Management workflows

Astra Automation

NetApp
August 10, 2022

Table of Contents

Management workflows

Before you begin

You can use these workflows as part of administering the applications within an Astra managed cluster.

These workflows can be expanded and enhanced by NetApp at any time and so you should review them periodically.

General preparation

Before using any of the Astra workflows, make sure to review Prepare to use the workflows.

Workflow categories

The management workflows are organized in different categories to make it easier to locate the one you want.

Category	Description
Application control	These workflows allow you to control the managed and unmanaged applications. You can list the apps as well as create and remove a managed app.
Application protection	You can use these workflows to protect your managed applications through snapshots and backups.
Cloning and restoring apps	These workflow describe how to clone and restore your managed applications.
Support	There are several workflows available to debug and support your applications as well as the general Kubernetes environment.

Additional considerations

There are a several additional considerations when using the management workflows.

Cloning an app

There are a few things to consider when cloning an application. The parameters described below are part of the JSON input.

Source cluster identifier

The value of `sourceClusterID` always identifies the cluster where the original app is installed.

Cluster identifier

The value of `clusterID` identifies the cluster where the new app will be installed.

- When cloning within the same cluster, `clusterID` and `sourceClusterID` have the same value.
- When cloning across clusters, the two values are different and `clusterID` should be the ID of the target cluster.

Namespaces

The `namespace` value must be different than the original source app. Further, the namespace for the clone cannot exist and Astra will create it.

Backups and snapshots

You can optionally clone an application from an existing backup or snapshot using the `backupID` or `snapshotID` parameters. If you don't provide a backup or snapshot, Astra will create a backup of the application first and then clone from the backup.

Restoring an app

Here are a few things to consider when restoring an application.

- Restoring an application is very similar to the clone operation.
- When restoring an app, you must provide either a backup or snapshot.

App control

List the unmanaged apps

You can list the applications that are currently not managed by Astra. You might do this as part of selecting an app to be managed.

The REST endpoint used in these workflows returns all the Astra applications by default. You can use the `filter` query parameter on the API call to request only the unmanaged apps be returned. As an alternative, you can omit the filter parameter to return all the apps and then examine the `managedState` field in the output to determine which apps are in the `unmanaged` state.

List only the apps with managedState equal to unmanaged

This workflow uses the `filter` query parameter to return only the unmanaged apps.

1. List the unmanaged applications

Perform the following REST API call.

HTTP method	Path
GET	/account/{account_id}/topology/v1/apps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
filter	Query	No	Use a filter to specify which apps should be returned.

Parameter	Type	Required	Description
include	Query	No	Optionally select the values you want returned in the response.

Curl example: Return the name, id, and managedState for the unmanaged apps

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/apps?filter=man
agedState%20eq%20'unmanaged'&include=name,id,managedState' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

JSON output example

```
{
  "items": [
    [
      "maria",
      "eed19f78-0884-4792-bb7a-313258c6b0b1",
      "unmanaged"
    ],
    [
      "test-postgres-app",
      "1ee6235b-cda1-45cb-8d4c-630bdb8b41a5",
      "unmanaged"
    ],
    [
      "postgres1-postgresql",
      "e591ee59-ea90-4a9f-8e6c-d2b6e8647096",
      "unmanaged"
    ],
    [
      "kube-system",
      "077a2f73-4b51-4d04-8c6c-f63b3b069755",
      "unmanaged"
    ],
    [
      "trident",
      "5b6fc28f-e308-4653-b9d2-6d66a764d2e1",
      "unmanaged"
    ],
    [
      "postgres1-postgresql-clone",
      "06be05c5-763e-4d73-bd06-1f27f5f2e130",
      "unmanaged"
    ]
  ],
  "metadata": {}
}
```


List all the apps and select the unmanaged apps

This workflow returns all the apps. You must examine the output to determine which are unmanaged.

1. List all the applications

Perform the following REST API call.

HTTP method	Path
GET	/account/{account_id}/topology/v1/apps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
include	Query	No	Optionally select the values you want returned in the response.

Curl example: Return all data for all apps

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/apps' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Curl example: Return the name, id, and managedState for all apps

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/apps?include=na
me,id,managedState' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

JSON output example

```
{
  "items": [
    [
      "maria",
      "eed19f78-0884-4792-bb7a-313258c6b0b1",
      "unmanaged"
    ],
    [
      "mariadb-mariadb",
      "8da20fff-c69c-4170-bb0d-e4f91c5a1333",
      "managed"
    ],
    [
      "test-postgres-app",
      "1ee6235b-cda1-45cb-8d4c-630bdb8b41a5",
      "unmanaged"
    ],
    [
      "postgres1-postgresql",
      "e591ee59-ea90-4a9f-8e6c-d2b6e8647096",
      "unmanaged"
    ],
    [
      "kube-system",
      "077a2f73-4b51-4d04-8c6c-f63b3b069755",
      "unmanaged"
    ],
    [
      "trident",
      "5b6fc28f-e308-4653-b9d2-6d66a764d2e1",
      "unmanaged"
    ],
    [
      "postgres1-postgresql-clone",
      "06be05c5-763e-4d73-bd06-1f27f5f2e130",
      "unmanaged"
    ],
    [
      "davidns-postgres-app",
      "11e046b7-ec64-4184-85b3-debcc3b1da4d",
      "managed"
    ]
  ],
  "metadata": {}
}
```

2. Select the unmanaged applications

Review the output of the API call and manually select the apps with `managedState` equal to `unmanaged`.

List the managed apps

You can list the applications that are currently managed by Astra. You might do this as part of finding the snapshots or backups for a specific app.

1. List the applications

Perform the following REST API call.

HTTP method	Path
GET	/account/{account_id}/k8s/v1/managedApps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
include	Query	No	Optionally select the values you want returned in the response.

Curl example: Return all data for all apps

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Curl example: Return the name, id, and state for all apps

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps?include=
name,id,state' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

JSON output example

```
{
  "items": [
    [
      "test-postgres-app",
      "1ee6235b-cda1-45cb-8d4c-630bdb8b41a5",
      "running"
    ]
  ],
  "metadata": {}
}
```


Get a managed app

You can retrieve all the resource variables describing a single managed application.

Before you begin

You must have the ID of the managed app you want to retrieve. If needed you can use the workflow List the managed apps to locate the application.

1. Get the application

Perform the following REST API call.

HTTP method	Path
GET	/accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
managed app id	Path	Yes	ID value of the managed application to retrieve.

Curl example: Return all data for the application

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Manage an app

You can create a managed application based on an application already known to Astra.

When an application is managed, you can protect it by taking regular backups and snapshots.

Before you begin

You must have the ID of the discovered app you want to manage. If needed you can use the workflow List the unmanaged apps to locate the application.

1. Manage the application

Perform the following REST API call.

HTTP method	Path
POST	/account/{account_id}/k8s/v1/managedApps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
JSON	Body	Yes	Provides the parameters needed to identify the application to be managed. See the example below.

JSON input example

```
{
  "type": "application/astra-managedApp",
  "version": "1.1",
  "id": "7da20fff-c69d-4270-bb0d-a4f91c5a1333"
}
```

Curl example: Manage an app

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Content-Type: application/astra-managedApp+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'  --d @JSONinput
```

Unmanage an app

You can remove a managed app when it's no longer needed. Removing a managed application also deletes the associated schedules.

Before you begin

You must have the ID of the managed app you want to unmanage. If needed you can use the workflow List the managed apps to locate the application.

The application's backups and snapshots are not automatically removed when it is deleted. If you no longer need the backups and snapshots, you should delete them before removing the application.

1. Unmanaged the app

Perform the following REST API call.

HTTP method	Path
DELETE	/accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
managed app id	Path	Yes	Identifies the managed application to remove.

Curl example: Remove a managed app

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

App protection

List the snapshots

You can list the snapshots that have been taken for a specific managed application.

Before you begin

You must have the ID of the managed app you want to list the snapshots for. If needed you can use the workflow List the managed apps to locate the application.

1. List the snapshots

Perform the following REST API call.

HTTP method	Path
GET	/accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appSnaps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
managed app id	Path	Yes	Identifies the managed application owning the listed snapshots.
count	Query	No	If `count=true` the number of snapshots is included in the metadata section of the response.

Curl example: Return all snapshots for the app

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appSnaps' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Curl example: Return all snapshots for the app and the count

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appSnaps?count=true' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

JSON output example

```
{
  "items": [
    {
      "id": "dc2974ae-f71d-4c81-91b5-f96cf72dc3ba",
      "metadata": {
        "createdBy": "fb093413-b6fc-4a64-a48a-afc32ada8537",
        "creationTimestamp": "2021-06-04T21:23:14Z",
        "modificationTimestamp": "2021-06-04T21:23:14Z",
        "labels": []
      },
      "snapshotAppAsset": "4547658d-cc06-4c1d-ad8a-4a05274d0db0",
      "snapshotCreationTimestamp": "2021-06-04T21:23:47Z",
      "name": "test-postgres-app-snapshot-20210604212213",
      "state": "completed",
      "stateUnready": [],
      "type": "application/astra-appSnap",
      "version": "1.0"
    }
  ],
  "metadata": {
    "count": 1
  }
}
```


List the backups

You can list the backups that have been created for a specific managed application.

Before you begin

You must have the ID of the managed app you want to list the backups for. If needed you can use the workflow List the managed apps to locate the application.

1. List the backups

Perform the following REST API call.

HTTP method	Path
GET	/accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appBackups

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
managed app id	Path	Yes	Identifies the managed application owning the listed backups.

Curl example: Return all backups for the app

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appBackups' --header 'Accept: */*' --header 'Authorization:
Bearer <API_TOKEN>'
```

JSON output example


```
{
  "items": [
    {
      "type": "application/astra-appBackup",
      "version": "1.0",
      "id": "ed39fdb0-12db-497b-9e46-20036c1fb0d2",
      "name": "mariadb-mariadb-backup-20210617175900",
      "state": "completed",
      "stateUnready": [],
      "bytesDone": 0,
      "percentDone": 100,
      "metadata": {
        "labels": [],
        "creationTimestamp": "2021-06-17T17:59:09Z",
        "modificationTimestamp": "2021-06-17T17:59:09Z",
        "createdBy": "fb093413-b6fc-4a64-a48a-afc32ada8537"
      }
    }
  ],
  "metadata": {}
}
```


Create a snapshot for a managed app

You can create a snapshot for a specific managed application.

Before you begin

You must have the ID of the managed app you want to create a snapshot for. If needed you can use the workflow List the managed apps to locate the application.

1. Create a snapshot

Perform the following REST API call.

HTTP method	Path
POST	/accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appSnaps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
managed app id	Path	Yes	Identifies the managed application where the snapshot will be created.
JSON	Body	Yes	Provides the parameters for the snapshot. See the example below.

JSON input example

```
{
  "type": "application/astra-appSnap",
  "version": "1.0",
  "name": "snapshot-david-1"
}
```

Curl example: Create a snapshot for the app

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appSnaps' --header 'Content-Type: application/astra-appSnap+json'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --d
@JSONinput
```

Create a backup for a managed app

You can create a backup for a specific managed application. You can use the backup to restore or clone the app.

Before you begin

You must have the ID of the managed app you want to create a backup for. If needed you can use the workflow List the managed apps to locate the application.

1. Create a backup

Perform the following REST API call.

HTTP method	Path
POST	/accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appBackups

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
managed app id	Path	Yes	Identifies the managed application where the backup will be created.
JSON	Body	Yes	Provides the parameters for the backup. See the example below.

JSON input example

```
{
  "type": "application/astra-appBackup",
  "version": "1.0",
  "name": "backup-david-1"
}
```

Curl example: Create a backup for the app

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appBackups' --header 'Content-Type: application/astra-
appBackup+json' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>' --d @JSONinput
```

Delete a snapshot

You can delete a snapshot associated with a managed application.

Before you begin

You must have the following:

- ID of the managed app that owns the snapshot. If needed you can use the workflow List the managed apps to locate the application.
- ID of the snapshot you want to delete. If needed you can use the workflow List the snapshots to locate the snapshot.

1. Delete the snapshot

Perform the following REST API call.

HTTP method	Path
DELETE	/accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appSnaps/{appSnap_id}

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
managed app id	Path	Yes	Identifies the managed application owning the snapshot.
snapshot id	Path	Yes	Identifies the snapshot to be deleted.

Curl example: Delete a single snapshot for the app

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appSnaps/<SNAPSHOT_ID>' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

Delete a backup

You can delete a backup associated with a managed application.

Before you begin

You must have the following:

- ID of the managed app that owns the backup. If needed you can use the workflow List the managed apps to locate the application.
- ID of the backup you want to delete. If needed you can use the workflow List the backups to locate the snapshot.

1. Delete the backup

Perform the following REST API call.

> You can force the deletion of a failed backup using the optional request header as described below.

HTTP method	Path
DELETE	/accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appBackups/{appBackup_id}

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
managed app id	Path	Yes	Identifies the managed application owning the backup.
backup id	Path	Yes	Identifies the backup to be deleted.
force delete	Header	No	Used to force the deletion of a failed backup.

Curl example: Delete a single backup for the app

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appBackups/<BACKUP_ID>' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

Curl example: Delete a single backup for the app with the force option

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appBackups/<BACKUP_ID>' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>' --header 'Force-Delete: true'
```

Cloning and restoring an app

Clone a managed app

You can create a new application by cloning an existing managed app.

Before you begin

Note the following about this workflow:

- An app backup or snapshot is not used
- The clone operation is performed within the same cluster

> To clone an app to a different cluster, you need to update the `clusterId` parameter in the JSON input as appropriate for your environment.

1. Select the managed app to clone

Perform the workflow List the managed apps and select application you want to clone. Several of the resource values are needed for the REST call used to clone the app.

2. Clone the app

Perform the following REST API call.

HTTP method	Path
POST	/account/{account_id}/k8s/v1/managedApps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
JSON	Body	Yes	Provides the parameters for the cloned app. See the example below.

JSON input example

```
{
  "type": "application/astra-managedApp",
  "version": "1.0",
  "name": "postgres1-postgresql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "davidns-postgres-app",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

Curl example: Clone an app

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Content-Type: application/astra-managedApp+json' --header '*/*'
--header 'Authorization: Bearer <API_TOKEN>' --d @JSONinput
```

Clone a managed app from a snapshot

You can create a new application by cloning it from an app snapshot.

Before you begin

Note the following about this workflow:

- An app snapshot is used
- The clone operation is performed within the same cluster

To clone an app to a different cluster, you need to update the `clusterId` parameter in the JSON input as appropriate for your environment.

1. Select the managed app to clone

Perform the workflow List the managed apps and select application you want to clone. Several of the resource values are needed for the REST call used to clone the app.

2. Select the snapshot to use

Perform the workflow List the snapshots and select snapshot you want to use.

3. Clone the app

Perform the following REST API call.

HTTP method	Path
POST	/account/{account_id}/k8s/v1/managedApps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
JSON	Body	Yes	Provides the parameters for the cloned app. See the example below.

JSON input example

```
{
  "type": "application/astra-managedApp",
  "version": "1.0",
  "name": "postgres1-postgresql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "davidns-postgres-app",
  "snapshotID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

Curl example: Clone an app from a snapshot

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Content-Type: application/astra-managedApp+json' --header '*/*'
--header 'Authorization: Bearer <API_TOKEN>' --d @JSONinput
```

Clone a managed app from a backup

You can create a new managed application by cloning it from an app backup.

Before you begin

Note the following about this workflow:

- An app backup is used
- The clone operation is performed within the same cluster

> To clone an app to a different cluster, you need to update the `clusterId` parameter in the JSON input as appropriate for your environment.

1. Select the managed app to clone

Perform the workflow List the managed apps and select application you want to clone. Several of the resource values are needed for the REST call used to clone the app.

2. Select the backup to use

Perform the workflow List the backups and select backup you want to use.

3. Clone the app

Perform the following REST API call.

HTTP method	Path
POST	/account/{account_id}/k8s/v1/managedApps

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
JSON	Body	Yes	Provides the parameters for the cloned app. See the example below.

JSON input example

```
{
  "type": "application/astra-managedApp",
  "version": "1.0",
  "name": "postgres1-postgresql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "davidns-postgres-app",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

Curl example: Clone an app from a backup

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Content-Type: application/astra-managedApp+json' --header '*/*'
--header 'Authorization: Bearer <API_TOKEN>' --d @JSONinput
```

Restore a managed app from a backup

You can restore a managed application by creating a new app from a backup.

1. Select the managed app to restore

Perform the workflow List the managed apps and select application you want to clone. Several of the resource values are needed for the REST call used to clone the app.

2. Select the backup to use

Perform the workflow List the backups and select backup you want to use.

3. Restore the app

Perform the following REST API call. You must provide the ID for either a backup (as shown below) or snapshot.

HTTP method	Path
PUT	/account/{account_id}/k8s/v1/managedApps/{app_id}

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
JSON	Body	Yes	Provides the parameters for the cloned app. See the example below.

JSON input example

```
{
  "type": "application/astra-managedApp",
  "version": "1.2",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

Curl example: Restore an app in place from a backup

```
curl --location -i --request PUT
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<APP_ID>
' --header 'Content-Type: application/astra-managedApp+json' --header
'*/*' --header 'ForceUpdate: true' --header 'Authorization: Bearer
<API_TOKEN>' --d @JSONinput
```

Support

List the notifications

You can list the notifications for a specific Astra account. You might do this as part of monitoring the system activity or debugging an issue.

1. List the notifications

Perform the following REST API call.

HTTP method	Path
GET	/account/{account_id}/core/v1/notifications

Additional input parameters

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl examples for this step.

Parameter	Type	Required	Description
filter	Query	No	Optionally filter the notifications you want returned in the response.
include	Query	No	Optionally select the values you want returned in the response.

Curl example: Return all notifications

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/notifications'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Curl example: Return the description for notifications with severity of warning

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/notifications?filte
r=severity%20eq%20'warning'&include=description' --header 'Accept: */*'
--header 'Authorization: Bearer <API_TOKEN>'
```

JSON output example

```
{
  "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": {}
}
```

Delete a failed app

You might be unable to remove a managed app if it has a backup or snapshot in a failed state. In this case you can manually remove the app using the workflow described below.

1. Select the managed app to delete

Perform the workflow List the managed apps and select application you want to remove.

2. List the existing backups for the app

Perform the workflow List the backups.

3. Delete all the backups

Delete all the app backups by performing the workflow Delete a backup for each backup in the list.

4. List the existing snapshots for the app

Perform the workflow List the snapshots.

5. Delete all the snapshots

Perform the workflow Delete a snapshot from each snapshot in the list.

6. Remove the application

Perform the workflow Unmanage an app to remove the application.