アプリを管理、保護します

Astra Control Service

NetApp
June 04, 2024

目次

アプリを管理、保護します

アプリの管理を開始します

お先にどうぞ "Kubernetes クラスタを Astra Control に追加"をクリックして、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を定義します。

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

アプリケーション管理の要件

Astra Control には、次のアプリケーション管理要件があります。

- ライセンス：10個を超えるネームスペースを管理するには、Astra Controlサブスクリプションが必要です。
- 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、アプリケーションデータの完全な管理機能がない可能性があります。Astra Control では、次の Kubernetes リソースが収集されます。

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

サポートされているアプリインストール方法

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

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

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

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

  これらのパターンに続くいくつかのオペレータアプリを次に示します。

  - "Apache K8ssandra"

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

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

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

クラスタにアプリをインストールします

お先にどうぞ "クラスタが追加されました" Astra Controlを使用すると、アプリケーションをインストールしたり、クラスタ上の既存のアプリケーションを管理したりできます。1つ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。

Astra Controlは、Astra Controlでサポートされているストレージクラス上にストレージがある場合にのみ、ステートフルアプリケーションを管理する。Astra Control Serviceは、Astra Control Provisionerまたは汎用CSIドライバでサポートされるすべてのストレージクラスをサポートします。

- "GKE クラスタのストレージクラスについて説明します"
- "AKS クラスタのストレージクラスについて学習します"
- "AWSクラスタのストレージクラスについて説明します"

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

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

作業を開始する前に

- KubernetesクラスタをAstra Controlに追加。
- クラスタにインストールされているアプリケーションが1つ以上あります。 サポートされているアプリケーションのインストール方法については、こちらをご覧ください。
- Astra Controlに追加したKubernetesクラスタ上の既存のネームスペース。
- （オプション）すべてのにKubernetesラベルを付けます "サポートされるKubernetesリソース"。

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

このタスクについて

- 開始する前に、を理解しておく必要があります "標準ネームスペースとシステムネームスペースの管理"。
- Astra Controlのアプリケーションで複数の名前空間を使用する場合は、を検討してください "ネームスペースの制約を持つユーザロールの変更" アプリケーションを定義する前に。
- Astra Control APIを使用してアプリケーションを管理する方法については、を参照してください "Astra の自動化と API に関する情報"。

アプリケーション管理オプション

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

アプリケーションとして管理するリソースを定義します

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

アプリケーションを定義することで、クローン、スナップショット、バックアップなどのAstra Control操作に

含めるものをより細かく制御できます。

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

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

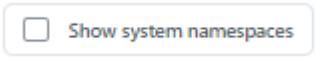
ネームスペースのリソースをアプリケーションとして定義することで、ネームスペース内のすべてのKubernetesリソースをAstra Control管理に追加できます。この方法は、アプリケーションを個別に定義するよりも望ましい方法です "特定のネームスペース内のすべてのリソースを管理および保護することを意図しています" 同様の方法で、共通の間隔で実行します。

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 々のネームスペースに複数のアプリケーションが定義されます。マルチネームスペース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]。

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

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[関連アプリケーション]列に表示されます

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

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

手順

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

spec.includedClusterScopedResources

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

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

spec.includedNamespaces

アプリケーション内のこれらのリソースに名前空間とリソースを含めます。

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

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

YAMLの例：

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

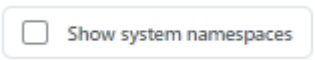
4. データを入力した後、 `astra_mysql_app.yaml` 正しい値を持つファイルを作成し、CRを適用します。

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

システムネームスペースについて教えてください。

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

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



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。

Snapshot とバックアップでアプリケーションを保護

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

の詳細を確認してください "Astra Controlによるデータ保護"。

アプリケーションデータの保護に関連する次のタスクを実行できます。

- [保護ポリシーを設定します]
- Snapshot を作成します
- [バックアップを作成します]
- ONTAP NAS経済性に優れた運用向けのバックアップとリストアを実現
- [変更不可のバックアップの作成]
- Snapshot とバックアップを表示します
- Snapshot を削除します
- [バックアップをキャンセルします]
- [バックアップを削除します]

保護ポリシーを設定します

保護ポリシーは、定義されたスケジュールでスナップショット、バックアップ、またはその両方を作成することでアプリケーションを保護します。Snapshot とバックアップを毎時、日次、週次、および月単位で作成し、保持するコピーの数を指定できます。保護ポリシーは、Astra Control Web UIまたはカスタムリソース（CR）ファイルを使用して定義できます。

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

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

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

Web UIを使用して保護ポリシーを設定する

手順

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. [* データ保護 *] を選択します。
3. ［保護ポリシーの設定］ を選択します。
4. 毎時、毎日、毎週、および毎月の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。

   [Snapshot とバックアップを毎時、毎日、毎週、または毎月作成する設定ポリシーの例のスクリーンショット。]

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

[テクニカルプレビュー**] CR**を使用した保護ポリシーの設定

手順

1. カスタムリソース（CR）ファイルを作成して名前を付けます。 `astra-control-schedule-cr.yaml`。Astra Control環境、クラスタ構成、データ保護のニーズに合わせて、かっこ<>の値を更新します。
   - <CR_NAME>：このカスタムリソースの名前。環境に適した一意の適切な名前を選択します。
   - <APPLICATION_NAME>：バックアップするアプリケーションのKubernetes名。
   - <APPVAULT_NAME>：バックアップコンテンツを格納するAppVaultの名前。
   - <BACKUPS_RETAINED>：保持するバックアップの数。ゼロは、バックアップを作成しないことを示します。
   - <SNAPSHOTS_RETAINED>：保持するSnapshotの数。ゼロは、スナップショットを作成しないことを示します。
   - <GRANULARITY>：スケジュールを実行する頻度。指定可能な値と必須の関連フィールドは次のとおりです。
     - `hourly` （次を指定する必要があります： `spec.minute`）
     - `daily` （次を指定する必要があります： `spec.minute` および `spec.hour`）
     - `weekly` （次を指定する必要があります： `spec.minute`、 `spec.hour`および `spec.dayOfWeek`）

▪ `monthly`（次を指定する必要があります： `spec.minute`、 `spec.hour`および `spec.dayOfMonth`）

- <DAY_OF_MONTH>：_（オプション）_スケジュールを実行する日にち（1 ~ 31）。このフィールドは、粒度が次の値に設定されている場合は必須です。 `monthly`。
- <DAY_OF_WEEK>：_（オプション）_スケジュールを実行する曜日（0 ~ 7）。0または7の値は日曜日を示します。このフィールドは、粒度が次の値に設定されている場合は必須です。 `weekly`。
- <HOUR_OF_DAY>：_（オプション）_スケジュールを実行する時間（0 ~ 23）。このフィールドは、粒度が次の値に設定されている場合は必須です。 `daily`、 `weekly`または `monthly`。
- <MINUTE_OF_HOUR>：_（オプション）_スケジュールを実行する分（0~59）。このフィールドは、粒度が次の値に設定されている場合は必須です。 `hourly`、 `daily`、 `weekly`または `monthly`。

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

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

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

結果

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

Snapshot を作成します

オンデマンド Snapshot はいつでも作成できます。

このタスクについて

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

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

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

Web UIを使用したSnapshotの作成

手順

1. 「 * アプリケーション * 」を選択します。
2. 目的のアプリケーションの * アクション * 列のオプションメニューから、 * スナップショット * を選択します。
3. スナップショットの名前をカスタマイズし、*次へ*を選択します。
4. [技術プレビュー]ストレージバケットのリストからスナップショットのデスティネーションバケットを選択します。
5. Snapshot の概要を確認し、「 * Snapshot * 」を選択します。

[テクニカルプレビュー] CRを使用したスナップショットの作成

手順

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

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

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

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

結果

スナップショットプロセスが開始されます。スナップショットは'ステータスが* Healthy である場合に成功し

ます（ 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。

Azure NetApp Files ストレージでホストされているアプリケーションをバックアップするときは、ストレージスペースがどのように処理されるかに注意してください。を参照してください "アプリケーションのバックアップ" を参照してください。

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

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

このタスクについて

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

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

Web UIを使用したバックアップの作成

手順

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

[テクニカルプレビュー] CRを使用したバックアップの作成

手順

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

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

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

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

結果

Astra Control ：アプリケーションのバックアップを作成

- ネットワークに障害が発生している場合や、処理速度が異常に遅い場合は、バックアップ処理がタイムアウトする可能性があります。その結果、バックアップは失敗します。
- 実行中のバックアップをキャンセルする必要がある場合は、の手順に従ってください [バックアップをキャンセルします]。バックアップを削除するには、完了するまで待ってから、の手順を実行します [バックアップを削除します]。
- データ保護処理（クローン、バックアップ、リストア）が完了して永続ボリュームのサイズを変更したあと、新しいボリュームのサイズが UI に表示されるまでに最大 20 分かかります。データ保護処理にかかる時間は数分です。また、ストレージバックエンドの管理ソフトウェアを使用してボリュームサイズの変更を確認できます。

ONTAP NAS経済性に優れた運用向けのバックアップとリストアを実現

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

作業を開始する前に

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

構成手順用に展開

1. ONTAPストレージバックエンドで次の手順を実行します。

   a. をホストしているSVMを検索します。 `ontap-nas-economy`-アプリケーションのボリュームベース。

   b. ボリュームを作成するONTAPに接続されている端末にログインします。

   c. SVMのSnapshotディレクトリを非表示にします。

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

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

   +

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

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

   a. ontap-nas-economyベースでアプリケーションに関連付けられている各PVのsnapshotディレクトリを有効にします。

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

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

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

   対応：

   ```
   snapshotDirectory: "true"
   ```

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

結果

Astra Controlを使用して、アプリケーションのバックアップとリストアを実行できるようになります。

各PVCは、他のアプリケーションでバックアップおよびリストアに使用することもできます。

変更不可のバックアップの作成

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

作業を開始する前に

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

- * Amazon Web Services *： "バケットの作成時にS3オブジェクトロックを有効にし、デフォルトの保持モードを「governance」にデフォルトの保持期間を設定する"。
- * Google Cloud *： "保持ポリシーを使用してバケットを設定し、保持期間を指定する"。
- * Microsoft Azure *： "コンテナレベルのスコープで時間ベースの保持ポリシーを使用してBLOBストレージバケットを構成する"。
- * NetApp StorageGRID *： "バケットの作成時にS3オブジェクトロックを有効にし、デフォルトの保持モードを「compliance」にデフォルトの保持期間を設定する"。

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

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

手順

1. 「 * アプリケーション * 」を選択します。
2. 目的のアプリケーションの*アクション*列のオプションメニューから、*バックアップ*を選択します。
3. バックアップ名をカスタマイズする。
4. 既存のスナップショットからアプリケーションをバックアップするかどうかを選択します。このオプションを選択すると、既存の Snapshot のリストから選択できます。
5. ストレージバケットのリストから、バックアップのデスティネーションバケットを選択します。Write Once Read Many（WORM）バケット名の横にステータスが「Locked」と表示されます。

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

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

結果

Astra Controlがアプリケーションの変更不可のバックアップを作成

- ネットワークに障害が発生している場合や、処理速度が異常に遅い場合は、バックアップ処理がタイムアウトする可能性があります。その結果、バックアップは失敗します。
- 同じアプリケーションの書き換え不能な2つのバックアップを同じバケットに同時に作成しようとすると、Astra Controlによって2つ目のバックアップが開始されなくなります。最初のバックアップが完了してから、別のバックアップを開始してください。
- 実行中の変更不可のバックアップはキャンセルできません。
- データ保護処理（クローン、バックアップ、リストア）が完了して永続ボリュームのサイズを変更したあと、新しいボリュームのサイズが UI に表示されるまでに最大 20 分かかります。データ保護処理にかかる時間は数分です。また、ストレージバックエンドの管理ソフトウェアを使用してボリュームサイズの変更を確認できます。

Snapshot とバックアップを表示します

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

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

手順

1. 「 * アプリケーション」を選択し、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。

   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参照するには、「* Backups」を選択します。

Snapshot を削除します

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

手順

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

結果

Astra Control がスナップショットを削除します。

バックアップをキャンセルします

実行中のバックアップをキャンセルすることができます。

バックアップをキャンセルするには、バックアップが実行されている必要があります `Running` 状態。にあるバックアップはキャンセルできません `Pending` 状態。

実行中の変更不可のバックアップはキャンセルできません。

手順

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

バックアップを削除します

不要になったスケジュール済みまたはオンデマンドのバックアップを削除します。

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。

保持期間が終了する前に変更不可のバックアップを削除することはできません。

手順

1. 「 * アプリケーション」を選択し、アプリケーションの名前を選択します。
2. [* データ保護 *] を選択します。
3. 「 * Backups * 」を選択します。
4. 目的のバックアップの [* アクション *] 列の [オプション] メニューから、 [* バックアップの削除 *] を選択します。
5. 削除を確認するために「 delete 」と入力し、「 * はい、バックアップを削除 * 」を選択します。

結果

Astra Control がバックアップを削除する。

[技術プレビュー]クラスタ全体を保護する

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

れ、Pythonスクリプトを使用してオーケストレーションされます。

動作の仕組み

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

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

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

例：完全なクラスタ保護

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

例：ラベルベースの保護

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

すべてのネームスペースのスケジュールされたバックアップを作成する

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

手順

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

特定のネームスペースのスケジュールされたバックアップを作成する

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

手順

1. クラスタにネットワークでアクセスできるマシンに、次のファイルをダウンロードします。
   - "コンポーネント.yaml CRDファイル"

- "protectCluster.py Pythonスクリプト"

2. ツールキットを設定してインストールするには、次の手順に従います。 "付属の手順に従います。"。

アプリケーションのリストア

Astra Control を使用すると、スナップショットまたはバックアップからアプリケーションをリストアできます。同じクラスタにアプリケーションをリストアする場合、既存の Snapshot からのリストアは高速です。Astra Control UI またはを使用できます "Astra Control API" 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。

作業を開始する前に

- 最初にアプリケーションを保護する:アプリケーションを復元する前に、アプリケーションのスナップショットまたはバックアップを作成することを強くお勧めします。これにより、リストアに失敗した場合に、スナップショットまたはバックアップからクローンを作成できます。
- デスティネーションボリュームの確認：別のストレージクラスにリストアする場合は、ストレージクラスで同じ永続ボリュームアクセスモード（ReadWriteManyなど）が使用されていることを確認してください。デスティネーションの永続ボリュームアクセスモードが異なると、リストア処理は失敗します。たとえば、ソースの永続ボリュームがRWXアクセスモードを使用している場合は、Azure Managed Disks 、AWS EBS、Google Persistent Disk、など、RWXを提供できないデスティネーションストレージクラスを選択します `ontap-san`を指定すると、リストア処理は失敗します。原因は失敗します。永続ボリュームのアクセスモードの詳細については、を参照してください "Kubernetes" ドキュメント
- 必要なスペースを確保するための計画：NetApp ONTAP ストレージを使用するアプリケーションのインプレースリストアを実行すると、リストアしたアプリケーションで使用されるスペースが2倍になることがあります。In Placeリストアを実行したあとに、リストアしたアプリケーションから不要なSnapshotを削除して、ストレージスペースを解放します。
- サポートされるストレージクラスドライバ：Astra Controlでは、次のドライバに基づくストレージクラスを使用したバックアップのリストアがサポートされます。
  - `ontap-nas`
  - `ontap-nas-economy`
  - `ontap-san`
  - `ontap-san-economy`
- （**ontap-nas-economy**ドライバのみ）バックアップとリストア： `ontap-nas-economy` ドライバを使用して、 "ONTAPストレージバックエンドのSnapshotディレクトリが非表示になっている"。このディレクトリを非表示にしないと、アプリケーション（特にNFSv3を使用している場合）へのアクセスが失われる可能性があります。

> 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。

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。
     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。

     以前に削除したネームスペースにリストアすると、同じ名前の新しいネームスペースがリストアプロセスで作成されます。以前に削除したネームスペースでアプリケーションを管理する権限を持つユーザは、新しく作成したネームスペースに手動で権限を復元する必要があります。

   - 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。この手順を使用して、アプリケーションを別のストレージクラスに移行することもできます。
     i. 復元されたアプリの名前を指定します。
     ii. リストアするアプリケーションのデスティネーションクラスタを選択します。
     iii. アプリケーションに関連付けられている各ソースネームスペースのデスティネーションネームスペースを入力します。

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

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

     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。

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

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

CRを使用したバックアップからのリストア

手順

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

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

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

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

<stdin>の未解決ディレクティブ-include::../_include/selective-restore-cr.adoc[]

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

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

CRを使用したバックアップから元のネームスペースへのリストア

手順

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

- <BACKUP_PATH>：バックアップコンテンツが格納されているAppVault内のパス。例：

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

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

<stdin>の未解決ディレクティブ-include::../_include/selective-restore-cr.adoc[]

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

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

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

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

CRを使用した**Snapshot**からのリストア

手順

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

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

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

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

<stdin>の未解決ディレクティブ-include::../_include/selective-restore-cr.adoc[]

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

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

CRを使用した**Snapshot**から元のネームスペースへのリストア

手順

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

- <BACKUP_PATH>：バックアップコンテンツが格納されているAppVault内のパス。例：

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

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

<stdin>の未解決ディレクティブ-include::../_include/selective-restore-cr.adoc[]

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

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

結果

Astra Control は、指定した情報に基づいてアプリケーションを復元します。アプリケーションをインプレースでリストアした場合、既存の永続ボリュームのコンテンツが、リストアしたアプリケーションの永続ボリュームのコンテンツに置き換えられます。

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

ネームスペースの名前/ IDまたはネームスペースのラベルでネームスペースの制約を受けているメンバーユーザは、同じクラスタの新しいネームスペース、または組織のアカウントに含まれる他のクラスタにアプリケーションをクローニングまたはリストアできます。ただし、同じユーザが、クローニングまたはリストアされたアプリケーションに新しいネームスペースからアクセスすることはできません。クローン処理またはリストア処理で新しいネームスペースが作成されたあと、アカウントの管理者/所有者はメンバーユーザアカウントを編集し、影響を受けるユーザのロールの制約を更新して、新しいネームスペースへのアクセスを許可できます。

アプリケーションのリストア中にリソースをフィルタリングします

にフィルタルールを追加できます "リストア" リストアされたアプリケーションに含める、またはリストアされたアプリケーションから除外する既存のアプリケーションリソースを指定する処理。指定した名前空間、ラ

ベル、またはGVK（GroupVersionKind）に基づいて、リソースを含めたり除外したりできます。

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。

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 ネームスペース間でワークロードを移動できます。

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。

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、ソースクラスタを含むクラウドインスタンスのデフォルトバケットが割り当てられていることを確認する必要があります。ソースクラウドインスタンスにデフォルトのバケットセットがないと、クラスタ間のクローニング処理は失敗します。
- 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。

クローンの制限事項

- 明示的なストレージクラス：ストレージクラスを明示的に設定したアプリケーションを導入し、そのアプリケーションのクローンを作成する必要がある場合、ターゲットクラスタには元 々 指定されたストレージクラスが必要です。ストレージクラスを明示的に設定したアプリケーションを、同じストレージクラス

を含まないクラスタにクローニングすると、失敗します。

- * ontap-nas-economy-basedアプリケーション*：アプリケーションのストレージクラスが `ontap-nas-economy` ドライバ。ただし、 "ONTAP NAS経済性に優れた運用向けのバックアップとリストアを実現"。
- クローンとユーザーの制約：名前空間の名前/ IDまたは名前空間のラベルによって名前空間の制約を持つメンバーユーザーは、同じクラスタ上の新しい名前空間、または組織のアカウント内の他の任意のクラスタに対して、アプリケーションのクローンまたはリストアを実行できます。ただし、同じユーザが、クローニングまたはリストアされたアプリケーションに新しいネームスペースからアクセスすることはできません。クローン処理またはリストア処理で新しいネームスペースが作成されたあと、アカウントの管理者/所有者はメンバーユーザアカウントを編集し、影響を受けるユーザのロールの制約を更新して、新しいネームスペースへのアクセスを許可できます。
- 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やアプリケーションのリストア中に、使用するバケットを指定できます。クラスタ間でクローニングする場合、デフォルトバケットを指定する必要がありますが、同じクラスタ内でクローニングする場合、バケットの指定は任意です。
  - クラスタ間でクローニングする場合、クローン処理のソースクラスタを含むクラウドインスタンスには、デフォルトのバケットセットが必要です。
  - クローンのバケットを変更するオプションはありません。どのバケットを使用するかを制御する必要がある場合は、どちらかを選択できます "バケットのデフォルト設定を変更する" または、を実行します "バックアップ" その後にを押します "リストア" 個別。
- * 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。

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ネーム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ている各ソースネームスペースは、デスティネーションネーム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ネームスペースが作成されます。指定するデスティネーションネームスペースがデスティネーションクラスタに存在していないことを確認してください。

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

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

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

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

結果

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

クローン処理またはリストア処理で新しいネームスペースが作成されたあと、アカウントの管理者/所有者はメンバーユーザアカウントを編集し、影響を受けるユーザのロールの制約を更新して、新しいネームスペースへのアクセスを許可できます。

アプリケーション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 Serviceでは、実行可能なタイミングに基づいて、次のタイプの実行フックがサポートされます。

- 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に、実行フックにフィルタを追加して、フックが一致するコンテナを管理できます。フィルタは、すべてのコンテナで同じコンテナイメージを使用し、各イメージを別の目的（Elasticsearchなど）に使用するアプリケーションに便利です。フィルタを使用すると、一部の同一コンテナで実行フックが実行されるシナリオを作成できます。1つの実行フックに対して複数のフィルタ

を作成すると、それらは論理AND演算子と結合されます。実行フックごとに最大10個のアクティブフィルタを使用できます。

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

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

カスタム実行フックに関する重要な注意事項

アプリケーションの実行フックを計画するときは、次の点を考慮してください。

> 実行フックは、実行中のアプリケーションの機能を低下させたり、完全に無効にしたりすることが多いため、カスタム実行フックの実行時間を最小限に抑えるようにしてください。
> 実行フックが関連付けられている状態でバックアップまたはスナップショット操作を開始した後'キャンセルした場合でも'バックアップまたはスナップショット操作がすでに開始されていればフックは実行できますつまり、バックアップ後の実行フックで使用されるロジックは、バックアップが完了したとは見なされません。

- 新しいAstra Control環境では、実行フック機能はデフォルトで無効になっています。
  - 実行フックを使用する前に、実行フック機能を有効にする必要があります。
  - 所有者ユーザまたは管理者ユーザは、現在のAstra Controlアカウントで定義されているすべてのユーザの実行フック機能を有効または無効にできます。を参照してください [実行フック機能を有効にする] および [実行フック機能を無効にする] 手順については、を参照し
  - 機能の有効化ステータスは、Astra Controlのアップグレード中も維持されます。
- 実行フックは、スクリプトを使用してアクションを実行する必要があります。多くの実行フックは、同じスクリプトを参照できます。
- Astra Controlでは、実行フックが実行可能なシェルスクリプトの形式で記述されるようにするスクリプトが必要です。
- スクリプトのサイズは96KBに制限されています。
- Astra Controlは、実行フックの設定と一致条件を使用して、スナップショット、バックアップ、または復元操作に適用できるフックを決定します。
- 実行フックの障害はすべて'ソフトな障害ですフックが失敗しても'他のフックとデータ保護操作は試行されますただし、フックが失敗すると、 * アクティビティ * ページイベントログに警告イベントが記録されます。
- 実行フックを作成、編集、または削除するには、 Owner 、 Admin 、または Member 権限を持つユーザーである必要があります。
- 実行フックの実行に 25 分以上かかる場合 ' フックは失敗し ' 戻りコードが N/A のイベント・ログ・エントリが作成されます該当する Snapshot はタイムアウトして失敗とマークされ、タイムアウトを通知するイベントログエントリが生成されます。
- 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、*アクティビティ*ページイベン

トログに保存されます。ただし、スケジュールされたデータ保護処理については、フック障害イベントだけがイベントログに記録されます（スケジュールされたデータ保護処理自体によって生成されたイベントは記録されたままです）。

実行順序

データ保護操作を実行すると、実行フックイベントが次の順序で実行されます。

1. 適用可能なカスタムプリオペレーション実行フックは、適切なコンテナで実行されます。カスタムのプリオペレーションフックは必要なだけ作成して実行できますが、操作前のこれらのフックの実行順序は保証も構成もされていません。
2. データ保護処理が実行されます。
3. 適用可能なカスタムポストオペレーション実行フックは、適切なコンテナで実行されます。必要な数のカスタムポストオペレーションフックを作成して実行できますが、操作後のこれらのフックの実行順序は保証されず、設定もできません。

同じ種類の実行フック（スナップショット前など）を複数作成する場合、これらのフックの実行順序は保証されません。ただし、異なるタイプのフックの実行順序は保証されています。たとえば、すべての異なるタイプのフックを持つ構成の実行順序は次のようになります。

1. 予備フックが実行されます
2. スナップショット前フックが実行されます
3. スナップショット後フックが実行されます
4. バックアップ後のフックが実行されます
5. 復元後のフックが実行されます

シナリオ番号2のこの設定の例は、の表を参照してください [フックが実行されるかどうかを確認します]。

本番環境で実行スクリプトを有効にする前に、必ず実行フックスクリプトをテストしてください。'kubectl exec' コマンドを使用すると、スクリプトを簡単にテストできます。本番環境で実行フックを有効にしたら、作成されたSnapshotとバックアップをテストして整合性があることを確認します。これを行うには、アプリケーションを一時的なネームスペースにクローニングし、スナップショットまたはバックアップをリストアしてから、アプリケーションをテストします。

フックが実行されるかどうかを確認します

次の表を使用して、アプリケーションでカスタム実行フックが実行されるかどうかを判断します。

アプリケーションの高レベルの処理は、すべてスナップショット、バックアップ、またはリストアの基本的な処理のいずれかを実行することで構成されることに注意してください。シナリオによっては、クローニング処理はこれらの処理のさまざまな組み合わせで構成されるため、クローン処理を実行する実行フックはさまざまです。

In Placeリストア処理では既存のSnapshotまたはバックアップが必要になるため、これらの処理ではSnapshotまたはバックアップフックは実行されません。

開始してスナップショットを含むバックアップをキャンセルし'実行フックが関連付けられている場合は'一部のフックが実行され'ほかのフックが実行されないことがありますつまり、バックアップ後の実行フックでは、バックアップが完了したとは判断できません。キャンセルしたバックアップに関連する実行フックがある場合は、次の点に注意してください。

- バックアップ前およびバックアップ後のフックは常に実行されます。
- バックアップに新しいスナップショットが含まれており'スナップショットが開始されている場合は'スナップショット前フックとスナップショット後フックが実行されます
- スナップショットの開始前にバックアップがキャンセルされた場合は'スナップショット前フックとスナップショット後フックは実行されません

シナリオ（**Scenario**）	操作	既存の**Snapshot**	既存のバックアップ	ネームスペース	クラスタ	スナップショットフックが実行されます	バックアップフックが実行されます	フックを元に戻します
1.	クローン	N	N	新規	同じ	Y	N	Y
2.	クローン	N	N	新規	違う	Y	Y	Y
3.	クローンまたはリストア	Y	N	新規	同じ	N	N	Y
4.	クローンまたはリストア	N	Y	新規	同じ	N	N	Y
5.	クローンまたはリストア	Y	N	新規	違う	N	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。

実行フック機能を有効にする

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

手順

1. 「 * アプリケーション」に移動し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。
3. *実行フックを有効にする*を選択します。

   アカウント>*機能設定*タブが表示されます。

4. Execution Hooks*ペインで、設定メニューを選択します。
5. [有効] を選択します。
6. 表示されるセキュリティ警告を確認します。
7. [はい、実行フックを有効にする]*を選択します。

実行フック機能を無効にする

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

手順

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

既存の実行フックを表示します

アプリケーションの既存のカスタム実行フックを表示できます。

手順

1. 「 * アプリケーション」に移動し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。

   有効または無効になっているすべての実行フックを結果リストに表示できます。フックのステータス、一致するコンテナの数、作成時間、および実行時間（プリ/ポストオペレーション）を確認できます。を選択

できます + アイコンをクリックして、実行するコンテナのリストを展開します。このアプリケーションの実行フックに関連するイベントログを表示するには、*アクティビティ*タブに移動します。

既存のスクリプトを表示します

アップロードされた既存のスクリプトを表示できます。このページでは、使用中のスクリプトと、使用中のフックを確認することもできます。

手順

1. 「アカウント」に移動します。
2. [スクリプト]タブを選択します。

   このページには、アップロードされた既存のスクリプトのリストが表示されます。[使用者*]列には、各スクリプトを使用している実行フックが表示されます。

スクリプトを追加します

各実行フックは、スクリプトを使用してアクションを実行する必要があります。実行フックが参照できるスクリプトを1つ以上追加できます。多くの実行フックは同じスクリプトを参照できます。これにより、1つのスクリプトを変更するだけで多くの実行フックを更新できます。

手順

1. 実行フック機能が 有効。
2. 「アカウント」に移動します。
3. [スクリプト]タブを選択します。
4. 「 * 追加」を選択します。
5. 次のいずれかを実行します。
   - カスタムスクリプトをアップロードする。
     i. [ファイルのアップロード（ Upload file ）] オプションを選択します。
     ii. ファイルを参照してアップロードします。
     iii. スクリプトに一意の名前を付けます。
     iv. （オプション）他の管理者がスクリプトについて知っておく必要があるメモを入力します。
     v. 「スクリプトを保存」を選択します。
   - クリップボードからカスタムスクリプトを貼り付けます。
     i. [貼り付け（Paste）]または[タイプ（* type）]オプションを選択する
     ii. テキストフィールドを選択し、スクリプトテキストをフィールドに貼り付けます。
     iii. スクリプトに一意の名前を付けます。
     iv. （オプション）他の管理者がスクリプトについて知っておく必要があるメモを入力します。
6. 「スクリプトを保存」を選択します。

結果

新しいスクリプトが、[スクリプト]タブのリストに表示されます。

スクリプトを削除します

不要になって実行フックで使用されなくなったスクリプトは、システムから削除できます。

手順

1. 「アカウント」に移動します。
2. [スクリプト]タブを選択します。
3. 削除するスクリプトを選択し、「アクション」列のメニューを選択します。
4. 「 * 削除」を選択します。

> スクリプトが1つまたは複数の実行フックに関連付けられている場合、*Delete*アクションは使用できません。スクリプトを削除するには、まず関連する実行フックを編集し、別のスクリプトに関連付けます。

カスタム実行フックを作成します

アプリケーションのカスタム実行フックを作成してAstra Controlに追加できます。を参照してください [実行フックの例] フックの例を参照してください。実行フックを作成するには、 Owner 、 Admin 、または Member のいずれかの権限が必要です。

> 実行フックとして使用するカスタムシェルスクリプトを作成する場合は、特定のコマンドを実行するか、実行可能ファイルへの完全パスを指定する場合を除き、ファイルの先頭に適切なシェルを指定するようにしてください。

手順

1. 実行フック機能が 有効。
2. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
3. [実行フック *] タブを選択します。
4. 「 * 追加」を選択します。
5. [フックの詳細*（Hook Details *）]領域で、次の
   a. *操作*ドロップダウンメニューから操作タイプを選択して、フックをいつ実行するかを決定します。
   b. フックの一意の名前を入力します。
   c. （オプション）実行中にフックに渡す引数を入力し、各引数を入力した後で Enter キーを押して、それぞれを記録します。
6. （オプション）フックフィルタの詳細（* Hook Filter Details *）領域で、実行フックが実行されるコンテナを制御するフィルタを追加できます。
   a. [フィルタの追加]を選択します。
   b. [フックフィルタータイプ*]列で、フィルターを適用する属性をドロップダウンメニューから選択します。
   c. **[Regex]**列に、フィルタとして使用する正規表現を入力します。Astra Controlでは、を使用します "正規表現2（RE2）正規表現の正規表現構文"。

正規表現フィールドに他のテキストが含まれていない属性（ポッド名など）の正確な名前でフィルタリングすると、サブストリングの一致が実行されます。正確な名前とその名前だけを照合するには、完全に一致する文字列の一致構文を使用します（例：`^exact_podname$`）。

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。

7. 完了したら、「次へ」を選択します。
8. [* スクリプト * （ * Script * ）] 領域で、次のいずれかを実行します。
   - 新しいスクリプトを追加します。
     i. 「 * 追加」を選択します。
     ii. 次のいずれかを実行します。
     - カスタムスクリプトをアップロードする。
       I. [ファイルのアップロード（ Upload file ）] オプションを選択します。
       II. ファイルを参照してアップロードします。
       III. スクリプトに一意の名前を付けます。
       IV. （オプション）他の管理者がスクリプトについて知っておく必要があるメモを入力します。
       V. 「スクリプトを保存」を選択します。
     - クリップボードからカスタムスクリプトを貼り付けます。
       I. [貼り付け（Paste）]または[タイプ（* type）]オプションを選択する
       II. テキストフィールドを選択し、スクリプトテキストをフィールドに貼り付けます。
       III. スクリプトに一意の名前を付けます。
       IV. （オプション）他の管理者がスクリプトについて知っておく必要があるメモを入力します。
   - リストから既存のスクリプトを選択します。

   このスクリプトを使用するように実行フックに指示します。
9. 「 * 次へ * 」を選択します。
10. 実行フックの設定を確認します。
11. 「 * 追加」を選択します。

実行フックの状態を確認します

スナップショット、バックアップ、または復元操作の実行が終了したら、操作の一部として実行された実行フックの状態を確認できます。このステータス情報を使用して、実行フックを保持するか、変更するか、削除するかを決定できます。

手順

1. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
2. [データ保護]タブを選択します。
3. 実行中のSnapshotを表示するには「* Snapshots」を選択し、実行中のバックアップを表示するには「* Backups」を選択します。

   フック状態*は、操作完了後の実行フックランのステータスを示します。状態にカーソルを合わせると、詳細を確認できます。たとえば、スナップショット中に実行フック障害が発生した場合、そのスナップショットのフック状態にカーソルを合わせると、失敗した実行フックのリストが表示されます。各失敗の理由を確認するには、左側のナビゲーション領域の*アクティビティ*ページを確認します。

スクリプトの使用状況を表示します

どの実行フックがAstra Control Web UIの特定のスクリプトを使用しているかを確認できます。

手順

1. 「 * アカウント * 」を選択します。
2. [スクリプト]タブを選択します。

   スクリプトのリストにある* Used by *列には、リスト内の各スクリプトを使用しているフックの詳細が表示されます。

3. 目的のスクリプトの[使用者*]列の情報を選択します。

   より詳細なリストが表示され、スクリプトを使用しているフックの名前と、それらが実行されるように構成されている操作のタイプが示されます。

実行フックを編集します

実行フックを編集して、その属性、フィルタ、または使用するスクリプトを変更できます。実行フックを編集するには、Owner、Admin、またはMemberのいずれかの権限が必要です。

手順

1. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。
3. 編集するフックの*アクション*列のオプションメニューを選択します。
4. 「 * 編集 * 」を選択します。
5. 各セクションを完了したら、「次へ」を選択して、必要な変更を行います。
6. [保存（ Save ）] を選択します。

実行フックを無効にします

アプリケーションのスナップショットの前または後に実行を一時的に禁止する場合は、実行フックを無効にできます。実行フックを無効にするには、 Owner 、 Admin 、または Member のいずれかの権限が必要です。

手順

1. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。
3. 無効にするフックの * アクション * 列のオプションメニューを選択します。
4. [**Disable**] を選択します。

実行フックを削除します

不要になった実行フックは完全に削除できます。実行フックを削除するには、 Owner 、 Admin 、または Member のいずれかの権限が必要です。

手順

1. 「 * アプリケーション」を選択し、管理アプリの名前を選択します。
2. [実行フック *] タブを選択します。
3. 削除するフックの * アクション * 列のオプションメニューを選択します。
4. 「 * 削除」を選択します。
5. 表示されたダイアログで、「delete」と入力して確定します。
6. [はい]を選択し、実行フックを削除します。*

を参照してください。

- "NetApp Verda GitHubプロジェクト"

著作権に関する情報

商標に関する情報