# NetBackup for Cassandra Administrator's Guide

UNIX, Windows, and Linux

Release 10.0

VERITAS™

# NetBackup™ for Cassandra Administrator's Guide

Last updated: 2022-02-27

## 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

# Introduction

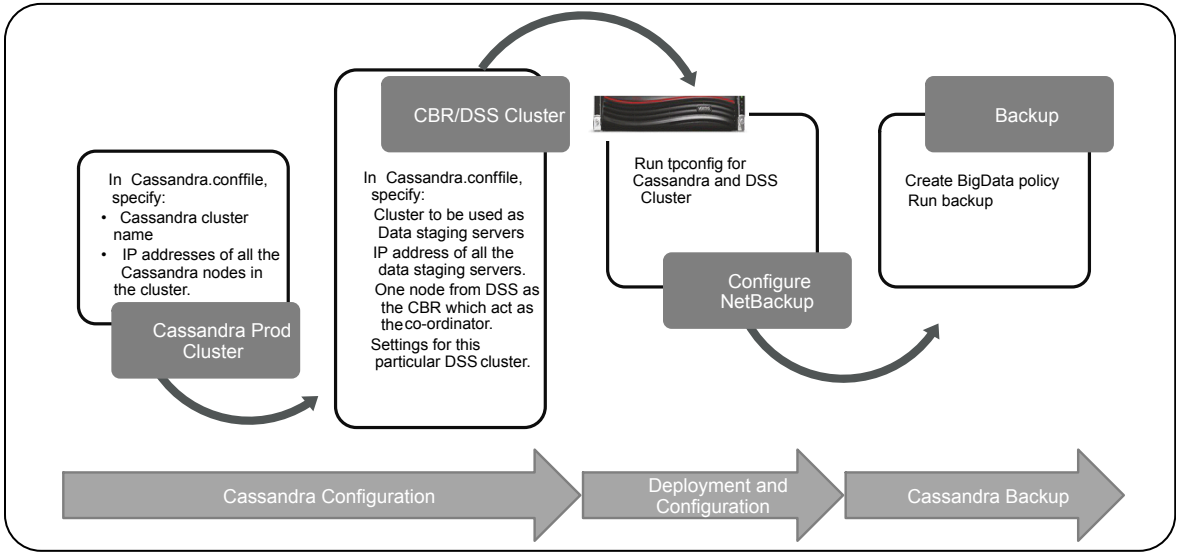
This chapter includes the following topics:

- Protecting Cassandra data using NetBackup
- Protecting Cassandra
- NetBackup for Cassandra terminologies

## Protecting Cassandra data using NetBackup

The NetBackup enables you to protect your Cassandra clusters that are deployed on-premises.

**Figure 1-1** Architectural overview



The following table describes the purpose of different components of the Cassandra backup and recovery solution.

**Table 1-1**

| Components | Purpose |
|---|---|
| Cassandra cluster | Represents the Cassandra production cluster that you want to protect. |
| Data staging servers | During a backup or restore, Cassandra keyspace are streamed in-parallel between the Cassandra cluster and the data staging servers.<br>The data staging servers, represent a staging cluster. You need to provision the nodes wherein, they are used depending on the size of data that needs to be backed up or restored. |
| Backup host | The Cassandra Backup Recovery (CBR) solution, uses the BigData policy with application type **cassandra**.<br>The BigData policy uses this backup host.<br>The media server that is used to configure storage server for the CBR solution must be used as backup host.<br>**Note:** You can also use NetBackup client as a backup host. |
| NetBackup primary server | All the jobs are executed from the NetBackup primary server. |

**Table 1-1** *(continued)*

| Components | Purpose |
|---|---|
| Data reduction | As part of data reduction the following tasks are performed:<br>■ Efficient reconciliation<br>Data for same keys from different nodes are transferred to the same node in the backup nodes.<br>Reconciliations happen in-parallel within each data staging servers without any inter-node communication.<br>■ Record synthesis<br>While iterating over the records, columns of the same key from different SStables are merged.<br>■ Semantic Deduplication<br>Stale and duplicate records (replicas) are identified and removed. |

- The data is backed up in parallel streams wherein the data nodes stream data blocks simultaneously to multiple data staging servers and from there to multiple backup hosts. The job processing is accelerated due to multiple backup hosts and parallel streams. The data staging servers help in optimizing the data being backed up thus achieving data deduplication.
- The communication between the Cassandra cluster and NetBackup is enabled using the Cassandra backup and recovery component that gets deployed on the data staging servers and the Cassandra cluster.
- For NetBackup communication, you need to configure a BigData policy and add the related backup hosts.
- You can configure a NetBackup media server, client, or primary server as a backup host. Also, based on Cassandra data size, you can add or remove backup hosts and data staging servers. You can scale up your environment easily by adding more backup hosts.
- The communication between the Cassandra cluster, data staging servers, and backup hosts happens over SSH.
- The NetBackup Parallel Streaming Framework enables a thin client-based, agentless backup wherein the backup-restore operations are performed on the backup hosts. The NetBackup thin client binary (Cassandra backup and recovery component) is automatically pushed to the Cassandra cluster during the

backup-recovery operations. This Cassandra backup and recovery component is automatically removed after the backup-recovery operations complete.

**Note:** Agent management is not required on the Cassandra cluster nodes.

# Protecting Cassandra

On a very high level, you need:

- NetBackup primary server
- NetBackup media server
- A backup host that is NetBackup primary, NetBackup media server or a NetBackup client.

Refer to the NetBackup compatibility list for the supported primary and media server configurations. The backup host that is NetBackup media server or a NetBackup client for Cassandra is supported only on an RHEL. NetBackup Appliance, NetBackup Flex Appliance and NetBackup FlexScale is supported as a NetBackup primary, media server, or as a client that can act as a backup host.

You need to follow the high-level steps for protecting Cassandra cluster:

1. Verify pre-requisites for Cassandra protection.
2. Run `tpconfig` on the NetBackup primary server.
3. Create `cassandra.conf` file with configuration details on the primary server.
4. Add required paths and hosts in the Allowed list.

# NetBackup for Cassandra terminologies

The following table defines the terms you come across using NetBackup to protect Cassandra cluster.

**Table 1-2** NetBackup terminologies

| Terminology | Definition |
|---|---|
| Cassandra Backup Recovery component | The NetBackup thin client which gets deployed on data staging servers and Cassandra cluster to aid in backup and restore operations. |

**Table 1-2** NetBackup terminologies *(continued)*

| Terminology | Definition |
|---|---|
| Data staging servers | NetBackup requires a set of servers for backup of Cassandra cluster in addition to the NetBackup primary, and backup hosts. These servers are typically 5% of the total number of servers in the Cassandra cluster. These servers are used to deduplicate the data from Cassandra cluster during backup and optimize the backup process. They are also used as staging-server for the data to be backed up and restored. |
| Parallel streams | The NetBackup parallel streaming framework allows data blocks from multiple nodes to be backed up using multiple backup hosts simultaneously. |
| Backup host | The backup host acts as a proxy client. All the backup and the restore operations are executed through the backup host.<br>You can configure media servers, clients, or a primary server as a backup host.<br>The backup host is also used as destination client during restores. |
| BigData policy | The BigData policy is introduced to:<br>■ Specify the application type.<br>■ Allow backing up distributed multi-node environments.<br>■ Associate backup hosts.<br>■ Perform workload distribution. |
| Application Cluster | ■ Application cluster is the Cassandra production cluster name.<br>■ Cluster name must be a single word with no white spaces in between words and must be the actual cluster name used in the `Cassandra.yaml` file on the production nodes. |

Chapter 2

# Configuring Cassandra Backup and Recovery solution

This chapter includes the following topics:



## Operating system and platform compatibility

NetBackup supports only RHEL platform for Backup host and Cassandra production. NetBackup also requires a set of staging servers which also need to be RHEL. For details, see *Software Compatibility List*

# Prerequisites and the best practices for protecting Cassandra

### Prerequisites

# Configuring NetBackup for Cassandra

#### Tpconfig

Run the `tpconfig` command on the NetBackup primary server.

---

**Note:** The path to access the `tpconfig` command is `/usr/openv/volmgr/bin/` for UNIX and `<install_path>\Volmgr\bin\` for Windows.

---

# Adding Cassandra credentials in NetBackup

To establish a seamless communication between Cassandra clusters and NetBackup for successful backup and restore operations, you must add and update Cassandra credentials in NetBackup.

Use the `tpconfig` command to add credentials in NetBackup primary server.

For Cassandra you need to provide the SHA 256 RSA fingerprint when you add the credentials. To obtain the RSA key run the `cat /etc/ssh/ssh_host_rsa_key.pub |awk '{print $2}' |base64 -d |sha256sum |awk '{print $1}'` command.

#### To add credentials in NetBackup

1. Run `tpconfig` command from the following directory paths:

   On UNIX systems, `/usr/openv/volmgr/bin/`

   On Windows systems, `install_path\Volmgr\bin\`

2. Run the `tpconfig --help` command. A list of options which are required to add, update, and delete Cassandra credentials is displayed.

3 All the servers/nodes in the cassandra cluster must support one non-root host user id which can be used by NetBackup to connect to all the node using `ssh`. This host user id and its password must be specified in the command of `tpconfig` while configuring the cassandra cluster.

```
./tpconfig -add -application_type cassandra -application_server
cassandra cluster name -application_server_user_id app user id
-password app password -host_user_id host user -host_password
host password -host_RSA_key host rsa key
```

**Note:** `Host_user_Id` is the OS user (non -root) on te Cassandra nodes. And -application_sever_user_id is the Cassandra shell user.

4 Similarly, one non-root host user id must be supported on all the nodes of the DSS cluster. This host user id and its password must be specified in the command of `tpconfig` while configuring the DSS cluster.

```
./tpconfig -add -application_type cassandra -application_server
DSS cassandra cluster name -application_server_user_id DSS app
user id -password DSS app password -host_user_id DSS host user
-host_password DSS host password -host_RSA_key DSS host rsa key
-requiredport 80
```

command.

**Note:** `Host_user_Id` is the OS user (non -root) on te DSS nodes. And -application_sever_user_id is the DSS Cassandra shell user.

5 Run the `tpconfig -dappservers` command to verify if the NetBackup primary server has the Cassandra credentials added.

# Configuring BigData policy for Cassandra plug-in

Use the following procedure to create a BigData policy with the NetBackup Administration Console.

**To create a BigData policy with the NetBackup Administration Console**

1 In the **NetBackup Administration Console**, in the left pane, expand **NetBackup Management > Policies**.

2 On the **Actions** menu, click **New > Policy**.

3 Type a unique name for the new policy in the **Add a New Policy** dialog box.

Click **OK**.

4. On the **Attributes** tab, select **BigData** as the policy type.
5. On the **Attributes** tab, select the storage unit for BigData policy type.
6. On the **Schedules** tab, click **New** to create a **Full Backup** schedule.
7. On the **Clients** tab, enter the name of the Cassandra cluster.

   **Note:** The Production cluster name added as a Client is case-sensitive. It must not have any white space, special characters, or non-English characters.

8. On the **Backup Selections** tab, enter the following parameters and their values as shown:
   - *Application_Type=cassandra*
     The parameter values are case-sensitive.
   - *Backup_Host=IP_address or hostname*
     You can specify multiple backup hosts.
   - Add the key word /ALL_KEYSPACES
9. Click **OK** to save the changes.

# Setting up `cassandra.conf` file on the primary server

To protect a Cassandra cluster you need to create a configuration file on the primary server which has the configuration details of the Cassandra cluster. Create the in the following path:

- UNIX: `/usr/openv/var/global/`
- WINDOWS: *`install path`*`\NetBackup\var\global\`

**Note:** The file name `cassandra.conf` must have all characters in lower case. This file is a JSON file and can be edited anytime manually and saved at the same location. Verify the JSON format with an online formatter, to avoid any JSON formatting errors when reading this file in NetBackup.

This file can have entries for multiple Cassandra clusters. All the Cassandra clusters must be listed in this file whether they are being backed up or being used for doing an alternate restore.

Sample `Cassandra.conf`file:

```
{
  "productionCluster": {
    "multi_72": {
      "nodes": [
        "10.221.104.71",
        "10.221.104.72",
        "10.221.104.73",
        "10.221.104.74",
        "10.221.104.77"
      ],
      "prodClusternodekeyHashes": {
        "10.221.104.71": "7b69ed1bbe095b2c5fcd34c26806793f8740ebcb24e0c7
bbd9a9bbae9e848923",
        "10.221.104.72": "a41dfc6a7b33f5fa02d7226e871a900666cd65beeca148
a77d0aabe9ed33e7ff",
        "10.221.104.73": "1a41c78e68effd51e6eaf8cde265421cb81475bf836593
8be146a271f444ce35",
        "10.221.104.74": "ebec0750d15ea1f0dfca993e8425d0106ef5aa0bf6e30d
5bfa6a3aad84313bbd",
        "10.221.104.77": "ba8f8b33a46bc88780288d87b5cb32116773a3929c2f4c

f33bd324e9516c5fdb"
      },
      "dataCenterName": "datacenter1",
      "nodeDownThresholdPercentage": 25,
      "dssClusterName": "dss_multi_72"
    },
    "multi_82": {
        "nodes": [
          "10.221.104.171",
          "10.221.104.172"
        ],
        "prodClusternodekeyHashes": {
          "10.221.104.171": "8a69ed1bbe095b2c5fcd34c26806793f8740ebcb24e0c
7bbd9a9bbae9e848964",
          "10.221.104.172": "b21dfc6a7b33f5fa02d7226e871a900666cd65beeca14
8a77d0aabe9ed33e7ab"
        },
        "dataCenterName": "datacenterwest",
        "nodeDownThresholdPercentage": 20,
        "dssClusterName": "dss_multi_82"
    }
  },
```

```
  "dssCluster": {
    "dss_multi_72": {
      "dssClusterInfo": {
        "cbrNode": "10.221.104.75",
        "nodes": [
          "10.221.104.75",
          "10.221.104.76"
        ],
        "dssClusternodekeyHashes": {
          "10.221.104.75": "14d0288c869d7021a2c855124c4ee5367e3cb6ede8ffc4d
74a883ff655ba0c57",
          "10.221.104.76": "ebd134c712ba8c2f8a75ba3c2ce1baf80bbbe199ed50476
e2c36f8e84adce294"
        }
      },
      "settings": {
        "jobCleanupTimeoutSec": 3600,
        "dssMinRam": "90909",
        "dssMinStoragePerBkupNode": "10485",
        "concurrentCompactions": "8",
        "sstableloaderMemsize": "4096M",
        "concurrentTransfers": "2",
        "scriptHome": "/tmp/.backups",
        "workingDir": "/home",
        "dssDist": "/tmp/cbrpack",
        "cph": "1",
        "optThreshold": "32",
        "securityMode": "userProvided",
        "verbose": "5",
        "maxLogSize": "1",
        "maxStreamsPerBackupHost": "10"
      }
    },
    "dss_multi_82": {
        "dssClusterInfo": {
          "cbrNode": "10.221.104.175",
          "nodes": [
            "10.221.104.175",
            "10.221.104.176"
          ],
          "dssClusternodekeyHashes": {
            "10.221.104.175": "28d0288c869d7021a2c855124c4ee5367e3cb6ede8ffc4
d74a883ff655ba0c21",
```

```
            "10.221.104.176": "a8d134c712ba8c2f8a75ba3c2ce1baf80bbbe199ed5047
6e2c36f8e84adce214"
          }
        },
        "settings": {
          "jobCleanupTimeoutSec": 28800,
          "dssMinRam": "90909",
          "dssMinStoragePerBkupNode": "10485",
          "concurrentCompactions": "8",
          "sstableloaderMemsize": "4096M",
          "concurrentTransfers": "2",
          "scriptHome": "/tmp/.backups",
          "workingDir": "/home",
          "dssDist": "/tmp/cbrpack",
          "cph": "1",
          "optThreshold": "32",
          "securityMode": "userProvided",
          "verbose": "5",
          "maxLogSize": "1",
          "maxStreamsPerBackupHost": "10"
        }
    }
  }
}
```

Enter the RSA key of the CBR node. To obtain the RSA key, log in to the CBR node with the host credentials you plan to use with the Data staging servers and run the `cat /etc/ssh/ssh_host_rsa_key.pub |awk '{print $2}' |base64 -d |sha256sum |awk '{print $1}'` command.

**Table 2-1**

| Key | Description |
|---|---|
| `productionCluster` | Lists one or more Cassandra under this key as a subkey. |
| `nodes` | Lists all the nodes in the Cassandra cluster. The values of this key must always be an IPv4 address and can be listed as a list which is comma separated. |

**Table 2-1** *(continued)*

| Key | Description |
|---|---|
| `prodClusternodeKeyHashes` | Lists all the nodes in the nodes key with the public SHA 256 RSA key.<br>The RSA key can be obtained by logging on to the node using the host credentials you plan to use with the Data staging servers or the production node and run the following command `cat /etc/ssh/ssh_host_rsa_key.pub \|awk '{print $2}' \|base64 -d \|sha256sum \|awk '{print $1}'` |
| `dssClusterName` | Lists the name of the dss cluster to be used for the production cluster backup. |
| `dssClusterInfo` | Specifies the details of the DSS cluster. |
| `cbrNode` | Specifies the CBR node IPv4 address which is used as the coordinator node on the DSS cluster. |
| `dssClusternodekeyHashes` | Lists all the nodes in the nodes key under the dssClusterInfo with the public RSA key. |
| `dssClusterName` | Lists the name of the DSS cluster. |
| `settings` | Contains all the settings of the DSS cluster which are used for that DSS cluster. |
| `dssMinRam` | Sets minimum RAM requirement for Data optimization on data staging server. |
| `dssMinStoragePerBkupNode` | Sets minimum storage requirement for cata optimization on data staging server. |
| `concurrentCompactions` | Sets maximum number of compactions which can run concurrently. |
| `sstableloaderMemsize` | Sets heap memory size for Cassandra sstableloader. |
| `concurrentTransfers` | The value for concurrent transfers which is used for parallel data transfer from Production to Data staging server. Default value is 8. |

**Table 2-1** *(continued)*

| Key | Description |
|---|---|
| `scriptHome` | The path of directory which is used for CBR package installation on the Cassandra nodes.<br>**Note:** The path must exists on both Prod cluster and DSS nodes and have full access to the host user account configured with NetBackup for nodes. |
| `workingDir` | The path of the directory used for Cassandra data processing. This path contains the schema files, binary files, and DB files. |
| `dssDist` | The path is used as the thin-client distribution directory on the data staging servers.<br>**Note:** The path must exist on all DSS nodes and must have full access to the host user account configured with NetBackup for DSS nodes. |
| `cph` | Number of connections per backup host from Data staging server cluster. The default value is 8. |
| `optThreshold` | The Optimization Threshold value which refers to the maximum number of column family to be optimized at same time. Value of Optimization Threshold ranges between 4 to 32. |
| `securityMode` | User needs to provide the key value as *userProvided*. This ensures that your RSA keys are validated at the time of connecting to concern nodes. |
| `verbose` | Sets the logging level for CBR logging. Value of verbose ranges between 1 to 5 only. |
| `maxLogSize` | Sets maximum size value for log files. |
| `maxStreamsPerBackupHost` | Sets maximum number of Streams per Backup Host. It is recommended that total number of streams for the job should match the number of DSS nodes or number of backup hosts. |

**Table 2-1** *(continued)*

| Key | Description |
|---|---|
| `dataCenterName` | Data center name to use for the backup. NetBackup only backups from the nodes in this data center. Specify the name of the data center that is geographically co-located with the media/backup host of NetBackup for better performance in backup and restores. If this field is left empty NetBackup backs up data from all data centers in the cluster irrespective of their geographic location. Hence if your cluster has data centers based on geography do specify the data center co-located with the NetBackup media/backup host to get efficient backups and restores. |
| `nodeDownThresholdPercentage` | Specify the percentage of nodes from the Cassandra cluster that can go down. If there are more nodes that are found to be down than this percentage NetBackup fails the backup. This is to ensure that the user can define the percentage of nodes that can be down and NetBackup can still continue to backup the cluster. |
| `jobCleanupTimeoutSec` | This parameter is the timeout in seconds to allow the next backup to continue on this cluster. Specify this timeout to the typical time it takes to backup this cluster. If a job fails and NetBackup was not able to do the clean up this time out value will be used to force a cleanup the next job executes for the same cluster. The next job cleans up the remanent meta data after this timeout value has passed else the next job wont do the cleanup. |

Chapter 3

# Performing backups and restores of Cassandra

This chapter includes the following topics:



## Backing up a Cassandra cluster

You can either schedule a backup job or run a backup job manually. See, NetBackup Administrator's Guide, Volume I

For Cassandra, full and differential incremental backups are supported.

For overview of the backup process, See "Protecting Cassandra data using NetBackup" on page 5.

## Pre-requisites for Cassandra Restore

NetBackup supports recovery of Cassandra entire cluster, keyspace, or column family level. The backup images which are selected for recovery determine the point in time of recovery.

- Backup host, data staging-server, Cassandra clusters must be on RHEL. For details of supported versions, refer to *Software compatibility list*.

- Ensure to have enough free space on all the Data staging servers in the DSS cluster to run a restore operation. Free space must be two times greater than the largest object being recovered.

  **Note:** You can query the catalogs to find the object details before running a recovery. If enough space is not available on the DSS cluster, NetBackup fails the recovery job.

- Make sure that Cassandra service is up and running on all the data staging servers.
- Ensure to have enough free space on the target Cassandra cluster.
- The target Cassandra cluster must be fully functional with access control to the appropriate users.

  **Note:** The Cassandra user account that is configured in NetBackup is used to restore the data. The recovered data is accessible to this account after the restore operation is complete.

- For NetBackup 10.0, backup and restore are supported by CLI. Create policy, submit backup and job monitoring is supported by java GUI.

Recovery to original Cassandra cluster, keyspace, column family.

- To recover Cassandra data back to the original location, ensure that the original Cassandra cluster is up and running and also, all the nodes are connected.
- The images which need to be recovered must be identified.
- Ensure that all the images of one backup operation are selected.
- Ensure to run `bpclimagelist` command on the NetBackup primary server and get a list of images for a particular Cassandra cluster.

  ```
  bpclimagelist –client <Cassandra cluster name> -ct 44 –K –L
  ```

  The output shows a list of backup images for the given Cassandra cluster.
- Whenever you upgrade cassendra or make any schema change, initiate a full backup before any incremental backup job.
- Choose the images from the `bpclimagelist` command such that all the images of one full backup are selected. From the list of images for restore identify the lesser timestamp as the start time and the higher timestamp as end time.

- To check the contents of the images you selected please run the following command on NetBackup primary server.

```
bplist –C <Cassandra cluster name>
-t 44
–R
–l
–s <start time MM/DD/YYYY HH:MM>
-e <end time MM/DD/YYY HH:MM>
/
```

  The output shows a list of backup up files in the backup images which are selected as per start and end time.

- When you can see the key spaces and column families that you want to restore, then run the restore command on the NetBackup primary server.
- You must specify a rename file to the `bprestore` command for Cassandra restores. Create a file with the following contents as the rename file and pass the path of this file to `bprestore` command.

```
• Rename file:
{
 "recoveryOptions" : "BIGDATA_CASSANDRA"
}
```

- You must also need to provide the restore selections in case you want to restore the entire cluster specify the restore selection as follows restore selection file.

```
Restore selection file
{
 "restoreSelections" : { }
}
```

- To do the actual restore operation you need to run the following command on the NetBackup primary server `Bprestore` command.

```
-S Master_Server_Name
-C  <Cassandra cluster name>  (Client_Name specified during Backup)
-D  <Restore host name>
-s mm/dd/yyyy hh:mm
-e mm/dd/yyyy hh:mm  (Date Time range)
-t 44 (For Bigdata Policy Type -t 44)
-f <Restore selection file>
```

```
-R <Rename file>
-cassandra_restore
```

**Note:** `-p` option is not applicable for Cassandra.

- Restore selections for a granular restore operation specifies the keyspace and column family.

```
Restore selection
{
 "restoreSelections" : {
  "<keyspace name>" : ["<column family name>"]
  }
}
```

# Configurations for Cassandra Restore

- The restore operation requires a Data Staging Server cassandra cluster to stage the backup image. It is then restored to the Cassandra production cluster. Hence configure a DSS cluster along with the target Cassandra cluster.

  **Note:** If the restore is to the original Cassandra cluster and it is already configured in NetBackup skip the following steps.

- To configure an alternate Cassandra cluster to restore the data, do the following:

  `Tpconfig` of Cassandra cluster:

  - Provide the credentials of the target cluster using the `tpconfig` command.`/tpconfig -add -application_type cassandra -application_server <application cluster name> -application_server_user_id <cassandra user id> -password <cassandra user password> -host_user_id <host user id> -host_password <host password> -host_RSA_key <host RSA key.`
  - Provide the credentials of the target DSS cluster using the `tpconfig` command.

    **Note:** This command is the same as above with the DSS cluster names in it.

`Cassandraconfig` of target cluster:

- If your target Cassandra cluster is different from the backup source, add the Cassandra configuration details in the `cassandra.conf` file on the primary server .
  For example:

```
bprestore -S emidas105.vxindia.veritas.com
-C Test_Cluster72 -D emidas105.vxindia.veritas.com
-s 03/09/2021 17:17 -e 03/09/2021 17:17 -t 44 -L /
input/cassandra_progress.log -f /input/cassandra_filelist_cluster
 -R /input/cassandra_rename_cluster -cassandra_restore
```

- Cassandra thin client creates temporary files in `/tmp` on Cassandra production nodes.
- The restore operation creates number of child jobs depending on:
  - Number of specified backup hosts in the restore operation
  - Streams per backup host
  - Number of DSS nodes

**Note:** Number of jobs = minimum ((backup hosts * streams per backup host), number of DSS nodes)

# Restore combinations

The following are supported restore combinations.

**Table 3-1** Restore combination

| Restore Combination. | Sample selection file | Sample rename file. |
|---|---|---|
| To restore all the key spaces and column families to the original cluster. | `{ "restoreSelections": {} }` | `{ "overwrite": "true" }` |

**Table 3-1** Restore combination *(continued)*

| Restore Combination. | Sample selection file | Sample rename file. |
|---|---|---|
| To restore all the key spaces and column families but rename one or more key spaces. Restore to the original cluster. | <pre>{<br>"restoreSelections": {}<br>}</pre> | <pre>{<br>"overwrite": "true",<br>"alternateRecoveryOptions": [<br>{<br>"keyspace": {<br>"name": "ks_oldname",<br>"newName": "ks_newname"<br>},<br>"columnFamilies": [],<br>"strategy": {}<br>}<br>]<br>}</pre> |
| To restore all key spaces and column families and change the strategy and / or replication factor of one or more key spaces on the original cluster. | <pre>{<br>"restoreSelections": {}<br>}</pre> | <pre>{<br>"overwrite": "true",<br>"alternateRecoveryOptions": [<br>{<br>"keyspace": {<br>"name": "ks_name"<br>},<br>"columnFamilies": [<br>],<br>"strategy": {<br>"name": "simpleStrategy",<br>"replica": "5"<br>}<br>}<br>]<br>}</pre> |

**Table 3-1** Restore combination *(continued)*

| Restore Combination. | Sample selection file | Sample rename file. |
|---|---|---|
| To restore all key spaces and column families to the original cluster and rename one or more key spaces along with change in the strategy or the replication factor. | <pre>{<br>"restoreSelections": {}<br>}</pre> | <pre>{<br>"overwrite": "true",<br>"alternateRecoveryOptions": [<br>{<br>"keyspace": {<br>"name": "ks_name",<br>"newName": "ks_newname"<br>},<br>"columnFamilies": [],<br>"strategy": {<br>"name": "simpleStrategy",<br>"replica": "9"<br>}<br>}<br>]<br>}</pre> |
| To restore one or more key spaces and all their column families to the original cluster. | <pre>{<br>"restoreSelections": {<br>"ks_name": [<br>]<br>}<br>}</pre> | <pre>{<br>"overwrite": "true"<br>}</pre> |

**Table 3-1** Restore combination *(continued)*

| Restore Combination. | Sample selection file | Sample rename file. |
|---|---|---|
| To restore one or more key spaces and all their column families to the original cluster but with a different keyspace name. The selection should include all the key spaces that you want to rename. | { "restoreSelections": { "ks_name": [ ] } } | { "overwrite": "true", "alternateRecoveryOptions": [ { "keyspace": { "name": "old_ks_name", "newName": "new_ks_name" }, "columnFamilies": [], "strategy": {} } ] } |
| To restore one or more key spaces and all their column families to the original cluster. Also, change the strategy of all these key spaces or specify a different replication factor. The selection should include all the key spaces that you want to rename. | { "restoreSelections": { "ks_name": [ ] } } | { "overwrite": "true", "alternateRecoveryOptions": [ { "keyspace": { "name": "ks_oldname" }, "columnFamilies": [ ], "strategy": { "name": "simpleStrategy", "replica": "9" } } ] } |

**Table 3-1** Restore combination *(continued)*

| Restore Combination. | Sample selection file | Sample rename file. |
|---|---|---|
| To restore one or more key spaces and all their column families to the original cluster but with a different keyspace name. Also change the strategy of all these key spaces or specify a different replication factor. The selection should include all the key spaces that you want to rename. | <pre>{<br>"restoreSelections": {<br>"ks_name": [<br>]<br>}<br>}</pre> | <pre>{<br>"overwrite": "true",<br>"alternateRecoveryOptions": [<br>{<br>"keyspace": {<br>"name": "ks_oldname",<br>"newName": "ks_newname"<br>},<br>"columnFamilies": [],<br>"strategy": {<br>"name": "simpleStrategy",<br>"replica": "9"<br>}<br>}<br>]<br>}</pre> |
| To restore one or more key spaces and specific column families in each of these key spaces to the original cluster. | <pre>{<br>"restoreSelections": {<br>"ks_name": [<br>"cf_name"<br>]<br>}<br>}</pre> | <pre>{<br>"overwrite": "true"<br>}</pre> |

**Table 3-1** Restore combination *(continued)*

| Restore Combination. | Sample selection file | Sample rename file. |
|---|---|---|
| To restore one or more key spaces and specific column families in each of these key spaces to its original cluster but with a different column family name for each of the column families which are restored. Each column family being renamed should be specified in the selection. The strategy or the replication factor cannot be changed when restoring specific column families. | <pre>{<br>"restoreSelections": {<br>"ks_name": [<br>"cf_name"<br>]<br>}<br>}</pre> | <pre>{<br>  "overwrite": "true",<br>  "alternateRecoveryOptions": [<br>    {<br>      "keyspace": {<br>        "name": "ks_name"<br>      },<br>      "columnFamilies": [<br>        {<br>          "name": "cf_name",<br>          "newName": "cf_newname"<br>        }<br>      ],<br>      "strategy": {}<br>    }<br>  ]<br>}</pre> |

**Note:** Whenever NetBackup restores a keyspace, it restores with durable writes set to true. If you want to change this attribute, you can change it in Cassandra, after the restore is complete.

Chapter 4

# Troubleshooting

This chapter includes the following topics:



## About NetBackup for Cassandra debug logging

NetBackup maintains process-specific logs for the various processes that are involved in the backup and restore operations. Examining these logs can help you to find the root cause of an issue.

These log folders must already exist in order for logging to occur. If these folders do not exist, you must create them.

**Table 4-1** NetBackup logs related to Cassandra

| Log Folder | Messages | Logs reside on. |
|---|---|---|
| `/usr/openv/volmgr /debug/tpcommand/` | Credentials configuration | Primary server |
| `install_path/NetBackup/logs/nbaapidiscv` | Discovery | Backup host |
| `install_path/NetBackup/logs/bptm` | Policy validation, backup, and restore operations. | Media server |
| `install_path/NetBackup/logs/bpdm` | Image cleanup, verification, import, and duplication logs. | Backup host |

**Table 4-1** NetBackup logs related to Cassandra *(continued)*

| Log Folder | Messages | Logs reside on. |
|---|---|---|
| `install_path/logs/nbrmms` | Configuration management, access to media server resources, monitoring, and event notifications. | Media server |
| /usr/openv/NetBackup/logs/nbaapi.req_handler | Recovery orchestrator, debug logs and thin client debug logs | Backup host |
| `user/openv/NetbCakup/logs/bpbrm` | Debug logs of Backup restore manager | Media server |

For more details, refer to the NetBackup Logging Reference Guide.

# Errors while backing up the jobs

Stale processes on backup jobs:

In some cases, canceling the child job for a Cassandra backup operation may keep some stale processes running on the CBR node and backup hosts. These needs to be killed manually before the next job is triggered.

The following are the processes on:

- Backup host: `bpbkarv`
- CBR node: `nbcbr backup start`

To kill these processes:

- Step to identify PID: `ps -ef | grep <bpbkarv>`
- ps -ef | grep <nbcbr backup start>.
- To kill the identified -9 <pid_obtained_from_the_above_commands>.

# Common errors

**Table 4-2**

| Error | Description |
|---|---|
| Error 6654: | Check the case in the `App_type`. It must be lower case.<br><br>If the backup host is NetBackup client, it must be mentioned in primary server's `bp.conf` as **APP_PROXY_SERVER = NetBackup client name**.<br><br>Ensure that the password length is not beyond the limited characters. |
| Error 6661: Unable to find the configuration parameter. | Validate parameters in the `cassandra.conf`file. For details, See "Configurations for Cassandra Restore" on page 23. |
| Error 3237: backup fails | Ensure that the entire of `RSA config,, host config` in the `cassadnra.con` file are correct. |