NetBackup™ for Kubernetes Administrator's Guide

Release 10.5

VERITAS™

Last updated: 2024-09-30

Legal Notice

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054

http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

NB.docs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1

Overview of NetBackup for Kubernetes

This chapter includes the following topics:

Overview

The NetBackup web UI provides the capability for backups and restores of Kubernetes applications in the form of namespaces. The protectable assets in the Kubernetes clusters are automatically discovered in the NetBackup environment and administrators can select one or more protection plans that contain the wanted schedule, backup, and retention settings.

The NetBackup web UI lets you perform the following operations:

- Add Kubernetes cluster for protection.
- View discovered namespaces.
- Manage permissions for roles
- Set resource limits to optimize load on your infrastructure and network.
- Manage protection and intelligent group to protect Kubernetes assets.
- Restore namespaces and persistent volumes to same or alternate Kubernetes cluster.
- Monitor backup and restore operations.
- Image expiration, image import, and image copy operations.

Features of NetBackup support for Kubernetes

Table 1-1 NetBackup for Kubernetes

Feature	Description
Auto NetBackup Kubernetes Agent Configuration	Adds Kubernetes cluster and configurations such as storage class and volume snapshot class, and data mover configuration can be done with automated deployment supported.
Integration with NetBackup role-based access control (RBAC)	The NetBackup web UI provides RBAC roles to control which NetBackup users can manage Kubernetes operations in NetBackup. The user does not need to be a NetBackup administrator to manage Kubernetes operations.
Licensing	Capacity-based licensing.
Protection plans	The following benefits are included: ■ Use a single protection plan to protect multiple Kubernetes namespaces. The assets can be spread over multiple clusters. ■ You are not required to know the Kubernetes commands to protect the Kubernetes assets.
Intelligent management of Kubernetes assets	NetBackup automatically discovers the namespaces, persistent volumes, persistent volume claims, and so on, in the Kubernetes clusters. You can also perform manual discovery. After the assets are discovered, the Kubernetes workload administrator can select one or more protection plans to protect them.
Kubernetes specific credentials	Kubernetes service accounts used to authenticate and manage the clusters.
Discovery ■ Full discovery ■ Incremental discovery	Discovery using **Discover now** option is always a full discovery. Discovery when a new cluster is added to the NetBackup is always a full discovery. Once the Kubernetes cluster is added, auto discovery cycle is triggered to discover all the assets available on the Kubernetes cluster. The first auto discovery of the day is a full discovery and subsequent auto discoveries are incremental.
Backup features ■ Snapshot only backups ■ Backup from snapshot	The following features are available for backup: ■ Backups are managed entirely by the NetBackup server from a central location. Administrators can schedule automatic, unattended backups for namespaces on different Kubernetes clusters. ■ The NetBackup web UI supports backup and restore of namespaces from one interface. ■ Backup schedule configuration for full backups. ■ Manual backups and snapshot only backups. ■ Resource throttling for each cluster to improve the performance of backups. ■ NetBackup can perform backups of Kubernetes namespaces with snapshot methodology, achieving faster recovery time objectives.

Table 1-1 NetBackup for Kubernetes *(continued)*

Feature	Description
Restore features ■ Restore from snapshot ■ Restore from backup copy	The following features are available for restore: ■ Restore Kubernetes namespaces and persistent volumes to different locations. ■ Restore to a different Kubernetes cluster flavor using restore from backup copy with parallel restore jobs.
Client side data deduplication support	Client side data deduplication support feature is enabled for Kubernetes. For more details, refer to the *About client-side deduplication* section in the *NetBackup Deduplication Guide*.
Auto Image Replication (AIR)	The backups that are generated in one NetBackup Kubernetes cluster can be replicated to storage in one or more target NetBackup domains. This also is referred to as AIR. The ability to replicate backups to storage in other NetBackup domains. The Auto Image Replication (A.I.R.) is supported for all schedule types.
Protection of Stateful applications	Kubernetes application using persistent volumes to maintain there states can be protected. Backup and restore of Persistent Volume Claims (PVCs) of mode file system and/or block for the Container Storage Interface (CSI) providers which supports the following features: ■ PVC snapshot capability ■ PVC volume provisioning based on the Network File System (NFS) or other non-block storage ■ Backup and restore of a namespace with mixed volumes (VolumeMode: Filesystem and Block) is supported for NetBackup 10.3 and later.
Import and verify	Import is a two step operation, the first step recreates the catalog entries for the backups that are on the specified media. Once the second phase import has been completed catalog entries for files were backed up by those images will be created. Verify: NetBackup can verify the contents of a backup by comparing its contents to what is recorded in the NetBackup catalog.
Federal Information Processing Standards (FIPS) support for Red Hat platforms	NetBackup Kubernetes on Red Hat platform provides support to FIPS compliant communication.
Accelerator backup support for Kubernetes	NetBackup supports accelerator backup for Kubernetes workloads and it reduces the backup time.
OpenShift Virtualization support for Kubernetes workload	NetBackup version 10.4.1 and later, provides backup and restore support for namespaces with one or more virtual machines running on Kubernetes clusters.

Chapter 2

Deploying and configuring the NetBackup Kubernetes operator

This chapter includes the following topics:

Prerequisites for NetBackup Kubernetes Operator deployment

Before deploying the NetBackup Kubernetes Operator, you must install the Helm chart and provide space for persistent volume.

To install the latest Helm version, run the following commands:

1. `$ curl -fsSL -o get_helm.sh` https://raw.githubusercontent.com/helm
2. `$ chmod 700 get_helm.sh`
3. `$ ./get_helm.sh`

Note: You must deploy the operator in each cluster, where you want to deploy NetBackup.

To install a new Helm chart

1 To list all the helm charts in a namespace, run the command:

```
- helm list -n <namespace>
```

2 To uninstall an older plug-in, run the command:

- helm uninstall `<plugin-name> -n <namespace>`

3 To install a new plug-in, run the command:

- helm install `<plugin-name> <chart-path> -n <namespace>`

Here is the Helm chart and tree structure layout:

```
netbackupkops-helm-chart/
├── charts
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   └── _helpers.tpl
└── values.yaml
```

Directory structure:

```
tar --list -f netbackupkops-10.3.tar.gz
veritas_license.txt
netbackupkops.tar
netbackupkops-helm-chart/
netbackupkops-helm-chart/Chart.yaml
```

```
netbackupkops-helm-chart/values.yaml
netbackupkops-helm-chart/.helmignore
netbackupkops-helm-chart/templates/
netbackupkops-helm-chart/templates/deployment.yaml
netbackupkops-helm-chart/templates/_helpers.tpl
netbackupkops-helm-chart/charts/
```

Deploy service package on NetBackup Kubernetes operator

Configuring the Helm chart

You can use the Helm chart to deploy the NetBackup Kubernetes operator.

You must upgrade a helm chart to upgrade NetBackup Kubernetes operator.

Note: Before installing a new plug-in, you must uninstall the older plug-in.

To deploy NetBackup Kubernetes operator:

1. Download the tar package from Veritas Support website: https://www.veritas.com/content/support
2. Extract the package to the home directory. The `netbackupkops-helm-chart` folder should be in the home directory.
3. To list all cluster contexts, run the command: `kubectl config get-contexts`
4. To switch to the cluster where you want to deploy the operator service, run the command:

   ```
   kubectl config use-context <cluster-context-name>
   ```

5. To change the current directory to your home directory, run the command: `cd ~`
6. NetBackup supports any Container Image Repositories complied to OCI standards. you can use any tools to push the operators and data mover images.

   If you use a private docker registry, follow the instructions in this step to create a secret `nb-docker-cred` in NetBackup namespace. Otherwise, skip to the next step.

   - To log on to the private docker registry, run the command: `docker login -u <user name><repo-name>`

After log in, the `config.json` file containing the authorization token is created or updated. To view the `config.json` file, run the command: `cat ~/.docker/config.json`
The output looks like:

```
{
   "auths": {
       "https://index.docker.io/v1/": {
           "auth": "c3R...zE2"
       }
   }
}
```

- To create a secret named as `netbackupkops-docker-cred` in the NetBackup namespace, run the command:
  ```
  kubectl create secret generic netbackupkops-docker-cred \
  --from-file=.dockerconfigjson=.docker/config.json \
  --type=kubernetes.io/dockerconfigjson -n netbackup
  ```
  You can provide any namespace to create a secret.
- To check if the secret `netbackupkops-docker-cred` is created in the NetBackup namespace, run the command:
  ```
  kubectl get secrets -n netbackup
  ```
- To load the image to the docker cache and push the image to the docker image repository, run the commands:
  - Load the tar file for Netbackup Kubernetes Operator.
    ```
    <docker load -i <nameof the tar file> ./>
    ```
  - Tag the loaded docker image as per requirement.
    ```
    docker tag <imagename:tagof the loadedimage>
    <repo-name/image-name:tag-name>
    ```
  - Push the image to a repository from where Kubernetes can fetch the image at the time of NetBackup Kubernetes Operator deployment.
    ```
    docker push <repo-name/image-name:tag-name>
    ```

Note: In the example **Docker** is used for reference. You can use any other CLI tool which provides equivalent functionality.

---

7 Edit the netbackupkops-helm-chart/values.yaml in a text editor,

- replace the value for image in the manager section, with your image name and tag repo-name/image-name:tag-name.
- Change the value of replicas to 0.

Note: Setting replicas to 0 as we are following manual steps to configure the Netbackup Kubernetes Operator.

8 Sizing for metadata persistent volume is required. The default persistent volume size for Kubernetes operator is 10Gi. The persistent volume size is configurable.

You can change the value for storage from 10Gi to a higher value before deploying the plugin. This leads to the nbukops pod have the size of the PVC mounted in the pod.

You can specify metadata persistent volume size in values.yaml.

Persistent Volume Claim in deployment.yaml under helm-chart looks like this :

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  labels:
    component: netbackup
  name: {{ .Release.Namespace }}-netbackupkops
  namespace: {{ .Release.Namespace }}
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

- During fresh installation while configuring the Helm Chart. You can modify the size of PVC storage in the deployment.yaml of the netbackupkops-helm-chart which leads to creation of the initial PVC size.
- Post installation, updating the PVC size (dynamic volume expansion) is supported by few storage vendors. For more information, refer to https://kubernetes.io/docs/concepts/storage/persistent-volumes

Note: The default size of persistent volume can be resized to larger value without losing the data. You are recommended to add the storage provider that supports volume expansion.

9 To deploy the NetBackup Kubernetes operator service, run the command:

```
helm install <release name of the deployment>
./netbackupkops-helm-chart -n <namespace which runs NetBackup
operator service>
```

Example: `helm install veritas-netbackupkops ./netbackupkops-helm-chart -n netbackup`

- You can change the release name of the deployment as required.
- The `-n` option is required to specify the namespace in which NetBackup operator service and NetBackup is intended to run.

10 To check the status of the deployment, run the command:

```
helm list -n <namespace which runs NetBackup operator service >
```

Example:

```
helm list -n netbackup
```

11 To check the release history, run the command:

```
helm history veritas-netbackupkops -n
<namespace which runs NetBackup operator service>.
```

Example:

```
helm history veritas-netbackupkops -n netbackup
```

Port requirements for Kubernetes operator deployment

Following table shows the port requirements for the Kubernetes operator deployment. If firewall exists between the various hosts, you must open the required communication ports.

Table 2-1 Ports that must be open in a NetBackup Kubernetes cluster environment

Source	Port number	Destination
Primary server	TCP port 443	Kubernetes cluster
Media server	TCP port 443 (new in NetBackup 10.0).	Kubernetes cluster

Table 2-1 Ports that must be open in a NetBackup Kubernetes cluster environment *(continued)*

Source	Port number	Destination
Note: Review the Kubernetes configuration to ensure that the Kubernetes API server port has not been changed from 443 to a non-default port; often 6443 or 8443.		
Kubernetes cluster	TCP port 443 (applicable in NetBackup version 9.1, but not in version 10.0 or later).	Primary server
Note: NetBackup Kubernetes Operator (KOps) and datamover pods have additional requirements (new in NetBackup 10.0).		
Kubernetes cluster	TCP port 1556 outbound	Primary server
Kubernetes cluster	TCP port 1556 outbound	Media server
Kubernetes cluster	TCP port 13724 bi-directional if using Resilient Network.	Primary and media server

Upgrade the NetBackup Kubernetes operator

You can upgrade the NetBackup Kubernetes operator deployment using Helm commands.

Example:

```
helm upgrade veritas-netbackupkops ./netbackupkops-helm-chart -n
netbackup
```

Add note to backup configmap values if they are changed. Upgrade resets the helm values to default. Old configmap must be patched again after upgrade.

Important notes

- All components (NBU Primary, Media, Kubernetes operators, and Data mover) must be same version.
- Existing policies continue to take backups but must be restored manually until the Kubernetes operator is updated.

Note: This is applicable to the NetBackup version 9.1 to 10.x upgrade

To upgrade the NetBackup Kubernetes operator

1. Download the tar package from Veritas Support website: https://www.veritas.com/support
2. Extract the package to the home directory. The `netbackupkops-helm-chart` folder must be in the home directory.
3. To list all cluster contexts, run the command: `kubectl config get-contexts`
4. To switch to the cluster where you want to deploy the operator service, run the command:`kubectl config use-context <cluster-context-name>`
5. To change the current directory to your home directory, run the command: `cd ~`
6. NetBackup supports any Container Image Repositories complied to OCI standards. you can use any tools to push the operators and data mover images. If you use a private docker registry, follow the instructions in this step to create a secret `nb-docker-cred` in NetBackup namespace. Otherwise, skip to the next step.
   - To load the image to the docker cache and push the image to the docker image repository, run the commands:
     - Load the tar file for Netbackup Kubernetes operator.
       `<docker load -i <nameof the tar file> ./>`
     - Tag the loaded docker image as per requirement.
       `docker tag <imagename:tagof the loadedimage> <repo-name/image-name:tag-name>`
     - Push the image to a repository from where Kubernetes can fetch the image at the time of NetBackup Kubernetes operator deployment.
       `docker push <repo-name/image-name:tag-name>`

   Note: In the example, a docker is used for reference. You can use any other CLI tool that provides equivalent functionality.

7. Edit the `netbackupkops-helm-chart/values.yaml` in a text editor:
   - Replace the image value in the manager section with your image name and tag in the format `reponame/image-name:tag-name`.

- Replace the datamover image in the `netbackup_config_pod` section with datamover image name and tag.

8 To upgrade the NetBackup Kubernetes operator, run the command:

```
helm upgrade <plugin-name> <chart-path> -n <namespace>
```

Example:

```
helm upgrade veritas-netbackupkops ./netbackupkops-helm-chart -n
netbackup
```

Note: Upgrading the NetBackup Kubernetes operator will reset the Helm values to their defaults. Ensure that you back up the old configmap and reapply any patches if the values change after the upgrade.

Delete the NetBackup Kubernetes operator

You can delete a NetBackup Kubernetes operator deployment from a cluster.

```
helm uninstall <plugin-name> -n <Netbackup Kubernetes Operator
Namespace>
```

Note: By uninstalling the plugin, the NetBackup Kubernetes operator PVC is also deleted which has metadata elated to snapshot based backups.

NetBackup Kubernetes operator deletion can result loss of metadata volume, which also hosts the snapshot metadata. If any snapshots are already performed, then restore from snapshot copy operation fails in the absence of metadata.

In NetBackup 9.1, you must first delete the older snapshots manually then delete the associated Velero snapshots.

In NetBackup 10.0, you cannot perform expiration of Velero managed snapshots which were created using NetBackup 9.1. When the backup images are expired in NetBackup, the catalog is automatically cleared. But you must delete the snapshot on Kubernetes server manually.

For more details on manual image expiration operation, see https://www.veritas.com/content/support.

Note: Without expiring the snapshots or without deleting persistent volume snapshots, if Kubernetes Operator is uninstalled then orphan volume snapshot will be lying around Kubernetes Cluster.

Configure NetBackup Kubernetes data mover

You need to configure data mover for the NetBackup Kubernetes workload. Download the correct version of the data mover image:

You need to configure Netbackup Kubernetes Operator namespace to support Backup from Snapshot and Restore from Backup. (Backup copy). Download the correct version of the data mover image:`veritasnetbackup-datamover-10.3.tar`for your release version, from the download center. See https://www.veritas.com/content/support

To configure data mover

1. To push the data mover image to image registry, run the command:

   ```
   docker login -u <user name> <repo-name>
   ```

2. Enter the password upon prompt. Skip this step if you are already logged in
3. Run `docker load -i <name of the datamover image file>`
4. Run `docker tag <datamover image name:tag of the loaded datamover image> <repo-name/image-name:tag-name>`
5. `docker push <repo-name/image-name:tag-name>`

   Note: In the example, docker is used for reference. You can use any CLI tool which has equivalent capabilities.

6. Ensure that the configmap with primary server name, have image value set to *<repo-name/image-name:tag-name>* pushed in step no 4.

   Example,

   ```
   apiVersion: v1
   data:
     datamover.properties: image=<image-repo>/datamover:<datamover tag>
     version: "1"
   kind: ConfigMap
   metadata:
     name: <Primary Server Name>
     namespace: <Netbackup Kubernetes Operator Namespace Name>
   ```

For more details on Configmap, refer to the *Kubernetes operators supported configuration parameters* section in the *NetBackup for Kubernetes Administrator's Guide*..

Automated configuration of NetBackup protection for Kubernetes

Pre-requisites

Before configuring the NetBackup on the Kubernetes workload, you must run a NetBackup server with access to ports 443, 1556, and 13724.

NetBackup Kubernetes operator and data mover images must be uploaded to container registry accessible from the Kubernetes cluster.

You need to create a secret to be consumed for automated deployment.

To create an API key

1. Open the NetBackup web UI.
2. On the left, click **Security > Access keys**.
3. Click the **API keys** tab.
4. Click **Add**.
5. On the Kubernetes cluster, create a new secret, **nb-config-deploy-secret.yaml,** with the following content.

```
apiVersion: v1
kind: Secret
metadata:
  name: <kops-namespace>-nb-config-deploy-secret
  namespace: <kops-namespace>
type: Opaque
stringData:
  apikey: <Enter the value of API key from the earlier step>
```

6. Apply the secret. Run the command `kubectl apply -f nb-config-deploy-secret.yaml`.

Preinstallation

1. Edit the following fields in `netbackupkops-helm-chart/values.yaml`.
   - containers.manager.image: Container registry URL for pulling the NetBackup Kubernetes controller image.
   - imagePullSecrets name: name of the image pull secret if the container registry requires authentication to pull images.
   - nbprimaryserver: Configured name of NetBackup primary server.

- nbsha256fingerprint: Fetch sha256 fingerprint from the NetBackup web UI. On the left click **Security > Certificates**. Click **Certificate Authority**.
- k8sCluster: FQDN for Kubernetes cluster API server.
- k8sPort: Port on which the Kubernetes API server is listed.

The information is available on the UI console of the Kubernetes cluster.

2. If it is not present, run the following command to get Kubernetes cluster and Kubernetes port:

   `# kubectl cluster-info` Kubernetes control plane runs at https://<Kubernetes FQDN>:6443

   - datamoverimage: Container registry URL to pull data mover image.
   - Storage parameters are required for snapshot and backup from snapshot operations. At least one of the Block or Filesystem storage parameters are mandatory.

3. To get the storage classes, run the following command:

   ```
   # kubectl get storageclasses
   ```

   - storageclassblock: Storage class that is used for provisioning block volumes.
   - storageclassfilesystem: Storage class that is used for provisioning file system volumes.

4. To get the volume snapshot class, run the following command:

   ```
   # kubectl get volumesnapshotclasses
   ```

   - volumesnapshotclassblock: Volume snapshot class for creating block volume snapshots.
   - volumesnapshotclassfilesystem: Volume snapshot class for creating file system volume snapshots.

5. Mapping between the storageclass and the snapshot class is managed through the storageMap. If a new storage option is added to the cluster, it can also be updated in the configmap for backup-operator-configuration after installation.

   - storageMap is a dictionary of key, value fields where key is storage class and its value is a tuple consisting of (snapshotClass, storageClassForBackupDataMovement, storageClassForRestoreFromBackup) This field is mandatory to specify mapping between storage class and snapshot class.
   - snapshotclass must be created with same provisioner as storage class and it must be capable of snapshotting the storage class. All storage classes should have their entry for snapshotclass.

- storageClassForBackupDataMovement is used for creating temporary PVC for datamover. It must be compatible with original storage class PVC created using snapshot of original storage class must be readable when created using this storage class. Datamover reads data from this PVC and sends it to NetBackup media server. storageClassForRestoreFromBackup is used to restore from media server backup. It must be compatible with original storage class and come from same provisioner.
- One snapshot class can be used for snapshotting multiple compatible storage classes.
- Template

```
storageMap:
  <key - storage class name>:
    snapshotClass: [mandatory field to specify volumesnapshotclass for
    storageClassForBackupDataMovement: <optional, storage class used t
NetBacup media server>
    storageClassForRestoreFromBackup: <optional, storage class used to
to k8s cluster>

Note: storageClassForBackupDataMovement and storageClassForRestoreFrom
with key storage class if they are configured different from key storag
fields original
storage class would be used. These values can be changed later in backu

Example for openshift storage classes. cephfs storage class should have
storageMap:
  ocs-storagecluster-cephfs:
    storageClassForBackupDataMovement: ocs-storagecluster-cephfs
    storageClassForRestoreFromBackup: ocs-storagecluster-cephfs
    snapshotClass: ocs-storagecluster-cephfsplugin-snapclass
  ocs-storagecluster-ceph-rbd:
    snapshotClass: ocs-storagecluster-rbdplugin-snapclass
```

Install

To install helm, run the following command:

```
# helm install veritas-netbackupkops <path to
netbackupkops-helm-chart> -n <kops namespace>
```

Debug

To get the config-deploy pod from the Kubernetes operator namespace, run the following command:

```
# kubectl get pod -n <kops namespace> | grep "config-deploy"
```

Logs

To check the logs from the pod <namespace>-netbackup-config-deploy, run the following command:

```
# kubectl logs <pod-name> -n <kops namespace>
```

Log level

It sets the log level of the configuration pod. Values can be set to `DEBUG`, `INFO`, or `ERROR`. Default value is set to `INFO`.

Note: For more details, refer to the *NetBackup Kubernetes Quick Start Guide*.

Configure settings for NetBackup snapshot operation

You need to configure storage classes and volume snapshot class in order to protect the Persistent Volume Claims. After Netbackup Kubernetes Operator is installed, the following configuration needs to be done.

1. Identify the storage classes pointing to CSI plugin or if there is no storage class present then define a new storage class pointing to the CSI plugin.
2. Identify the Volume Snapshot Class pointing to CSI Plugin or If there is no volume snapshot class present then define a new volume snapshot class pointing to CSI plugin.

   Define a *VolumeSnapshotClass* class consisting of CSI driver details.
3. Label the CSI Storage classes on the kubernetes cluster.
   - The storage class labels *netbackup.veritas.com/default-csi-storage-class=true* is used to label where storage class provisions volumes based on raw block (volumeMode=Block).
   - The storage class label *netbackup.veritas.com/default-csi-filesystem-storage-class=true* is used to label where storage class provisions volumes based on file system (volumeMode=FileSystem).

   Note: You can add both labels on a single storage class. If the storage class supports Block volume backed by raw block and the Filesystem volume.

   ---

4. Label the CSI Volume Snapshot Class for Netbackup usage
   - The needs*volumesnapshotclasslabel* *netbackup.veritas.com/default-csi-storage-class=true* is label that is required to add to all the CSI volumesnapshotclass which user wants to use for snapshot operation.

Note: Snapshot of a namespace consisting of persistent volume fails with an error message : Failed to create snapshot of the Kubernetes namespace. The snapshot operation may fail due to multiple reasons, for example a valid volumesnapshot class for the driver with valid label volumesnapshotclass is not found.

Kubernetes operators supported configuration parameters

Note: To get the configuration value, you can run the command: `kubectl get configmaps <namespace>-backup-operator-configuration -n <namespace> -o yaml > {local.file}`

Table 2-2 Kubernetes operators supported configuration parameters in <namespace>-backup-operator-configuration

Configuration	Description	Default value	Possible value
daemonsets	If value is set to false this resource are discovered and backed up. But the resources will not be visible in the **Resource** section when you click on Kubernetes namespace in NetBackup web UI.	true	true, false

Table 2-2 Kubernetes operators supported configuration parameters in <namespace>-backup-operator-configuration *(continued)*

Configuration	Description	Default value	Possible value
deployments	If value is set to false this resource are discovered and backed up. But the resources will not be visible in the **Resource** section when you click on Kubernetes namespace in NetBackup web UI.	true	true, false
pods	If value is set to false this resource are discovered and backed up. But the resources will not be visible in the **Resource** section when you click on Kubernetes namespace in NetBackup web UI.	true	true, false
replicasets	If value is set to false this resource are discovered and backed up. But the resources will not be visible in the **Resource** section when you click on Kubernetes namespace in NetBackup web UI.	true	true, false

Table 2-2 Kubernetes operators supported configuration parameters in <namespace>-backup-operator-configuration *(continued)*

Configuration	Description	Default value	Possible value
secrets	If value is set to false this resource are discovered and backed up. But the resources will not be visible in the **Resource** section when you click on Kubernetes namespace in NetBackup web UI.	true	true, false
services	If value is set to false this resource are discovered and backed up. But the resources will not be visible in the **Resource** section when you click on Kubernetes namespace in NetBackup web UI.	true	true, false
namespace	Kubernetes operator is deployed in the namespace. If value is set to false this resource are discovered and backed up. But the resources will not be visible in the **Resource** section when you click on Kubernetes namespace in NetBackup web UI.	Any name given to a namespace.	NetBackup namespace.

Table 2-2 Kubernetes operators supported configuration parameters in <namespace>-backup-operator-configuration *(continued)*

Configuration	Description	Default value	Possible value
cleanStaleCRDurationMinutes	Time duration after a CR job is invoked to clean stale CRs. The interval after which stale custom resource cleanup job is triggered. For long running Backup from snapshot and Restore from backup jobs, you need to increase the cleanStaleCRDurationMinutes value.	1440 minutes	Any values in minutes
ttlCRDurationMinutes	TTL CR duration	30240 minutes	30240 minutes
fipsMode	Configuration to enable FIPS_MODE in Netbackup Kubernetes Operator and data mover.	DISABLE	ENABLE, DISABLE
livesnessProbeInitialDelay	Probe initial delay period.	60 seconds	60 minutes
livenessProbeTimeoutInSeconds	On loaded machine, liveness probe execution might take more than 1 second, User can increase this value in-order to stop failures due liveness probe timeout errors.	1 Second	1 to 5 Seconds.
livenessProbePeriodInSeconds	Probe period.	180 seconds	Any value in seconds
checkNbcertdaemonStatusDurationMinutes	NB certificate daemon status duration.	1440 minutes	1440 minutes

Table 2-2 Kubernetes operators supported configuration parameters in <namespace>-backup-operator-configuration *(continued)*

Configuration	Description	Default value	Possible value
collectDataMoverLogs	Due to high memory usage in datamover logs collection, it is recommended to enable the logs only when you are debugging, troubleshooting, or restarting the pods. Before enabling the logs for datamover, ensure to increase the memory limits for NetBackup Kubernetes pod to at least 2 GB or more. After the debugging or troubleshooting is done, you can reset to the previous or the default value. **Note:** Granular support is provided for collecting datamover logs only in case of failed jobs. It provides an additional level of granularity layer, All/FailedOnly/Off.	Failed	All, Failed, None
maxRetentionDataMoverLogsInHours	Maximum retention for datamover logs.	24 hours	72 hours
maxRetentionDataMoverInHours	It removes all the datamover resources that are older than the specified time.	24 hours	Any value in hours

Table 2-2 Kubernetes operators supported configuration parameters in <namespace>-backup-operator-configuration *(continued)*

Configuration	Description	Default value	Possible value
cleanStaleCertFilesDurationMinutes	The interval after which stale certificate files cleanup job is triggered.	60 minutes	1440 minutes
maxRetentionInDiscoveryCacheHours	It is the time in hours that decides the time interval for keeping the discovery cache.	24 hours	48 hours
pollingTimeoutInMinutes	It is the timeout that keeps retrying till it expires and fails.	15 minutes	Any value in minutes
pollingFrequencyInSecs	Polling frequency.	5 seconds	Any value in seconds
nbcertPrerequisteDirectoryAndFiles	NBCA prerequisites.	Certificate name	Certificate name

Prerequisites for backup from snapshot and restore from backup operations

1. Label a valid storage class for NetBackup usage. Add the following labels based on the VolumeModes (Block or Filesystem) that the storage class supports.
   - For a Filesystem based Persistent volume claim provisioning Storage class: *veritas.com/default-csi-filesystem-storage-class=true*
   - For a Block based Persistent volume claim provisioning Storage class: *veritas.com/default-csi-storage-class=true*

   If the NetBackup-labeled storage class is not found then the Backup from snapshot and Restore from backup copy fails with an error message: `No eligible storage classes found`.

   To label the storage classes, run the following commands that are shown in the examples:

   Example 1. Run the command:`# kubectl get sc`

Name	Provisioner
ocs-storagecluster-ceph-rbd (default)	openshift-storage.rbd.csi.ceph.com
ocs-storagecluster-ceph-rgw	openshift-storage.ceph.rook.io/bucket
ocs-storagecluster-ceph-rbd	openshift-storage.cephfs.csi.ceph.com
Openshift-storage.noobaa.io	openshift-storage.noobaa.io/obc
thin	kubernetes.io/vsphere-volume

Reclaim policy	Volume binding mode	Allow volume expansion	Age
Delete	Immediate	True	2d2h
Delete	Immediate	False	2d2h
Delete	Immediate	True	2d2h
Delete	Immediate	False	2d2h
Delete	Immediate	False	19h

Note: You need a storage class with volume binding mode set to *Immediate*. If the PVC volume binding mode is *WaitForFirstConsumer* then it affects the creation of the snapshot from the PVC. This situation can cause the backup jobs to fail.

Example 2. Run the command:`# kubectl get sc ocs-storagecluster-ceph-rbd --show-labels`

Name	Provisioner	Reclaim policy
ocs-storagecluster-ceph-rbd (default)	openshift-storage.rbd.csi.ceph.com	Delete

Volume binding mode	Allow volume expansion	Age	Label
Immediate	True	2d2h	netbackup.veritas.com/default-csi-storage-class=true

Example 3. Run the command: `oc label storageclass ocs-storagecluster-cephfs netbackup.veritas.com/default-csi-storage-class=true`

```
storageclass.storage.k8s.io/ocs-storagecluster-cephfs labeled
```

Example 4. Run the command:`kubectl get sc ocs-storagecluster-cephfs --show-labels`

Name	Provisioner	Reclaim policy
ocs-storagecluster-cephfs	openshift-storage.cephfs.csi.ceph.com	Delete

Volume binding mode	Allow volume expansion	Age	Label
Immediate	True	2d2h	netbackup.veritas.com/default-csi-storage-class=true

2. Label a valid volume snapshot class for NetBackup usage, add the following label: *netbackup.veritas.com/default-csi-volume-snapshot-class=true*. If the NetBackup labeled *VolumeSnapshotClass* class is not found, then backup from snapshot job for metadata image and restore jobs fails with an error message: *Failed to create snapshot of the Kubernetes namespace*.

   To label the volume snapshot classes, run the following commands given the examples:

   Example 1. Run the command:`# kubectl get volumesnapshotclass`

Name	Driver
ocs-storagecluster-cephfsplugin-snapclass	openshift-storage.cephfs.csi.ceph.com
ocs-storagecluster-rbdplugin-snapclass	openshift-storage.rbd.csi.ceph.co

Deletion policy	Age
Delete	2d2h
Delete	2d2h

   Example 2. Run the command:`# kubectl get volumesnapshotclass ocs-storagecluster-cephfsplugin-snapclass --show-labels`

Name	Driver
ocs-storagecluster-cephfsplugin-snapclass	openshift-storage.cephfs.csi.ceph.com

Deletion policy	Age
Delete	2d2h

Example 3. Run the command: `# kubectl label volumesnapshotclass ocs-storagecluster-cephfsplugin-snapclass netbackup.veritas.com/default-csi-volume-snapshot-class=true`

```
volumesnapshotclass.snapshot.storage.k8s.io/ocs-storagecluster-cephfsplug
```

Example 4. Run the command: `# kubectl get volumesnapshotclass ocs-storagecluster-cephfsplugin-snapclass --show-labels`

Name	Driver
ocs-storagecluster-cephfsplugin-snapclass	openshift-storage.cephfs.csi.ceph.com

Deletion policy	Age	Labels
Delete	2d2h	netbackup.veritas.com/default-csi-volume-snapshot-class=true

3. Each primary server which runs the backup from snapshot and restore from backup copy operations, needs to create a separate *ConfigMap* with the primary server's name.

   In the following `configmap.yaml` example:

   - `backupserver.sample.domain.com` and `mediaserver.sample.domain.com` are the host names of the NetBackup primary and media server.
   - IP: `10.20.12.13` and IP: `10.21.12.13` are the IP addresses of the NetBackup primary and media server.

```
apiVersion: v1
data:
  datamover.hostaliases: |
        10.20.12.13=backupserver.sample.domain.com
        10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
        image=reg.domain.com/datamover/image:latest
  version: "1"
```

```
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: kops-ns
```

- Copy the `configmap.yaml` file details.
- Open the text editor and paste the `yaml` file details.
- Save the file with the `yaml` file extension to the home directory from where the Kubernetes clusters are accessible.

4. Specify `datamover.properties: image=reg.domain.com/datamover/image:latest` with correct data mover image.
5. Specify `datamover.hostaliases`, if the primary server and the media servers that are connected to the primary server have short names and host resolution failing from the data mover. Provide a mapping of all the host names to the IPs for the primary and the media servers.
6. Create a secret as described in detail in the Point 6 in the *Deploy service package on NetBackup Kubernetes operator* section to use a private docker registry.

   Once the secret is created, add the following attributes while creating a configmap.yaml file.

   ```
   datamover.properties: |
   image=repo.azurecr.io/netbackup/datamover:10.0.0049
   imagePullSecret=secret_name
   ```

7. Create the `configmap.yaml` file. Run the command: `kubectl create -f configmap.yaml`.
8. If the Kubernetes operator is not able to resolve the primary server with the short names, refer to the following guidelines.
   - If you get the following message when you fetch the certificates:*EXIT STATUS 8500: Connection with the web service was not established*. Then, verify the host name resolution state from the `nbcert` logs.
   - If the host name resolution fails, then update the `values.yaml` file with `hostAliases`.
   - In the following `hostAliases` example:
     - `backupserver.sample.domain.com` and `mediaserver.sample.domain.com` are the host names of the NetBackup primary and media server.

- IP: `10.20.12.13` and IP: `10.21.12.13` are the IP addresses of NetBackup primary and media server.

```
hostAliases:
- hostnames:
  - backupserver.sample.domain.com
  ip: 10.20.12.13
- hostnames:
  -  mediaserver.sample.domain.com
  ip: 10.21.12.13
```

Copy, paste the `hostAliases` example details in the text editor and add to the `hostAliases` in the deployment.

Note: The `hostAliases` section must be added in the default file `./netbackupkops-helm-chart/values.yaml`.

`hostAliases` example:

```
2104 hostAliases;
- ip:10.15.206.7
hostnames:
- lab02-linsvr-01.demo.sample.domain.com
- lab02-linsvr-01
- ip:10.15.206.8
hostnames:
- lab02-linsvr-02.demo.sample.domain.com
- lab02-linsvr-02
imagePullSecrets:
- name:  {{ .values.netbackupKops.imagePullSecrets.name}}
```

9. Create a secret with fingerprint and authorization token.

   For more information about creating the secret and backupservercert, refer to the section *Deploying certificates on NetBackup Kubernetes operator* in the *NetBackup for Kubernetes Administrator's Guide*.

10. Create a backupservercert request to fetch certificates.

    For more information, refer to *Deploying certificates on NetBackup Kubernetes operator*in the *NetBackup for Kubernetes Administrator's Guide*.

    For more information, refer to the *NetBackup Security and Encryption Guide*.

Note: This step is mandatory to have successful backup from snapshot and restore from backup copies.

DTE client settings supported in Kubernetes

The **DTE_CLIENT_MODE** option specifies the data-in-transit encryption (DTE) mode that is set on the datamover via backupserver specific configmap. Data-in-transit encryption of backup images is carried out based on the global DTE mode and the client DTE mode.

Update the backupserver specific configmap and add DTE_CLIENT_MODE key to it. This key can take following values:

- AUTOMATIC
- ON
- OFF

For more information on the DTE_CLIENT_MODE, refer to the *DTE_CLIENT_MODE for clients* section in the *Veritas NetBackup™ Administrator's Guide, Volume I* .

Following is the configmap with DTE_CLIENT_MODE setting added:

```
apiVersion: v1
data:
  datamover.hostaliases: |
        10.20.12.13=backupserver.sample.domain.com
        10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
        image=reg.domain.com/datamover/image:latest
        DTE_CLIENT_MODE=ON
  version: "1"
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: kops-ns
```

Customization of datamover properties

You can customize datamover properties by passing key-value pairs in the backup server specific configmap.

Table 2-3 Datamover properties

Key Name	Possible Values
VXMS_VERBOSE	Range:[0,99]
VERBOSE	Range:[0,5]
DTE_CLIENT_MODE	■ AUTOMATIC ■ ON ■ OFF

To update the configmap, add the key value pairs as follows:

```
apiVersion: v1
data:
  datamover.properties: |
        image=reg.domain.com/datamover/image:latest
        VERBOSE=5
        DTE_CLIENT_MODE=OFF
        VXMS_VERBOSE=5
  version: "1"
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: kops-ns
```

Troubleshooting NetBackup servers with short names

1. If NetBackup Kubernetes operator is not able to resolve backup server or media server based on short names, perform the following steps:
   - While fetching certificates if you get a message, *EXIT STATUS 8500: Connection with the web service was not established*. Then confirm from the nbcert logs whether hostname resolution successful or not. If it has failed, then perform the following steps:
   - Update the Kubernetes operator `deployment.yaml` and add the `hostAliases` in the deployment.
   - In the following `hostAliases` example,
     - backupserver.sample.domain.com and mediaserver.sample.domain.com are the hostnames of NetBackup primary and media server.

- IP: 10.20.12.13 and IP: 10.21.12.13 are the IP addresses of NetBackup primary and media server.

```
hostAliases:
- hostnames:
  - backupserver.sample.domain.com
  ip: 10.20.12.13
- hostnames:
  -  mediaserver.sample.domain.com
  ip: 10.21.12.13
```

Copy, paste the hostAliases example details in the text editor and add to the hostAliases in the deployment.

2 If data mover is not able to resolve short names of backup server or media server. To resolve this issue, perform the following steps:

- Update configmap with backup server name.
- Add *datamover.hostaliases* field, map with IP addresses to the hostname.
- In the following `configmap.yaml` example,
  - backupserver.sample.domain.com and mediaserver.sample.domain.com are the hostnames of NetBackup primary and media server.
  - IP: 10.20.12.13 and IP: 10.21.12.13 are the IP addresses of NetBackup primary and media server.

```
apiVersion: v1

data:
  datamover.hostaliases: |
        10.20.12.13=backupserver.sample.domain.com
        10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
        image=reg.domain.com/datamover/image:latest
  version: "1"
kind: configmap
metadata:
  name: backupserver.sample.domain.com
  namespace: kops-ns
```

  - Copy the `configmap.yaml` file details.
  - Open the text editor and past the yaml file details.
  - Then, save it with the yaml file extension to the home directory from where the Kubernetes clusters are accessible.

- To create the `configmap.yaml` file, run the command `kubectl create -f configmap.yaml`.
- if you update the `configmap.yaml` which is already created then run the command to update configmap. `kubectl apply -f configmap.yaml`

Data mover pod schedule mechanism support

Specify the following fields in the backup server ConfigMap to schedule data mover pods on the nodes.

1. **nodeSelector**: nodeSelector is the effortless way to constrain pods to the nodes with specific labels.

   Example:

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup

data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com

  datamover.properties: |

    image=reg.domain.com/datamover/image:latest

  datamover.nodeSelector: |

    kubernetes.io/hostname: test1-l94jm-worker-k49vj

    topology.rook.io/rack: rack1
```

```
  version: "1"
```

2. **nodeName**: nodeName is a direct form of node selection than affinity or nodeSelector. It allows you to specify a node on which a pod is scheduled for backup, overriding the default schedule mechanism.

   Example:

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup

data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com

  datamover.properties: |

    image=reg.domain.com/datamover/image:latest

  datamover.nodeName : test1-l94jm-worker-hbblk

  version: "1"
```

3. **Taint and Toleration**: Toleration allows you to schedule the pods with similar taints. Taint and toleration work together to ensure that the pods are scheduled onto appropriate nodes. If one or more taints are applied to a node. Then that node must not accept any pods which does not tolerate the taints.

   Example:

```
apiVersion: v1

kind: ConfigMap
```

```
metadata:
  name: backupserver.sample.domain.com
  namespace: netbackup
data:
  datamover.hostaliases: |
    10.20.12.13=backupserver.sample.domain.com
    10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
  datamover.tolerations: |
    - key: "dedicated"
      operator: "Equal"
      value: "experimental"
      effect: "NoSchedule"
  version: "1"
```

4. **Affinity and Anti-affinity**: Node affinity functions like the nodeSelector field but it is more expressive and allows you to specify soft rules. Inter-pod affinity/anti-affinity allows you to constrain pods against labels on the other pods.

   Examples:

   - **Node Affinity**:

     ```
     apiVersion: v1
     kind: ConfigMap
     ```

```
metadata:
  name: backupserver.sample.domain.com
  namespace: netbackup
data:
  datamover.hostaliases: |
    10.20.12.13=backupserver.sample.domain.com
    10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
  datamover.affinity: |
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
            - test1-l94jm-worker-hbblk
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
          matchExpressions:
```

```
          - key: beta.kubernetes.io/arch
            operator: In
            values:
            - amd64
  version: "1"
```

- **Pod Affinity**

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: netbackup
data:
  datamover.hostaliases: |
    10.20.12.13=backupserver.sample.domain.com
    10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
  datamover.affinity: |
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
```

```
            matchExpressions:
            - key: component
              operator: In
              values:
              - netbackup
          topologyKey: kubernetes.io/hostname
    version: "1"
```

5. **topologySpreadConstraints**: Topology spread constraints are used to control the behavior of the pods that are spread across your cluster among failure-domains such as regions, zones, nodes, and other user-defined topology domains.

   Example:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: netbackup
data:
  datamover. hostaliases: |
    10.20.12.13=backupserver.sample.domain.com
    10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
  datamover.topologySpreadConstraints : |
```

```
  - maxSkew: 1
    topologyKey: kubernetes.io/hostname
    whenUnsatisfiable: DoNotSchedule
version: "1"
```

- **Labels**: Labels are the key/value pairs attached to the objects, such as pods. Labels intends to identify the attributes of an object which are significant and relevant to users. Labels can organize and select subsets of objects. Labels which are attached to objects at creation time are subsequently added and modified at any time.
  Example:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: netbackup
data:
  datamover.hostaliases: |
    10.20.12.13=backupserver.sample.domain.com
    10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
  datamover.labels: |
    env: test
    pod: datamover
```

```
  version: "1"
```

- **Annotations**: User can use either labels or annotations to attach metadata to Kubernetes objects. You cannot use Annotations to identify and select objects.
  Example:

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup

data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com

  datamover.properties: |

    image=reg.domain.com/datamover/image:latest

  datamover.annotations: |

    buildinfo: |-

      [{

            "name": "test",

            "build": "1"

      }]

    imageregistry: "https://reg.domain.com/"
```

```
version: "1"
```

Validating accelerator storage class

NetBackup supports the accelerator enabled backups and it can be enabled by setting *acceleratorTracklogPvcStorageClass* key in values.yaml with appropriate storage class during installation or upgrade.

A storage class must allow to create a file mode PVC.

Example,*acceleratorTracklogPvcStorageClass: ocs-storagecluster-ceph-rbd*

During installation and upgrade, NetBackup Kubernetes operator creates a file mode PVC and a pod to check if the given storage class is valid.

- If volume binding mode of the storage class is **Immediate**, it only creates a PVC and installation is successful if the PVC is in Bound state.
- If volume binding mode of the storage class is **WaitForFirstConsumer** then, it creates a datamover pod with a PVC.
- NetBackup Kubernetes Operator installation is successful, if the PVC is in Bound state and pod is in Running state.

Chapter 3

Deploying certificates on NetBackup Kubernetes operator

This chapter includes the following topics:

Deploy certificates on the Kubernetes operator

You need to deploy certificates for secure communication between the datamover and the NetBackup media servers.

Note: You must deploy the certificates before you can perform **Backup from Snapshot** and **Restore from Backup** operations.

The Cluster must be added and discovered successfully before creating the BackupServerCert as it relies on the NetBackup passing some `clusterInfo` in order to set the status as Success.

Certificates supported for datamover communication

Datamover facilitates data movement within the NetBackup environment, it communicates with the media servers over Transport Layer Security (TLS). For more details, refer to the *About secure communication in NetBackup* section in

NetBackup™ Security and Encryption Guide. Datamover needs a host-id-based certificate, or an ECA-signed certificate issued by NetBackup primary server for communication. A new custom resource definition BackupServerCert is introduced to enable certificate deployment operation in NBCA (NetBackup Certificate Authority) or ECA (External Certificate Authority) mode.

Custom resource specification looks like this:

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-sample-nbca
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: primary.server.sample.com
  certificateOperation: Create | Update | Remove
  certificateType: NBCA | ECA
  nbcaAttributes:
    nbcaCreateOptions:
      secretName: "Secret name consists of token and fingerprint"
    nbcaUpdateOptions:
      secretName: "Secret name consists of token and fingerprint"
      force: true | false
    nbcaRemoveOptions:
      hostID: "hostId of the nbca certificate. You can view on Netbackup UI"
  ecaAttributes:
    ecaCreateOptions:
      ecaSecretName: "Secret name consists of cert, key, passphrase, cacert"
      copyCertsFromSecret: true | false
      isKeyEncrypted: true | false
    ecaUpdateOptions:
      ecaCrlCheck: DISABLE | LEAF | CHAIN
      ecaCrlRefreshHours: [0,4380]
```

Perform Host-ID-based certificate operations

Ensure that the primary server is configure in the NBCA mode. To check if the NBCA mode is on, run the command: `/usr/openv/netbackup/bin/nbcertcmd -getSecConfig -caUsage`.

The output looks like this:

```
NBCA: ON
ECA: OFF
```

HostID based certificate specification looks like this:

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-sample
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: primaryserver.sample.domain.com
  certificateOperation: Create | Update | Remove
  certificateType: NBCA
  nbcaAttributes:
    nbcaCreateOptions:
      secretName: "Secret name consists of token and fingerprint"
    nbcaUpdateOptions:
      secretName: "Secret name consists of token and fingerprint"
      force: true
    nbcaRemoveOptions:
      hostID: "hostId of the nbca certificate. You can view on Netbackup UI"
```

Table 3-1 HostID based certificate operations

Operation type	Options and comments
Create	secretName: Name of the secret which contains a token and fingerprint.
Remove	hostID: Host identification of the NBCA certificate.
Update	secretName: Name of the secret which contains a token and fingerprint.

Creating a HostID based certificate for Kubernetes operator

You can create a HostID based certificate for Kubernetes operator using the following procedure.

To create HostID based certificate for Kubernetes operator

1. On the backup server run the following command and get the SHA-256 fingerprint.

   ```
   /usr/openv/netbackup/bin/nbcertcmd -listCACertDetails
   ```

2. To create an authorization token, refer to the *Creating authorization tokens* section in the *NetBackup™ Security and Encryption Guide*.

3. To create a reissue token, if required, refer to the *Creating a reissue token* section in the *NetBackup™ Security and Encryption Guide*.

4. Create a secret with token and fingerprint.

5. Provide a token as it is mandatory irrespective of security level.

   `Token-fingerprint-secret.yaml` looks like this:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: secret-name
     namespace: kops-ns
   type: Opaque
   stringData:
     token: "Authorization token | Reissue token"
     fingerprint: "SHA256 Fingerprint"
   ```

   - Copy the `Token-fingerprint-secret.yaml` file text.
   - Open the text editor and paste the yaml file text.
   - Then, save the text with the yaml file extension to the home directory from where the Kubernetes clusters are accessible.

6. To create the `Token-fingerprint-secret.yaml` file, run the command:

   ```
   kubectl create -f Token-fingerprint-secret.yaml
   ```

7. Create a `backupservercert` object with the

   `nbcaCreateOptions` and then specify a secret name.

   `nbca-create-backupservercert.yaml` looks like this:

   ```
   apiVersion: netbackup.veritas.com/v1
   kind: BackupServerCert
   metadata:
     name: backupserver-nbca-create
     namespace: kops-ns
   spec:
   ```

```
clusterName: cluster.sample.com:port
backupServer: backupserver.sample.domain.com
certificateOperation: Create
certificateType: NBCA
nbcaAttributes:
  nbcaCreateOptions:
    secretName: nbcaSecretName with token and fingerprint
```

- Copy the `nbca-create-backupservercert.yaml` file text.
- Open the text editor and paste the yaml file text.
- Then, save the text with the yaml file extension to the home directory from where the Kubernetes clusters are accessible.

8 To create the `nbca-create-backupservercert.yaml` file, run the command:
`kubectl create -f nbca-create-backupservercert.yaml`

9 Once the certificate is created, check custom resource status. If the custom resource status is successful, you can run **Backup from Snapshot** jobs.

Note: You need to check that the BackupServerCert custom resource status is successful before initiating **Backup from Snapshot** or **Restore from Backup Copy** operations.

Note: To renew host ID based certificate: NetBackup host ID certificate checks if it's due for renew after 24 hours cycle. Certificates get automatically renewed 180 days (6 months) before expiration date.

Note: Ensure to check whether the NetBackup primary server clock and the NetBackup Kubernetes operator clock are in sync. For more details on the `CheckClockSkew` errors, refer to the *Implication of clock skew on certificate validity* section in the *NetBackup™ Security and Encryption Guide*.

Removing primary server certificate from Kubernetes operator

You can remove a certificate from a primary server if the server is not used for running the backup and restore operations.

To remove primary server certificate from Kubernetes operator.

1. Log on to the NetBackup web UI and get a hostID for the certificate that you want to remove.

   To get the HostID for the certificate, refer to the *Viewing host ID-based certificate details* section in the *NetBackup™ Security and Encryption Guide*.

2. Create a backupservercert with operation type remove.

   `nbca-remove-backupservercert.yaml` file looks like this:

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupserver-nbca-domain.com
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Remove
  certificateType: NBCA
  nbcaAttributes:
    nbcaRemoveOptions:
      hostID: nbcahostID
```

   - Copy the `nbca-remove-backupservercert.yaml` file text.
   - Open the text editor and paste the yaml file text.
   - Then, save the text with the yaml file extension to the home directory from where the Kubernetes clusters are accessible.

3. To create the `nbca-remove-backupservercert.yaml` file, run the command:
   `kubectl create -f nbca-remove-backupservercert.yaml`

4. To revoke the certificate, refer to the *Revoking a host ID-based certificate* section in the *NetBackup™ Security and Encryption Guide*.

Note: Once the `nbca-remove-backupservercert.yaml` is applied, certificates are removed from the Kubernetes operator's local certificate store. But it's still present and valid in the NetBackup database. So, the certificate needs to be revoked.

Updating primary server certificates

Following is the scenario when you may want to update the certificates assuming that the certificates are readable and present in the Kubernetes operator:

When certificates present on the Netbackup Kubernetes operator are revoked, then certificates can be reissued with update operation. To resolve this issue, either you can update the server certificate or you can remove the server certificate and then create a new certificate.

Note: If update certificate operation fails, you must remove the certificate first and then create a new certificate.

To update a primary server certificate on Kubernetes operator:

1. Create a backupservercert object with the update operation:

   `nbca-update-backupservercert.yaml` file looks like this:

   ```
   apiVersion: netbackup.veritas.com/v1
   kind: BackupServerCert
   metadata:
     name: backupserver-nbca-update
     namespace:kops-ns
   spec:
     clusterName: cluster.sample.com:port
     backupServer: backupserver.sample.domain.com
     certificateOperation: Update
     certificateType: NBCA
     nbcaAttributes:
       nbcaUpdateOptions:
       secretName: "Name of secret containing
   token and fingerprint"
       force: true
   ```

   - Copy the `nbca-update-backupservercert.yaml` file text.
   - Open the text editor and paste the yaml file text.
   - Then, save the text with the yaml file extension to the home directory from where the Kubernetes clusters are accessible.

2. To create the `nbca-udpate-backupservercert.yaml` file, run the command:
   `kubectl create -f nbca-update-backupservercert.yaml`

3. Once the backupservercert object is created, then check the custom resource status.

Perform ECA certificate operations

Before performing External Certificate Authority (ECA) create, update, and remove operations; you must configure the backup server in ECA mode.

To check if the ECA mode is on, run the command:
`/usr/openv/netbackup/bin/nbcertcmd -getSecConfig -caUsage`.

The output looks like this:

```
NBCA: ON
ECA: ON
```

To configure the backup server in ECA mode, refer to the *About external CA support in NetBackup* section in the *NetBackup™ Security and Encryption Guide*

ECA certificate specification looks like this:

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-sample-eca
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: primaryserver.sample.domain.com
  certificateOperation: Create | Update | Remove
  certificateType: ECA
  ecaAttributes:
    ecaCreateOptions:
      ecaSecretName: "Secret name consists of cert, key, passphrase, cacert"
      copyCertsFromSecret: true | false
      isKeyEncrypted: true | false
    ecaUpdateOptions:
      ecaCrlCheck: DISABLE | LEAF | CHAIN
      ecaCrlRefreshHours: range[0,4380]
```

Table 3-2 ECA certificate operations

Operation type	Options and comments
Create	■ secretName: Name of secret containing cert, key, passphrase, cacert. ■ copyCertsFromSecret: Possible values are true and false. This option is added as the External CA is common across all primary servers. Same certificates can be enrolled to Kubernetes operator for all primary servers. Thus, there is no need to copy certs and keys every time. Copying of certificates and keys can be controlled with this option. If ECAHealthCheck fails due to something wrong with certs and keys, then the certificates must be copied again. ■ isKeyEncrypted; If the private key is encrypted, set this field as true else set it as false.
Remove	NA
Update	■ ecaCrlCheck: Lets you specify the revocation check level for external certificates. Possible values are DISABLE, LEAF, and CHAIN. ■ ecaCrlRefreshHours specifies the time interval in hours to download Certificate Revocation Lists. Possible values range between 0-4380

Creating ECA signed certificate

NetBackup supports Kubernetes operator on multiple primary servers for ECA. If the external CA is common across primary servers. It is mandatory to use Certificate Revocation List distribution point for fetching Certificate Revocation List dynamically during the communication.

To create ECA signed certificate

1. Use the Certificate Revocation List distribution point to fetch Certificate Revocation List.
2. Keep ECA signed certificate chain, private key, and passphrase (if required) ready in your home directory.
3. To identify different formats (like, DER, PEM and so on) that are supported for each of the files mentioned in step 2. For more information, refer to the *Configuration options for external CA-signed certificates* section in the *NetBackup™ Security and Encryption Guide*.
4. Create a secret using the files mentioned in step 3.

- To create a secret if private key is unencrypted, run the command: `kubectl create secret generic <Name of secret> --from-file=cert_chain=<File path to ECA signed certificate chain> --from-file=key=<File path to private key> --from-file=cacert=<File path to External CA certificate> -n <Namespace where kops is deployed>`
- To create a secret if private key is encrypted, run the command: `kubectl create secret generic <Name of secret> --from-file=cert_chain=<File path to ECA signed certificate chain> --from-file=key=<File path to private key> --from-file=cacert=<File path to External CA certificate> --from-file=passphrase=<File path to passphrase of encrypted private key> -n <Namespace where kops is deployed>`

Directory structure looks like this:

```
├── cert_chain.pem
├── private
|   |___key.pem
|   |___passphrase.txt
|___trusted
      |__cacerts.pem
```

cert_chain.pem is ECA signed certificate chain

private/key.pem is private key

private/passphrase.txt is passpharse for private key

trusted/cacerts.pem is External CA certificate

- To create a secret of name eca-secret when private key is unencrypted, run the command:

```
kubectl create secret generic
eca-secret--from-file=cert_chain=cert_chain.pem
--from-file=key=private/key.pem
--from-file=cacert=trusted/cacerts.pem -n kops-ns
```

- To create a secret of name eca-secret when private key is encrypted, run the command:

```
kubectl create secret generic eca-secret
--from-file=cert_chain=cert_chain.pem
--from-file=key=private/key.pem
--from-file=cacert=trusted/cacerts.pem
```

```
--from- file=passphrase=private/passphrase.txt
-n kops-ns
```

5 Once the secret is created, then create a `backupservercert` object custom resource.

`eca-create-backupservercert.yaml` file looks like this:

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-eca-create
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Create
  certificateType: ECA
  ecaAttributes:
    ecaCreateOptions:
      ecaSecretName: eca-secret
      copyCertsFromSecret: true
      isKeyEncrypted: false
```

- Copy the `eca-create-backupservercert.yaml` file text.
- Open the text editor and paste the yaml file text.
- Then, save the text with the yaml file extension to the home directory from where the Kubernetes clusters are accessible.

6 To copy certificate and keys to the Kubernetes operator, do any of the following:

- Set `copyCertsFromSecret` as true
- Set `copyCertsFromSecret` as false to avoid copying certificates and keys existing on the Kubernetes Operator.

Note: ECA is common across all primary server thus Kubernetes operator require one set of certificates and keys that can be enrolled with all primary servers as required. No need to copy certificates and keys every time unless there's issue with the previous copied certificates and keys.

Note: If `ecaHealthCheck` fails due to any reason related to certificates and keys (corrupted or expired or changed ECA) then you identify the reason for failure and perform a copy of a valid certificate using a flag.

7 If private key is encrypted, set `isKeyEncrypted` flag as true or else false for unencrypted key. Ensure passphrase is provided in secret if private key is encrypted.

8 Set `ecaSecretName` with the secret name, created `backupservercert yaml` in step 5.

9 To create the `eca-create-backupservercert.yaml` file, run the command:
`kubectl create -f eca-create-backupservercert.yaml`

10 Once the `backupservercert` custom resource is created, check the custom resource status.

11 To view the external certificate details on the NetBackup web UI, refer to the *View external certificate information for the NetBackup hosts in the domain* section in the *NetBackup™ Web UI Administrator's Guide*.

Removing the ECA signed certificate

You can remove the ECA signed certificate from the primary server.

To remove ECA signed certificate

1 Create a `backupservercert` with operation as remove and certificate type as ECA.

`eca-remove-backupservercert.yaml` file looks like this:

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-eca-remove
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Remove
  certificateType: ECA
```

- Copy the `eca-remove-backupservercert.yaml` file text.
- Open the text editor and paste the yaml file text.

- Then, save the text with the yaml file extension to the home directory from where the Kubernetes clusters are accessible.

2 To create the `eca-remove-backupservercert.yaml file`, run the command:
`kubectl create -f eca-remove-backupservercert.yaml`

3 Once the object is created, then you need to check the custom resource status. If failed, then you can take necessary actions.

These steps removes the external certificate details with respect to the specified primary server from the local certificate store. The certificate is neither deleted from the system nor from the NetBackup database.

If you want to disable ECA then refer to the *Disabling an external CA in a NetBackup domain* section in the *NetBackup™ Security and Encryption Guide*

If you enrolled ECA on the Kubernetes operator for a backup server but later reinstalled the backup server which supports just NBCA. Then, you have to remove ECA enrolment from Kubernetes operator because during *nbcertcmd* communication with backupserver CA support might get compared and if it mismatches then an error occurs.

Updating the ECA signed certificate

There are certain options that are configurable in ECA. You can configure these options through the update operations.

To update the ECA signed certificate

1 Create a `backupservercert` object with operation type update.

`eca-update-backupservercert.yaml` file looks like this:

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-eca-update
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Update
  certificateType: ECA
  ecaAttributes:
    ecaUpdateOptions:
      ecaCrlCheck: DISABLE | LEAF | CHAIN
      ecaCrlRefreshHours: [0,4380]
```

- Copy the `eca-update-backupservercert.yaml` file text.

- Open the text editor and paste the yaml file text.
- Then, save the text with the yaml file extension to the home directory from where the Kubernetes clusters are accessible.

2 To create the `eca-update-backupservercert.yaml` file, run the command:
`kubectl create -f eca-update-backupservercert.yaml`

3 The ECA_CRL_CHECK option lets you specify the revocation check level for external certificates of the host. It also lets you disable the revocation check for the external certificates. Based on the check, the revocation status of the certificate is validated against the Certificate Revocation List (CRL) during host communication. For more information, refer to the *ECA_CRL_CHECK for NetBackup servers and clients* section in the *NetBackup™ Security and Encryption Guide*.

4 The ECA_CRL_REFRESH_HOURS option specifies the time interval in hours to download the CRLs from the URLs that are specified in the peer host certificate's Certificate Revocation List distribution points (CDP). For more information, refer to the *ECA_CRL_REFRESH_HOURS for NetBackup servers and clients* section in the *NetBackup™ Security and Encryption Guide*

Identify certificate types

NetBackup helps you identify the certificate types enrolled on the Kubernetes operator.

To identify the certificate type

1 To list the Kubernetes operator pods, run the command: `kubectl get pods -n <namespace of Kubernetes operator>`

2 Log on to the Kubernetes operator with administrator rights and run the command:

```
kubectl exec pod/nbu-controller-manager-7c99fb8474-hzrsl -n
<namespace of Kubernetes operator> -c netbackupkops -it -- bash
```

3 To list backup servers which have NBCA certificate for Kubernetes, run the command:

```
/nbcertcmdtool/nbcertcmdtool -atLibPath/nbcertcmdtool/

-standalone -installDir "/usr/openv" -listCertDetails -NBCA
```

The output looks like this:

```
Master Server : masterserver.sample.domain.com
Host ID : b06738f0-a8c1-47bf-8d95-3b9a41b7bb0a
Issued By : /CN=broker/OU=NBCANBKOps
Serial Number : 0x508cdf4500000008
Expiry Date : Dec 22 05:46:32 2022 GMT
SHA-1 Fingerprint : 4D:7A:D9:B9:61:4E:93:29:B8:93:0B:E0:
07:0A:28:16:46:F6:39:C6
SHA-256 Fingerprint : C2:FA:AC:B5:21:6B:63:49:30:AC:4D:5E:
61:09:9A:8C:C6:40:4A:44:B6:39:7E:2B:B3:36:DE:D8:F5:D1:3D:EF
Key Strength : 2048
Subject Key Identifier : AC:C4:EF:40:7D:8D:45:B4:F1:89:DA:FB:
E7:FD:0F:FD:EC:61:12:C6
Authority Key Identifier : 01:08:CA:40:15:81:75:7B:37:9F:51:78:
B2:6A:89:A1:44:2D:82:2B
```

4 To list of backup servers which have ECA certificate for Kubernetes, run the command:

```
/nbcertcmdtool/nbcertcmdtool -atLibPath/nbcertcmdtool/

-standalone -installDir"/usr/openv" -listCertDetails -ECA
```

The output looks like this:

```
Subject Name : CN=ECA-KOPS,O=Veritas,OU=ECANBKOps
 Issued By : CN=ICA-2,O=Veritas,OU=ECANBKOps
 Serial Number : 0x56cf16040258d3654339b7f39817de89240d58
 Expiry Date : Dec 16 05:48:16 2022 GMT
 SHA-1 Fingerprint : 70:DE:46:72:57:56:4E:47:DB:82:8B:8D:A3:
 4B:BB:F9:8D:2C:B7:8E
 SHA-256 Fingerprint : E0:69:5F:79:A6:60:DB:7B:69:76:D3:A8:
 E6:E1:F2:0D:8C:6C:E6:4E:C4:5D:A4:77:17:5A:C2:42:89:74:15:7D
 Key Strength : 2048
 Subject Key Identifier : F0:E7:1F:8C:50:FD:4D:25:40:69:77:6C:
 2A:35:72:B6:1D:8E:E5:17
 Authority Key Identifier : D7:53:57:C7:A6:72:E3:CB:73:BD:48:51:
 2F:CB:98:A3:0B:8B:BA:5C
 Master Server : masterserver.sample.domain.com
 Host ID : b85ba9bf-02a8-439e-b787-ed52589c37d1
```

Chapter 4

Managing Kubernetes assets

This chapter includes the following topics:

Add a Kubernetes cluster

Before you add a Kubernetes cluster in NetBackup, you must install and configure the Kubernetes operator in the cluster. Or else the validation of the cluster fails which further leads to fail the add cluster operation.

After Kubernetes operator configuration, you can add Kubernetes clusters in NetBackup and discover all the assets inside the cluster automatically.

To add a cluster

1. On the left click **Kubernetes**, under **Workloads**.
2. Click the **Kubernetes clusters** tab, click **Add**.
3. In the **Add Kubernetes cluster** page, enter the following:
   - **Cluster name**: Enter a name for the cluster. The name should be a DNS resolvable value or an IP address. Example: cluster.sample.domain.com.
   - **Port**: Enter the Kubernetes API server port number.

- **Controller namespace**: Enter the namespace where the NetBackup Kubernetes operator is deployed in the Kubernetes cluster. Example: kops-ns.

4 Click **Next**. In the **Manage credentials** page, you can add credentials to the cluster.

- To use an existing credential, choose **Select from an existing credential**, and click **Next**. In the next page, select the required credentials, and click **Next**.
- To create a new credential, click **Add credential**, and click **Next**. In the **Manage credentials** page, enter the following:
  - **Credential name**: Enter a name of the credential.
  - **Tag**: Enter a tag to associate with the credential.
  - **Description**: Enter a description of the credential.
  - To add Kubernetes clusters in NetBackup you need a Certification Authority (CA) certificate and a token. CA certificate and a token of the backup service account is required for authorization and authentication of Kubernetes cluster. To get a CA certificate and token, run the following command in the Kubernetes cluster: `kubectl get secret <[namespace-name]-backup-server-secret> -n <namespace name> -o yaml..`
    - **Token**: Enter the authentication token value in Base64 encoded form.
    - **CA certificate**: Enter the CA certificate file contents.

5 Click **Next**.

The credentials are validated and on successful validation, the cluster is added. After the cluster is added, autodiscovery runs to discover available assets in the cluster.

Note: In NetBackup Kubernetes version 10.1, edit cluster operation fails with an error message. The recommended action to resolve this issue is to first delete the cluster and add the cluster again.

Configure settings

The Kubernetes settings let you configure the various aspects of the Kubernetes deployment.

Change resource limits for Kuberentes resource types

About resource limit settings

With this setting you can control the number of backups that can be performed simultaneously on Kubernetes clusters. Kubernetes have two different default values to run the snapshot and backup from snapshot jobs 1 and 4 respectively.

Examples:

To run a snapshot only backup job, if you protect 20 assets, and you have set the limit to 5. Then only 5 assets can perform backup simultaneously, rest of the 15 assets stand in a queue. After one of the first 5 assets completes the backup, an asset from the queue takes its place.

To run a snapshot job, the default value for the resource limit is 1. Indicating that only one backup job per cluster can be in progress, while the rest of the assets are the queued state.

Configure this setting is recommended for optimized use of your system and network resources. The settings apply to all Kubernetes backups for the selected primary server.

To set the resource limit

1. On the left, **Workloads > Kubernetes**.
2. On top right, click **Kubernetes settings > Resource limits**.
3. Do any of the following to set the resource limits:
   - Click **Edit**, next to **Backup jobs per Kubernetes cluster**. By default, the limit is 1.
     This defines the amount of namespaces processed in parallel per cluster. Also, applies to the first operation to create a backup using snapshots for Backup from Snapshot jobs.
     By default, the resource limit is 1 for the Backup jobs per cluster.
   - Click **Edit**, next to **Backup from Snapshot Jobs per Kubernetes Cluster**.
     This defines the amount of namespaces backed up in parallel after snapshot creation per cluster. Each Backup from Snapshot will start a data mover pod in the NetBackup namespace on the cluster to process the data.
     By default, the resource limit is 4 for the Backup from Snapshot jobs per cluster.
4. In the **Edit Kubernetes cluster** dialog:
   - Enter a value in the **Global** field, to set a global limit for all the clusters. This limit denotes the number of *Backup* and *Backup from Snapshot* jobs that are performed simultaneously on a cluster.

- You can add individual limits to the clusters that override the global limit for that cluster. To set individual limits to the clusters, click **Add**.
- You can select the cluster available from the list and then enter a limit value for the selected cluster. You can add limits to each available cluster in your deployment.
- Click **Save** to save the changes.

Note: In the NetBackup 10.0 release, the data mover pods exceed the Kubernetes resource limit settings.

See “Datamover pods exceed the Kubernetes resource limit ” on page 125.

Configure autodiscovery frequency

Autodiscovery keeps a count of the NetBackup protected assets in your clusters. This setting lets you set the frequency by which NetBackup runs auto discovery to locate new assets in your clusters. Gather count of the assets that are removed or deleted from the clusters.

Possible values are between 5 minutes to one year. The default value is 30 minutes.

To set the autodiscovery frequency

1. On the left, click **Workloads > Kubernetes**.
2. On top-right, click **Kubernetes settings > Autodiscovery**.
3. Click **Edit**, near **Frequency**.
4. Enter the number of hours after which NetBackup runs autodiscovery. Click **Save**.

Run full and incremental discovery

Once the Kubernetes cluster is added, autodiscovery cycle is triggered to discover all the assets available on the Kubernetes cluster. The first autodiscovery of the day is a full discovery and subsequent autodiscoveries are incremental.

To run a discovery

1. On the left, click **Workloads > Kubernetes**.
2. In the **Kubernetes clusters** list, locate the cluster name. Then click **Actions > Discover now**.

Here, incremental discovery fetches only those NetBackup assets which are changed in the cluster since last discovery run. Therefore, the first discovery is full and all subsequent ones are incremental discovery.

Configure permissions

Using manage permissions, you can assign different access privileges to the user roles. For more information see the *Managing role-based access control* chapter in the *NetBackup Web UI Administrator's Guide*.

Add protection to the assets

The **Namespaces** tab (**Workloads** > **Kubernetes**), lets you monitor the assets in your Kubernetes clusters, see their protection status, and easily add protection to any unprotected assets. You can also take a quick backup of asset using the backup now feature. This feature creates a one-time backup of the selected asset without affecting any scheduled backups.

The Namespaces tab displays with all the discovered and imported Kubernetes assets that NetBackup can protect. This tab displays the following information:

- **Namespaces**: Display name of the asset.
- **Cluster**: The cluster to which the asset belongs.
- **Protected by**: Name of the protection plan applied to the asset.
- **Last successful backup**: Date and time of the last successful backup of the asset.

You can perform the following action in the **Namespaces** tab.

To add protection to an unprotected asset

1. On the left, click **Workloads > Kubernetes**.
2. Select the option in the rows of the assets. Click **Add protection** on top right. Alternatively, click the Actions menu in the row of the asset and click **Add protection**.
3. Select a protection plan from the list and click **Next**. In the next page, click **Protect**.

To quickly back up an asset

1. Select the option in the rows of the assets, click **Backup now** on top right. Alternatively, click the Actions menu in the row of the asset and click **Backup now**.
2. In the next page,
   - If you backup an already protected asset, select a protection plan from the list of plans to which the asset is already subscribed, and click **Start backup**.

- If you are backing up an unprotected asset, select a protection plan from the available plans for the asset, click **Start backup**.

Scan for malware

NetBackup version 10.4 provides support for scanning Kubernetes assets for malware through the Kubernetes workload.

Assets by workload type

This section describes the procedure for scanning VMware, Universal shares, Kubernetes and Cloud VM assets for malware.

Ensure that you meet the following prerequisites:

- The backups were performed with a storage server at NetBackup 10.1 or later.
- Backup images are stored on MSDP storage only with instant access capability, for the supported policy type only.
- The last backup must be successful.
- You must have an RBAC role with permissions to perform malware scans.

To scan the supported assets for malware, perform the following:

1. On left, select the supported workload under **Workloads**.
2. Select the resource which has backups completed (for example, VMware/Cloud VM, Universal shares, Kubernetes and so on).
3. Select **Actions > Scan for malware**.
4. On the **Malware scan** page, perform the following:
   - Select the date range for the scan by selecting **Start date/time** and **End date/time**.
   - Select **Scanner host pool**
   - From the **Select current status of malware scan** list select one of the following:
     - **Not scanned**
     - **Not infected**
     - **Infected**
     - **All**

5 Click **Scan for malware**.

Note: The malware scanner host can initiate a scan of three images at the same time.

6 After the scan starts, you can see the **Malware Scan Progress** on **Malware Detection**, the following fields are visible:

- **Not scanned**
- **Not infected**
- **Infected**
- **Failed**

  Note: Any backup images that fail validation are ignored.

- **In progress**
- **Pending**

Chapter 5

Managing Kubernetes intelligent groups

This chapter includes the following topics:

About intelligent group

You can create and protect a dynamic group of assets by defining the intelligent asset groups based on a set of filters called queries. NetBackup selects the Kubernetes namespaces based on the queries and then adds them to the group. An intelligent group automatically reflects changes in the asset environment and eliminates the need to manually revise the list of assets in the group when the assets are added or removed from the environment.

When you a apply protection plan to an intelligent group, all the assets satisfying the query conditions are automatically protected.

Note: You can create, update, or delete the intelligent groups only if your role has the necessary RBAC permissions for the assets that you require to manage. The NetBackup security administrator can grant you access for an asset type (clusters, namespaces, and VMGroup). Refer to the *NetBackup Web UI Administrator's Guide*.

Create an intelligent group

Use the following procedure to create an intelligent group for Kubernetes workload in the NetBackup web UI.

Note: With Virtualization support in Kubernetes, you can create intelligent groups with filtering the namespaces based on specific resource kinds. Virtual machines, Persistent volume, and Persistent volume claims are resource kinds available to filter.

To create an intelligent group

1. On the left click **Kubernetes**, under **Workloads**.
2. Click the **Intelligent groups** tab and then, click **+ Add**.
3. Enter a name and description for the group.
4. Under **Clusters** section, click **Add clusters**
5. In the **Add clusters** window, select one or multiple clusters from the list and click **Select** the selected clusters are added to the intelligent group.

   Note: Intelligent group can be created across multiple clusters. Ensure that you have the required permissions to add clusters in the group. To view and manage the group, the group administrator must have the view and manage permission for the selected clusters and groups.

   ---

6. Under the **Select assets** section, do one of the following:
   - Select **Include all assets**.
     This option uses a default query to select all assets for backup when the protection plan runs.
   - To select only the assets that meet specific conditions, create your own query: Click **Add condition**.
   - To add label conditions for the assets, click **Add label condition** to add

7 To add a condition, use the drop-downs to select a keyword and operator and then enter a value.

To change the effect of the query, click **+ Condition** and click **AND** or **OR**, then select the keyword, operator, and value for the condition.

Note: To add label conditions, click **Add label condition** enter the label key and value.

Note: You can choose to have only a label key in the condition without the label value. As value is optional parameter to add a label condition.

Note: To add sub-query, click **Add sub-query**. You can add multiple level sub-queries.

---

8 To test the query, click **Preview**.

The query-based selection process is dynamic. Changes in the Kubernetes cluster can affect which assets the query selects when the protection plan runs. As a result, the assets that the query selects later when the protection plan runs may not be identical to those currently listed in the preview.

Note: When using queries in **Intelligent groups**, the NetBackup web UI might not display an accurate list of assets that match the query if the query condition has non-English characters.

Using the `not equals` filter condition on any attribute returns assets including those that have no value (null) present for the attribute.

Note: When you click **Preview** or you save the group, the query options are treated as case-sensitive when the assets are selected for the group.

---

9 To save the group without adding it to a protection plan, click **Add**.

10 To save the group with adding it to a protection plan, click **Add and protect**.

11 To subscribe the group to a protection plan, click **Add protection**.

Select the group and apply a protection plan to it, click **Protect**.

The selected asset group is successfully subscribed to the protection plan.

Limitations while adding label conditions to the assets

If you have a combination of conditions and labels, then you must first define a namespace condition and then a label condition.

Note: For conditions, only a namespace value is allowed.

Delete an intelligent group

To delete an intelligent group

1. On the left, click **Kubernetes**, under **Workloads**.
2. Locate the group, under the **Intelligent groups** tab.
3. If the group is not protected, select it and click **Delete**.
4. If the group is protected, select it, and then click **Remove protection** to remove all protection plans.
5. Then select that group under the **Intelligent groups** tab and click **Delete**.

Edit an intelligent group

You can edit the name and description details of an intelligent group. You can edit certain settings for a protection plan, including schedule backup windows and other options.

To edit an intelligent group

1. On the left, click **Kubernetes**, under **Workloads** .
2. On the **Intelligent groups** tab, click the group that you want to edit the protection for.
3. Do one of the following:
   - Click **Edit name and description** to edit name and description of the selected group and then click **Save**.
   - On the **Assets** tab, click **Edit** to add or remove the cluster. You can update the query condition for the selected asset and then, click **Save**.
     You can edit the cluster list in the group, add or remove the clusters from the group. You can also modify the query condition for the selected asset group.
   - On the **Permissions** tab, click **Add** to update the permissions for available roles and then, click **Save**.

Chapter 6

Managing Kubernetes policies

This chapter includes the following topics:

Create a policy

Use the following procedure to create a backup policy using policy type Kubernetes within the NetBackup web UI.

To create a policy

1. On the left, select **Protection > Policies**.
2. Click **Add**.
3. On the **Attributes** tab, do the following:
   - Enter policy name in the **Policy name** field.
   - Select the **Policy type** as Kubernetes.
   - Select the **Policy storage** that you want to use.
   - Select or configure any other policy attributes.
4. On the **Schedules** tab, configure all the necessary schedules. For example, Full and incremental schedules.
5. On the **Kubernetes** tab, do any of the following:
   - Select the **Intelligent group** to add new or existing intelligent group that you want to protect.
   - Select the **Namespace** to add namespaces from list that you want to protect.

6 On the **Resource kind and label selection** tab, do any of the following:

- Select the **Include all resource kinds in the backup**option to include all resource kinds in the backup.
- Select the **Exclude the following resource kinds from the backup** option and enter manually the resource kind or select the resource kinds you want to exclude the backup.
- In the **Label selection**, click **Add +** to the label queries you want to add.

7 Click **Create**.

Chapter 7

Protecting Kubernetes assets

This chapter includes the following topics:

Protect an intelligent group

You can create the Kubernetes specific protection plans for your Kubernetes workloads. Then you can subscribe an intelligent group to a protection Plan.

Use the following procedure to subscribe an intelligent group to a protection plan.

Note: The RBAC role that is assigned to you must give you access to the intelligent groups that you want to manage and to the protection plans that you want to use.

To protect an intelligent group

1. On the left, click **Kubernetes**.
2. On the **Intelligent groups** tab, click the box for the groups and then, click **Add protection**.
3. Select a protection plan and click **Next**.
4. Select a group and click **Protect** to subscribe to a protection plan.

'Backup now' option for immediate protection

Apart from the scheduled protection plans, you can also use the **Backup now** option to backup a group immediately, to safeguard against any unplanned circumstances.

Remove protection from an intelligent group

You can unsubscribe an intelligent group from a protection plan. When an intelligent group is unsubscribed from a protection plan, backups are no longer performed.

To remove protection from an intelligent group

1. On the left, click **Kubernetes**, under **Workloads**.
2. On the **Intelligent groups** tab, click the group that you want to remove the protection for.
3. Click **Remove Protection > Yes**.

Configure backup schedule

You can add backup schedule in the **Attributes** tab of the **Add backup** schedule dialog, while creating a protection plan for the Kubernetes workloads.

See the *Managing protection plans* section of the *NetBackup Web UI Administrator's Guide*, for details of how to create a protection plan.

To add backup schedule for the Kubernetes backup job

1. On the left, click **Protection** > **Protection plans** and then click **Add**.
2. In **Basic properties**, enter a **Name**, **Description**, and select **Kubernetes**, from the **Workload** drop-down list.
3. Click **Next**. In **Schedules**, click **Add schedule**.

   In the **Add backup schedule** tab, you can configure the options for retaining the backup and the snapshot.

4 From the **Recurrence** drop-down, specify the frequency of the backup.

5 In the **Snapshot and backup copy** options, do any of the following:

- Select **Create backup from snapshot** option, to configure backup from snapshot for the protection plan. Specify retention period for the backup from snapshot using the **Keep backup for** drop-down.

  Note: Only full backup schedules are supported on the Kubernetes workloads. You can set the backup duration in hours, days, weeks, months, and years.

  By default, four weeks is the backup retention duration.

  Note: You must select the **Create backup from snapshot** option to enable the replicate and duplicate options for backup copy.

- If you do not select **Create backup from snapshot** option, then by default, **Snapshot only storage** backup will get configured to run the backup jobs.
- Select **Create a replica copy (Auto Image Replication) of the backup from snapshot** option to create a replica copy of the backup.
- Select **Create a duplicate copy of the backup from snapshot** option to create a duplicate copy of the backup.

6 Continue creating the schedule in the **Start window** tab, as described in the *Managing protection plans* section of the *NetBackup Web UI Administrator's Guide*

7 Continue to configure the **Storage options** for backup from snapshot, as described in the *Managing protection plans* section of the *NetBackup Web UI Administrator's Guide*

Configure backup options

You can configure backup options for a protection plan.

See the *Managing protection plans* section of the *NetBackup Web UI Administrator's Guide*, for details of how to create a protection plan.

To configure the backup options while configuring a protection plan

1 In the **Backup options** page, under the **Resource kind selection** section,

- By default, **Include all resource kinds in the backup** option is selected to include all resource kinds for the backup job.

- Select **Exclude the following resource kinds from the backup** option to exclude the resource kinds from the backup job. Click **Select** to choose the resource kinds from the static list. The selected resource kinds are displayed in the text field or you can manually enter the custom resource definition (CRD) with correct format (type.group). You can delete the selected resource kinds from the exclude list.

  In case, the custom resource kind definitions are not present in the static list then you can enter custom resource definition (CRD) manually. For example: demo.nbu.com.

Note: Exclude list of resource kinds takes precedence in terms of mapping the resources over the labels selected for backup.

---

2 Under the **Labels selection** section, click **Add** to add the labels to map its associated resources for the backup, enter the label prefix and key, and then select a operator. All associated resources of the included labels are mapped for the backup job.

Following are the four operators which you can add to a label:

- Enter a label key equal to a value.
- Enter a label key which already exists, without any values.
- Enter a label key which is in a set of values.
- Enter a label key not in a set of values.

You can add multiple values for in/not in operators in the set of values with comma separated.

Note: Selected labels must be present at the time of backup to ensure that the conditions are applied successfully.

Note: Label selection must only be exclusive of selecting any resource kind which doesn't contradict between multiple label conditions.

Review page displays the excluded list of resource kinds and the selected labels for inclusions, and the selected storage units selected.

Note: You can edit or delete the protection plan created for Kubernetes workloads.

You cannot customize the protection plan created for Kubernetes workloads.

Configure backups

NetBackup allows you to run two types of backup jobs in Kubernetes workload: Snapshots only and Backup from Snapshot. Follow the steps to configure a backup job for Kubernetes operator.

To perform backup on Kubernetes workload

1. On the left, click **Protection** > **Protection plans** and then click **Add**.
2. In **Basic properties**, enter a **Name**, **Description**, and select **Kubernetes**, from the **Workload** drop-down list.
3. Click **Next**. In **Schedules**, click **Add schedule**.

   In the **Add backup schedule** tab, you can configure the options for retaining the backup and the snapshot.
4. From the **Recurrence** drop-down, specify the frequency of the backup.
5. In the **Snapshot and backup copy** options, do any of the following:
   - Select **Create backup from snapshot** option, to configure backup from snapshot for the protection plan. Specify retention period for the backup from snapshot using the **Keep backup for** drop-down.

     Note: Only full backup schedules are supported on the Kubernetes workloads. You can set the backup duration in hours, days, weeks, months, and years. By default, four weeks is the backup retention duration.

     Note: You must select the **Create backup from snapshot** option to enable the replicate and duplicate options for backup copy.
   - If you do not select **Create backup from snapshot** option, then by default, **Snapshot only storage** backup is configured to run the backup jobs.
   - Select**Create a replica copy (Auto Image Replication) of the backup from snapshot** option to create a replica copy of the backup.
   - Select **Create a duplicate copy of the backup from snapshot** option to create a duplicate copy of the backup.

6 Continue creating the schedule in the **Start window** tab, as described in the *Managing protection plans* section of the *NetBackup Web UI Administrator's Guide*

7 Continue to configure the **Storage options** for backup from snapshot, as described in the *Managing protection plans* section of the *NetBackup Web UI Administrator's Guide*

- While selecting a storage for **Backup from Snapshot** option, the selected storage unit must have the media servers of NetBackup version 10.0 or later.
- Media server managing the storage must have access to the selected Kubernetes clusters.
- Media server must be able to connect with the API server. The port corresponding to the API server must be open for the outbound connection from the media server. The datamover pod must be able to connect to the media server.

Configure Auto Image Replication (A.I.R.) and duplication

The backups that are generated in one NetBackup domain can be replicated to storage in one or more target NetBackup domains. This process is referred to as Auto Image Replication (A.I.R.).

NetBackup Kubernetes supports Auto Image Replication from a Media Server Deduplication Pool (MSDP) in one NetBackup domain to a Media Server Deduplication Pool (MSDP) in another domain. NetBackup uses storage lifecycle policies (SLP) in the source domain and the target domain to manage A.I.R. operations.

The Auto Image Replication (A.I.R.) supports all schedule types which includes Differential Incremental, Cumulative Incremental, and Automatic schedules.

For more information about configuring Auto Image Replication, refer to the *About NetBackup replication* chapter in the *NetBackup Administrator's Guide, Volume I.*

Note: A Kubernetes A.I.R. configuration requires NetBackup primary and media servers of version 10.0.1 or later.

To configure Auto Image Replication (A.I.R.) and duplication for Kubernetes backups

1 Configure an Auto Image Replication between two NetBackup primary servers.

- Establish the trust relationship between two primary servers for interdomain operations.
  - Log on to the source primary server, on the left, click **Hosts > Host properties** to build a connection between a source and target primary server.
  - Select a source primary server. If necessary, and click **Connect**. Then click **Edit primary server**.
  - Click **Servers**. On the **Trusted primary servers** tab, click **Add** to add a source server.
    - Click **Validate Certificate Authority**, then click **Next** to proceed with the certificate authority validation.
    - To create a trusted primary server, select from the following options:
      - Select **Specify authentication token of the trusted primary server** to add an existing token or create a new token for the source primary server.
      - Select **Specify credentials of the trusted primary server** to add user credentials for the source primary server.
    - Click **Create trust**.

    The database for the host properties is updated successfully.
  - Click **Save**.

2. Configure a Media Server Deduplication Pool (MSDP) storage in the source primary server and add a replication target in MSDP disk pool.
   - On the left, click **Storage > Disk storage**.
   - Add an MSDP storage and disk pool.
   - Click the **Disk pools** tab and click **Add**.
     - Select a trusted primary server and a target storage server.
     - Add user credentials for the replication target server in the **Username** and **Password** fields.
   - Click **Add**.
3. Create SLP with **Import** operation in the target primary server.
   - On the left, click **Storage > Storage lifecycle policies**. Then click **Add**.
   - In the **Storage lifecyle policy name** field, enter the policy name and then, click **Add**.
   - From the **Operation** list, select **Import**.

- In the **Destination storage** list select an MSDP storage unit.
- Click **Create**.

4 Create Kubernetes protection plan with the **Create backup from snapshot** option to enable the replicate copy option.

On the left, click **Protection > Protection plans**. On the **Schedules** tab, click **Add schedule**.

5 In the **Snapshot and backup copy options**section, select **Create backup from snapshot** option to enable the replicate and duplicate copy options.

6 Select **Create a replica copy (Auto Image Replication) of the backup from snapshot** option, and set a time duration to retain the replica copy.

Note: Auto Image Replication can only be created on the trusted NetBackup primary servers.

---

7 Select **Create a duplicate copy of the backup from snapshot** option and set a time duration to retain the duplicate copy.

8 Click **Add**.

9 Continue creating the schedule in the **Start window** tab, as described in the*Managing protection plans* section of the *NetBackup Web UI Administrator's Guide*.

10 Click **Next**.

11 On the **Storage options** tab, select the storage units to backup from snapshot, replicate, or duplicate copy.

Note: For Backup from snapshot and duplication, you can add simple storage units. But for replication, you must add a trusted storage unit with an import storage lifecycle policies (SLPs).

---

12 To the right of the selected backup options, click **Edit** to modify selected the storage units for backup.

- For the replica copy option, select the primary server for replication copy. Then click **Next**.
- Select an import storage lifecycle policy that is defined in the trusted server and then click **Use selected replication target**.

13 Continue with the steps in the wizard.

Configure storage units

You can configure all types of storage units for backup in a protection plan.

Note: All storage types supported in Storage Lifecycle Policy (SLP) are supported for backup jobs.

To configure a storage unit for backup

1. On the left, click **Storage > Storage units**.
2. Click the **Storage units** tab and then click **Add** to add a storage unit configuration.
3. Select the storage type from the list.
4. Select a **Category** and then click **Start**.
5. Enter the storage unit name in the **Name** field.
6. In the **Maximum concurrent jobs** field, choose the maximum number for the backup jobs.
7. In the **Maximum fragment size** field, choose the maximum number for the storage unit fragment size and then click **Next**.
8. In **Disk pool**, select the disk pool you want to use in the storage unit and then click **Next**.
9. The **On demand only** option specifies whether the storage unit is available exclusively on demand. A policy or schedule must be explicitly configured to use this storage unit.
10. On the **Media server** tab, select the media servers that you want to use and then click **Next**. You can have NetBackup select your media server automatically or you can select your media servers manually using the radio buttons.
    - All media servers must of NetBackup version 10.0 or later
    - All media server managing the storage must have access to the selected Kubernetes clusters.
    - Media server must be able to connect with the API server. The port corresponding to the API server must be open for the outbound connection from the media server. The datamover pod must be able to connect to the media server.

11 Review the setup of the storage unit and then click **Save**.

12 To check details of a scheduled backup or backup now job, In the **Activity monitor** tab, click the **Job ID**, to view the backup job details. For file mode, you can see the total number of backed up files for every image in the **Job Details** section.

Volume mode support

NetBackup Kubernetes supports with the following capabilities:

- Backup and restore of Persistent Volume Claims (PVCs) of mode file system and/or block for the Container Storage Interface (CSI) providers which supports the following features:
  - PVC snapshot capability.
  - PVC volume provisioning based on the Network File System (NFS) or other non-block storage.
  - PVC volume provisioning based on block storage.

Note: Backup and restore of a namespace with mixed volumes (VolumeMode: Filesystem and Block) is now supported from NetBackup 10.3 onwards.

Configure application consistent backup

Some pods that are running applications, such as databases, which require additional procedures to obtain application consistent backups.

Application consistent backups require a mechanism to understand the application metadata, its state in memory, and the persistent data that resides on the persistent storage. To achieve a healthy state during restore, an application consistent backup across all these Kubernetes resources helps streamline the recovery process. These procedures are not required if only a crash consistent backup is required.

The application has vendor-documented steps to pause Input and Output (I/O) operations to perform an application consistent snapshot. This varies from one application to another, so the custom nature of these procedures is important. The content of these procedures is the customer's responsibility.

For protecting Kubernetes workloads with NetBackup, the method to achieve application consistent snapshots is to apply application pod annotations that leverage backup hooks. Kubernetes annotations are simply metadata which can be applied to any Kubernetes resources. Hooks within Kubernetes are user-defined actions and can be any command or multiple commands. Within your Kubernetes

infrastructure, apply these annotations and hooks to any application pod that requires a quiesce state.

Backup hooks are used for both pre (before the snapshot) and post (after the snapshot) processing. In the context of data protection, this usually means that a netbackup-pre-backup hook calls a quiesce procedure or command, and the netbackup-post-backup hook calls an un-quiesce procedure or command. Each set of hooks specifies the command, as well as the container where it is applied. Note that the commands are not executed within a shell on the containers. Thus, a full command string with the directory is used in the given examples.

Identify the applications that require application consistent backups and apply the annotation with a set of backup hooks as part of the configuration for Kubernetes data protection.

Add an annotation to a pod, use the Kubernetes User Interface (UI). Alternatively, use the `kubectl` annotate function on the Kubernetes cluster console for a specific pod or label. The methods to apply annotations may vary depending on the distribution, therefore the following examples focuses on the `kubectl` command, based on its wide availability in most distributions.

Additionally, annotations can be added to the base Kubernetes objects, such as the deployment or replica set resources to ensure the annotations are included in any newly deployed pods. The Kubernetes administrator can update annotations dynamically.

Labels are key-value pairs which are attached to the Kubernetes objects such as Pods, or Services. Labels are used as attributes for objects that are meaningful and relevant to the user. Labels can be attached to objects at creation time and subsequently added and modified at any time. Kubernetes offers integrated support for using these labels to query objects and perform bulk operations on selected subsets. Each object can have a set of key-value labels defined. Each Key must be unique for a given object.

As an example of formatting and syntax of the label metadata:

```
"metadata": {"labels": {"key1":"value1","key2":"value2"}}
```

Either specify the pod name specifically, or a label that applies to the desired group of pods. If multiple annotation arguments are used, then specify the correct JSON format, such as a JSON array: '["item1","item2","itemn"]'`# kubectl annotate pod [ {pod_name} | -l {label=value}] -n {the-pods-namespace_name} [annotation syntax - see following]`

This method can be combined with `&&` to join multiple commands if some applications require multiple commands to achieve the desired result. The commands specified are not provided by Veritas, and the user must manually customize the application pod. Replace {values} with the actual names used in your environment.

Note: All`kubectl` commands must be defined in a single line. Be careful when you copy or paste the following examples.

After upgrading to NetBackup 10.2, update the annotations to these new netbackup-pre and netbackup-post backup hooks that now include the "netbackup" prefix:

```
netbackup-pre.hook.back.velero.io/command
netbackup-pre.hook.backup.velero.io/container
netbackup-post.hook.back.velero.io/command
netbackup-post.hook.backup.velero.io/container
```

MongoDB example using the pod name

Following are the commands to lock and unlock a MongoDB 4.2.23 database:

```
# mongo --eval "db. fsyncLock ()"

# mongo --eval "db.fsyncUnlock()"
```

This translates into the following single command to set both the pre and post backup hooks for MongoDB. Note the special syntax to escape special characters as well the brackets ([]), single and double quotes and commas (,) used as part of the JSON format:

```
# kubectl annotate pod {mongodb-pod-name} -n {mongodb namespace}
netbackup-pre.hook.back.velero.io/command='["/bin/bash", "-c", "mongo
--eval \"db.fsyncLock()\""]'
netbackup-pre.hook.backup.velero.io/container={mongodb-pod-name}
netbackup-post.hook.backup.velero.io/command='["/bin/bash","-c","mongo
--eval \"db.fsyncUnlock()\""]'
netbackup-post.hook.backup.velero.io/container={mongodb-pod-name}
```

MySQL example using the labels

Following are the commands to quiesce and un-quiesce the MySQL database:

```
# mysql -uroot -ppassword -e "flush tables with read lock"

# mysql -uroot -ppassword -e "unlock tables"
```

This translates into the following single command to set both the pre and post backup hooks for MySQL. In this example, we used a label instead of a pod name, so the label can annotate multiple pods at once. Note the special syntax to escape special characters as well the brackets ([]), single and double quotes and commas (,) used as part of the JSON format:

```
# kubectl annotate pod -l label=value -n {mysql namespace}
netbackup-pre.hook.backup.velero.io/command='["/bin/bash", "-c",
"mysql -uroot -ppassword -e \"flush tables with read lock\""]'
netbackup-pre.hook.backup.velero.io/container={mysql container name}
netbackup-post.hook.backup.velero.io/command='["/bin/bash", "-c",
"mysql -uroot -ppassword -e \"unlock tables\""]'
netbackup-post.hook.backup.velero.io/container={mysql container name}
```

Postgres example using labels

Following are the commands to quiesce and un-quiesce the PostgreSQL database:

```
# Psql -U postgres -c "SELECT pg_start_backup('tagvalue');"
```

```
# psql -U postgres -c \"SELECT pg_stop_backup();"
```

This translates into the following single command to set both the pre and post backup hooks for Postgres. In this example, we used a label instead of a pod name, so the label can annotate multiple matching pods at once. Labels can be applied to any Kubernetes object, and in this case, we are using them to provide another way to modify a specific container and select only certain pods. Note the special syntax to escape special characters as well the brackets ([]), single and double quotes and commas (,) used as part of the JSON format:

```
# kubectl annotate pod -l app=app-postgresql -n {postgres namespace}
netbackup-pre.hook.backup.velero.io/command='["/bin/bash", "-c",
"psql -U postgres -c \"SELECT
pg_start_backup(quote_literal($EPOCHSECONDS));\""]'
netbackup-pre.hook.backup.velero.io/container={postgres container
name} netbackup-post.hook.backup.velero.io/command='["/bin/bash",
"-c", "psql -U postgres -c \"SELECT pg_stop_backup();\""]'
netbackup-post.hook.backup.velero.io/container={postgres container
name}
```

NGINX application example without container hook

Following are the commands to quiesce and un-quiesce the Nginx application:

```
# /sbin/fsfreeze --freeze /var/log/nginx
```

```
# /sbin/fsfreeze --unfreeze /var/log/nginx
```

This translates into the following single command to set both the pre and post backup hooks for NGINX. In this example, we will omit the container hooks, and this will modify the first container that matches the pod name by default. Note the special syntax to escape special characters as well the brackets ([]), single and double quotes and commas (,) used as part of the JSON format:

```
# kubectl annotate pod {nginx-pod-name} -n {nginx namespace}
netbackup-pre.hook.backup.velero.io/command='["/sbin/fsfreeze",
"--freeze", "/var/log/nginx"]'
netbackup-post.hook.backup.velero.io/command='["/sbin/fsfreeze",
"--unfreeze", "/var/log/nginx"]'
```

Cassandra example

Following are the commands to quiesce and un-quiesce the Cassandra database:

```
# nodetool flush
```

```
# nodetool verify
```

This translates into the following single command to set both the pre and post backup hooks for Cassandra. Note the special syntax to escape special characters as well the brackets ([]), single (‘’), and double quotes (“”) and commas (,) used as part of the JSON format:

```
# kubectl annotate pod {cassandra-pod} -n {Cassandra namespace}
netbackup-pre.hook.backup.velero.io/command='["/bin/bash", "-c",
"nodetool flush"]'
netbackup-pre.hook.backup.velero.io/container={cassandra-pod}
netbackup-post.hook.backup.velero.io/command='["/bin/bash", "-c",
"nodetool verify"]'
netbackup-post.hook.backup.velero.io/container={cassandra-pod}
```

Note: The examples provided are only initial guide, and specific requirements for each workload must include the collaboration between backup, workload, and Kubernetes administrators.

At present, Kubernetes do not support an on-error hook. If the user-specified command fails, the backup snapshot does not proceed.

The default timeout value for the command to return an exit status is 30 seconds. But this value can be changed with the following hooks as annotations to the pods:

```
netbackup-pre.hook.backup.velero.io/timeout=#in-seconds#
```

```
netbackup-post.hook.backup.velero.io/timeout=#in-seconds#
```

Chapter 8

Managing image groups

This chapter includes the following topics:

About image groups

For every Kubernetes recovery point, an image group is created. An image group may include multiple images depending upon number of eligible persistent volume claims in a namespace.

A separate image is created for metadata and one image is created for every persistent volume claim.

Recovery point detail API is used to get the details about all the backup ids, resource names, copy completion status of an image group.

To support the backup from snapshot operation on the Kubernetes workload, multiple backup images are created to perform backup from snapshot for a single namespace.

For Kubernetes backup operation, a separate backup image is created for every persistent volume. All the images that are created must be grouped together to perform certain (restore, delete, import and so on) operations successfully.

Image expire

To reclaim the storage space occupied by the expired images, you need to delete those images.

Following are the important points related to image expiration.

For a recovery point consisting of multiple images:

- If you have expired a single image in an image group, then it does not lead to automatic expiration of remaining images. You must explicitly expire all images in an image group.
- If you have expired a few images then the recovery point will be incomplete. Restore operation is not supported for incomplete recovery point.
- If you have changed the expiration time for any of the images, then the expiration time for rest of the images must be changed. Otherwise, the expiration time for the images corresponding to recovery point gets skewed, leading to incomplete recovery point at some point in time.

Image import

Kubernetes recovery point may consist of multiple images. To perform restore operation, all the images corresponding to the recovery point must be imported. Otherwise, the recovery point is marked as incomplete and restore is not performed.

For more information, refer to the *About importing backup images* section in the *NetBackup™ Administrator's Guide, Volume I*

Image copy

You can create an image copy with two types of backup operations:

1. **Snapshot** is the default copy and is marked as copy no 1.
2. **Backup from snapshot** is marked as copy no 2.

Whenever any backup-now operation or scheduled backup triggers, **Snapshot** is taken. But, **Backup from snapshot** is optional as it depends whether **Backup from snapshot** option is selected or not while creating a protection plan.

An image group consists of asset images for metadata and Persistent Volume Claims (PVC). Every copy has one image for namespace and one image for each PVC present in the namespace.

Recovery point detail API is used to identify the copy completion status of an image. This API also details all the backup ids and resource name present in the respective copy. This complete or incomplete status of the image copy helps in restore functionality as an error is thrown if someone tries to restore the asset from an incomplete image copy.

Incomplete image copy

Following are the conditions for an incomplete image:

1. When the snapshot job or backup from snapshot job is in progress then the corresponding copy is shown as incomplete copy.
2. If backup activity of any PVC fails, then the copy is marked as incomplete.

3. If the child image of a copy gets expired (with more than 1 child), then the copy is marked as incomplete.

Chapter 9

Protecting Rancher managed clusters in NetBackup

This chapter includes the following topics:

Add Rancher managed RKE cluster in NetBackup using automated configuration

Follow the steps to add Rancher managed RKE cluster in NetBackup using automated configuration.

To add Rancher managed RKE cluster in NetBackup using automated configuration

---

Note: Extract the Global Rancher Management server certificate. This CA cert can be a default generated cert by rancher or configured by using a different/external CA (Certifying Authority) during the management servers creation.

---

1. Extract the CA cert: Navigate to the **Rancher Management Server UI**> Open the left side panel **Global Settings** > Under **CA Certs**, click the **Show CA Certs** button. Extract the complete CA cert value in a temporary file.

   Note: Make sure you extract the complete value which includes the starting and ending lines.

2. The CA certificate value is added in the secret which is created before Kubernetes operators helm install
3. To Extract the token: Open the **Rancher Management Server UI** > Open the left side panel > Under the **Explore Cluster** Section > Navigate to the cluster you want to protect > Click the **Download KubeConfig** icon on the top right corner.
4. Download the cluster's **KubeConfig** using the icon and the token field is present inside the file.
5. Extract the token: value without the double quotes " " from this downloaded Kubeconfig file.
6. This configuration process relies on a secret with the following naming pattern `(<kops-namespace>-nb-config-deploy-secret).`

   The secret have the values that are extracted in steps 1 & 3.
7. Create a yaml file `nb-config-deploy-secret. yaml` with the following format and enter the values in all the fields.

```
apiVersion: v1
kind: secret
metadata:
   name: <kops-namespce>-nb-config-deploy-secret
   namespace:  <kops-namespace>
type: Opaque
stringData:
#All the 3 fields are mandatory here to add a Rancher managed RKF2 cluste
     apikey: A_YoUkgYQwPLUkmyj9Q6A1-6RX8RNY-PtYX0SukbqCwIK-osPz8qVm9zCL9p
     k8stoken: kubeconfig-user-mvvgcm8sq8:nrscvn8hj46t24r2tjrxd2kn8tzo2bg
     k8scacert: |
-------BEGIN CERTIFICATE-----
MIIDDDCCAfSgAwIBAgIBATANBgkqhkiG9w0BAQwIgYDVQQDDBtpbmdy
ZXNzLW9wZXJhdG9yQDE2ODc1MzY4NjgWHhcNMjMwNjIzMTYxNDI3WhcNMjUwNjIy
XtXqbaBGrXIuCCo90mxv4g==
-------END CERTIFICATE------
```

8. Run the command: `kubectl apply -f nb-config-deploy-secret. yaml`
9. For the rest of the inputs in the values.yaml file of your helm chart refer the *Automated Configuration* section of the *Kubernetes Quick Start* guide and enter all the values which are necessary for a complete setup.
10. Run Helm install command on the NetBackup Kubernetes operator chart if all the necessary simplified install inputs are added in the `values.yaml` file, and the automated configuration pod **<kops-namespace>-netbackup-config-deploy** should start up.
11. Observe the **<kops-namespace>-netbackup-config-deploy logs** to identify if the updated secret value is picked up by the config-deploy pod.
12. Once the **config-deploy** pod performs its tasks the cluster is successfully added in NetBackup, and a discovery request is in progress or successfully completed. Perform another credential validation and manual discovery from the NetBackup web UI to ensure the process is working fine.

Add Rancher managed RKE cluster manually in NetBackup

Follow the steps to add Rancher managed RKE cluster manually in NetBackup.

Kubernetes credential creation for NetBackup

Navigate to the NetBackup web UI > **Credential Management** > **Named Credential** > **Add** > **Add credentials** > select the credential store as NetBackup > select the Kubernetes in the **Category** field, enter the token and CA certificate which were extracted from the Global Rancher Management platform UI earlier and then save this credential.

To add Rancher managed RKE cluster manually in NetBackup

1. **External CA Cert**: The External CA certificate is required for NetBackup to communicate successfully with the cluster, if there is a different CA (Certifying Authority) used to configure the certificates for external access.
   - Navigate to the **Rancher Management Server UI** > Open the left side panel **Global Settings** > Under **cacerts**, click the **showcacerts** button. Extract this complete CA certificate value in a temporary file
   - For example, `<cacert-value-file>`
2. **Service account CA Cert**:

Note: You must do the following the step as there is a different CA (Certifying Authority) configured for external access of the Kubernetes API server compared to the service account CA cert which is available within the cluster. Hence, these two CA certificates must be combined.

To get the service account CA certificate, run the following commands on the Linux cluster host.

- Get the service account secret name available on the Kubernetes operator's namespace using the following command:

  ```
  kubectl describe serviceaccount <kopsnamespace>-backup-server
  -n <kopsnamespace> | grep Tokens | cut -d ":" -f 2
  ```

- Get the CA certificate in the base 64 decoded form from this service account secret using this command:

  ```
  kubectl get secret <output-from-previous-command> -n
  <kopsnamespace> -o jsonpath='{. data.ca\.crt}' | base64 -d
  ```

  Entire output of this command must be appended to the temporary file which we created in step 1.

3 Append the output that was generated after Step 2 at the end of the `<cacert-value-file>` file. The necessary external and internal CA cert values have are extracted and available in the file **<cacert-value-file>**. The CA cert values are base 64 decoded form which you have to encode again while creating credentials on NetBackup.

4 **Token**: **Rancher Management Server** UI > Open the left side panel > Under the **EXPLORE CLUSTER** section > Navigate to the cluster you want to protect > **Kubeconfig** icon on the top right corner.

- Extract the token: value without the double quotes " " from the downloaded Kubeconfig file (using the **Download KubeConfig**) into a temporary file **<token-value-file>**.
- Both these fields **token** and **cacert** are required in the base64 encoded form to add in the NetBackup credentials for Kubernetes.
- To get the base64 encoded version of both these extracted values using the following base64 command:

  #Use a Linux VM to encode the values for this step #Note: the flag -w0 has the zero digit and not a 0 Symbol.
  #For CA cert:
  Cat <cacert-value-file>| base64 -w0
  Paste this output in the CA certificate field in the NetBackup credentials creation page.

#For Token:

Paste this output in the Token field in the NetBackup web UI's credentials creation page.

- Use these values in the NetBackup web UI > **Credential management** > **Named Credentials** > **Add** to add the valid Rancher credentials in NetBackup.
- Once the credentials are created, add the Kubernetes cluster in NetBackup using the name shown in the following cluster-info output.

To get cluster information output run the following commands

1. The cluster info output must be in the following example format: `[root@master-0~] # kubectl cluster-info`
2. Kubernetes control plane runs at https://<rancher-hostname>/k8s/clusters/c-m-zjrfft56
3. CoreDNS runs at https://<rancher-hostname>/k8s/clusters/c-m-zjrfft56/api/v1/

   namespaces/kube-system/services/rke2-coredns-rke2-coredns:udp-53/proxy
4. Extract the entire API server endpoint (https:// included) from the output mentioned which should be in the following pattern: https://<rancher-hostname>/k8s/clusters/c-m-zjrfft56
5. Add the entire rancher cluster name into NetBackup web UI > **Workloads** > **Kubernetes** > **Kubernetes clusters** >**Add** .
6. On the **Add Kubernetes cluster** page, select a option associated with URL or Endpoints to allow cluster addition based on the endpoints which contain (https://).

   Note: You cannot edit the cluster names added using the endpoint-based approach. You can only delete and re-add such cluster names.
7. Enter the cluster info output which is extracted above into the input field on the NetBackup web UI (Endpoint or URL).
8. Proceed ahead and select or create the credentials which were prepared in steps 1 to 4.
9. Once the credentials are validated and a cluster is added successfully. It will trigger an automated validation and discovery.
10. After a successful automated discovery, user attempts a manual credential validation and discovery to ensure that everything is working fine.

11. Add a Rancher managed cluster in NetBackup.
12. Create the backup server certificate secret and the data mover configmap to setup Backup from Snapshot (BFS) function.

    Then, proceed with the rest of the configuration steps as per the recommended setup guide.

Chapter 10

Recovering Kubernetes assets

This chapter includes the following topics:

Explore and validate recovery points

NetBackup version 10.0 onwards supports recovery of Kubernetes assets using restore from snapshot and restore from backup copy operations.

Note: After recovery, the newly created namespaces, persistent volumes, and other resources get new system-generated UIDs.

NetBackup helps you to perform backup image validation through complete or incomplete state of the backup copy in the Kubernetes workload. NetBackup does not let you run a restore operation from an incomplete backup copy.

The recovery point corresponding to Kubernetes namespace consists of multiple images. The recovery point may be incomplete as the copy for some of the images might not be available. Such recovery points are marked as incomplete.

To perform validation of recovery point

1. On the left, click **Kubernetes**, under **Workloads**
2. In the **Namespaces** tab, click the namespace of the asset that you want to recover.

3 Click the **Recovery points** tab.

4 The **Recovery points** tab shows you all the recovery points with the date, time, and copies of the backup.

Click the number of copies button next to the recovery point, to view the location, default copy, copy type, and complete state.

Complete state helps you to validate the selected recovery point to run the restore operation.

There can be multiple reasons for incomplete backup copy, backup in progress, image expiration, hardware failure, or network communication issues.

Restore from snapshot

NetBackup features a restore from snapshot function where you can restore all the backup images in a recovery point, using a single restore job. You can view the restore from snapshot job in the Activity monitor.

To restore from a snapshot

1 On the left, click **Workloads > Kubernetes**.

2 On the **Namespace** tab, click the namespace of the asset that you want to recover.

3 Click the **Recovery points** tab.

The **Recovery points** tab shows you all the recovery points with the date, time, and copies of the backup. You can set filters to filter the displayed recovery points. Click the date in the **Date** column, to view the details of the recovery point. The **Recovery points details** dialog shows the resources that were backed up, like config maps, secrets, persistent volumes, pod, and so on. For details about these resources, see https://kubernetes.io/docs/reference/kubernetes-api/

4 Locate the recovery point that you want to restore.

5 In the **Copies** column, click the **# copies** button. For example, if there are two copies, the button displays as **2 copies**.

6 In the list of copies, locate the **Snapshot** copy. Then click **Actions > Restore**.

Note: Enable the **Restore** option for all infected copies by selecting **Allow the selection of recovery points that are malware-affected** option.

---

7 In the **Recovery target** page, target cluster is auto populated.

Note: Alternate cluster restore is not supported for snapshot copy.

8 Under **Specify destination namespace**, select any of the following options for restore:

- **Use original namespace** to use the original namespace that was backed up for restore. By default, this option is selected.
- **Use alternate namespace** to use an alternate namespace for restore and then, click **Next**.

9 Under **Select resource types to recover**, select any of the following resource types to restore:

- **All resource types** to recover all resource types. By default, this option is selected.
- **Recover selected resource types** to recover only the selected resource types.

Note: Select resource types to recover option is for advance users. If you are not careful in selecting the resources that you want to restore, you may not get a fully functional namespace after restoring.

10 Under **Select Persistent volume claims to recover**, select any of the following persistent volume claims to recover:

- **All Persistent volume claims** to recover all persistent volume claims. By default, this option is selected.
- **Recover selected Persistent volume claims** to recover selected persistent volume claims.

Note: If you do not select any option in **Recover selected resource types**, then include empty persistent volume claims option is selected and no persistent volume claims is restored.

If you do not select any options in the **Recover selected persistent volume claims** then in the **Recovery options** section, it includes empty persistent volume claims and no persistent volume claims is restored.

Note: Restore only persistent volume enables toggle in the selected persistent volume claims to restore only the persistent volume. This setting does not create a corresponding persistent volume claim.

11. Click on the **Failure strategy** section to view the failure strategy options to recover.
12. Under **Select failure strategy to recover**, select any of the following failure strategies to recover:

Note: If a failure occurs during the restore of metadata or PVCs, the restore job runs according to the failure strategy selected.

- **Fail Fast** to terminate the restore for any failure occurrence.
- **Proceed Ahead** to continue restoring the next PVC. If the parent image (first image) restore fails then the restore job terminates.
- **Retry** to specify a retry count for the metadata or PVC restore. If the restore fails even after retries, then the restore job terminates.

Note: The selected failure strategy is displayed in the **Activity monitor**.

13. Click **Next**.
14. On the **Recovery options** page, click **Start recovery** to submit the recovery entry.
15. In **Activity monitor**, click the **Job ID** to view the restore job details.

Note: NetBackup Kubernetes restore uses single job to restore all the persistent volume claims and a namespace. You can view logs on the **Activity monitor** to track the restore of persistent volumes, persistent volume claims, or metadata.

Restore from backup copy

A NetBackup restore from backup happens in parallel if there are multiple PVCs in the selected namespace. When you start a restore the job creates a parent-child hierarchy (if the namespace has at least one PVC to restore). The parent job acts as an orchestrator and monitors the status of child jobs. The first child job restores the metadata, after which PVCs are restored in parallel.

Note: If metadata restore fails, no further jobs are submitted for restore operation. Once metadata is restored successfully, PVCs are restored parallel in batches.

You can follow the same procedure that is explained in restoring from snapshot, select the copy type as **Backup**. You can also restore to alternate target cluster.

To restore from a backup copy

1. On the left, click **Workloads > Kubernetes**.
2. On the **Namespace** tab, click the namespace of the asset that you want to recover. Click the **Recovery points**tab.
3. The **Recovery points** tab shows you all the recovery points with the date, time, and copies of the backup. You can set filters to filter the displayed recovery points. Click the date in the **Date** column to view the details of the recovery point. The **Recovery points details** dialog shows the resources that were backed up, like `ConfigMaps`, secrets, persistent volumes, pod, and so on. For details about these resources, see https://kubernetes.io/docs/reference.
4. Locate the recovery point that you want to restore.
5. In the **Copies** column, click the **# copies** button. For example, if there are two copies, the button displays as **2 copies**.
6. In the list of copies, locate the **Backup** copy. Then click **Actions > Restore**.

   Note: Enable the **Restore** option for all infected copies by selecting the **Allow the selection of recovery points that are malware-affected** option.

7. In the **Recovery target** page, to recover the asset to the same cluster source are auto populated. Click **Next**.
8. Under **Specify destination namespace**, select from the options:
   - Select **Use original namespace** to use the original namespace.
   - Select **Use alternate namespace** and enter the alternate namespace. Click **Next**.
9. Under **Select resource types to recover**, select from the following resource types:
   - Select **All resource types** to recover all resource types.
   - Select **Recover selected resource types** to recover only the selected resource types.

10 Under **Select Persistent volume claims to recover**, select from the following options:

- Select **All Persistent volume claims** to recover all persistent volume claims.
- Select **Recover selected Persistent volume claims** to recover selected persistent volume claims.

Note: If you do not select any option in **Recover selected resource types**, then include empty persistent volume claims option is selected and no persistent volume claims are restored.

If you do not select any options in the **Recover selected persistent volume claims**, then in the **Recovery options** section it includes empty persistent volume claims and no persistent volume claims are restored.

Note: Restore only persistent volume enables the toggle in the selected persistent volume claims to restore only the persistent volume. This setting does not create a corresponding persistent volume claim.

---

11 Click on the **Failure strategy** section to view the failure strategy options to recover.

12 Under **Select failure strategy to recover**, select any of the following failure strategies to recover:

Note: On occurrence of any failure while restoring metadata or PVCs, a restore job runs as per the failure strategy selected.

- **Fail Fast** to terminate the restore for any failure occurrence.
  - This restore failure strategy helps you to terminate the restore job when the first failure occurs.
  - All the remaining active restore jobs in the current batch are allowed to complete and no further batches are submitted for restore.
- **Proceed Ahead** to continue restoring the next PVC. If the parent image (first image) restore fails, then the restore job terminates.
  - This strategy helps you to proceed ahead with the restore of remaining PVCs, if any of the PVC restores fail for the ongoing batch.
  - If metadata restore fails, the final job is marked as failed and no PVCs are submitted for restore.

- In this case, the final job status which is marked as partial success and a list of PVCs with failed status appears in the **Activity Monitor** tab for the parent job.

- **Retry** to specify a retry count for metadata or PVC restore. If the restore fails even after retries, then the restore job terminates.
  - This failure strategy helps you to retry the restore job of failed PVC/metadata which is configurable at the start of the restore job.
  - If the restore job fails despite the maximum number of retries, the job which is marked as failed and no further batches are submitted for restore.

  Note: The selected failure strategy is displayed in the **Activity monitor**.

- Click **Next**.

13 Click **Start recovery** to submit the recovery entry.

14 In the **Activity monitor**, click the **Job ID** to view the restore job details.

15 On the **Job Details** page, click the **Details** tab. The restore job sequence (pre-restore, data movement, and then the post-restore job) is displayed.

Note: You can cancel the parent job to cancel the restore operation. The parent job terminates all the active child restore jobs.

Configuration change

The batch size for parallel PVC restore is configurable in bp.conf. User can add the key `KUBERNETES_RESTORE_FROM_BACKUP_COPY_PARALLEL_RESTORE_BATCH_SIZE` in `bp.conf` file to set the desired batch size. This key is optional and has the value of 5 if it is not defined.

The minimum value that can be assigned for batch size is 1, whereas the maximum value is 100.

You can use the `bpsetconfig` command on the NetBackup primary server to update the batch size.

Chapter 11

About incremental backup and restore

This chapter includes the following topics:

Incremental backup and restore support for Kubernetes

NetBackup Kubernetes versions 10.4 and later provide backup support for differential, cumulative, and automatic schedules.

Incremental backups reduce the backup window significantly in NetBackup. In this method, NetBackup backs up only the data that has been changed since the subsequent full backup.

Incremental backup support

Incremental backup supports only the file system type persistent volumes. The block type persistent volumes backups are always full backup irrespective of the schedule types.

Note: Snapshot copy is always a full backup due to storage class limitation. Apart from snapshot copy, backup from snapshot, duplicate copies have incremental support.

Restore jobs

Restore from a complete recovery point performs point in time restore. All the data till that recovery point is restored.

If the **Complete** field displays **No**, you cannot restore from that recovery point.

Image chain validation

Image chain validation operation is performed for recovery point copies, the validation is reflected in the **Complete** field of recovery point of each backup copy.

The **Complete** field is set to **Yes** when all the related images of the recovery points are present.

Note: The **Complete** field is marked as **No** (Complete = No) if the incremental backups chain is incomplete or if any image is missing from the image group.

Auto Image Replication (A.I.R.) limitation

A.I.R. is supported only for the full schedule backup jobs. A.I.R. function is not supported for differential incremental, cumulative incremental, or automatic schedules.

Restore from manual import

Manually imported incremental images can be restored from a valid recovery point (Complete = Yes).

Troubleshooting for manual import

After manual import, if the recovery point is marked as `Incomplete`, then the image chain might be broken because of the images that were missed for the manual import operation.

To recreate an image chain for manual import operation

1. Open the file `/usr/openv/netbackup/logs/bpdbm/root{dateformat}.log` and find the line `previous backup relationship. To restore the relationship`, know which images are missed from the manual import operation.
2. Import the missed out images and run the following command to create a new image chain.

   ```
   `bpimage -update -id <backupid> -previous_backupid <previous backup id>`
   ```

ctime and mtime flags

USE_CTIME_FOR_INCREMENTALS option for NetBackup clients:

- The USE_CTIME_FOR_INCREMENTALS entry changes how NetBackup determines whether or not a file has changed. This entry causes the client

software to use both modification time and inode change time during incremental backups to determine if a file has changed (mtime and ctime).

DO_NOT_RESET_FILE_ACCESS_TIME option for NetBackup clients:

- The DO_NOT_RESET_FILE_ACCESS_TIME entry specifies that if a file is backed up, its access time (Atime) displays the time of the backup. By default, NetBackup preserves the access time. NetBackup resets the previous value of the backup.
- To set the data mover properties: The user must set or update the flag in the NetBackup primary server-specific ConfigMap that is created under the NetBackupKOps namespace on the Kubernetes cluster.
- Example:

```
apiVersion: v1
data:
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
    VERBOSE=5
    VXMS_VERBOSE=5
    USE_CTIME_FOR_INCREMENTALS=YES
    DO_NOT_RESET_FILE_ACCESS_TIME=YES
     version: "1"
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: <NetBackupKOps-Namespace>
```

Protection plan

Following the schedules are supported in NetBackup Kubernetes workload.

- Automatic
- Full
- Differential Incremental
- Cumulative Incremental

A protection plan with different schedules can be configured as follows.

To backup with different schedule types

1. In a protection plan, select backup type (Full, Differential Incremental, Cumulative Incremental, Automatic).
2. Under configure schedule for snapshot, provide values for recurrence and snapshot retention.

   Provide values for Backup from snapshot retention period.
3. In the start window tab, set a schedule time and then click **Add**.
4. Under the **Schedules and retention** section, click **Add schedule** to add multiple schedules in the same protection plan (Differential Incremental, Cumulative Incremental).
5. Select a storage and follow the remaining steps to run a backup job.

Automatic schedule

- In the case of automatic type schedule, based on recurrence of snapshot schedule get resolved after creation of protection plan.
- If recurrence is less than one week, then a single differential and full schedule is created.

Recommendations

- Follow the recommendation for Retention values in a protection plan for incremental schedules.
  - To perform a restore from recovery point for any copies (snapshot, backup from snapshot, duplicate), it is recommended to keep the retention duration of the copy for a longer period.
  - For example, to restore from a backup copy the retention of Backup from Snapshot must be more than a Snapshot copy. Otherwise, the backup copy expires and the recovery point is marked as `COMPLETE = NO`.
  - In such cases, the warnings appear in the NetBackup web UI as follows:
    - `It is recommended to set backup retention period more than snapshot retention period.`
    - `It is recommended to set duplication retention period more than the backup retention period.`
- Always add a full backup schedule along with cumulative backup schedule. Otherwise, every Cumulative backup is performed as a Full backup.
- By default, the Backup from Snapshot option is always enabled for incremental backup types.

Chapter 12

Enabling accelerator based backup

This chapter includes the following topics:

About NetBackup Accelerator support for Kubernetes workloads

NetBackup Accelerator reduces the backup time for Kubernetes cluster backups.

For Kubernetes backups, the Accelerator feature is activated when you select a storage type that supports Accelerator. For example, MSDP, OpenStorage, CloudStorage, and MSDP-C (Azure and AWS), and Kubernetes clusters that support Accelerator-enabled backups.

Note: Accelerator enabled backup is supported only for the file mode PVCs.

Enable accelerator support for a specific Kubernetes cluster

The NetBackup Kubernetes Operator `values.yaml` has an entry

`acceleratorTracklogPvcStorageClass:` `None`

To enable Accelerator, specify a valid storage class name to generate the track logs for Accelerator backups. Storage class helps to create a file mode volume which is usable on any of the worker nodes within the Kubernetes cluster.

Note: If the *acceleratorTracklogPvcStorageClass* is set to None and an Accelerator-enabled storage is selected, then the Accelerator backup jobs do not run. After upgrade to NetBackup 10.4 release, the default value of *acceleratorTracklogPvcStorageClass* is None.

For more details, refer to the *Validating accelerator storage class*section in the *NetBackup for Kubernetes Administrator's Guide*.

Resource throttling and storage requirements for Accelerator track log storage class

- The default value for the number of Backup from Snapshot jobs per Kubernetes cluster is 4.
- If 4 Backup from Snapshot jobs with Accelerator run to back up 4 PVCs simultaneously, these jobs consume some storage.
- As per this calculation, each PVC requires some space for track log creation. Track log size in Bytes = 2*((Number of files in PVC * 200) + ((Total used disk space in PVC KiB/128KiB) * 20))
- The storage that is required to run 4 Backup from Snapshot jobs simultaneously = Sum of track log size of 4 PVCs.
  Therefore, the storage requirement changes if the number of Backup from Snapshot jobs per Kubernetes cluster is changed.
- Ensure that a sufficient storage is available before running the backups jobs. To avoid storage issues, elastic storage can be used.

Backup streams

NetBackup Accelerator creates the backup stream as follows:

- If the namespace has no previous backup, NetBackup performs a full backup.
- For the next backup job, NetBackup identifies data that is changed since the previous backup. Only the changed blocks and the header information are included in the backup, to create a full backup.
- Once the backup is done, bpbkar on the data mover updates the track log. Track log path inside data mover - *usr/openv/netbackup/track/<primary server>/<storage server>/<k8s cluster name>_<namespace uuid>_<pvc uuid>/<policy>/<backup selection>*

- This track log is then transferred to primary server in the inline style at the following location:
  /usr/openv/netbackup/db/track/<primary server>/<storage server>/<k8s cluster name>_<namespace uuid>_<pvc uuid>/<policy>/<backup selection>
- When the next Accelerator backup job is initiated, the track log is fetched from the primary server to identify the changed files. Then it is updated with the new content and transferred back to the primary server.

Controlling disk space for track logs on primary server

To proceed with an Accelerator enabled backup, NetBackup anticipates a disk-full situation due to track log processing. Accelerator track logs on the primary server can become a problem with less free space.

By default, NetBackup prevents an Accelerator backup, if there is less than 5 GB or 5% of free space on the system.

NetBackup provides two configuration settings to control the amount of free disk space on a host for an Accelerator backup to start:

- ACCELERATOR_TRACKLOG_FREE_SPACE_MB
- ACCELERATOR_TRACKLOG_FREE_SPACE_PERCENT

The default value for each setting is 5120 MB and 5%, respectively. To have backup fail fast if there is insufficient space, configure these values in the primary server's `bp.conf` file.

Effect of storage class behavior on Accelerator

The following are the affects of the storage class behavior on Accelerator:

- For Kubernetes Accelerator-enabled backups, NetBackup shows optimization based on the amount of data changed.
  However, the duration to complete a subsequent accelerator job can be similar to complete a full backup job.
- This situation happens due to the storage class behavior where the `INODE` and `CTIME` of the files are changed irrespective of the data or if the metadata of the file is changed.
- It's a result, due to internal implementation of the storage classes. For more details, refer to the *Red Hat Knowledgebase articles* on the customer portal site: https://access.redhat.com/solutions/7036388

About Accelerator forced rescan

NetBackup Accelerator forced rescan feature support helps you to prevent corrupt backup image issues. When Accelerator forced rescan is used, all the data from the selected backup target is backed up.

To perform Accelerator forced rescan job, run the command `ForcedRescan` manually. When accelerator forced rescan is used, all the data from the selected backup target is backed up.

This backup is similar to the first Accelerator backup for a policy. The duration of the backup is similar to a non-Accelerator full backup. Force rescan enhances safety, and establishes a baseline for the next Accelerator backup. This feature protects against any potential damage like failure of checksum verification.

Recommendations for using forced rescan:

- To manually initiate the backup with force rescan, run the following command in the command prompt or Linux terminal:

  ```
  bpbackup -i -p <policy_name> -s ForcedRescan
  ```

- For example,

  ```
  bpbackup -i -p msdp_10mins_FRS+5d990ab5-f791-474f-885a-ae0c30f31c98
  -s ForcedRescan
  ```

  Following API can be used to initiate the force rescan schedule:

  ```
  POST admin/manual-backup
  ```

Warnings and probable reason for Accelerator backup failures

Warning	Message	Recommended action
`Kubernetes accelerator backup feature is not supported on this NetBackup media server version, will perform a non-accelerated backup.`	Media server version is less than 10.4.	Upgrade the media server to 10.4 or later.

Warning	Message	Recommended action
`The client does not support accelerator due to older version of client or Storage class for track log PVC is not configured on the client. However, the client performs a non-accelerated backup.`	NetBackup Kubernetes operators version is less than 10.4 or Storage class for track log PVC is not configured on the client.	Upgrade the NetBackup Kubernetes operators and data mover to 10.4 or later. Ensure that storage class for track log PVC is properly configured in the Kubernetes operators.

Chapter 13

Enabling FIPS mode in Kubernetes

This chapter includes the following topics:

Enable Federal Information Processing Standards (FIPS) mode in Kubernetes

NetBackup Kubernetes 10.3 release provides FIPS support for RedHat based NetBackup deployments. All the Kubernetes workload component involved in NetBackup, Kubernetes operator and Data mover must run in FIPS mode. To achieve the FIPS support, there are certain requirements that needs to be met across all these components.

System requirements

Following are the system requirements for FIPS support in NetBackup Kubernetes workload.

Name	Parameters

NetBackup Primary and Media	■ Both Primary and Media must be deployed on the NetBackup10.2.1 with underlying RHEL-8system which is enabled with FIPS.
	■ RHEL OS version must be greater than RHEL8.
	■ You can check version of RedHat machine using the following command:
	`cat /etc/Redhat-release`
	■ You can check if underlying system have FIPS enabled using the following command:
	`FIPS-MODE-SETUP--CHECK`
	■ For more details, you can check man page entry of the following command:
	`fips-mode-setup`

Kubernetes Cluster	■ Kubernetes cluster must be deployed with FIPS enabled mode. ■ The process to deploy Kubernetes cluster in FIPS mode is vendor dependent. ■ For example, deploying Openshift with FIPS Enabled

Configuration parameters

Following are the configuration parameters for FIPS mode in NetBackup Kubernetes workload.

Configuration	Parameters
NetBackup Primary and Media	Enabling NetBackup process to run in FIPS mode: ■ Update <Netbackup-Installation-Path>/netbackup/bp.conf with the following key: *NB_FIPS_MODE = ENABLE* ■ For more information on NetBackup in FIPS mode, refer to the *Configure the FIPS mode in your NetBackup domain* section in the *NetBackup™ Security and Encryption Guide*

Configuration	Parameters
NetBackup Kubernetes Operator	Do any of the following to enable FIPS mode: ■ Update the value of parameter **fipsMode** to **ENABLE** in values.yaml file from the Helm Chart. ■ Update the value of parameter **NB_FIPS_MODE** to **ENABLE** in backup-operator.

Note: Make sure that all the systems on which NetBackup Kubernetes workload runs are FIPS compliant.

Troubleshooting for FIPS

Impact on the Automated Image Replication (AIR) operation:

- For AIR on FIPS enabled environment, you need to do the additional configuration.
- Update the <KB-Article>on the support site.
- Run the following commands in the command-line interface (CLI):

```
/usr/openv/java/jre/bin/keytool/keytool -storetype BCFKS
-providerpath
/usr/openv/wmc/webserver/lib/ccj-3.0.1.jar -providerclass
com.safelogic.cryptocomply.jcajce.provider.CryptoComplyFipsProvider
-importcert -trustcacerts -file <target CA certificate file (pem
encoded)> -keystore
NB_INSTALL_DIR/var/global/wsl/credentials/cacerts.bcfks -storepass
<password from the /usr/openv/var/global/jkskey file>
-alias <alias name of the trusted certificate entry to be added>
```

Chapter 14

About Openshift Virtualization support

This chapter includes the following topics:

OpenShift Virtualization support

NetBackup version 10.5, provides backup and restore support for namespaces with one or more virtual machines running on Kubernetes clusters.

In OpenShift clusters, virtualization is supported with the help of openshift-cnv operator (which is a RedHat version of KubeVirt).

With Virtualization support in Kubernetes, you can deploy virtual machines in Kubernetes environments using KubeVirt. KubeVirt is an extension for Kubernetes that allows you to manage virtual machines alongside containers. KubeVirt enables you to run virtual machines as Kubernetes pods, which provides a consistent management interface for containers and virtual machines.

You can create Kubernetes protection plan to protect namespace which have virtual machines and its related resources.

In an OpenShift environment, it is essential to have all required components installed and configured correctly in order to ensure a successful backup and restore of namespaces including virtual machines.

- Install OpenShift Virtualization Operator (openshift-cnv).

- (Optional) Install Container Data Importer (CDI), If you want to create a virtual machine using disk image from network source.

Intelligent group

With Virtualization support in Kubernetes, you can create intelligent groups with filtering the namespaces based on specific resource kinds. Following are resource kinds available to filter:

- Virtual Machines
- Persistent Volume
- Persistent Volume Claims

Application consistent virtual machines backup

You must annotate the virt-launcher pod with the NetBackup pre and post hooks if it is responsible for spawning virtual machines (VMs), as VMs cannot be created without it.

Commands to freeze and unfreeze the virtual machines:

- `/usr/bin/virt-freezer --freeze --name <vm-name> --namespace <namespace>`
- `/usr/bin/virt-freezer --unfreeze --name <vm-name> --namespace <namespace>`

```
# kubectl annotate pod -l vm.kubevirt.io/name=<vm-name>
-n <vm-namespace>
netbackup-pre.hook.backup.velero.io/command='["/usr/bin/virt-freezer",
"--freeze", "--name", "<vm-name>", "--namespace", "<vm-namespace>"]'

netbackup-pre.hook.backup.velero.io/container=compute

 netbackup-post.hook.backup.velero.io/command='

["/usr/bin/virt-freezer", "--unfreeze", "--name",

"<vm-name>", "--namespace", "<vm-namespace>"]'

  netbackup-post.hook.backup.velero.io/container=compute
```

Note: In order to achieve app consistency, it is necessary to install qemu-guest-agent on virtual machines to implement kubevirt-specific pre-exec and post-exec rules.

For more details, about configuration of NetBackup pre and post hook, see https://www.veritas.com/

Limitations and considerations for virtual machine backup operations

Backup with acceleration and incremental backups cannot be performed on PVCs that are exclusively used as virtual machine disks.

Troubleshooting for virtualization

- On the same cluster, a VM restore with a static MAC (Media Access Control address) defined will fail.

  There is a possibility that an existing VM MAC will be allocated to the source VM when a user does a virtual machine restore operation on the same cluster with an alternate namespace. The user must make sure that the MAC is not given to any VMs.

- During a cross-cluster restore of a namespace, if the storage class provided for the filesystem is different from that of the source PVC, it may result in a failed restore operation.

  To prevent this, the user must ensure that the target cluster has the same storage class on the source and target cluster.

Chapter 15

Troubleshooting Kubernetes issues

This chapter includes the following topics:

Error during the primary server upgrade: NBCheck fails

NetBackup primary server upgrade from version 9.1 to 10.0 fails with a non-critical NBCheck error.

Error message : The test found {{no. of policies}} active Kubernetes policy. This test fails if the NetBackup instance has any active Kubernetes policies.

Recommended action: To deactivate all the active Kubernetes policies on the primary server before upgrading NetBackup to 10.0. version.

For more details, see https://www.veritas.com/content/support

Error during an old image restore: Operation fails

Kubernetes restore operation fails for the older images which were created using NetBackup 9.1. version.

Error message: Restore operation is not supported on the backup images of NetBackup older than 10.0 version.

Recommended action: Restore the older image using Velero commands. Velero is an open-source tool to safely backup and restore, perform disaster recovery, and migrate Kubernetes cluster resources and persistent volumes. Thus, to restore old image from Velero, installation is a pre-requisite on the cluster.

Get the backup name / backup id from the NetBackup Administrator Web UI and use it in Velero commands to restore it.

For more details, see https://www.veritas.com/content/support

Error during persistent volume recovery API

On NetBackup Kubernetes operator version10.0, the persistent volume recovery API is removed and not supported. On the older versions of NetBackup this API was used to restore the persistent volume. So, if you have upgraded NetBackup 10.0 version, and using the persistent volume recovery API to restore, then the restore operation will fail.

Error message: Kubernetes persistent volume recovery API is no longer in use and has been removed from the product due to redesign at NetBackup Kubernetes recovery process.

Recommended action: In NetBackup Kubernetes operator version 10.0, NetBackup is upgraded to recover selected resources from backups. So, if you want to recover persistent-volume or persistent-volume claims then you can select the persistent volumes from NetBackup and recover on to the destination namespace.

For more details, see https://www.veritas.com/content/support

Error during restore: Final job status shows partial failure

Final restore job status is partially failed with few warnings specific to the resource RoleBinding.

The warnings displayed are specific to the resource RoleBinding for API groups `groupauthorization.openshift.io` and `rbac.authorization.Kubernetes.io` are seen. Because the RoleBinding are auto managed using the controller and gets created when we create a new namespace.

Recommended action: You can exclude the relevant RoleBinding resources from the restore or ignore the warnings created.

Error during restore on the same namespace

Restoring PVCs on an original namespace might fail, if the selected PVCs are already present in the namespace.

Recommended actions:

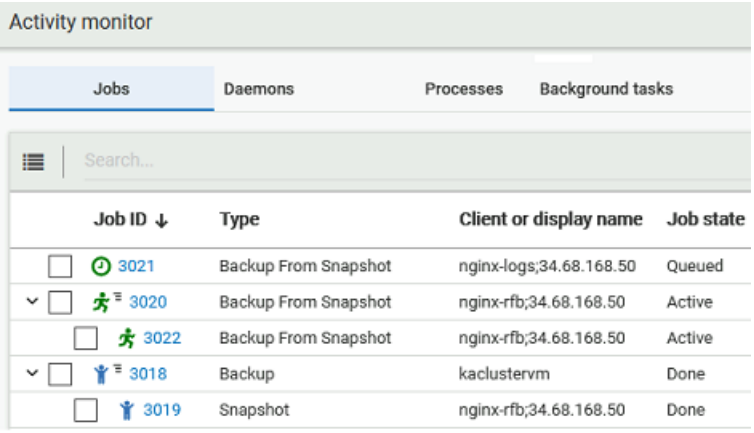
- You can use alternate namespace restore
- You can select the PVCs in the **Recovery option** which are not overlapping with the existing PVCs while running the restore operation.

Datamover pods exceed the Kubernetes resource limit

NetBackup controls the total number of in-progress backup jobs on Kubernetes workload using the two resource limit properties. In NetBackup version 10.0, datamover pods exceeds the **Backup** and **Backup From Snapshot** resource limits set for per Kubernetes cluster.

Following is the example with resource limit issue

Scenario no 1



Resource limit for Backup from Snapshot jobs per Kubernetes cluster is set to 1.

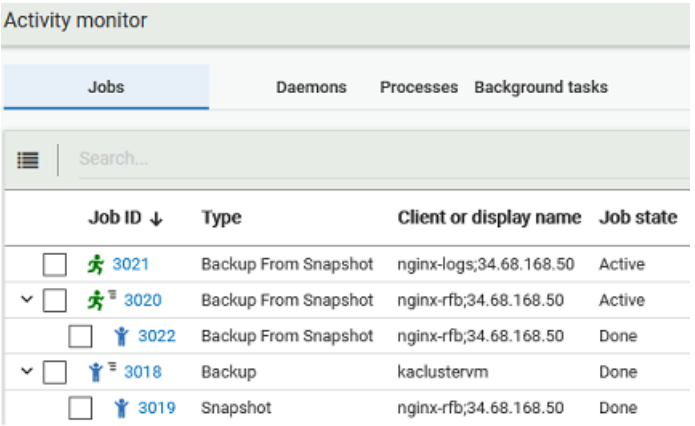
Job IDs 3020 and 3021 are the parent jobs for Backup from snapshot. The creation of the data mover pod and its cleanup process are part of the backup job life cycle.

Job ID 3022 is the child job, where the data movement takes place from the cluster to the storage unit.

Based on the resource limit setting, while job ID 3022 is in the running state, job ID 3021 will continue to be in queued state. Once, the backup job ID 3022 is completed, then the parent Job ID 3021 will start.

Notice that the job ID 3020 is still in progress, since we are in process to clean up the data mover pod and complete the life cycle of the parent job ID 3020.

Scenario no 2



At this stage, we may encounter that there are 2 data mover pods running simultaneously in the NetBackup Kubernetes operator deployment namespace. Because the data mover pod created as part of job ID 3020 is still not cleaned up, but we started with creation of data mover pod for job 3021.

In a busy environment, where multiple Backup from Snapshot jobs are triggered, a low resource limit value setting may lead to backup jobs spending most of the time in the queued state.

But if we have a higher resource limit setting, we may observe that the data mover pods might exceed the count specified in the resource limit. This may lead to resource starvation in the Kubernetes cluster.

While the data movement job like 3022 runs in parallel, cleanup activities are handled sequentially. This when combined with the time it takes to cleanup the datamover resource, if closer to the time it takes to backup the pvc/namespace data leads to longer delay in the completion of the jobs.

If the combined time duration for data movement and clean up resources is like the backup job. Then, the backup job of persistent volume or namespace data may lead to delay in the job completion.

Recommended action: Ensure to review your system resources and performance, to set the resource limit value accordingly. This measure will help you achieve the best performance for all backup jobs.

Error during restore: Job fails on the highly loaded cluster

Restore jobs fails on the heavily loaded Kubernetes cluster.

Error messages: ERR - Unable to initialize VxMS, cannot write data to socket, Connection reset by peer.

Error bpbrm (pid=712755) from client cluster.sample.domain.com: ERR - Unable to initialize VxMS.

Error bptm (pid=712795) cannot write data to socket, Connection reset by peer.

Recommended action: If you face this issue during restore operation, then you should run the restore operation on a less loaded cluster or when the cluster is idle.

Custom Kubernetes role created for specific clusters cannot view the jobs

When a custom RBAC role is created for Kubernetes workload with specific Kubernetes clusters then the system administrator must explicitly provide the permissions to view Kubernetes jobs, else all the Kubernetes specific jobs will not be visible.

If the system administrator does not provide the permission to view the Kubernetes jobs, then the user may view the following jobs:

- Only restore jobs in the hierarchy view.
- Only snapshot and restore jobs in the list view.

If a custom based Kubernetes role created is not able to view the jobs for specific Kubernetes clusters. Then perform the following steps to provide view permissions.

To provide view permissions

1. On the left click **Kubernetes** under **Workload**.
2. On the right click **Kubernetes setting>Manage permissions**.
3. Click the vertical ellipse next to the corresponding role and select **Edit**.
4. In the **Edit permissions**, select **Edit**, and **View jobs** permissions for the role and then, click **Save**.

   Kubernetes custom role user will be able to view backup, snapshot, restore and backup form snapshot jobs both in the hierarchical and list view.

Assumptions:

- If the setup is upgraded, then the user may view the following:
  - Only restore jobs in the hierarchy view from the existing jobs.
  - Only snapshot and restore jobs in the list view from the existing jobs.

- If a custom role for Kubernetes is created with permission to the selected Kubernetes clusters, then user can **cancel** and **restart** operations only on the snapshot jobs.

Openshift creates blank non-selected PVCs while restoring applications installed from OperatorHub

In an Openshift environment, where an application is installed via OperatorHub catalog sources. When a user tries to perform a selective PVC restore from such application namespace's backup. But instead, all the PVCs gets created.

This issue happens because, an Openshift environment provisions the non-selected PVCs with required size in the destination namespace.

Note: For such applications, the PVCs, are auto-provisioned as per the deployment configurations even if a user does not select them for restore.

NetBackup Kubernetes operator become unresponsive if PID limit exceeds on the Kubernetes node

In Linux systems, there is an initd or system process running as PID 1 to reap zombie processes. Containers that do not have such an initd process would keep spawning zombie processes.

After certain time period these zombie processes accumulates and then reaches the max limit of PIDs set on the Kubernetes node.

In NetBackup Kubernetes operator, nbcertcmdtool spawns child processes to carry out certificate-related operations. On completion of the operation, the processes get orphan and are not reaped. Eventually it hits the max PID limit and NetBackup Kubernetes operator becomes unresponsive.

Error message: login pod/nbukops-controller-manager-67f5498bbb-gn9zw -c netbackupkops -n nbukops ERRO[0005] exec failed: container_linux.go:380: starting container process caused: read init-p: connection reset by peer a command that is terminated with exit code 1.

Recommended actions:

- To fix the PID limit exceed issue, you can use the Initd script. Initd script acts as parent process or entry point script to the controller pod.

As a parent process it attaches zombie process to itself after the child process completion to terminate the persistent zombie process. It also helps you to shut down the container gracefully. Initd script is available in NBUKOPs build version 10.0.1.

- Use the following steps to remove the existing nbcertcmdtool zombie processes:

1. Describe the NetBackup operator pod and find the Kubernetes node on which the controller pod is running. Run the command:

   ```
   kubectl describe -c netbackupkops <NB k8s operator pod name> -n
   <namespace>
   ```

2. Log on to the Kubernetes node, run the command:

   ```
   kubectl debug node/nodename
   ```

3. Terminate the nbcertcmdtool zombie processes, run the command:

   ```
   ps -ef | grep "\[nbcertcmdtool\] <defunct>” | awk '{print $3}' |
   xargs kill -9
   ```

Note: These steps terminate all the zombie processes for that worker node. But it resolves the issue temporarily. For a permanent solution, you must deploy a new KOps build with Initd script.

Failure during edit cluster in NetBackup Kubernetes 10.1

Kubernetes edit cluster operation have an issue and does not work in the NetBackup Kubernetes 10.1 version.

Recommended action: To edit the cluster, you must first delete the Kubernetes cluster from the protection plan and then add the cluster again.

Backup or restore fails for large sized PVC

For large sized PVC, Backup from Snapshot and Restore from Snapshot/Restore from Backup fails when PVC does not get bound in configured polling timeout. This issue happens because hydration for large volume snapshots takes more time than default 15 minutes timeout.

Backup from Snapshot

Backup from snapshot fails for large sized PVC (example: 1.5 TB) with error code 34

Error messages:

Error nbcs (pid=250908) failed to setup the data mover instance for tracklog pvc operation.

Error nbcs (pid=250908) unable to initialize the tracklog data mover instance, data mover pod status: Pending reason:Failed message:Error: context deadline exceeded.

Restore from Snapshot or Restore from Backup

Restore from snapshot fails for large sized PVC (example: 100GB) with error code 5

Error message:

Error nbcs (pid=29228) timeout occurred while waiting for the persistent volume claim pvc-sample status to be in the bound phase

Recommended action:

Increase the polling timeout in the backup operator **configmap**.

- **Configmap name**: <kops-name>-backup-operator-configuration
- **Key to update**: pollingTimeoutInMinutes

Restore of namespace file mode PVCs to different file system partially fails

Restoration of namespace file mode PVCs to different file system results in partial success of namespace volumes. In this case, a restore of a file system objects (file/directory) to a file system other than the source file system results into failure of restoration of incompatible metadata. As a result, this operation shows up as partially successful restore.

Error message: 7:38:57 AM - Error bpbrm (pid=30171) client restore EXIT STATUS 1: the requested operation was partially successful.

Recommended action: Review the destination file system which reveals that the files are in place. As the files are restored, there is no actual problem with data. This partial failure is reported as an advisory that there are issues with the restore of metadata and the operator must be aware of it.

Restore from backup copy fails with image inconsistency error

Restore from backup copy fails with an image inconsistency error when older version of media server is used for storage. For example: Media server older version than 10.1.1 for storage and Netbackup version 10.1.1 is used for restore from backup copy.

Error message: Sep 22, 2022 3:12:55 PM - Info tar (pid=1459) done. status: 229: events out of sequence - image inconsistency Sep 22, 2022 3:12:55 PM - Error bpbrm (pid=16523) client restore EXIT STATUS 229: events out of sequence - image inconsistency

Recommended action: You must always use Primary, Media and Netbackup Kubernetes operators version 10.1.1 for Kubernetes file system based backups with all Kubernetes workflows.

Connectivity checks between NetBackup primary, media, and Kubernetes servers.

To check the connectivity between NetBackup primary and other hosts, you can refer to the following commands.

- Once the required ports are open to facilitate the communication with the NetBackup primary server, you can run the following commands.
- To check connectivity between NetBackup primary/media server and Kubernetes cluster, run the following command from Kubernetes operator's pod.

  ```
  curl -v telnet://<netbackup-server-host>:<port-no>
  ```
- To check connectivity between NetBackup primary server and Kubernetes cluster, run the following command from NetBackup primary host.

  ```
  curl -v telnet://<kubernetes-api-server-host>:<port-no>
  ```

Note: The responses to both these commands must indicate that a connection is established successfully.

Error during accelerator backup when there is no space available for track log

Backup from Snapshot job fails with error socket write failed (24) if there is no sufficient space available to store track log.

Error message: Duplicate existed with status 24 (socket write failed).

Recommended action: You must have a sufficient storage on the path where track log is stored to run a backup job successfully.

Error during accelerator backup due to track log PVC creation failure

Backup from Snapshot job fails, if there is an error in track log PVC creation. Track log PVC creation can fail due to multiple reasons as follows:

- Invalid storage class is provided.
- Sufficient space not available for PVC creation.

Recommended actions:

- Check if sufficient space is available to create a PVC with the required size.
- Check if the storage class is configured correctly on the Kubernetes cluster.

Error during accelerator backup due to invalid accelerator storage class

Track log PVC creation fails, if an invalid storage class is provided for accelerator backup jobs. An error occurs while waiting for the PVC status to be in bound phase.

Error message: StorageClass.storage.k8s.io cstor-storage-class-x2 not found

Jan 11, 2024 2:12:54 AM - Error nbcs (pid=92639) StorageClass.storage.k8s.io cstor-storage-class-x2 not found

Recommended action: Provide a valid storage class in the backup operator configmap and rerun the backup job.

Error occurred during track log pod start

Error message: Unable to initialize the track log data mover instance.

Recommended action: Check the pod creation logs on Kubernetes cluster using describe command for a detailed error.

Failed to setup the data mover instance for track log PVC operation

Error message: Failed to fetch track log data mover pod status and events.

Recommended action: Check the pod creation logs on Kubernetes cluster using describe command for detailed error.

Error to read track log storage class from configmap

Error message: Failed to get storage class for track log.

Recommended action: Check if NetBackup Kubernetes operator is correctly configured.