることを確認します

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

Astra Control Center をアップグレードします

1. Astra Control Centerカスタムリソース (CR) を編集します。

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

2. Astraのバージョン番号を変更します (astraVersion の内部 spec) をアップグレードするバージョンにアップグレードします。

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

3. イメージレジストリパスが、イメージをプッシュしたレジストリパスと一致す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以降にレジストリが変更されている場合。

```
imageRegistry:
  name: "[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
```

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	23.04.2-7	
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。

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

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

2. を削除するには、次のコマンドを使用します netapp-acc（またはカスタム名）ネームスペース：

```
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
```

3. リソースの削除を確認します。

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

結果

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

4. 監視エージェントが削除されたことを確認：

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

サンプル結果：

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

5. カスタムリソース定義（CRD）情報の削除：

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

結果

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

Astra Control Center をアンインストールしても、Traefik CRD をクリーンアップできない

Traefik CRD を手動で削除できます。CRD はグローバルリソースであり、削除するとクラスタ上の他のアプリケーションに影響を与える可能性があります。

手順

1. クラスタにインストールされている Traefik CRD を表示します。

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

応答

```
ingressroutes.traefik.containo.us          2021-06-23T23:29:11Z  
ingressroutetcps.traefik.containo.us      2021-06-23T23:29:11Z  
ingressrouteudps.traefik.containo.us      2021-06-23T23:29:12Z  
middlewares.traefik.containo.us           2021-06-23T23:29:12Z  
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"](#)

Astra Control REST APIで自動化

Astra Control REST API による自動化

Astra Control には REST API が搭載されており、Curl などのプログラミング言語やユーティリティを使用して Astra Control 機能に直接アクセスできます。また、Ansible などの自動化テクノロジーを使用して Astra Control 環境を管理することもできます。

Kubernetesアプリケーションをセットアップおよび管理するには、Astra Control Center UIまたはAstra Control APIのいずれかを使用できます。

詳細については、を参照してください "[Astra 自動ドキュメント](#)"。

知識とサポート

トラブルシューティング

発生する可能性のある一般的な問題を回避する方法について説明します。

["NetApp Knowledge Base for Astra"](#)

詳細については、こちらをご覧ください

- ["ネットアップにファイルをアップロードする方法 \(ログインが必要\)"](#)
- ["ネットアップにファイルを手動でアップロードする方法 \(ログインが必要\)"](#)

ヘルプを表示します

ネットアップでは、Astra Control をさまざまな方法でサポートしています。ナレッジベース (KB) 記事やDiscordチャンネルなど、24時間365日利用可能な無料のセルフサポートオプションをご用意しています。Astra Control アカウントには、Web チケット発行によるリモートテクニカルサポートが含まれています。



Astra Control Center の評価用ライセンスをお持ちの場合は、テクニカルサポートを受けることができます。ただし、NetApp Support Site (NSS) でケースを作成することはできません。フィードバック・オプションを使用してサポートに連絡するか、Discordチャンネルを使用してセルフサービスで連絡できます。

最初に実行する必要があります ["ネットアップのシリアル番号のサポートを有効にします"](#) これらの非セルフサービスサポートオプションを使用するには、次の手順に従います。チャットや Web でのチケット発行、ケース管理には、NetApp Support Site (NSS) の SSO アカウントが必要です。

セルフサポートオプション

メインメニューから * Support * タブを選択すると、Astra Control Center UI からサポートオプションにアクセスできます。

次のオプションは、24 時間 365 日無料で利用できます。

- ["* ナレッジベース * \(ログインが必要\)"](#) : Astra Control に関する記事、FAQ、またはトラブルシューティング情報を検索します。
- ["* ドキュメントセンター *](#) : 現在表示しているドキュメントサイトです。
- ["* Discord *で助けを得なさい"](#) : thePubカテゴリのAstraに移動して、他のユーザやエキスパートと交流しましょう。
- ["* サポートケースの作成 *](#) : トラブルシューティングのためにネットアップサポートに提供するサポートバンドルを生成
- ["* Astra Control についてのフィードバック *](#) : astra.feedback@netapp.com に電子メールを送信して、あなたの考え、アイデア、懸念を知らせてください。

ネットアップサポートへのサポートバンドルの日次アップロードを有効にします

Astra Control Centerのインストール中に、を指定した場合 `enrolled: true` の場合 `autoSupport Astra Control Center` カスタムリソース (CR) ファイル (`astra_control_center.yaml`) を指定すると、日次サポートバンドルが自動的にアップロードされます **"NetApp Support Site"**。

ネットアップサポートに提供するサポートバンドルを生成する

Astra Control Center を使用すると、管理者ユーザはバンドルを生成できます。バンドルには、ネットアップサポートに役立つログ、Astra 環境のすべてのコンポーネントに対するイベント、管理対象のクラスタとアプリケーションに関する指標、トポロジ情報などが含まれます。インターネットに接続している場合は、NetApp Support Site (NSS) に Astra Control Center の UI から直接サポートバンドルをアップロードすることができます。



Astra Control Center がバンドルを生成するのにかかる時間は、Astra Control Center のインストールのサイズと、要求されたサポートバンドルのパラメータによって異なります。サポートバンドルの生成要求時に指定した期間によって、バンドルの生成にかかる時間が決まります (たとえば、期間を短くするとバンドルの生成にかかる時間が短縮されます)。

作業を開始する前に

NSS へのバンドルのアップロードにプロキシ接続が必要かどうかを確認します。プロキシ接続が必要な場合は、Astra Control Center がプロキシサーバーを使用するように設定されていることを確認します。

1. [`* アカウント * > * 接続 *`] を選択します。
2. `* 接続設定 *` でプロキシ設定を確認します。

手順

1. NSS ポータルで、Astra Control Center UI の `* Support *` ページに記載されているライセンスシリアル番号を使用してケースを作成します。
2. Astra Control Center UI を使用して、サポートバンドルを生成するには、次の手順を実行します。
 - a. [`サポート * (Support *)`] ページの [`サポートバンドル (Support bundle)`] タイルで、 [`* 生成 (Generate)`] を選択します。
 - b. [`* サポートバンドルの生成 * (Generate a Support Bundle *)`] ウィンドウで、期間を選択します。

クイックタイムフレームまたはカスタムタイムフレームのいずれかを選択できます。



カスタムの日付範囲を選択したり、期間にカスタムの期間を指定したりできます。

- c. 選択したら、`* 確認 *` を選択します。
- d. [`生成時にNetApp Support Siteにバンドルをアップロードする *`] チェックボックスをオンにします。
- e. [`* バンドルの生成 *`] を選択します。

サポートバンドルの準備が完了すると、アラート領域の `* アカウント * > * 通知 *` ページ、`* アクティビティ *` ページ、および通知リストに通知が表示されます (UI の右上にあるアイコンを選択してアクセスできます)。

生成が失敗した場合は、バンドルの生成ページにアイコンが表示されます。アイコンを選択すると、メッ

セージが表示されます。



UI の右上にある通知アイコンには、バンドルの作成に成功したタイミング、バンドルの作成に失敗したタイミング、バンドルをアップロードできなかったタイミング、バンドルをダウンロードできなかったタイミングなど、サポートバンドルに関連するイベントに関する情報が表示されます。

エアギャップ設置がある場合

エアギャップのある設置の場合は、サポートバンドルが生成されたあとに次の手順を実行します。バンドルがダウンロード可能になると、[サポート *] ページの [サポートバンドル *] セクションの [生成 *] の横に [ダウンロード] アイコンが表示されます。

手順

1. ダウンロードアイコンを選択して、バンドルをローカルにダウンロードします。
2. 手動で NSS にバンドルをアップロードします。

これを行うには、次のいずれかの方法を使用します。

- 使用 "[NetApp Authenticated File Upload \(ログインが必要\)](#) "。
- NSS でケースにバンドルを直接添付します。
- NetApp Active IQ を使用する。

詳細については、こちらをご覧ください

- "[ネットアップにファイルをアップロードする方法 \(ログインが必要\)](#) "
- "[ネットアップにファイルを手動でアップロードする方法 \(ログインが必要\)](#) "

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

以前のリリースのドキュメントを参照できます。

- ["Astra Control Center 22.11ドキュメント"](#)
- ["Astra Control Center 22.08ドキュメント"](#)
- ["Astra Control Center 22.04 のドキュメント"](#)
- ["Astra Control Center 21.12ドキュメント"](#)
- ["Astra Control Center 21.08 ドキュメント"](#)

法的通知

著作権に関する声明、商標、特許などにアクセスできます。

著作権

["https://www.netapp.com/company/legal/copyright/"](https://www.netapp.com/company/legal/copyright/)

商標

NetApp、NetApp のロゴ、および NetApp の商標ページに記載されているマークは、NetApp, Inc. の商標です。その他の会社名および製品名は、それぞれの所有者の商標である場合があります。

["https://www.netapp.com/company/legal/trademarks/"](https://www.netapp.com/company/legal/trademarks/)

特許

ネットアップが所有する特許の最新リストは、次のサイトで入手できます。

<https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf>

プライバシーポリシー

["https://www.netapp.com/company/legal/privacy-policy/"](https://www.netapp.com/company/legal/privacy-policy/)

オープンソース

通知ファイルには、ネットアップソフトウェアで使用されるサードパーティの著作権およびライセンスに関する情報が記載されています。

- ["Astra Control Center 23.04.2に関するお知らせ"](#)
- ["Astra Control Center 23.04.0に関するお知らせ"](#)

Astra Control API ライセンス

<https://docs.netapp.com/us-en/astra-automation/media/astra-api-license.pdf>

著作権に関する情報

Copyright © 2023 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、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。