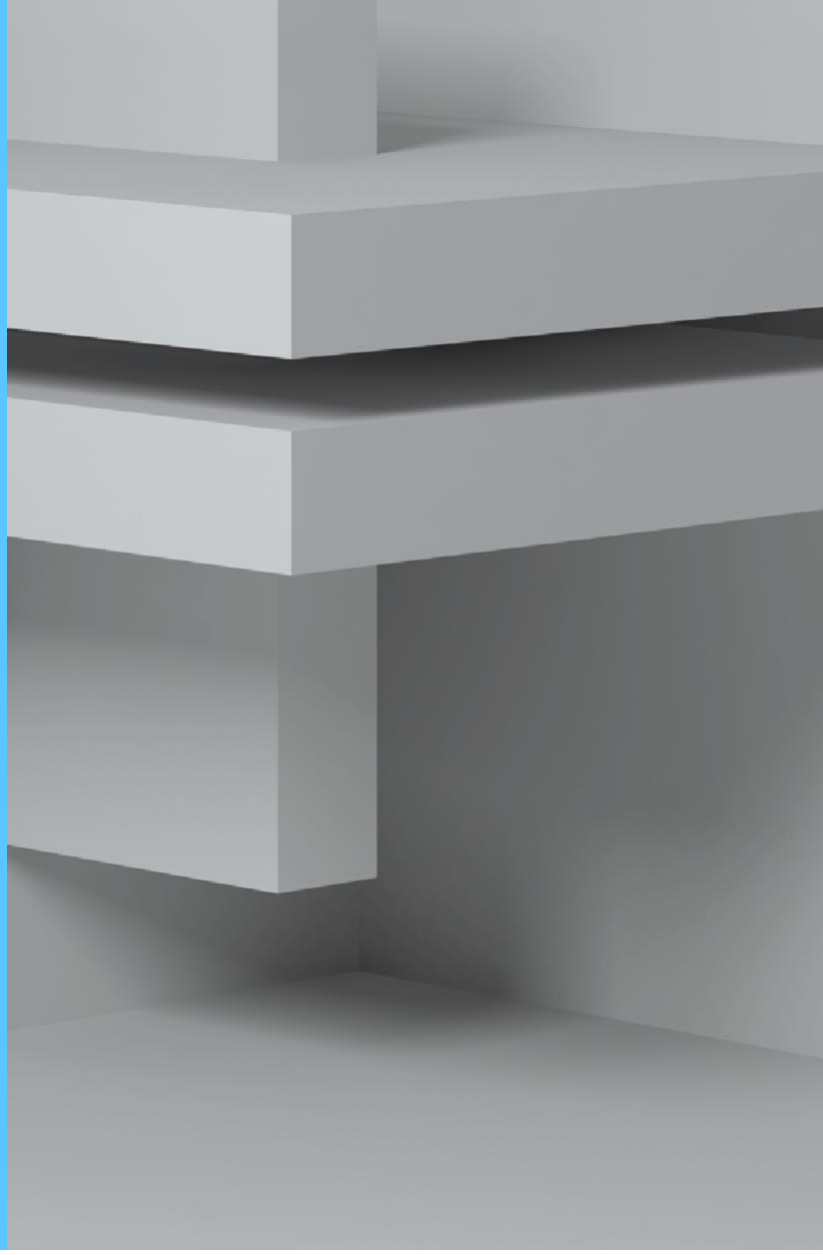
SOLUTION BRIEF

Application-aware data management for DataStax Enterprise with NetApp Astra Control and ONTAP storage

Backup, restore, clone, and DR of Cassandra clusters in OpenShift Kubernetes environment



N DataStax Red Hat

Key benefits

Realize faster time to value with DataStax Enterprise and Apache Cassandra with NetApp Astra Control when delivering business applications.

Clone DataStax Enterprise and Apache Cassandra clusters locally, and even migrate to different geographical locations, allowing for improved application unit and system testing.

Rapid recovery from a disaster, or going back to a point-in-time copy of a DataStax Enterprise or Apache Cassandra cluster.

Executive Summary

This document details the business advantages of application-aware data management solutions (such as backup and recovery, business continuity, and active database cluster cloning) when using NetApp Astra Control Center for containerized DataStax Enterprise and Apache Cassandra database server clusters on Red Hat® OpenShift® Container Platform clusters.

NetApp ONTAP serves as a persistent storage provider for containerized DataStax Enterprise and Apache Cassandra running on the OpenShift Container Platform (OCP). NetApp Astra Control seamlessly extends the data management benefits of ONTAP to data-rich Kubernetes applications such as Apache Cassandra.

Use Cases

Simplified Day One operations: Automatic storage provisioning by using Astra Trident and enabling the freedom to choose the Network Attached Storage protocol in on-premises with Astra Control Center.

Dispelling the network-attached storage (NAS) myth.

Network-attached storage performs at the same level as block storage but offers more features.

Immediate bootstrapping (recovery) of failed data nodes. Data nodes fail; operate a database service long enough and it will happen. With NetApp Data ONTAP NAS based PV and the Kubernetes platform, recovery from these events is nearly instantaneous, with zero or low impact on the production application.

DataStax Enterprise and Apache Cassandra Cluster data management with NetApp Astra Control

As enterprises rely on Cassandra to store and manage their business-critical data, policy-based snapshots and backups become a critical requirement for continued operations. In addition, the ability to clone DataStax Enterprise and Apache Cassandra clusters locally, and even migrate to a different geographical location is a very practical functionality. To improve the effectiveness of application development unit and system tests, we need to test with realistically sized and realistically complex data. DataStax Enterprise and Apache Cassandra cluster cloning is a much-needed testing feature when running atop OCP to accurately master and then replicate golden copies of an entire database server's contents. This capability aids in improved application system unit and system testing.

Rapid recovery from a disaster, or going back to a point-in-time copy of a DataStax Enterprise or Apache Cassandra cluster

Disaster can happen even when running DataStax Enterprise or Apache Cassandra clusters atop OCP. This can be caused by a data center failure or human error. Businesses must continue to run regardless of the situation. NetApp Astra Control enables DataStax Enterprise or Apache Cassandra to recover quickly in these cases by using NetApp Astra Control's application-aware backups that offer execution hooks to enable application consistency.

Database cluster portability is essential in such a situation to meet the business continuity requirements, whether it's on-demand (a special analytics project or need) or in a disaster recovery scenario (an entire site or availability zone fails).

NetApp Astra Control Center solution for DataStax Enterprise and Apache Cassandra offers the following key benefits:

- Automatic storage provisioning from ONTAP storage and storage class setup.
- Rich set of data management services, including data protection, business continuity, and disaster recovery, active cloning, activity log, and more.
- Consistent data management UI.
- Clear visualization of data protection status.
- Simple data protection management with support for application-consistent snapshots and backups
- Seamless cloning and migration.
- Health and performance monitoring of backend storage and applications stack.

About DataStax

DataStax is the leader in scale-out data and the company behind Apache Cassandra. DataStax is committed to Kubernetes as the cloud-native deployment and orchestration technology of choice for modern enterprises. DataStax Enterprise and Apache Cassandra end users benefit from rich data APIs, zero-downtime, and global scale with (Cassandra). When combined with Kubernetes and NetApp storage technology, enterprises can achieve transformational outcomes with the best-of-breed technologies for cloud-native operations. These benefits span across digital transformation initiatives of all types including new application development and application/service portfolio modernization.

DataStax also creates the open source DataStax Kubernetes Operator for Apache Cassandra, referenced in this article. This operator, a Kubernetes CRD, actively configures and manages DataStax Enterprise and Apache Cassandra clusters.

About NetApp Astra Control

NetApp Astra Control is an application-aware data management solution that manages, protects, and moves data-rich Kubernetes workloads in both public clouds and on-premises. Astra Control enables data protection, disaster recovery, and migration for your Kubernetes workloads leveraging NetApp's industry-leading data management technology for snapshots, backups, replication, and cloning. NetApp Astra Control is available in two deployment models:

NetApp Astra Control Service: A NetApp-managed service that provides application-aware data management of Kubernetes clusters in Google Kubernetes Engine (GKE) and Azure Kubernetes Service (AKS).

NetApp Astra Control Center: Self-managed software that provides application-aware data management of Kubernetes clusters running in your on-premises environment. For illustration purposes, this solution brief uses self-managed software to manage data protection needs of Apache Cassandra cluster.

About Red Hat OpenShift Container Platform

The Red Hat OpenShift Container Platform unites development and IT operations on a single platform to build, deploy, and manage applications consistently across on-premises and hybrid cloud infrastructures. Red Hat OpenShift is built on open-source innovation and industry standards, including Kubernetes and Red Hat Enterprise Linux CoreOS, the world's leading enterprise Linux distribution designed for container-based workloads. OpenShift is part of the Cloud Native Computing Foundation (CNCF) Certified Kubernetes program, providing portability and interoperability of container workloads.

Detailed Solution Overview

Solution Configuration

This reference solution is tested with the following configuration:

1. Two OpenShift Container Platform (OCP) clusters with each with following configuration:
2. Red Hat OpenShift 4.7 - 4.8
3. 3 control plane nodes (4 vCPU/16 GB RAM each)
4. 4 worker nodes (12 vCPU/ 32 GB RAM each)
5. Red Hat Enterprise Linux CoreOS 47.83 – 48.84

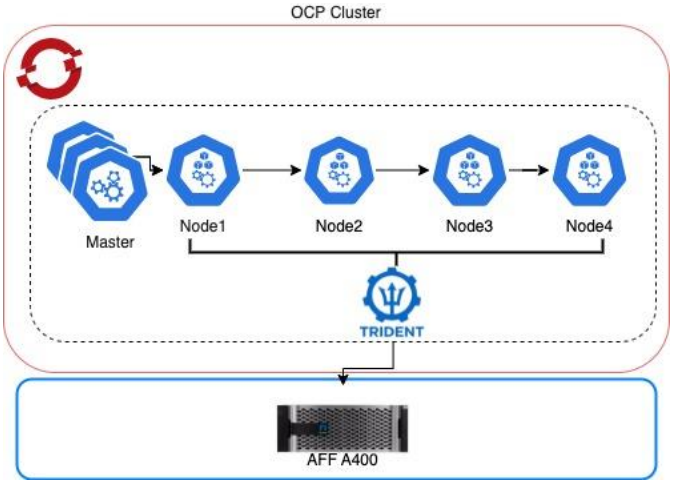


Figure 1 RedHat OpenShift Container Platform cluster config view

Three-node Apache Cassandra Cluster version 3.11.7 is installed with DataStax Kubernetes Operator for Apache Cassandra (available from https://github.com/datastax/cass-operator).

The Apache Cassandra Operator and the Apache Cassandra cluster is installed in a namespace called cass-operator. The Apache Cassandra Kubernetes Operator is deployed as namespace scoped.

The Apache Cassandra cluster nodes (pods) use Persistent Volume Claims (PVC) from a Storage Virtual machine (SVM) on an AFF A400 (ONTAP version 9.8) cluster.

NetApp Astra Trident automatically provisions the Kubernetes persistent volume claims from the SVM using NFS 4.1 for Apache Cassandra.

An object store (S3 bucket) from NetApp Storage Grid configured with Astra Control Center. You can also use ONTAP S3 (ONTAP version 9.8 or higher) or a generic S3 bucket.

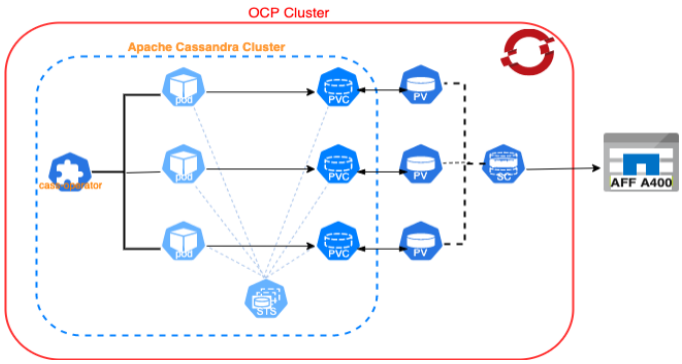


Figure 2 Apache Cassandra Cluster configuration

Simplified Day One operations: Automatic storage provisioning for Cassandra

Three-node Apache Cassandra Cluster version 3.11.7 is installed with DataStax Kubernetes Operator for Apache Cassandra (available from https://github.com/datastax/cass-operator). Tightly and natively integrated atop Kubernetes, DataStax designs and manufactures the open-source DataStax Kubernetes Operator for Apache Cassandra, a Kubernetes CRD. The critical business differentiators provided by Apache Cassandra derive from Cassandra's unique active/active network and process architecture that deliver always-on network partition fault tolerance, localized writes even for globally distributed applications, and more.

As an active/active and distributed operational data platform, the DataStax Kubernetes Operator for Apache Cassandra provides the Kubernetes primitives and Kubernetes Controller Manager with necessary sequencing and intelligent resource management, that deliver an always-on, distributed, secure, and safe enterprise data platform.

Prior to the deployment of the cass-operator, install NetApp Astra Trident on the OCP cluster where Cassandra will be deployed. Astra Trident is an open-source Kubernetes storage provisioner and it is configured to use ONTAP as the storage backend.
As part of Kubernetes cluster registration with Astra Control Center for application-aware data management, Astra Control automatically performs the following actions:

Astra Control Center creates role bindings.
Creates NetApp monitoring namespace to collect metrics and logs from the application pods and worker nodes.
Make one of the supported NetApp ONTAP based storage class as default.

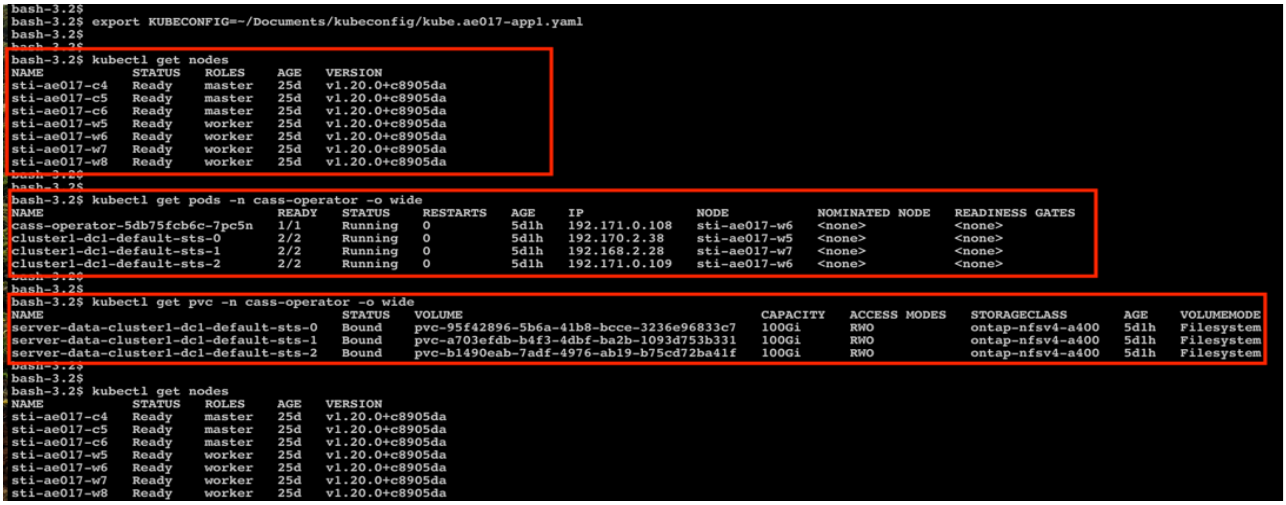


Figure 3 Apache Cassandra Cluster on Kubernetes

Dispelling the network attached storage(NAS) myth

The current version of the NetApp Astra software defaults to using NFS v3 for the storage classes it creates. For this test we created an additional storage class netapp-cvs-extreme-nfsv4 with mount options for NFSv4.1 as follows:

```
parameters:
  backendType: gcp-cvs
  selector:
  serviceLevel=extreme;storageClass=hardware
  provisioner: csi.trident.netapp.io
  reclaimPolicy: Retain
  volumeBindingMode:WaitForFirstConsumer
  mountOptions:
   - nfsvers=4.1
```

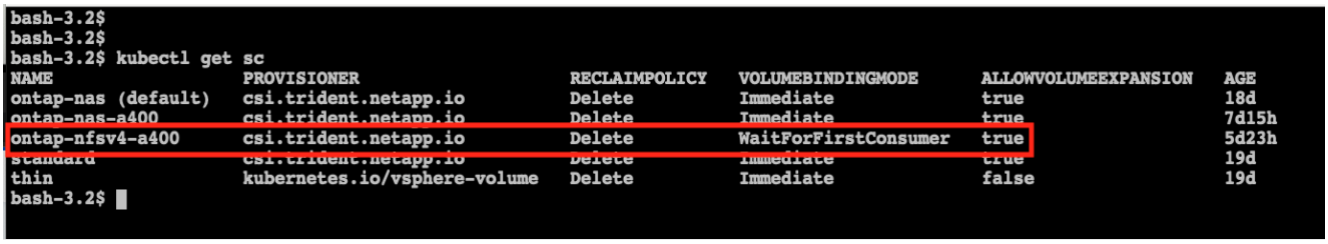


Figure 4 Storage Class used for Apache Cassandra

Persistent Volume Claims (PVCs) requested by Apache Cassandra Cluster nodes (pods) are now served from NetApp AFF A400 cluster (ONTAP version 9.8). The PVCs are using storage class name onatp-nfsv4-a400, which is configured to use nfs version 4.1 as the data protocol.

We used the *cassandra-stress* tool to validate the Cassandra performance and read/write IO flows.

Immediate bootstrapping (recovery) of failed data nodes

Anyone with experience operating database services knows that nodes can and will eventually fail. With NetApp ONTAP FlexVols, recovery from these events is nearly instantaneous, with zero or low impact to the production application.

We powered off one of the worker node sti-ae-017-w7 in OCP cluster sti-ae017-app1 (The Apache Cassandra nodes are running in worker nodes sti-ae-017-w5, sti-ae-017-w7 and sti-ae-017-w8) to simulate failure of an Apache Cassandra cluster node. The OCP scheduler detects the pod failure and automatically schedules the offline Apache Cassandra pod to another worker node sti-ae-017-w6. The Apache Cassandra failed node is brought up on the new worker node and completes the initialization. The new pod (Apache Cassandra node) is bound to the original Kubernetes persistent volume claim. Data created during the event will be resynced from whichever Apache Cassandra nodes in the cluster have the most up-to-date data.

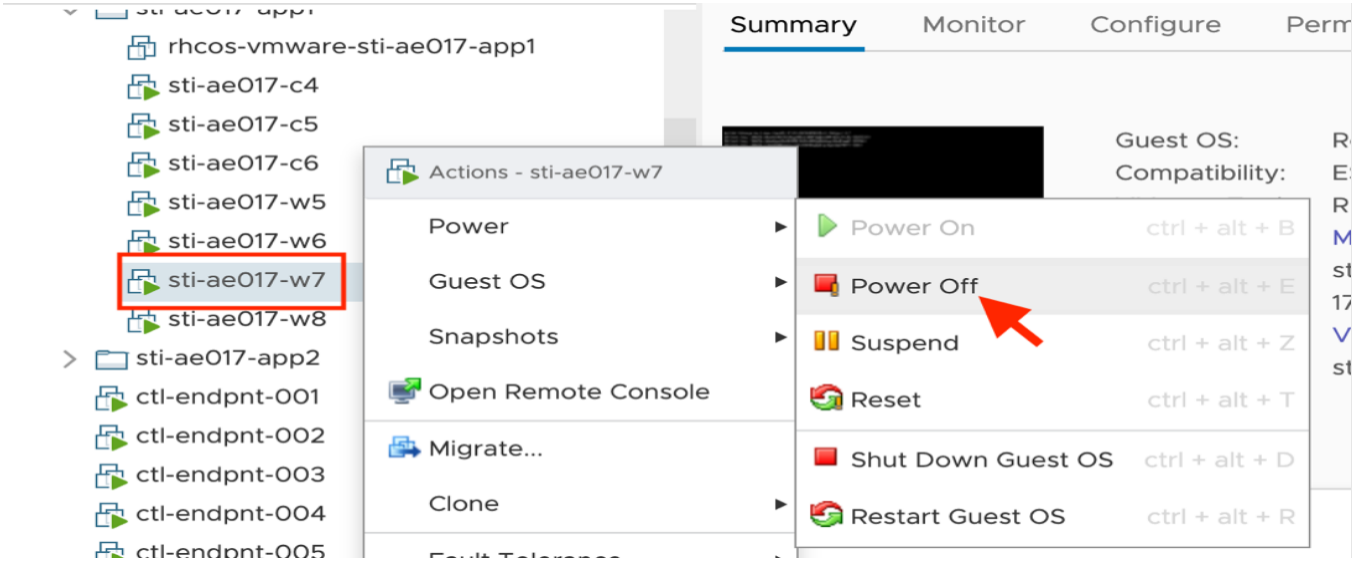


Figure 5 Powering-off a worker node (VM) to simulate Apache Cassandra Cluster node failure

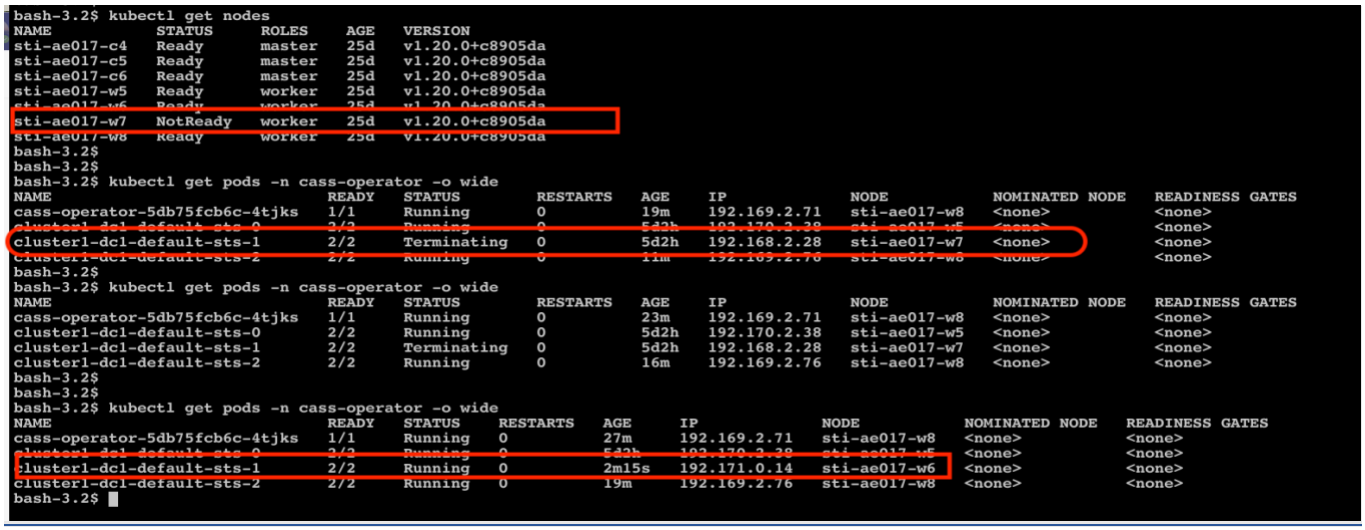


Figure 6 Apache Cassandra Cluster node resurfaces on a different Kubernetes worker node

DataStax Enterprise and Apache Cassandra Data Management with NetApp Astra Control

Apache Cassandra clusters can be installed in any custom namespace. NetApp Astra Control Center discovers the Apache Cassandra kubernetes operator and Apache Cassandra clusters within in the namespaces. As each Apache Cassandra cluster can have multiple pods, (Apache Cassandra cluster nodes), the recommended way to manage the application in NetApp Astra Control Center is by choosing the namespace hosting Apache Cassandra cluster as a management unit. Using NetApp Astra Control Center to manage your application addresses critical Day Two challenges like application-aware snapshots, backups, and disaster recovery.

We managed the OpenShift project cass-operator to leverage the application-aware data management features for Apache Cassandra Cluster on OpenShift Container Platform cluster with NetApp Astra Control:

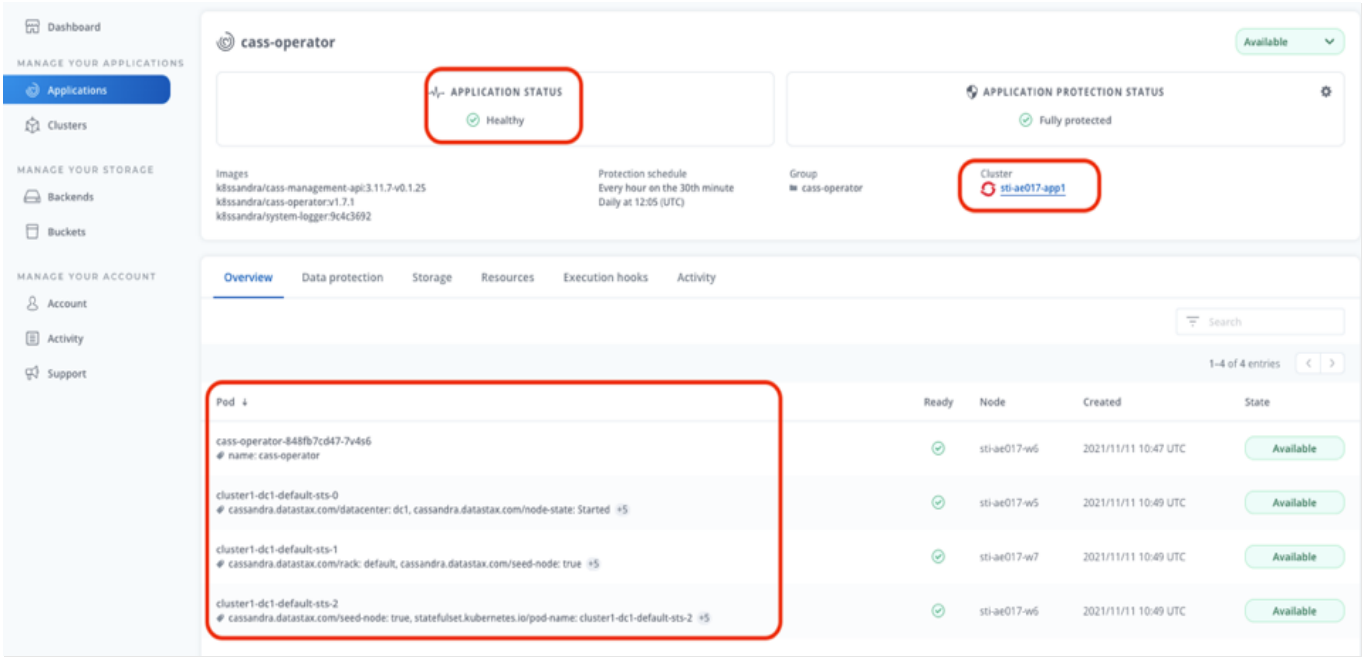


Figure 7 Managed Apache Cassandra Application in NetApp Astra Control

After the Apache Cassandra application is registered as a managed application with NetApp Astra Control, an administrator can take application snapshots, backups, and clones of that application, including its Kubernetes resources and associated Persistent Volumes.

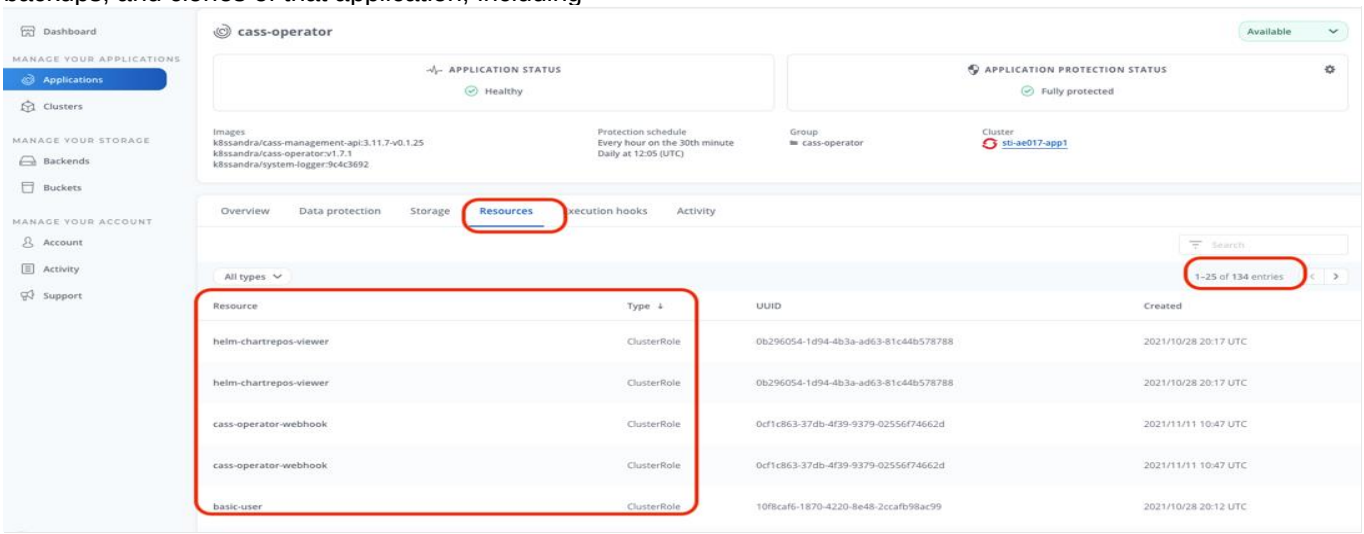


Figure 8 Kubernetes resources of the Apache Cassandra Cluster

Data generated by the DataStax Enterprise, and Apache Cassandra database nodes can be automatically protected with snapshots and backups. NetApp Astra Control creates snapshots and backups that preserve the application state, Kubernetes resources, and its volumes in one easily manageable unit. NetApp Astra Control stores application backups in a bucket on an S3 compatible object store configured for each application.

Databases like Apache Cassandra and other stateful applications benefit from application-consistent snapshots and backups. Pre- and post-snapshot execution hooks in NetApp Astra Control provide the ability to perform application-aware snapshots and backups by running custom scripts to quiesce applications. To create an application consistent state Apache Cassandra must flush the data written to memory to the underlying PVC for each node in the cluster.

Apache Cassandra includes the nodetool utility to flush the data from memtable before taking a snapshot or backup. Running this tool on every node in an Apache Cassandra Cluster before taking a snapshot or backup creates a consistent state across the Cassandra cluster. In the screenshots below, execution hooks in NetApp Astra Control facilitate running nodetool flush -- ks_cluster1 on each Cassandra node at the same time to flush the memtable to PVC before taking a snapshot and backup.

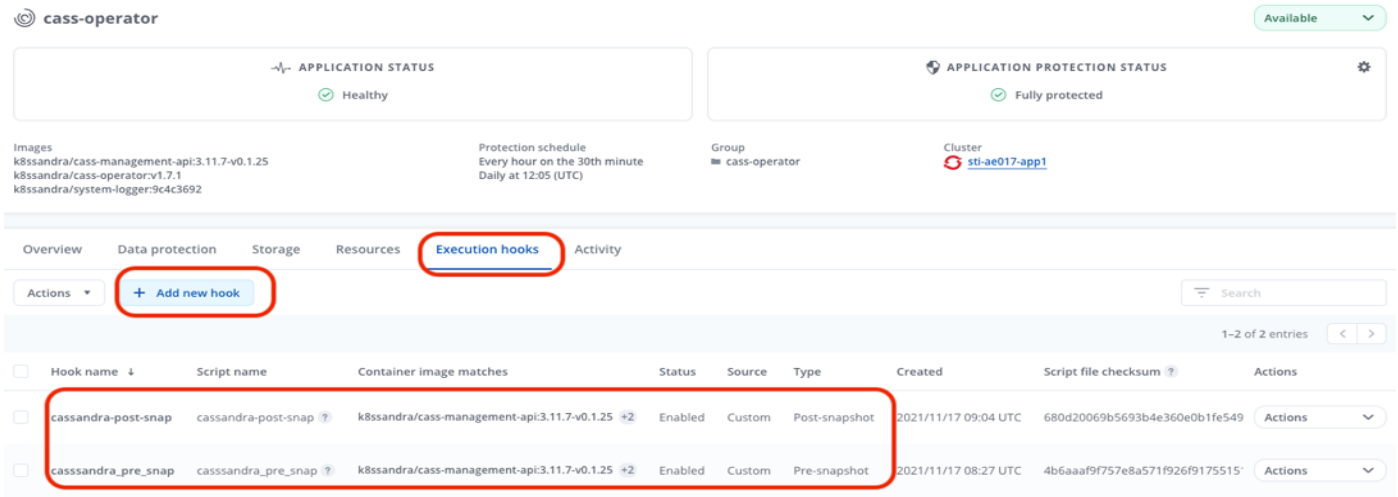


Figure 9 Apache Cassandra execution hooks in NetApp Astra Control

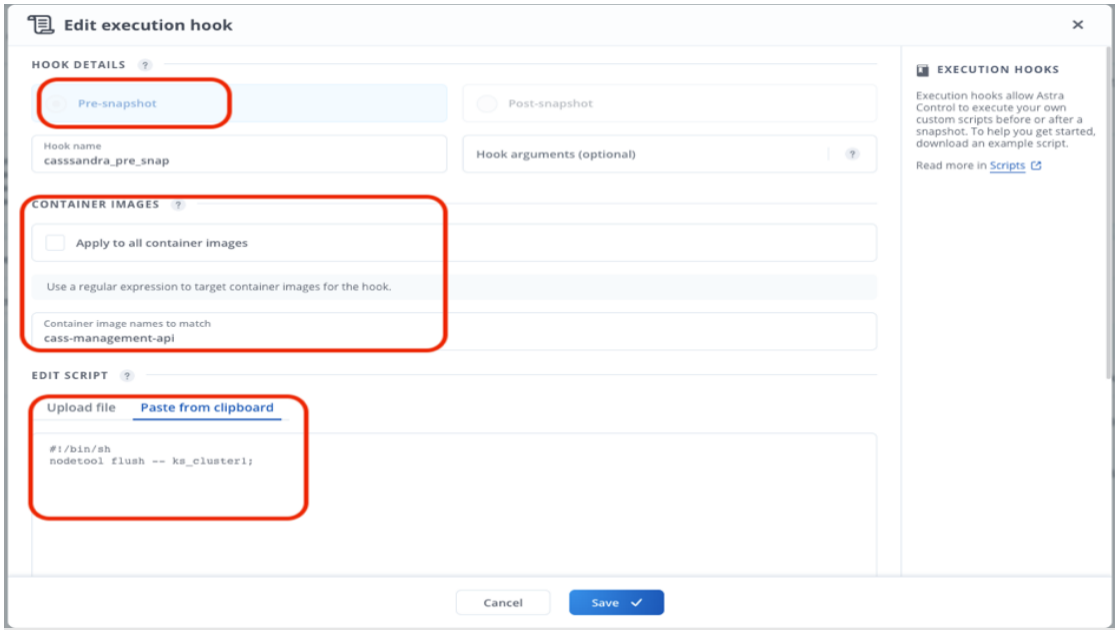


Figure 10 Pre-snapshot execution hook for Apache Cassandra in NetApp Astra Control

NetApp Astra Control supports both on-demand and scheduled snapshots and backups. When using execution hooks, the pre-snapshot script will run first before taking any snapshot or backup to achieve application consistency specific to each application.

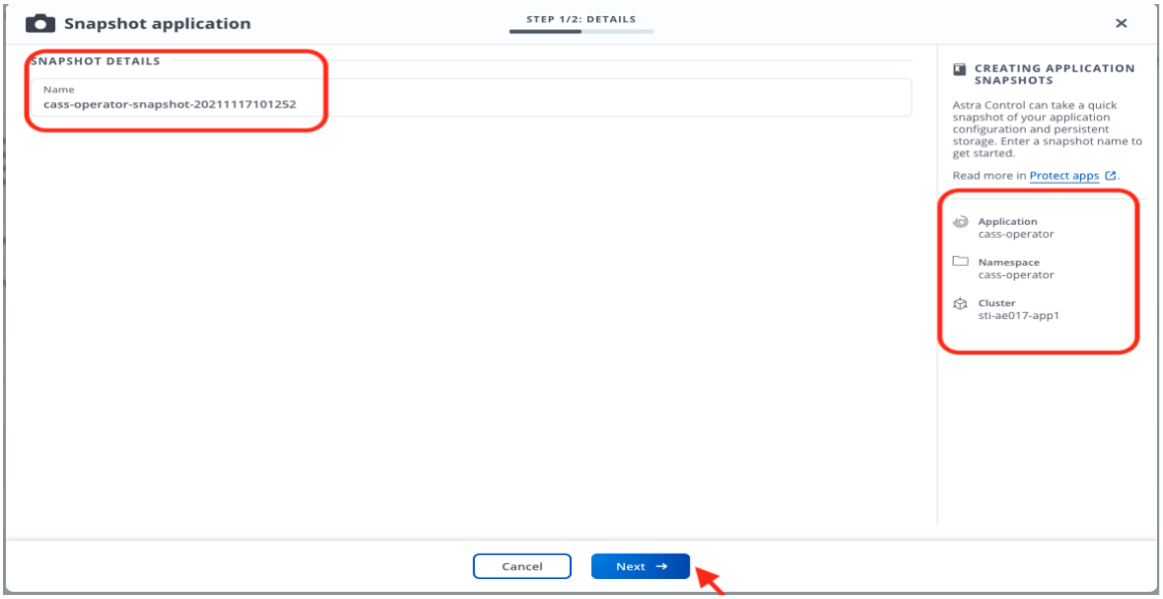


Figure 11 On-demand application snapshot in NetApp Astra Control

When taking on-demand backups, you have the option to create a backup from any existing snapshot OR create from a new snapshot.

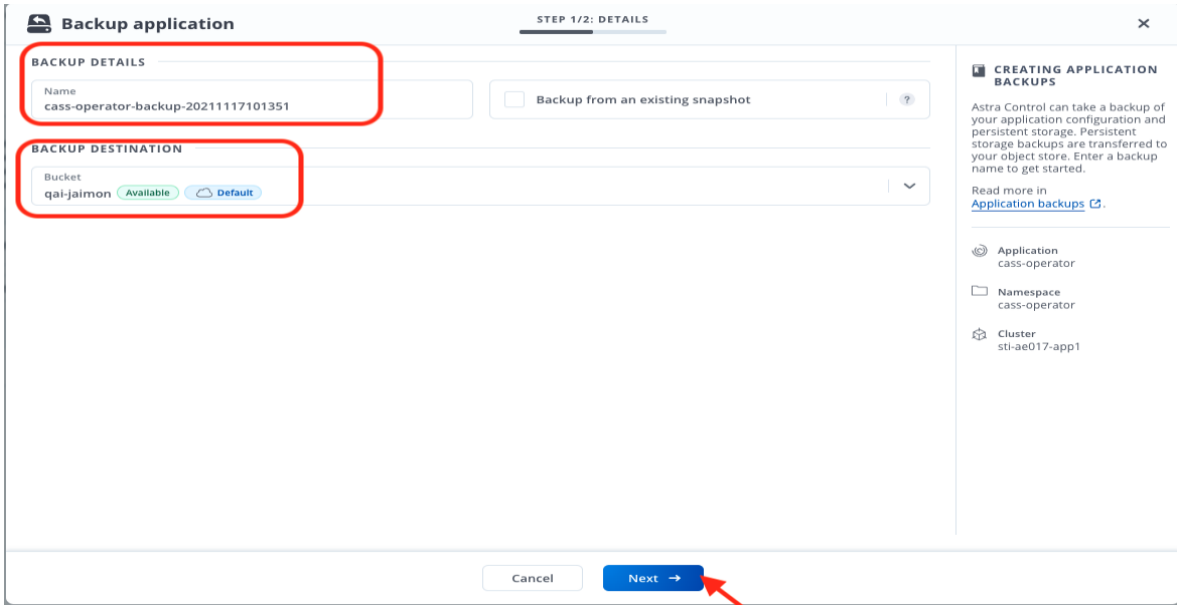


Figure 12 On-demand application backup in NetApp Astra Control

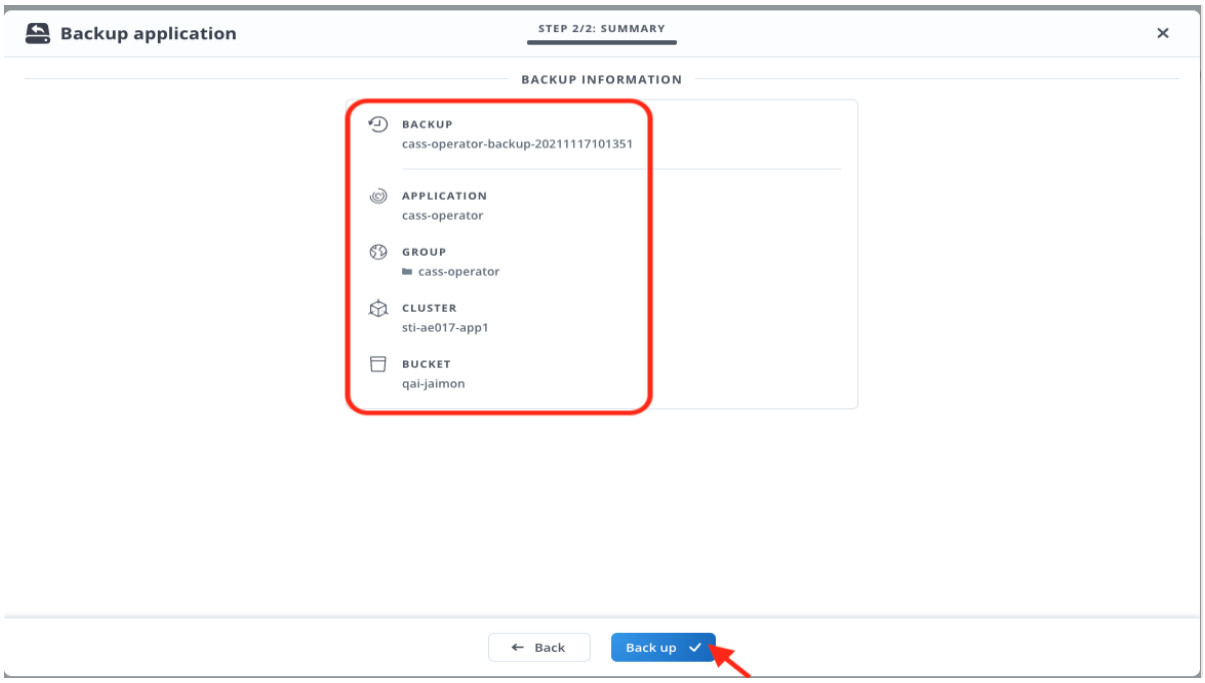


Figure 13 On-demand application backup in NetApp Astra Control

On NetApp Astra Control, snapshot and backup policies can be configured for the Apache Cassandra cluster. This includes selecting the frequency of snapshots/backups, and associated retention units to customize per needs. You can choose a different object store other than the default to store the scheduled application backups.

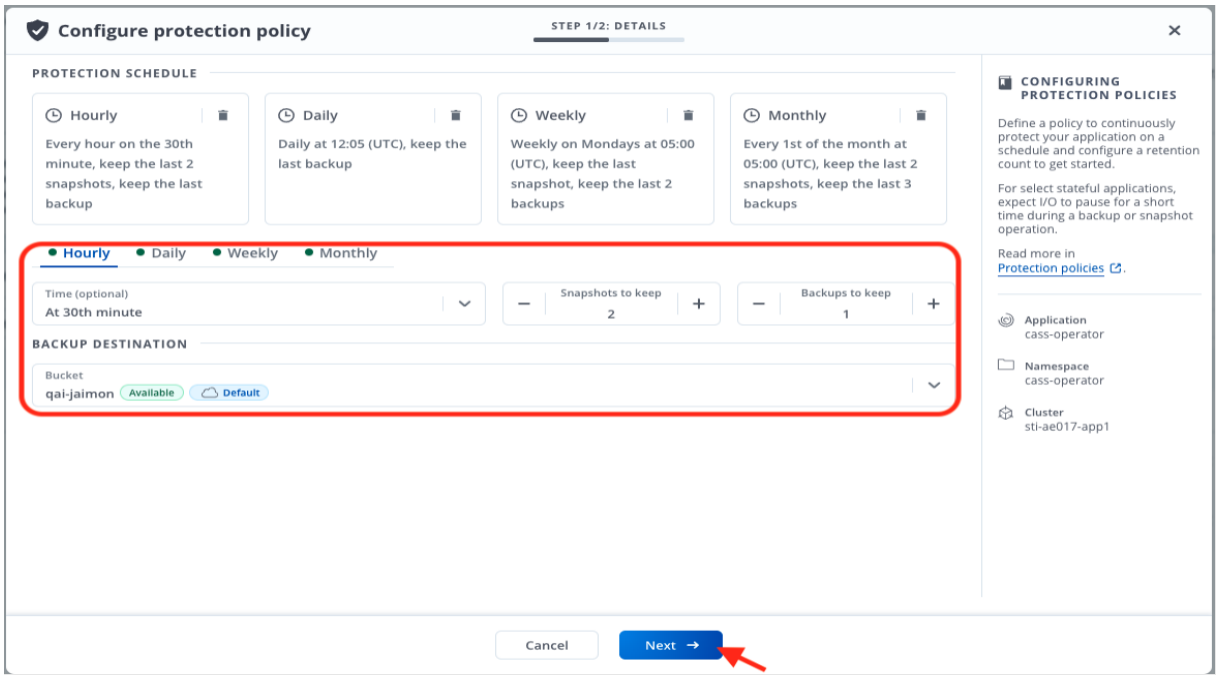


Figure 14 Protection policy in NetApp Astra Control

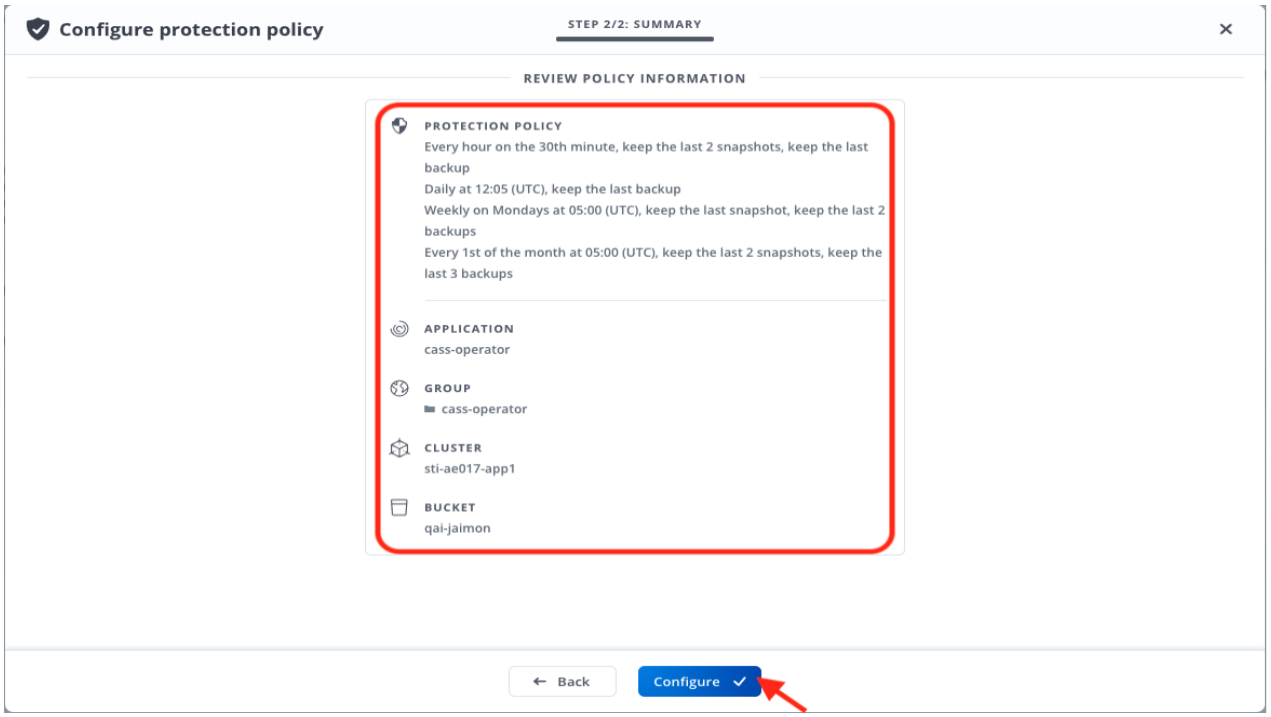


Figure 15 Protection policy in NetApp Astra Control

After reviewing the information, we set the protection policy for Apache Cassandra cluster. NetApp Astra Control now automatically takes snapshots and backups based on the schedule and follows the defined retention policy.

After a successful backup, the Apache Cassandra database cluster is protected against disasters like losing the Kubernetes cluster, or a human error like deleting a namespace or corrupting a database with a bad SQL/CQL (Apache Cassandra) operation.

Cloning Apache Cassandra to another OpenShift namespace within the same OCP cluster

You can use the clone option to redeploy Apache Cassandra to a new namespace, either within the same cluster or in a new cluster.

For example, suppose that your team needs a way to test the production database for a new use case without interrupting the production Apache Cassandra cluster. NetApp Astra Control can clone the Apache Cassandra database cluster to another namespace within the same Kubernetes cluster.

In our setup, Apache Cassandra is running on the OCP cluster sti-ae017-app1. You can use the clone option from the NetApp Astra Control Center UI and either specify new namespace or application name or use the automatic namespace and application names proposed by NetApp Astra Control. You can also select an existing snapshot or backup to go back to a point-in-time copy of the Apache Cassandra application.
We created a new database (keyspace) ks_cluster1 (replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}) table t1 and inserted sample data into the Apache Cassandra database to demonstrate the data management capabilities of NetApp Astra Control.

```
bash-3.2$ kubectl -n cass-operator exec -it cluster1-dc1-default-sts-0 -c cassandra -- sh
$
$
$ cqlsh -u cluster1-superuser -p T8ZcOjeaGFn137BCx2N_gxIzdr2mwYZ2QIfcvV_gyrrDSRoNwhy9YA
Connected to cluster1 at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.7 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cluster1-superuser@cqlsh>
cluster1-superuser@cqlsh> desc keyspaces;

system_schema  system                system_traces
system_auth    system_distributed    ks_cluster1

cluster1-superuser@cqlsh>
cluster1-superuser@cqlsh> use ks_cluster1;
cluster1-superuser@cqlsh:ks_cluster1>
cluster1-superuser@cqlsh:ks_cluster1> select * from t1;

 col1 | col2 | col3 | col4
------+------+------+------
  888 |  888 |  888 |  888
  000 |  000 |  000 |  000
  444 |  444 |  444 |  444
  999 |  999 |  999 |  999
  777 |  777 |  777 |  777
  111 |  111 |  111 |  111
  666 |  666 |  666 |  666
  333 |  333 |  333 |  333
  555 |  555 |  555 |  555
  222 |  222 |  222 |  222

(10 rows)
cluster1-superuser@cqlsh:ks_cluster1>
```

Figure 16 Test database and table view

Using NetApp Astra Control, you can clone and migrate the Apache Cassandra cluster with existing data.

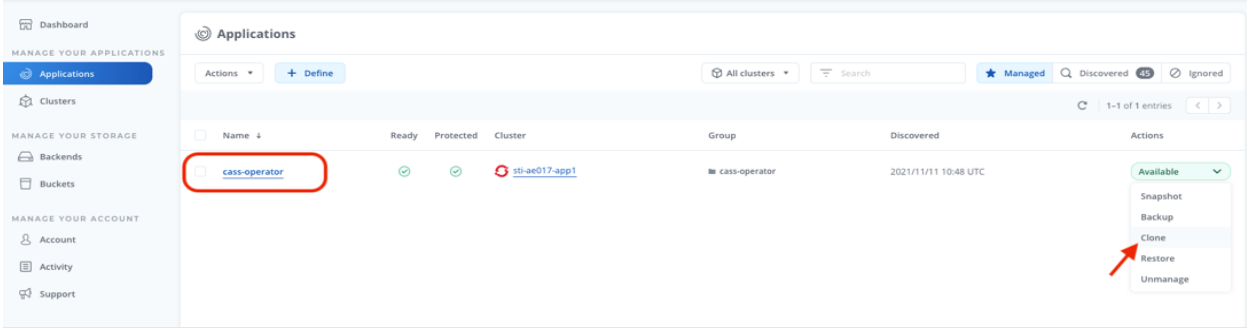


Figure 17 Cloning an application in NetApp Astra Control

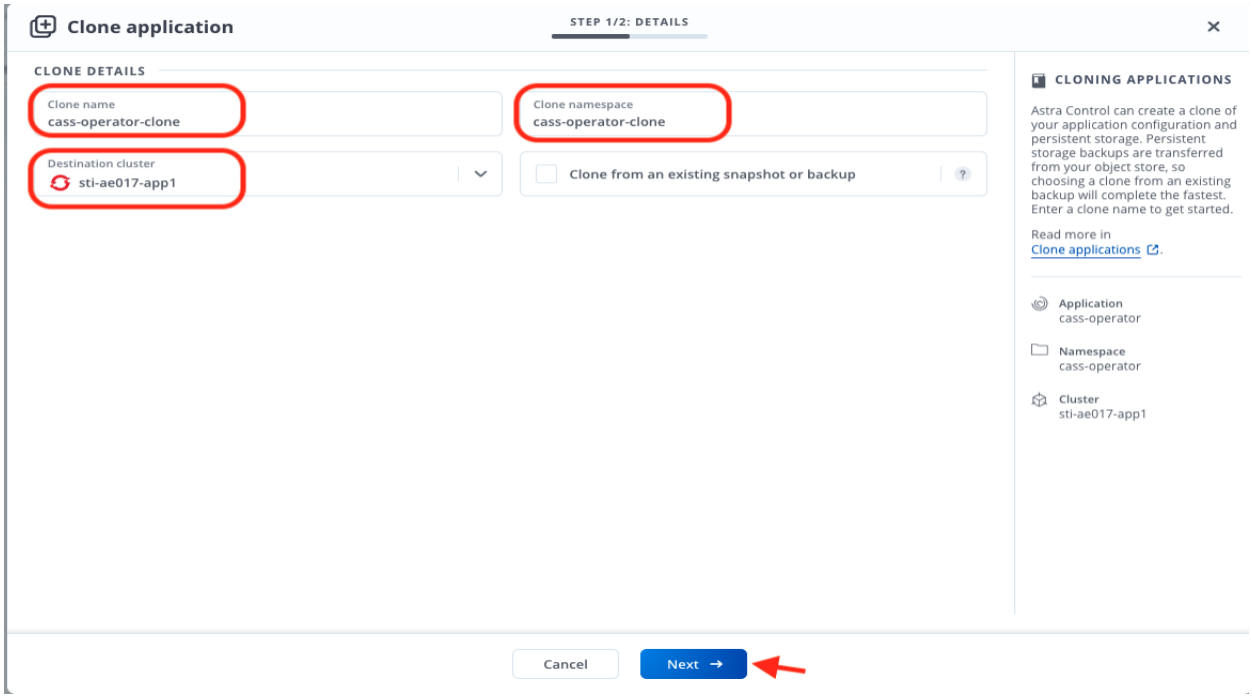


Figure 18 Migrate the clone to a selected namespace in the same cluster

The clone operation is initiated from the current state of the Apache Cassandra cluster. NetApp Astra Control runs the pre-snapshot execution hook script and initiates an application consistent snapshot for the Apache Cassandra cluster. When the snapshot creation is successfully completed, NetApp Astra Control restores the snapshot into the new namespace provided. This operation restores the DataStax Kubernetes Operator and the Apache Cassandra cluster clone on the new namespace. It may take a few minutes before the Apache Cassandra Cluster comes up on the destination namespace.

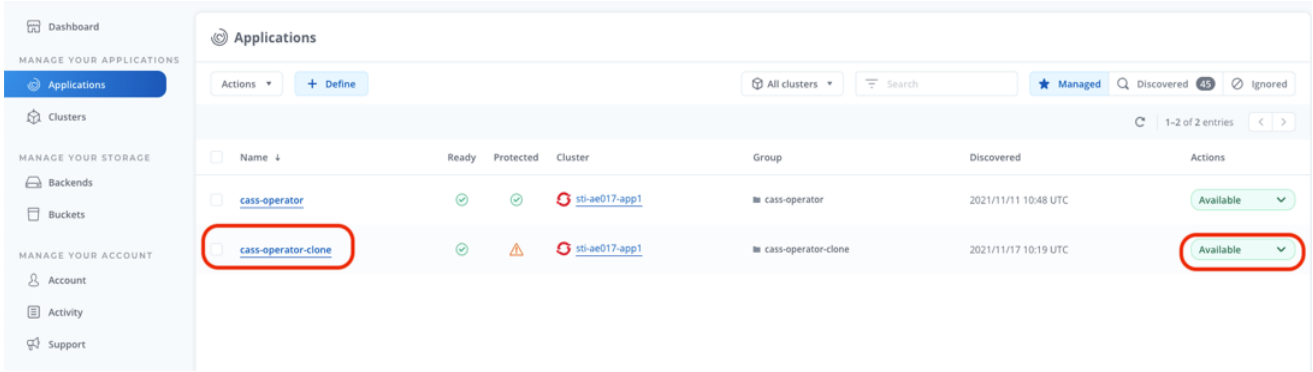


Figure 19 Cloned Apache Cassandra Cluster in a different namespace

Apache Cassandra clusters need to be backed up and restored with the CassandraDatacenter object within Kubernetes. After cloning the Apache Cassandra cluster, the DataStax Kubernetes Operator for Apache Cassandra will perform the necessary actions to verify that the Apache Cassandra database cluster is brought to a fully online and operational state.

```
bash-3.2$
bash-3.2$ kubectl get pods -n cass-operator-clone
NAME                                READY   STATUS    RESTARTS   AGE
cass-operator-6698bd5c47-9gchx      1/1     Running   0          3m45s
cluster1-dc1-default-sts-0          2/2     Running   0          3m17s
cluster1-dc1-default-sts-1          2/2     Running   0          3m17s
cluster1-dc1-default-sts-2          2/2     Running   0          3m17s
bash-3.2$
bash-3.2$ kubectl -n cass-operator-clone exec -it cluster1-dc1-default-sts-0 -c cassandra -- sh
$
$
$ cqlsh -u cluster1-superuser -p T8ZcOjeaGFn137BCx2N_gxIzdr2mwYZ2QIfcvV_gyrrDSRoNwhy9YA
Connected to cluster1 at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.7 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cluster1-superuser@cqlsh>
cluster1-superuser@cqlsh> desc keyspaces;

system_schema  system              system_traces
system_auth    system_distributed  ks_cluster1

cluster1-superuser@cqlsh>
cluster1-superuser@cqlsh> use ks_cluster1;
cluster1-superuser@cqlsh:ks_cluster1>
cluster1-superuser@cqlsh:ks_cluster1> select * from t1;

 col1 | col2 | col3 | col4
------+------+------+------
  888 |  888 |  888 |  888
  000 |  000 |  000 |  000
  444 |  444 |  444 |  444
  999 |  999 |  999 |  999
  777 |  777 |  777 |  777
  111 |  111 |  111 |  111
  666 |  666 |  666 |  666
  333 |  333 |  333 |  333
  555 |  555 |  555 |  555
  222 |  222 |  222 |  222

(10 rows)
cluster1-superuser@cqlsh:ks_cluster1>
```

Figure 20 Validating the data from the cloned Apache Cassandra Cluster

Clone Apache Cassandra cluster to a remote OpenShift Container Platform cluster

In this section, we clone a DataStax Enterprise or Apache Cassandra database server cluster from OCP cluster sti-ae017-app1 to another OCP cluster sti-ae017-ae1. We could also clone from an existing backup or snapshot.

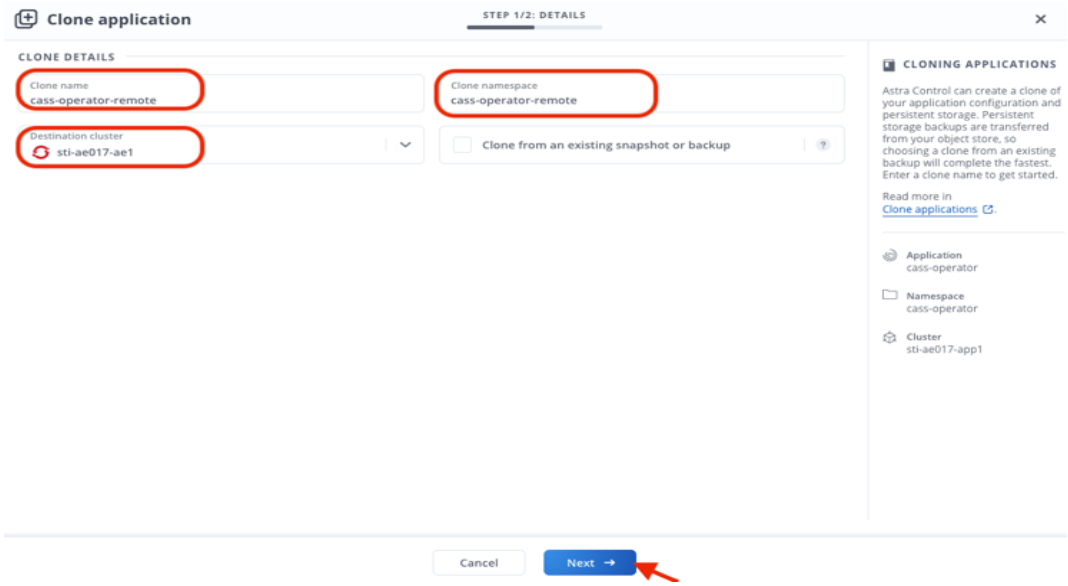


Figure 21 Clone to a different Kubernetes cluster in NetApp Astra Control

When cloning from the current state to a remote Kubernetes cluster, NetApp Astra Control creates an application consistent backup by using the pre and post execution hook scripts and restores the Apache Cassandra cluster to the destination cluster. Astra Control provisions a new Apache Cassandra clone in the destination cluster and automatically manages the application. After the completion, the Apache Cassandra cluster on the destination Kubernetes cluster has the same Kubernetes resources and data as the source cluster.

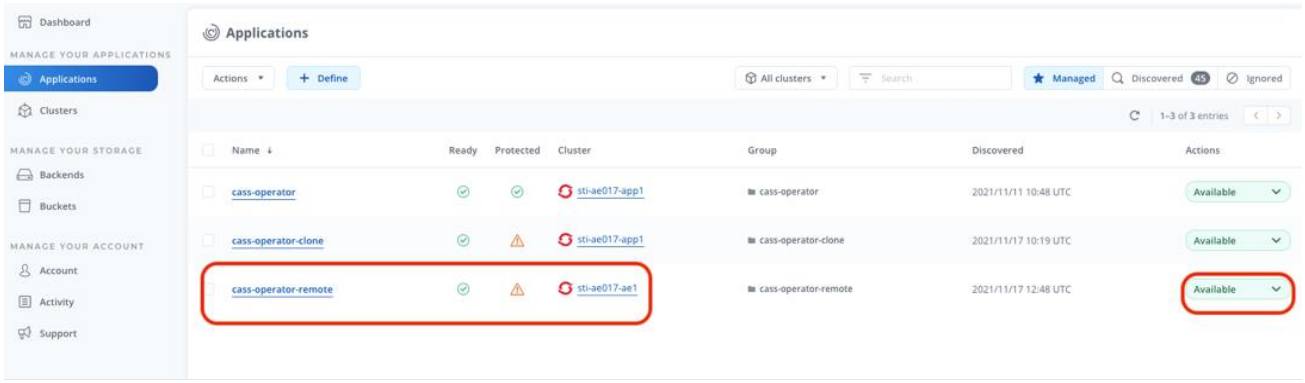


Figure 22 Apache Cassandra Cluster on the destination Kubernetes cluster

We validate that the cloned database and table match the data in the source Cassandra database.

```
bash-3.2$
bash-3.2$
bash-3.2$ kubectl -n cass-operator-remote exec -it cluster1-dc1-default-sts-0 -c cassandra -- sh
$
$ cqlsh -u cluster1-superuser -p T8ZcOjeaGFn137BCx2N_gxIzdr2mwYZ2QIfcvV_gyrrDSRoNWhy9YA
Connected to cluster1 at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.7 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cluster1-superuser@cqlsh> desc keyspaces;

system_schema  system              system_traces
system_auth    system_distributed  ks_cluster1

cluster1-superuser@cqlsh> use ks_cluster1;
cluster1-superuser@cqlsh:ks_cluster1>
cluster1-superuser@cqlsh:ks_cluster1>
cluster1-superuser@cqlsh:ks_cluster1> select * from t1;

 col1 | col2 | col3 | col4
------+------+------+------
  888 |  888 |  888 |  888
  000 |  000 |  000 |  000
  444 |  444 |  444 |  444
  999 |  999 |  999 |  999
  777 |  777 |  777 |  777
  111 |  111 |  111 |  111
  666 |  666 |  666 |  666
  333 |  333 |  333 |  333
  555 |  555 |  555 |  555
  222 |  222 |  222 |  222

(10 rows)
cluster1-superuser@cqlsh:ks_cluster1>
```

Figure 23 Validating data on the cloned Apache Cassandra cluster

Rapid recovery from a disaster or going back to a point-in-time copy of DataStax Enterprise and Apache Cassandra Clusters

Accidently deleting the wrong Kubernetes namespace can bring down the production database cluster and delete all its Kubernetes resources. Having an application-consistent backup, in this case, is significant, and using it to restore will bring the business back to life. NetApp Astra Control provides instantaneous restore of your DataStax Enterprise or Apache Cassandra cluster and application to the same or different namespace in a few clicks.

Earlier in this document, we cloned an Apache Cassandra cluster across OCP clusters. Here we are using a backup within NetApp Astra Control to go back to a point-in-time, application-consistent state.
Use the restore option and choose the most recent backup taken by NetApp Astra Control to redeploy the DataStax Enterprise or Apache Cassandra cluster either to the same namespace or to a new namespace in the original same OCP cluster or to a different cluster.

In this example, we use the most recent backup (cass-operator-x5v2w) to restore Apache Cassandra to the original namespace in the same cluster.

```
bash-3.2$ kubectl delete ns cass-operator
namespace "cass-operator" deleted
```

Figure 24 Simulation of an DR event for Apache Cassandra

NetApp Astra Control automatically detects that the registered cass-operator namespace has been removed.

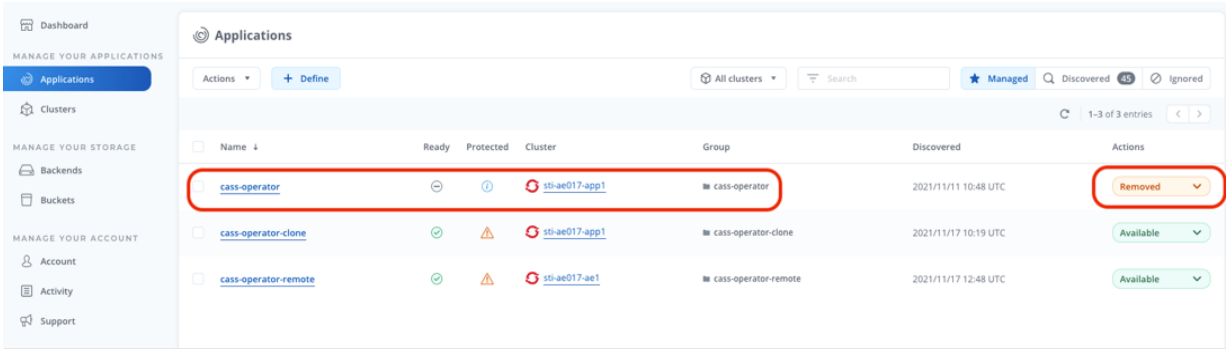


Figure 25 Status update on NetApp Astra Control

You can reinstate the application from a backup to the same namespace.

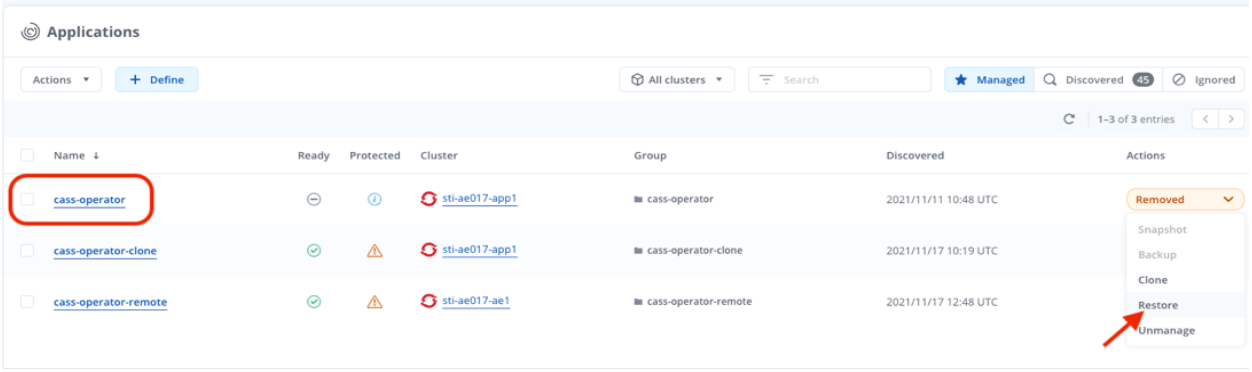


Figure 26 Restoring Apache Cassandra from a backup using NetApp Astra Control

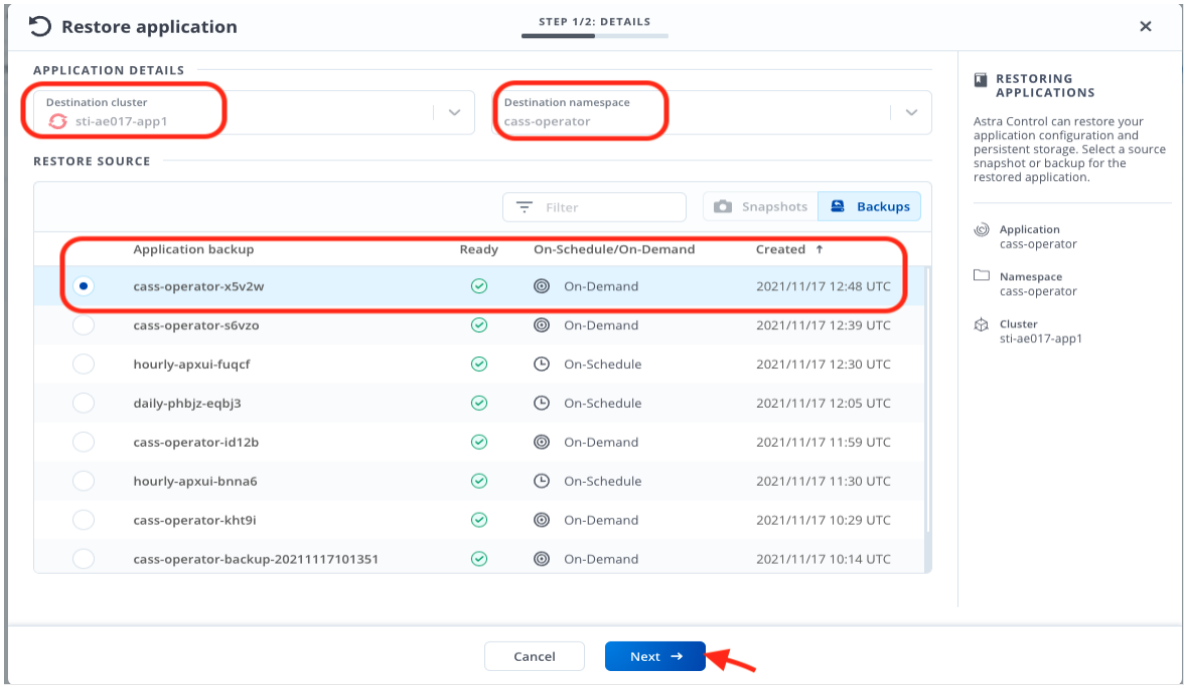


Figure 27 Restoring Apache Cassandra from a backup to the original namespace

We validate that the restored database and table match the data from the latest backup.

```
bash-3.2$ kubectl -n cass-operator exec -it cluster1-dc1-default-sts-0 -c cassandra -- sh
$
$
$ cqlsh -u cluster1-superuser -p T8ZcOjeaGFn137BCx2N_gxIzdr2mwYZ2QIfcvV_gyrrDSRoNwhy9YA
Connected to cluster1 at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.7 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cluster1-superuser@cqlsh>
cluster1-superuser@cqlsh> desc keyspaces;

system_schema  system               system_traces
system_auth    system_distributed   ks_cluster1

cluster1-superuser@cqlsh>
cluster1-superuser@cqlsh> use ks_cluster1;
cluster1-superuser@cqlsh:ks_cluster1>
cluster1-superuser@cqlsh:ks_cluster1> select * from t1;

 col1 | col2 | col3 | col4
------+------+------+------
  888 |  888 |  888 |  888
  000 |  000 |  000 |  000
  444 |  444 |  444 |  444
  999 |  999 |  999 |  999
  777 |  777 |  777 |  777
  111 |  111 |  111 |  111
  666 |  666 |  666 |  666
  333 |  333 |  333 |  333
  555 |  555 |  555 |  555
  222 |  222 |  222 |  222

(10 rows)
cluster1-superuser@cqlsh:ks_cluster1>
```

Figure 28 Validating data on the restored Apache Cassandra cluster

Application-consistent snapshots and backup in NetApp Astra Control helps you to go back to a point-in-time copy of your DataStax Enterprise or Apache Cassandra cluster.

Summary

This solution guide provided a step-by-step guide for validating the following key benefits NetApp Astra provides to DataStax Enterprise and Apache Cassandra:

- Automatic storage provisioning from ONTAP storage array.
- Rich set of application-aware data management functionality (snapshot revert, backup and restore, activity log, and active cloning) for data protection, disaster recovery, data audit, and migration use-cases.
- Consistent data management UI.
- Clear visualization of data protection status.
- Simple data protection management.
- Seamless portability and migration.

Start your free trial of NetApp Astra Control today by registering at https://cloud.netapp.com/astra-register.

Where can I learn more?

To learn more, visit the Astra website and the documentation on Astra Control

About NetApp

In a world full of generalists, NetApp is a specialist. We're focused on one thing, helping your business get the most out of your data. NetApp brings the enterprise-grade data services you rely on into the cloud, and the simple flexibility of cloud into the data center. Our industry-leading solutions work across diverse customer environments and the world's biggest public clouds.

As a cloud-led, data-centric software company, only NetApp can help build your unique data fabric, simplify and connect your cloud, and securely deliver the right data, services, and applications to the right people—anytime, anywhere. www.netapp.com

NetApp