



管理ワークフロー

Astra Automation

NetApp
August 11, 2025

This PDF was generated from https://docs.netapp.com/ja-jp/astra-automation/workflows/workflows_before.html on August 11, 2025. Always check docs.netapp.com for the latest.

目次

管理ワークフロー	1
管理ワークフローを使用するための準備	1
一般的な準備	1
ワークフローのカテゴリ	1
その他の考慮事項については	1
アプリコントロール	2
アプリを表示します	2
アプリ入手する	3
アプリを管理します	4
アプリの管理を解除します	5
アプリケーションの保護	6
Snapshot を一覧表示します	6
バックアップをリスト表示します	8
アプリケーションのスナップショットを作成します	10
アプリケーションのバックアップを作成する	11
Snapshot を削除します	12
バックアップを削除します	13
データ保護ポリシーを作成する	15
アプリケーションのクローニングとリストア	16
アプリケーションのクローンを作成します	16
スナップショットからアプリケーションをクローニングします	17
バックアップからアプリケーションをクローニングする	19
バックアップからアプリケーションをリストアする	20
ネームスペース	21
ネームスペースを表示します	21
サポート	23
通知を一覧表示します	23
失敗したアプリを削除します	25

管理ワークフロー

管理ワークフローを使用するための準備

これらのワークフローは、Astra管理対象クラスタでアプリケーションを管理する際に使用できます。



これらのワークフローは、ネットアップがいつでも拡張および強化できるため、定期的に確認する必要があります。

一般的な準備

いずれかの Astra ワークフローを使用する前に、を確認してください "ワークフローを使用する準備をします"。

ワークフローのカテゴリ

管理ワークフローは、目的のワークフローを見つけやすくするために、さまざまなカテゴリに分類されています。

カテゴリ	説明
アプリケーション制御	これらのワークフローでは、管理対象アプリケーションと管理対象外アプリケーションを制御できます。アプリをリスト表示したり、管理アプリを作成および削除したりできます。
アプリケーションの保護	これらのワークフローを使用して、Snapshot とバックアップを通じて管理対象アプリケーションを保護することができます。
アプリケーションのクローニングとリストア	ここでは、管理対象アプリケーションのクローニングとリストアのワークフローについて説明します。
サポート	アプリケーションと一般的な Kubernetes 環境をデバッグおよびサポートするためのワークフローがいくつかあります。

その他の考慮事項については

管理ワークフローを使用する場合は、さらにいくつかの点を考慮する必要があります。

アプリケーションのクローンを作成します

アプリケーションのクローニングでは、いくつかの点を考慮する必要があります。以下に、JSON 入力のパラメータを示します。

ソースクラスタの識別子

「sourceClusterID」の値は、元のアプリケーションがインストールされているクラスタを常に識別します。

クラスタ ID

「ClusterId」の値は、新しいアプリケーションをインストールするクラスタを識別します。

- 同じクラスタ内でクローニングする場合、「ClusterId」と「sourceClusterID」には同じ値があります。
- クラスタ間でクローンを作成する場合、2つの値は異なる値であり 'ClusterId' はターゲット・クラスタの ID である必要があります

ネームスペース

「namespace」の値は、元のソースアプリケーションとは異なる値にする必要があります。さらに、クローンのネームスペースを作成できず、Astra がそのクローンを作成します。

バックアップとスナップショット

オプションで 'backupId' または 'snapshotID' パラメータを使用して '既存のバックアップまたはスナップショットからアプリケーションのクローンを作成できます' バックアップまたはスナップショットを指定しない場合は、まずアプリケーションのバックアップを作成し、次にバックアップからクローンを作成します。

アプリケーションの復元

アプリケーションをリストアする際に考慮すべき事項をいくつか示します。

- アプリケーションのリストアは、クローニング処理とほぼ同じです。
- アプリケーションをリストアするときは、バックアップまたはスナップショットを指定する必要があります。

アプリコントロール

アプリを表示します

Astra が現在管理しているアプリケーションをリストできます。特定のアプリケーションの Snapshot やバックアップを検索する際に、この操作を行うことができます。

次の REST API 呼び出しを実行します。

HTTP メソッドとエンドポイント

この REST API 呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/account/{account_id}/k8s/v2/apps

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
含める	クエリ	いいえ	必要に応じて、応答で返す値を選択します。

curl の例：すべてのアプリケーションのすべてのデータを返します

```
curl --request GET \
--location "https://astra.netapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

curl の例：すべてのアプリケーションの名前、**ID**、および状態を返します

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps?include=name,id,
state" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

JSON 出力例

```
{
  "items": [
    [
      "mysql",
      "4ee2b8fa-3696-4f32-8879-399792f477c3",
      "ready"
    ],
    [
      "postgresql",
      "3b984474-e5c9-4b64-97ee-cdeb9bcd212e",
      "ready"
    ],
    ...
  ],
  "metadata": {}
}
```

アプリ入手する

単一のアプリケーションを記述するすべてのリソース変数を取得できます。

作業を開始する前に

取得するアプリケーションのIDが必要です。必要に応じて、ワークフローを使用できます "[アプリを表示します](#)" アプリケーションを検索します。

ステップ1：アプリケーションを取得する

次の REST API呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /k8s/v2/apps/ {app_id}

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
アプリケーションID	パス	はい。	取得するアプリケーションのID値。

curl の例：アプリケーションのすべてのデータを返します

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps/$APP_ID" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

アプリを管理します

特定のネームスペースすでにAstraが認識しているアプリケーションに基づいて、管理アプリケーションを作成できます。アプリケーションをAstraで管理または定義した場合、バックアップとスナップショットを作成してアプリケーションを保護できます。

手順1：ネームスペースを選択する

ワークフローを実行 "[ネームスペースを表示します](#)" ネームスペースを選択します。

手順2：クラスタを選択する

ワークフローを実行 "[クラスタをリストします](#)" をクリックし、クラスタを選択します。

ステップ3：アプリケーションを管理する

アプリケーションを管理するために次のREST API呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿 (Post)	/accounts / {account_id} /k8s/v2/apps

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
JSON	ボディ (Body)	はい。	管理対象アプリケーションを識別するために必要なパラメータを提供します。以下の例を参照してください。

curl の例：アプリケーションの管理

```
curl --request POST \
--location "https://astranetapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps" \
--include \
--header "Content-Type: application/astra-app+json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

JSON の入力例

```
{
  "clusterID": "7ce83fba-6aa1-4e0c-a194-26e714f5eb46",
  "name": "subtext",
  "namespaceScopedResources": [{"namespace": "kube-matrix"}],
  "type": "application/astra-app",
  "version": "2.0"
}
```

アプリの管理を解除します

管理対象アプリが不要になった場合は削除できます。管理アプリケーションを削除すると、関連付けられているスケジュールも削除されます。

作業を開始する前に

管理を解除するアプリのIDが必要です。必要に応じて、ワークフローを使用できます "[アプリを表示します](#)" アプリケーションを検索します。

アプリケーションのバックアップと Snapshot は、削除されても自動的には削除されません。バックアップと Snapshot が不要になった場合は、アプリケーションを削除する前に削除する必要があります。

アプリの管理を解除する

次のREST API呼び出しを実行してアプリケーションを削除します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
削除	/accounts / {account_id} /k8s/v2/apps/ {app_id}

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
アプリケーションID	パス	はい。	削除するアプリケーションを指定します。

curl の例：管理対象アプリケーションを削除します

```
curl --request DELETE \
--location
"https://astral.netapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps/$APP_ID" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

アプリケーションの保護

Snapshot を一覧表示します

特定のアプリケーションに対して作成されたスナップショットを一覧表示できます。

作業を開始する前に

スナップショットを表示するアプリケーションのIDが必要です。必要に応じて、ワークフローを使用できます "アプリを表示します" アプリケーションを検索します。

Snapshot を一覧表示します

次のREST API呼び出しを実行してSnapshotを表示します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /k8s/v1/apps/ {app_id} /appSnaps

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
アプリケーションID	パス	はい。	リストされたスナップショットを所有するアプリケーションを識別します。
カウント	クエリ	いいえ	'count=true' の場合 'スナップショットの数は応答のメタデータセクションに含まれます

curl の例：アプリケーションのすべてのスナップショットを返します

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/appSnaps"
" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

curl の例：アプリケーションのすべてのスナップショットとカウントを返します

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/appSnaps
?count=true" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

JSON 出力例

```
{  
  "items": [  
    {  
      "type": "application/astra-appSnap",  
      "version": "1.1",  
      "id": "1ce34da4-bb0a-4926-b925-4a5d85dda8c2",  
      "hookState": "success",  
      "metadata": {  
        "createdBy": "a530e865-23e8-4e2e-8020-e92c419a3867",  
        "creationTimestamp": "2022-10-30T22:44:20Z",  
        "modificationTimestamp": "2022-10-30T22:44:20Z",  
        "labels": []  
      },  
      "snapshotAppAsset": "0ebfe3f8-40ed-4bdc-88c4-2144fbda85a0",  
      "snapshotCreationTimestamp": "2022-10-30T22:44:33Z",  
      "name": "snapshot-david-1",  
      "state": "completed",  
      "stateUnready": []  
    }  
  ],  
  "metadata": {}  
}
```

バックアップをリスト表示します

特定のアプリケーション用に作成されたバックアップをリストできます。

作業を開始する前に

バックアップを表示するアプリケーションのIDが必要です。必要に応じて、ワークフローを使用できます "[アプリを表示します](#)" アプリケーションを検索します。

バックアップをリスト表示します

次の REST API 呼び出しを実行します。

HTTP メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /k8s/v1/apps/ {app_id} /appBackups

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使

用されます。

パラメータ	を入力します	必須	説明
アプリケーションID	パス	はい。	リストされたバックアップを所有する管理アプリケーションを指定します。

curl の例：アプリケーションのすべてのバックアップを返します

```
curl --request POST \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/appBackups" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

JSON 出力例

```
{  
  "items": [  
    {  
      "type": "application/astra-appBackup",  
      "version": "1.1",  
      "id": "8edeb4a4-fd8b-4222-a559-1013145b28fc",  
      "name": "backup-david-oct28-1",  
      "bucketID": "a443e58f-59bd-4d45-835a-1bc7813f659a",  
      "snapshotID": "dfe237cb-57b7-4576-af4d-00ba3a8f2828",  
      "state": "completed",  
      "stateUnready": [],  
      "hookState": "success",  
      "totalBytes": 205219132,  
      "bytesDone": 205219132,  
      "percentDone": 100,  
      "metadata": {  
        "labels": [  
          {  
            "name": "astra.netapp.io/labels/read-  
only/triggerType",  
            "value": "backup"  
          }  
        ],  
        "creationTimestamp": "2022-10-28T21:58:37Z",  
        "modificationTimestamp": "2022-10-28T21:58:55Z",  
        "createdBy": "a530e865-23e8-4e2e-8020-e92c419a3867"  
      }  
    }  
  ],  
  "metadata": {}  
}
```

アプリケーションのスナップショットを作成します

特定のアプリケーションのスナップショットを作成できます。

作業を開始する前に

スナップショットを作成するアプリケーションのIDが必要です。必要に応じて、ワークフローを使用できます
["アプリを表示します"](#) アプリケーションを検索します。

Snapshot を作成します

次の REST API 呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（Post）	/accounts / {account_id} /k8s/v1/apps/ {app_id} /appSnaps

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
アプリケーションID	パス	はい。	Snapshot を作成する管理対象アプリケーションを指定します。
JSON	ボディ（Body）	はい。	Snapshot のパラメータを提供します。以下の例を参照してください。

curl の例：アプリケーションのスナップショットを作成します

```
curl --request POST \
--location
"https://astral.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/appSnaps"
" \
--include \
--header "Content-Type: application/astra-appSnap+json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

JSON の入力例

```
{
  "type": "application/astra-appSnap",
  "version": "1.1",
  "name": "snapshot-david-1"
}
```

アプリケーションのバックアップを作成する

特定のアプリケーションのバックアップを作成し、そのバックアップを使用してアプリケーションのリストアやクローニングを行うことができます。

作業を開始する前に

バックアップするアプリケーションのIDが必要です。必要に応じて、ワークフローを使用できます "アプリを

[表示します](#) アプリケーションを検索します。

バックアップを作成します

次の REST API 呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（Post）	/accounts / {account_id} /k8s/v1/apps/ {app_id} /appBackups

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
アプリケーションID	パス	はい。	バックアップを作成するアプリケーションを指定します。
JSON	ボディ（Body）	はい。	バックアップのパラメータが表示されます。以下の例を参照してください。

curl の例：アプリケーションのバックアップを作成します

```
curl --request POST \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/appBackups" \
--include \
--header "Content-Type: application/astra-appBackup+json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

JSON の入力例

```
{
  "type": "application/astra-appBackup",
  "version": "1.1",
  "name": "backup-david-1"
}
```

Snapshot を削除します

アプリケーションに関連付けられているSnapshotを削除できます。

作業を開始する前に

次の情報が必要です。

- スナップショットを所有するアプリケーションのID。必要に応じて、ワークフローを使用できます "アブリを表示します" アプリケーションを検索します。
- 削除する Snapshot の ID。必要に応じて、ワークフローを使用できます "Snapshot を一覧表示します" をクリックしてください。

Snapshotを削除します。

次の REST API 呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
削除	/accounts / {account_id} /k8s/v1/apps/ {app_id} /appSnaps/ {appSnap_id}

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
アプリケーションID	パス	はい。	スナップショットを所有する管理アプリケーションを指定します。
Snapshot ID	パス	はい。	削除する Snapshot を指定します。

curl の例：アプリケーションのスナップショットを 1 つ削除します

```
curl --request DELETE \
--location
"https://astral.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/appSnaps
/<SNAPSHOT_ID>" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

バックアップを削除します

アプリケーションに関連付けられているバックアップを削除することができます。

作業を開始する前に

次の情報が必要です。

- ・バックアップを所有するアプリケーションのID。必要に応じて、ワークフローを使用できます "アプリを表示します" アプリケーションを検索します。
- ・削除するバックアップの ID。必要に応じて、ワークフローを使用できます "バックアップをリスト表示します" をクリックしてください。

バックアップの削除

次の REST API 呼び出しを実行します。



障害が発生したバックアップは、次に示すオプションの要求ヘッダーを使用して強制的に削除できます。

HTTP メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
削除	/accounts / {account_id} /k8s/v1/apps/ {app_id} /appBackups / {appBackup_id}

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
アプリケーションID	パス	はい。	バックアップを所有する管理アプリケーションを指定します。
バックアップ ID	パス	はい。	削除するバックアップを指定します。
強制的に削除します	ヘッダー	いいえ	失敗したバックアップを強制的に削除する場合に使用します。

curl の例：アプリケーションのバックアップを 1 つ削除します

```
curl --request DELETE \
--location
"https://astral.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/appBackups/<BACKUP_ID>" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

curl の例：force オプションを使用して、アプリケーションのバックアップを 1 つ削除します

```
curl --request DELETE \
--location
"https://astral.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/appBackups/<BACKUP_ID>" \
--include \
--header "Force-Delete: true" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

データ保護ポリシーを作成する

1つ以上のスケジュールに基づいてデータ保護ポリシーを作成できます。

ステップ1：アプリを選択する

ワークフローを実行 "アプリを表示します" をクリックし、目的のアプリケーションを選択します。

手順2：保護を作成する

次のREST API呼び出しを実行して、特定のアプリケーションの保護ポリシーを作成します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（Post ）	/accounts / {account_id} /k8s/v1/apps/ {app_id} /schedules

カールの例

```
curl --request POST \
--location
"https://astral.netapp.io/accounts/$ACCOUNT_ID/k8s/v1/apps/$APP_ID/schedules" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

JSON の入力例

```
{  
  "type": "application/astra-schedule",  
  "version": "1.3",  
  "name": "Backup Schedule",  
  "enabled": "true",  
  "granularity": "monthly",  
  "minute": "0",  
  "hour": "0",  
  "dayOfMonth": "1",  
  "snapshotRetention": "12",  
  "backupRetention": "12"  
}
```

アプリケーションのクローニングとリストア

アプリケーションのクローンを作成します

既存のアプリケーションをクローニングして、新しいアプリケーションを作成できます。

作業を開始する前に

このワークフローについては、次の点に注意してください。

- ・ アプリケーションのバックアップまたはスナップショットは使用されません
- ・ クローニング処理は同じクラスタ内で実行されます
- ・ 新しいアプリケーションは別のネームスペースに配置されます



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、 JSON 入力の「 clusterId 」 パラメータを環境に応じて更新する必要があります。

手順1：クローニングするアプリケーションを選択

ワークフローを実行 "アプリを表示します" をクリックし、クローニングするアプリケーションを選択します。アプリケーションのクローニングに使用される REST 呼び出しには、いくつかのリソース値が必要です。

ステップ2：アプリのクローンを作成する

次のREST API呼び出しを実行してアプリケーションをクローニングします。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿 (Post)	/accounts / {account_id} /k8s/v2/apps

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
JSON	ボディ (Body)	はい。	クローニングされたアプリケーションのパラメーターを提供します。以下の例を参照してください。

curl の例：アプリケーションのクローンを作成します

```
curl --request POST \
--location "https://astranetapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps" \
--include \
--header "Content-Type: application/astra-app+json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

JSON の入力例

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql-ns",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

スナップショットからアプリケーションをクローニングします

Snapshotからクローニングして新しいアプリケーションを作成できます。

作業を開始する前に

このワークフローについては、次の点に注意してください。

- ・ アプリケーションスナップショットが使用されます
- ・ クローニング処理は同じクラスタ内で実行されます



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、JSON 入力の「clusterId」パラメータを環境に応じて更新する必要があります。

手順1：クローニングするアプリケーションを選択

ワークフローを実行 "アプリを表示します" をクリックし、クローニングするアプリケーションを選択します。アプリケーションのクローニングに使用される REST 呼び出しには、いくつかのリソース値が必要です。

手順2：使用するSnapshotを選択する

ワークフローを実行 "Snapshot を一覧表示します" をクリックし、使用する Snapshot を選択します。

ステップ3：アプリケーションのクローンを作成する

次の REST API 呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（Post）	/accounts / {account_id} /k8s/v2/apps

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
JSON	ボディ（Body）	はい。	クローニングされたアプリケーションのパラメーターを提供します。以下の例を参照してください。

curl の例：Snapshot からアプリケーションをクローニングします

```
curl --request POST \
--location "'https://astra.netapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps'" \
--include \
--header "Content-Type: application/astra-app+json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

JSON の入力例

```
{  
  "type": "application/astra-app",  
  "version": "2.0",  
  "name": "mysql-clone2",  
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",  
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",  
  "namespace": "mysql",  
  "snapshotID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"  
}
```

バックアップからアプリケーションをクローニングする

新しいアプリケーションは、バックアップからクローニングして作成できます。

作業を開始する前に

このワークフローについては、次の点に注意してください。

- ・ アプリケーションのバックアップが使用されます
- ・ クローニング処理は同じクラスタ内で実行されます



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、JSON 入力の「clusterId」パラメータを環境に応じて更新する必要があります。

手順1：クローニングするアプリケーションを選択

ワークフローを実行 "アプリを表示します" をクリックし、クローニングするアプリケーションを選択します。アプリケーションのクローニングに使用される REST 呼び出しには、いくつかのリソース値が必要です。

手順2：使用するバックアップを選択する

ワークフローを実行 "バックアップをリスト表示します" 使用するバックアップを選択します。

ステップ3：アプリケーションのクローンを作成する

次の REST API 呼び出しを実行します。

HTTP メソッドとエンドポイント

この REST API 呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（Post）	/accounts / {account_id} /k8s/v2/qpps

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
JSON	ボディ (Body)	はい。	クローニングされたアプリケーションのパラメーターを提供します。以下の例を参照してください。

curl の例：バックアップからアプリケーションをクローニングします

```
curl --request POST \
--location "https://astranetapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps" \
--include \
--header "Content-Type: application/astra-app+json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

JSON の入力例

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone3",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

バックアップからアプリケーションをリストアする

アプリケーションをリストアするには、バックアップから新しいアプリケーションを作成します。

ステップ1：復元するアプリを選択する

ワークフローを実行 "[アプリを表示します](#)" をクリックし、クローニングするアプリケーションを選択します。アプリケーションのリストアに使用するREST呼び出しには、いくつかのリソース値が必要です。

手順2：使用するバックアップを選択する

ワークフローを実行 "[バックアップをリスト表示します](#)" 使用するバックアップを選択します。

ステップ3：アプリを復元する

次の REST API呼び出しを実行します。バックアップ（以下を参照）または Snapshot の ID を指定する必要があります。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
PUT	/accounts / {account_id} /k8s/v2/apps/ {app_id}

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
JSON	ボディ（Body）	はい。	クローニングされたアプリケーションのパラメーターを提供します。以下の例を参照してください。

curl の例：バックアップからアプリを所定の場所にリストアします

```
curl --request PUT \
--location
"https://astranetapp.io/accounts/$ACCOUNT_ID/k8s/v2/apps/$APP_ID" \
--include \
--header "Content-Type: application/astra-app+json" \
--header "ForceUpdate: true" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

JSON の入力例

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

ネームスペース

ネームスペースを表示します

使用可能なネームスペースをリストできます。

ネームスペースを表示するには、次のREST API呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts/{account_id}/topology/v1/Namespaces

curlの例：すべてのネームスペースのすべてのデータを返します

```
curl --request GET \
--location
"https://astral.netapp.io/accounts/$ACCOUNT_ID/topology/v1/namespaces" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

cURLの例：すべてのネームスペースについて、名前、状態、およびクラスタIDを返します

```
curl --request GET \
--location
"https://astral.netapp.io/accounts/$ACCOUNT_ID/topology/v1/namespaces?inclu
de=name,namespaceState,clusterID" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

JSON 出力例

```
{
  "items": [
    [
      "default",
      "discovered",
      "922f924a-a476-4a79-97f6-472571698154"
    ],
    [
      "kube-node-lease",
      "discovered",
      "922f924a-a476-4a79-97f6-472571698154"
    ],
    [
      "kube-public",
      "discovered",
      "922f924a-a476-4a79-97f6-472571698154"
    ]
  ]
}
```

```

] ,
[
  "kube-system",
  "discovered",
  "922f924a-a476-4a79-97f6-472571698154"
],
[
  "mysql",
  "discovered",
  "922f924a-a476-4a79-97f6-472571698154"
],
[
  "mysql-clone1",
  "discovered",
  "922f924a-a476-4a79-97f6-472571698154"
],
[
  "netapp-acc-operator",
  "discovered",
  "922f924a-a476-4a79-97f6-472571698154"
],
[
  "openshift",
  "discovered",
  "922f924a-a476-4a79-97f6-472571698154"
],
[
  "trident",
  "discovered",
  "922f924a-a476-4a79-97f6-472571698154"
]
],
"metadata": {}
}

```

サポート

通知を一覧表示します

特定のアストラアカウントの通知をリストできます。この処理は、システムアクティビティの監視や問題のデバッグの一環として実行できます。

次の REST API 呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /core/v1/notifications

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
フィルタ	クエリ	いいえ	必要に応じて、応答で返す通知をフィルタリングします。
含める	クエリ	いいえ	必要に応じて、応答で返す値を選択します。

curl の例：すべての通知を返します

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/notifications" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

curl の例：重大度が **WARNING** の通知について概要を返します

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/notifications?filter \
=severity%20eq%20'warning'&include=description" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

JSON 出力例

```
{  
  "items": [  
    [  
      "Trident on cluster david-ie-00 has failed or timed out;  
      installation of the Trident operator failed or is not yet complete;  
      operator failed to reach an installed state within 300.00 seconds;  
      container trident-operator not found in operator deployment"  
    ],  
    [  
      "Trident on cluster david-ie-00 has failed or timed out;  
      installation of the Trident operator failed or is not yet complete;  
      operator failed to reach an installed state within 300.00 seconds;  
      container trident-operator not found in operator deployment"  
    ]  
  ],  
  "metadata": {}  
}
```

失敗したアプリを削除します

バックアップまたはスナップショットが失敗した場合、管理アプリを削除できることがあります。この場合、以下に説明するワークフローを使用して、アプリケーションを手動で削除できます。

ステップ1：削除するアプリを選択する

ワークフローを実行 "[アプリを表示します](#)" をクリックし、削除するアプリケーションを選択します。

手順2：アプリケーションの既存のバックアップを一覧表示する

ワークフローを実行 "[バックアップをリスト表示します](#)"。

手順3：すべてのバックアップを削除する

ワークフローを実行して、すべてのアプリケーションバックアップを削除します "[バックアップを削除します](#)" リスト内のバックアップごとに、

ステップ4：アプリケーションの既存のスナップショットを一覧表示する

ワークフローを実行 "[Snapshot を一覧表示します](#)"。

手順5：すべてのスナップショットを削除する

ワークフローを実行 "[Snapshot を削除します](#)" リスト内の各 Snapshot から削除します。

手順6：アプリケーションを削除する

ワークフローを実行 "アプリの管理を解除します" アプリケーションを削除します。

著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を隨時、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 5225.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。