NetApp

Release notes

Astra Control Center

NetApp
February 02, 2022

Table of Contents

Release notes

We're pleased to announce the 21.12 release of Astra Control Center.

- What's in this release of Astra Control Center

- Resolved issues

- Known issues

- Known issues with Astra Data Store preview and this Astra Control Center release

- Known limitations

Follow us on Twitter @NetAppDoc. Send feedback about documentation by becoming a GitHub contributor or sending an email to doccomments@netapp.com.

What's new in this release of Astra Control Center

We're pleased to announce the latest 21.12 release of Astra Control Center.

14 December 2021 (21.12)

Updated release of Astra Control Center.

New features and support
- Application restore

- Execution hooks

- Support for applications deployed with namespace-scoped operators

- Additional support for upstream Kubernetes and Rancher

- Astra Data Store preview backend management and monitoring

- Astra Control Center upgrades

- Red Hat OperatorHub option for installation

Resolved issues
- Resolved issues for this release

Known issues and limitations
- Known issues for this release

- Known issues with Astra Data Store preview and this Astra Control Center release

- Known limitations for this release

5 August 2021 (21.08)

Initial release of Astra Control Center.

- What it is

- Understand architecture and components

- What it takes to get started

- [Install](#) and [setup](#)
- [Manage](#) and [protect](#) apps
- [Manage buckets](#) and [storage backends](#)
- [Manage accounts](#)
- [Automate with API](#)

Find more information

- [Known issues for this release](#)
- [Known limitations for this release](#)
- [Astra Data Store documentation](#)
- [Earlier versions of Astra Control Center documentation](#)

Resolved issues

These issues have been corrected in this release of the product.

Extra backups are retained as part of scheduled backup

Sometimes one or more backups in Astra Control Center are retained beyond the number specified to be retained in the backup schedule. These extra backups should be deleted as part of a scheduled backup but are not deleted and are stuck in a `pending` state.

Backup or clone fails for apps using PVCs with decimal units in Astra Control Center

Volumes created with decimal units fail using the Astra Control Center backup or clone process.

Astra Control Center UI slow to show changes to app resources such as persistent volume changes

After a data protection operation (clone, backup, restore) and subsequent persistent volume resize, there is up to a twenty-minute delay before the new volume size is shown in the UI. This delay in the UI can also occur when any app resources are added or modified. In this case, a data protection operation is successful within minutes and you can use the management software for the storage backend to confirm the change in volume size.

Incorrect cluster role binding created by Astra Control Center custom resource definition during installation

The patch to correct cluster role binding during installation is no longer required in this release.

ASUP collection stuck in a generating or uploading state

If an ASUP pod is stopped or restarted, an ASUP collection might become stuck in a generating or uploading state.

Operator-deployed apps and namespaces

An operator and the app it deploys must use the same namespace. Astra Control supports only one operator-deployed app per namespace.

Find more information

- Known issues
- Known limitations
- Known issues with Astra Data Store preview and this Astra Control Center release

Known issues with Astra Data Store preview and this Astra Control Center release

Known issues identify problems that might prevent you from using this release of the product successfully.

The following known issues affect the management of Astra Data Store with this current release of the Astra Control Center:

Astra Data Store preview cannot be used as a storage class for Astra Control Center due to MongoDB pod liveness probe failure

When you attempt to use Astra Data Store preview as the storage class provisioner during an Astra Control Center deployment, the MongoDB pod liveness probe fails, resulting in a deployment that will not complete.

To correct this issue, make the following changes in addition to the standard YAML changes when completing the Astra Control Center installation process:

1. Edit the Astra Control Center operator deployment YAML (astra_control_center_operator_deploy.yaml) to change the Helm install timeout:

```
- name: ACCOP_HELM_INSTALLTIMEOUT
  value: 20m
```

2. Edit the Astra Control Center custom resource (CR) file (astra_control_center_min.yaml) and include the highlighted additional values under spec:

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  additionalValues:
    polaris-mongodb:
      mongodb:
        livenessProbe:
          initialDelaySeconds: 400
      metrics:
        livenessProbe:
          initialDelaySeconds: 400
```

Astra Control Center shows Astra Data Store preview storage backend in `Unknown` state

Astra Control Center shows the Astra Data Store preview storage backend in an `Unknown` state from the Backends page in the UI. In this condition, the storage backend is actually available and can be communicated with. A component within the storage backend is likely in an unhealthy state and needs to be returned to a healthy state for the storage backend to show as `available`.

Find more information

- Resolved issues
- Known issues
- Known limitations
- Astra Data Store documentation

Known issues

Known issues identify problems that might prevent you from using this release of the product successfully.

The following known issues affect the current release:

- App with user-defined label goes into "removed" state
- Unable to stop running app backup
- During app restore from backup Trident creates a larger PV than the original PV
- Clone performance impacted by large persistent volumes
- App clones fail using a specific version of PostgreSQL
- App clones fail when using Service Account level OCP Security Context Constraints (SCC)
- Reusing buckets between instances of Astra Control Center causes failures
- Selecting a bucket provider type with credentials for another type causes data protection failures
- Backups and snapshots might not be retained during removal of an Astra Control Center instance
- Clone operation can't use other buckets besides the default
- Managing a cluster with Astra Control Center fails when default kubeconfig file contains more than one context
- 500 internal service error when attempting Trident app data management
- Custom app execution hook scripts time out and cause post-snapshot scripts not to execute
- Can't determine ASUP tar bundle status in scaled environment
- Snapshots eventually begin to fail when using external-snapshotter version 4.2.0
- Simultaneous app restore operations in the same namespace can fail
- Upgrade fails if source version uses a container image registry that does not require authentication and target version uses a container image registry that requires authentication
- Uninstall of Astra Control Center fails to clean up the monitoring-operator pod on the managed cluster
- Uninstall of Astra Control Center fails to clean up Traefik CRDs

App with user-defined label goes into "removed" state

If you define an app with a non-existent k8s label, Astra Control Center will create, manage, and then immediately remove the app. To avoid this, add the k8s label to pods and resources after the app is managed by Astra Control Center.

Unable to stop running app backup

There is no way to stop a running backup. If you need to delete the backup, wait until it has completed and then use the instructions in Delete backups. To delete a failed backup, use the Astra Control API.

During app restore from backup Trident creates a larger PV than the original PV

If you resize a persistent volume after creating a backup and then restore from that backup, the persistent volume size will match the new size of the PV instead of using the size of the backup.

Clone performance impacted by large persistent volumes

Clones of very large and consumed persistent volumes might be intermittently slow, dependent on cluster access to the object store. If the clone is hung and no data has been copied for more than 30 minutes, Astra Control terminates the clone action.

App clones fail using a specific version of PostgreSQL

App clones within the same cluster consistently fail with the Bitnami PostgreSQL 11.5.0 chart. To clone successfully, use an earlier or later version of the chart.

App clones fail when using Service Account level OCP Security Context Constraints (SCC)

An application clone might fail if the original security context constraints are configured at the service account level within the namespace on the OCP cluster. When the application clone fails, it appears in the Managed Applications area in Astra Control Center with status `Removed`. See the knowledgebase article for more information.

App clones fail after an application is deployed with a set storage class

After an application is deployed with a storage class explicitly set (for example, `helm install …-set global.storageClass=netapp-cvs-perf-extreme`), subsequent attempts to clone the application require that the target cluster have the originally specified storage class.
Cloning an application with an explicitly set storage class to a cluster that does not have the same storage class will fail. There are no recovery steps in this scenario.

Reusing buckets between instances of Astra Control Center causes failures

If you try to reuse a bucket used by another or previous installation of Astra Control Center, backup and restore operations will fail. You must use a different bucket or completely clean out the previously used bucket. You can't share buckets between instances of Astra Control Center.

Selecting a bucket provider type with credentials for another type causes data protection failures

When you add a bucket, select the correct bucket provider and enter the right credentials for that provider. For example, the UI accepts NetApp ONTAP S3 as the type and accepts StorageGRID credentials; however, this will cause all future app backups and restores using this bucket to fail.

Backups and snapshots might not be retained during removal of an Astra Control Center instance

If you have an evaluation license, be sure you store your account ID to avoid data loss in the event of Astra Control Center failure if you are not sending ASUPs.

Clone operation can't use other buckets besides the default

During an app backup or app restore, you can optionally specify a bucket ID. An app clone operation, however, always uses the default bucket that has been defined. There is no option to change buckets for a clone. If you want control over which bucket is used, you can either change the bucket default or do a backup followed by a restore separately.

Managing a cluster with Astra Control Center fails when default kubeconfig file contains more than one context

You cannot use a kubeconfig with more than one cluster and context in it. See the knowledgebase article for more information.

500 internal service error when attempting Trident app data management

If Trident on an app cluster goes offline (and is brought back online) and 500 internal service errors are encountered when attempting app data management, restart all of the Kubernetes nodes in the app cluster to restore functionality.

Custom app execution hook scripts time out and cause post-snapshot scripts not to execute

If an execution hook takes longer than 25 minutes to run, the hook will fail, creating an event log entry with a return code of "N/A". Any affected snapshot will timeout and be marked as failed, with a resulting event log entry noting the timeout.

Because execution hooks often reduce or completely disable the functionality of the application they are running against, you should always try to minimize the time your custom execution hooks take to run.

Can't determine ASUP tar bundle status in scaled environment

During ASUP collection, the status of the bundle in the UI is reported as either `collecting` or `done`. Collection can take up to an hour for large environments. During ASUP download, the network file transfer speed for the bundle might be insufficient, and the download might time out after 15 minutes without any indication in the UI. Download issues depend on the size of the ASUP, the scaled cluster size, and if collection time goes beyond the seven-day limit.

Snapshots eventually begin to fail when using external-snapshotter version 4.2.0

When you use Kubernetes snapshot-controller (also known as external-snapshotter) version 4.2.0 with Kubernetes 1.20 or 1.21, snapshots can eventually begin to fail. To prevent this, use a different supported version of external-snapshotter, such as version 4.2.1, with Kubernetes versions 1.20 or 1.21.

Simultaneous app restore operations in the same namespace can fail

If you try to restore one or more individually managed apps within a namespace simultaneously, the restore operations can fail after a long period of time. As a workaround, restore each app one at a time.

Upgrade fails if source version uses a container image registry that does not require authentication and target version uses a container image registry that requires authentication

If you upgrade an Astra Control Center system that uses a registry that doesn't require authentication to a newer version that uses a registry that requires authentication, the upgrade fails. As a workaround, perform the following steps:

1. Log in to a host that has network access to the Astra Control Center cluster.

2. Ensure that the host has the following configuration:

   ◦ `kubectl` version 1.19 or later is installed

   ◦ The KUBECONFIG environment variable is set to the kubeconfig file for the Astra Control Center cluster

3. Execute the following script:

```
namespace="<netapp-acc>"
statefulsets=("polaris-vault" "polaris-mongodb" "influxdb2" "nats"
"loki")
for ss in ${statefulsets[@]}; do
    existing=$(kubectl get -n ${namespace} statefulsets.apps ${ss} -o
jsonpath='{.spec.template.spec.imagePullSecrets}')
    if [ "${existing}" = "[{}]" ] || [ "${existing}" = "[{},{},{}]" ];
then
        kubectl patch -n ${namespace} statefulsets.apps ${ss} --type
merge --patch '{"spec": {"template": {"spec": {"imagePullSecrets":
[]}}}}'
    else
        echo "${ss} not patched"
    fi
done
```

You should see output similar to the following:

```
statefulset.apps/polaris-vault patched
statefulset.apps/polaris-mongodb patched
statefulset.apps/influxdb2 patched
statefulset.apps/nats patched
statefulset.apps/loki patched
```

4. Proceed with the upgrade using the Astra Control Center upgrade instructions.

Uninstall of Astra Control Center fails to clean up the monitoring-operator pod on the managed cluster

If you did not unmanage your clusters before you uninstalled Astra Control Center, you can manually delete the pods in the netapp-monitoring namespace and the namespace with the following commands:

Steps

1. Delete acc-monitoring agent:

```
oc delete agents acc-monitoring -n netapp-monitoring
```

Result:

```
agent.monitoring.netapp.com "acc-monitoring" deleted
```

2. Delete the namespace:

```
oc delete ns netapp-monitoring
```

Result:

```
namespace "netapp-monitoring" deleted
```

3. Confirm resources removed:

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

Result:

```
No resources found in netapp-monitoring namespace.
```

4. Confirm monitoring agent removed:

```
oc get crd|grep agent
```

Sample result:

```
agents.monitoring.netapp.com                    2021-07-21T06:08:13Z
```

5. Delete custom resource definition (CRD) information:

```
oc delete crds agents.monitoring.netapp.com
```

Result:

```
customresourcedefinition.apiextensions.k8s.io
"agents.monitoring.netapp.com" deleted
```

Uninstall of Astra Control Center fails to clean up Traefik CRDs

You can manually delete the Traefik CRDs:

Steps

1. Confirm which CRDs were not deleted by the uninstall process:

```
kubectl get crds |grep -E 'traefik'
```

Response

```
ingressroutes.traefik.containo.us              2021-06-23T23:29:11Z
ingressroutetcps.traefik.containo.us           2021-06-23T23:29:11Z
ingressrouteudps.traefik.containo.us           2021-06-23T23:29:12Z
middlewares.traefik.containo.us                2021-06-23T23:29:12Z
middlewaretcps.traefik.containo.us             2021-06-23T23:29:12Z
serverstransports.traefik.containo.us          2021-06-23T23:29:13Z
tlsoptions.traefik.containo.us                 2021-06-23T23:29:13Z
tlsstores.traefik.containo.us                  2021-06-23T23:29:14Z
traefikservices.traefik.containo.us            2021-06-23T23:29:15Z
```

2. Delete the CRDs:

```
kubectl delete crd ingressroutes.traefik.containo.us
ingressroutetcps.traefik.containo.us
ingressrouteudps.traefik.containo.us middlewares.traefik.containo.us
serverstransports.traefik.containo.us tlsoptions.traefik.containo.us
tlsstores.traefik.containo.us traefikservices.traefik.containo.us
middlewaretcps.traefik.containo.us
```

Find more information

- Resolved issues
- Known issues with Astra Data Store prreview and this Astra Control Center release
- Known limitations

Known limitations

Known limitations identify platforms, devices, or functions that are not supported by this release of the product, or that do not interoperate correctly with it. Review these limitations carefully.

The same cluster cannot be managed by two Astra Control Center instances

If you want to manage a cluster on another Astra Control Center instance, you should first unmanage the cluster from the instance on which it is managed before you manage it on another instance. After you remove the cluster from management, verify that the cluster is unmanaged by executing this command:

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

There should be no pods running in that namespace or the namespace should not exist. If either of those are true, the cluster is unmanaged.

Astra Control Center cannot manage two identically named clusters in the same cloud

If you try to add a cluster with the same name of a cluster that already exists in your cloud, the operation will fail. This issue most often occurs in a standard Kubernetes environment if you have not changed the cluster name default in Kubernetes configuration files.

As a workaround, do the following:

1. Edit your kubeadm-config ConfigMap:

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

2. Change the `clusterName` field value from `kubernetes` (the Kubernetes default name) to a unique custom name.

3. Edit kubeconfig (`.kube/config`).

4. Update cluster name from `kubernetes` to a unique custom name (`xyz-cluster` is used in the examples below). Make the update in both `clusters` and `contexts` sections as shown in this example:

   ```
   apiVersion: v1
   clusters:
   - cluster:
       certificate-authority-data:
   ExAmPLERb2tCcjZ5K3E2Njk4eQotLExAMpLEORCBDRVJUSUZJQ0FURS0txxxxXX==
       server: https://x.x.x.x:6443
     name: xyz-cluster
   contexts:
   - context:
       cluster: xyz-cluster
       namespace: default
       user: kubernetes-admin
     name: kubernetes-admin@kubernetes
   current-context: kubernetes-admin@kubernetes
   ```

Clones of apps installed using pass by reference operators can fail

Astra Control supports apps installed with namespace-scoped operators. These operators are generally designed with a "pass-by-value" rather than "pass-by-reference" architecture. The following are some operator apps that follow these patterns:

- Apache K8ssandra
- Jenkins CI

- [Percona XtraDB Cluster](#)

Note that Astra Control might not be able to clone an operator that is designed with a "pass-by-reference" architecture (for example, the CockroachDB operator). During these types of cloning operations, the cloned operator attempts to reference Kubernetes secrets from the source operator despite having its own new secret as part of the cloning process. The clone operation might fail because Astra Control is unaware of the Kubernetes secrets in the source operator.

Cluster is in `removed` state although cluster and network are otherwise working as expected

If a cluster is in `removed` state yet cluster and network connectivity appears healthy (external attempts to access the cluster using Kubernetes APIs are successful), the kubeconfig you provided to Astra Control might no longer be valid. This can be due to certificate rotation or expiration on the cluster. To correct this issue, update the credentials associated with the cluster in Astra Control using the [Astra Control API](#):

1. Run a POST call to add an updated kubeconfig file to the `/credentials` endpoint and retrieve the assigned `id` from the response body.

2. Run a PUT call from the `/clusters` endpoint using the appropriate cluster ID and set the `credentialID` to the `id` value from the previous step.

After you complete these steps, the credential associated with the cluster is updated and the cluster should reconnect and update its state to `available`.

OLM-enabled and cluster-scoped operator deployed apps not supported

Astra Control Center does not support apps that are deployed with Operator Lifecycle Manager (OLM)-enabled operators or cluster-scoped operators.

Cloning apps can only be done with same K8s distribution

If you clone an app between clusters, the source and destination clusters must be the same distribution of Kubernetes. For example, if you clone an app from an OpenShift 4.7 cluster, use a destination cluster that is also OpenShift 4.7.

S3 buckets in Astra Control Center do not report available capacity

Before backing up or cloning apps managed by Astra Control Center, check bucket information in the ONTAP or StorageGRID management system.

metalLB 0.11.0 is not supported

metalLB 0.11.0 is not a supported load balancer for Astra Control Center. For more information regarding supported load balancers, see [Astra Control Center requirements](#).

Apps deployed with Helm 2 are not supported

If you use Helm to deploy apps, Astra Control Center requires Helm version 3. Managing and cloning apps deployed with Helm 3 (or upgraded from Helm 2 to Helm 3) is fully supported. For more information, see [Astra Control Center requirements](#).

Astra Control Center does not validate the details you enter for your proxy server

Ensure that you enter the correct values when establishing a connection.

Data protection for Astra Control Center as app not yet available

This release does not support the ability to manage Astra as an app using snapshot, backup, or restore options.

Unhealthy pods affect app management

If a managed app has pods in an unhealthy state, Astra Control can't create new backups and clones.

Existing connections to a Postgres pod causes failures

When you perform operations on Postgres pods, you shouldn't connect directly within the pod to use the psql command. Astra Control requires psql access to freeze and thaw the databases. If there is a pre-existing connection, the snapshot, backup, or clone will fail.

Trident isn't uninstalled from a cluster

When you unmanage a cluster from Astra Control Center, Trident isn't automatically uninstalled from the cluster. To uninstall Trident, you'll need to follow these steps in the Trident documentation.

Find more information

- Resolved issues
- Known issues
- Known issues with Astra Data Store preview and this Astra Control Center release