NetBackup™ for MongoDB Administrator's Guide

Release 10.0

VERITAS™

NetBackup™ for MongoDB Administrator's Guide

Last updated: 2022-03-08

Legal Notice

Copyright © 2022 Veritas Technologies LLC. All rights reserved.

Veritas, the Veritas Logo, and NetBackup are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third-party software for which Veritas is required to provide attribution to the third party (“Third-party Programs”). Some of the Third-party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the Third-party Legal Notices document accompanying this Veritas product or available at:

https://www.veritas.com/about/legal/license-agreements

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Veritas Technologies LLC and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. Veritas Technologies LLC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

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 protecting MongoDB using NetBackup 7

About protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup 7
Protecting MongoDB data using NetBackup 12
NetBackup for MongoDB terminologies 14
Limitations 15
Prerequisites and the best practices for protecting MongoDB 16

Chapter 2 Verify the pre-requisites for the MongoDB plug-in for NetBackup 20

Operating system and platform compatibility 20
Prerequisites for configuring the MongoDB plug-in 20

Chapter 3 Configuring NetBackup for MongoDB 22

About the MongoDB configuration tool 22
Prerequisites for manually creating the mongodb.conf file 24
Configuring backup options for MongoDB using the mongodb.conf file 25
Including the configuration file path on NetBackup master server allowed list 32
Obtaining the RSA key of the MongoDB nodes 33
Adding MongoDB credentials in NetBackup 34
About the credential configuration file 35
How to add the MongoDB credentials in NetBackup 37
About the MongoDB roles for protecting the data 38
Using a non-root user as a host user 38
Managing backup hosts 39
Including a NetBackup client on NetBackup master server allowed list 40

Chapter 4 Backing up MongoDB using NetBackup 42

Backing up MongoDB data 42
Backing up a MongoDB cluster 44
Prerequisites for backing up a MongoDB cluster 44
Configuring NetBackup policies for MongoDB plug-in 46
Creating a BigData backup policy 46
Creating BigData policy using the NetBackup Administration Console 46
Using the Policy Configuration Wizard to create a BigData policy for MongoDB clusters 47
Using the NetBackup Policies utility to create a BigData policy for MongoDB clusters 47
Using NetBackup Command Line Interface (CLI) to create a BigData policy for MongoDB clusters 48

Chapter 5 Restoring or recovering MongoDB data using NetBackup 51

Restoring MongoDB data 51
Prerequisites for MongoDB restore and recovery 52
About the restore scenarios for MongoDB database from the BAR interface 55
High-level steps involved in the Restore and Recovery process 57
Using the BAR interface to restore the MongoDB data on the same cluster 58
Using the BAR interface to restore the MongoDB data on an alternate cluster 59
About restoring MongoDB data in a high availability setup on an alternate client 62
Recovering a MongoDB database using the command line 63
Creating or modifying the rename file 64
Using the command line to recover a MongoDB database 65
Manual steps after the recovery process 68

Chapter 6 Troubleshooting 69

About NetBackup for MongoDB debug logging 69
Known limitations for MongoDB protection using NetBackup 70

Appendix A Additional information 83

Sample MongodB configuration utility workflow to add and update MongodB credentials 83

Index 88

Chapter 1

Overview of protecting MongoDB using NetBackup

This chapter includes the following topics:

- About protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup
- Protecting MongoDB data using NetBackup
- NetBackup for MongoDB terminologies
- Limitations
- Prerequisites and the best practices for protecting MongoDB

About protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup

NetBackup supports the protection of the following MongoDB configurations:

- Sharded MongoDB cluster
- Replica set MongoDB cluster
- Standalone MongoDB cluster without replica sets

Protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup

Use the NetBackup for MongoDB plug-in to protect your sharded (MongoDB cluster with configuration server and shards), replica set, or standalone MongoDB cluster using the following high-level steps:

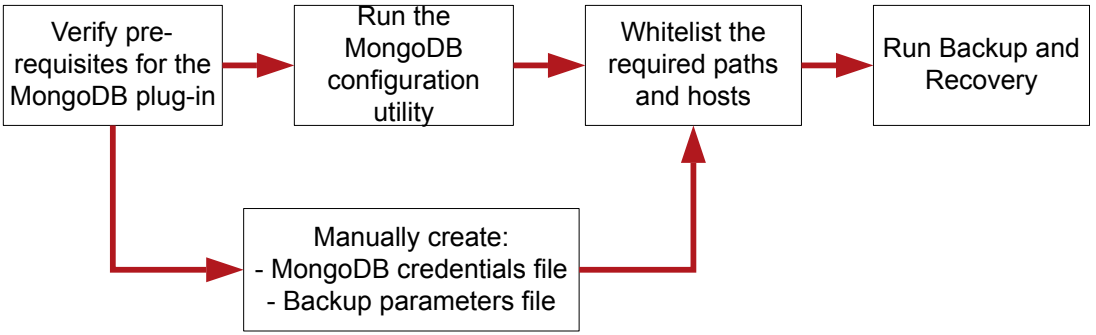


Table 1-1 Protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup

Step overview	Details
Use NetBackup to protect MongoDB	On a very high level, to protect MongoDB, you need: ■ NetBackup master server ■ NetBackup media server ■ A backup host (NetBackup media server or a NetBackup client) Refer to the NetBackup compatibility lists for the supported master and media server configurations. The backup host (NetBackup media server or a NetBackup client) is supported only on an RHEL or a SUSE host. NetBackup appliance including Flex appliance is also supported as a NetBackup master, media server, or as a client that can act as a backup host. Refer to the following topics to get a protection overview and the best practices: ■ See "Protecting MongoDB data using NetBackup" on page 12. ■ See "NetBackup for MongoDB terminologies" on page 14. ■ See "Prerequisites and the best practices for protecting MongoDB" on page 16.

Table 1-1 Protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup *(continued)*

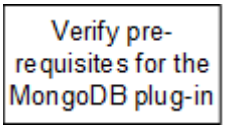
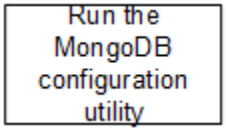
Step overview	Details
Verify the pre-requisites for the MongoDB plug-in 	Refer to the following topics before you use the plug-in: ■ See "Operating system and platform compatibility" on page 20. ■ See "Prerequisites for configuring the MongoDB plug-in" on page 20.
Run the MongoDB configuration tool 	Run MongoDB configuration tool to generate the following files automatically: ■ The file for the MongoDB cluster topology credentials. ■ The MongoDB configuration file that configures the global NetBackup parameters for the MongoDB cluster. You can access the MongoDB configuration tool using the `tpconfig` command line on the NetBackup master server. The path to access the `tpconfig` command is `/usr/openv/volmgr/bin/` for UNIX and `<install_path>\Volmgr\bin\` for Windows. For more information, See "About the MongoDB configuration tool" on page 22.

Table 1-1 Protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup *(continued)*

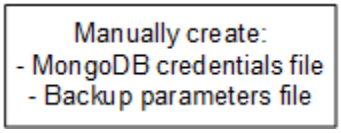
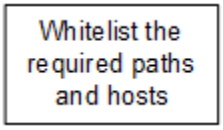
Step overview	Details
Configure the MongoDB plug-in and the communication between NetBackup and MongoDB **Note:** If you use the MongoDB configuration tool, a few these configuration steps are not required.  	Create a `mongodb.conf` file to configure backup options in NetBackup: ■ See "Prerequisites for manually creating the mongodb.conf file" on page 24. ■ See "Configuring backup options for MongoDB using the mongodb.conf file " on page 25. ■ See "Including the configuration file path on NetBackup master server allowed list" on page 32. **Note:** If you use the MongoDB configuration tool, these steps are not required. Get the RSA key of the MongoDB node for adding MongoDB credentials to NetBackup: ■ See "Obtaining the RSA key of the MongoDB nodes" on page 33. **Note:** If you use the MongoDB configuration tool, these steps are not required. Add the MongoDB credentials to NetBackup to facilitate communication: ■ See "Adding MongoDB credentials in NetBackup" on page 34. ■ See "About the credential configuration file" on page 35. ■ See "How to add the MongoDB credentials in NetBackup" on page 37. **Note:** If you use the MongoDB configuration tool, these steps are not required. Give the appropriate permissions to a NetBackup user in MongoDB: ■ See "About the MongoDB roles for protecting the data" on page 38. To use a non-root user or a user without root permissions as a host user: ■ See "Using a non-root user as a host user" on page 38. Identify and configure a backup host. ■ See "Managing backup hosts" on page 39. ■ To use NetBackup client as a backup host, include the NetBackup client on the master server on the allowed list. See "Including a NetBackup client on NetBackup master server allowed list" on page 40.inlcu

Table 1-1 Protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup *(continued)*

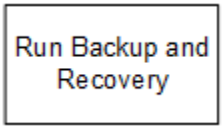
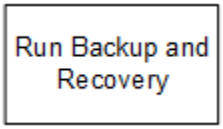
Step overview	Details
Back up your MongoDB database using NetBackup 	Overview of the backup process: ■ See "Backing up MongoDB data" on page 42. ■ See "Backing up a MongoDB cluster" on page 44. Prerequisites or the best practices for backing up MongoDB databases: ■ See "Prerequisites for backing up a MongoDB cluster" on page 44. Creating and using a backup policy: ■ See "Creating a BigData backup policy" on page 46. ■ NetBackup Administration Console See "Creating BigData policy using the NetBackup Administration Console" on page 46. Create a BigData policy using: ■ Policy configuration wizard See "Using the Policy Configuration Wizard to create a BigData policy for MongoDB clusters" on page 47. ■ NetBackup policies utility See "Using the NetBackup Policies utility to create a BigData policy for MongoDB clusters" on page 47. ■ NetBackup command line interface See "Using NetBackup Command Line Interface (CLI) to create a BigData policy for MongoDB clusters " on page 48.

Table 1-1 Protecting a sharded, replica set, or standalone MongoDB cluster using NetBackup *(continued)*

Step overview	Details

Restore and recover the MongoDB database
 
	Overview of the restore and the recovery process:
	■ See "Restoring MongoDB data" on page 51.
	Prerequisites or the best practices for backing up MongoDB databases:
	■ See "Prerequisites for MongoDB restore and recovery" on page 52.
	Understanding the restore and the recovery scenarios for MongoDB:
	■ See "About the restore scenarios for MongoDB database from the BAR interface" on page 55.
	■ See "High-level steps involved in the Restore and Recovery process" on page 57.
	■ Restore and recovery using the BAR UI:
	  ■ See "Using the BAR interface to restore the MongoDB data on the same cluster" on page 58.
	  ■ See "Using the BAR interface to restore the MongoDB data on an alternate cluster" on page 59.
	■ Restore and recovery using the command line interface:
	  ■ See "Recovering a MongoDB database using the command line" on page 63.
	  ■ See "Creating or modifying the rename file" on page 64.
	  ■ See "Using the command line to recover a MongoDB database" on page 65.
	  ■ Manual steps after the recovery process
	  See "Manual steps after the recovery process" on page 68.

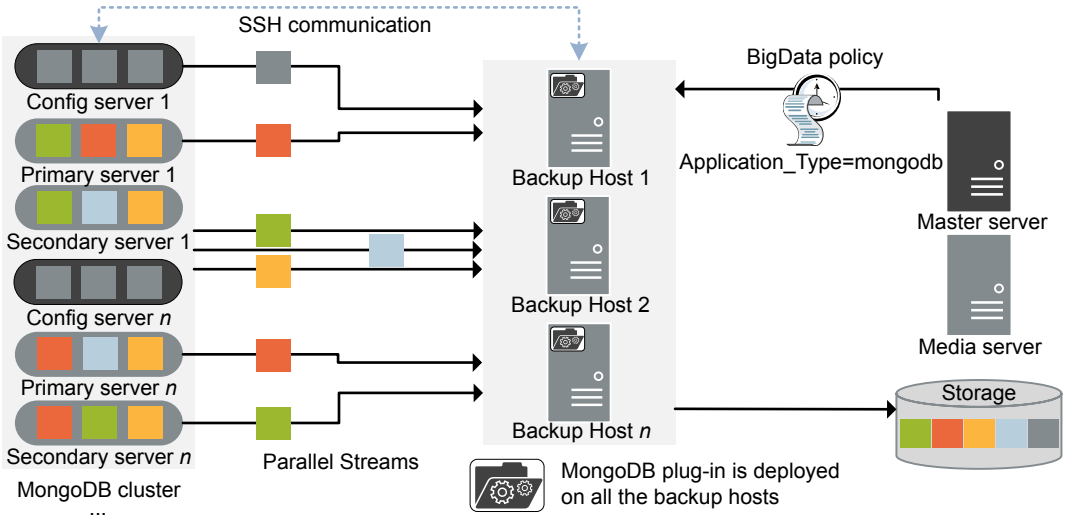
Protecting MongoDB data using NetBackup

Using the NetBackup Parallel Streaming Framework (PSF), MongoDB data can now be protected using NetBackup.

The following diagram provides an overview of how MongoDB data is protected by NetBackup.

Also, review the definitions of terminologies.See "NetBackup for MongoDB terminologies" on page 14.

Figure 1-1 Architectural overview



As illustrated in the diagram:

- The data is backed up in parallel streams wherein the data nodes stream data blocks simultaneously to multiple backup hosts. The job processing is accelerated due to multiple backup hosts and parallel streams.
- The communication between the MongoDB cluster and the NetBackup is enabled using the NetBackup plug-in for MongoDB.
- 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 master server as a backup host. Also, depending on the number of replica sets or sharding, you can add or remove backup hosts. You can scale up your environment easily by adding more backup hosts.
- The communication between the configuration server, secondary nodes and the backup hosts happens over SSH.
- The NetBackup Parallel Streaming Framework enables a thin client-based, agentless backup wherein the backup and restore operations run on the backup hosts. The NetBackup thin client binary is automatically pushed to the MongoDB cluster nodes during the backup and recovery operations. This thin client is automatically removed after the backup and recovery operations complete.

There is no agent management required on the cluster nodes. Also, NetBackup is not affected by the MongoDB cluster upgrades or maintenance.

For more information:

- See "Backing up MongoDB data" on page 42.
- See "Restoring MongoDB data" on page 51.
- See "Limitations" on page 15.
- For information about the NetBackup Parallel Streaming Framework (PSF) refer to the *NetBackup Administrator's Guide, Volume I*.

NetBackup for MongoDB terminologies

The following table defines the terms you come across when using NetBackup for protecting MongoDB cluster.

Table 1-2 NetBackup terminologies

Terminology	Definition
Compound job	A backup job for MongoDB data is a compound job. ■ The backup job runs a discovery job for getting information of the data to be backed up. ■ Child jobs are created for each backup host that performs the actual data transfer. ■ After the backup is complete, the job cleans up the snapshots on the backup nodes, removes the thin client and is marked complete.
Discovery job	When a backup job is executed, first a discovery job is created. The discovery job communicates with the config server and gathers information of the shards that need to be backed up and the associated nodes. At the end of the discovery, the job populates a workload discovery file that NetBackup then uses to distribute the workload amongst the backup hosts.
Child job	For backup, a separate child job is created for each backup host to transfer data to the storage media. A child job can transfer data blocks from multiple secondary nodes.
Workload discovery file	During discovery, when the backup host communicates with the config server, a workload discovery file is created. The file contains information about the data files to be backed up and the associated data nodes.
Workload distribution file	After the discovery is complete, NetBackup creates a workload distribution file for each backup host. These files contain information of the data that is backed up by the respective backup host.

Table 1-2 NetBackup terminologies *(continued)*

Terminology	Definition
Parallel streams	The NetBackup parallel streaming framework allows data blocks from multiple secondary 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. You can configure media servers, clients, or a master server as a backup host. The backup host is also used as destination client during restores.
BigData policy	The BigData policy is introduced to: ■ Specify the application type. ■ Allow backing up distributed multi-node environments. ■ Associate backup hosts. ■ Perform workload distribution.
Application server	■ Sharded MongoDB cluster: Application server is the MongoDB primary config server. ■ Replica set MongoDB cluster: Application server is the primary node of MongoDB. ■ Standalone cluster: Application server is the standalone node.
Primary config server	In a high-availability scenario, the primary config server is the MongoDB instance running in a primary role on a config server replica set. The primary config server must have at least one associated `mongos` service running on the same host.
Fail-over config server	In a high-availability scenario, the config server other than the primary config server that is specified as `alternate_config_server` in the `mongodb.conf` file is referred as the fail-over config server.

Limitations

Consider the following limitations before you deploy the MongoDB plug-in:

- For highly available MongoDB cluster, if fail-over happens during a backup, then the job fails.
- IP address is not supported for the application server and backup host field. You must enter the FQDN, host name, or the short name of the application server or the backup host.
- Encrypted MongoDB environments are not supported.

- English-only MongoDB environments are supported.
- Extended Access Control Lists (ACL) are not recovered after a recovery operation.
- Recovery is not supported for a shrunk MongoDB cluster.
- For standalone MongoDB nodes without replica sets, incremental backups are not supported.
- MongoDB password less service account usage is not supported.
- Protection of MongoDB environments that are deployed or managed using the MongoDB Ops Manager is not supported.
- If you change the Feature Compatibility Version between the full and differential incremental backups, the backups fail.
- The backup of a sharded MongoDB environment can be taken only as a sharded backup configuration and not a replica set or any other backup configuration.
- NetBackup does not support backup and restore of MongoDB deployments which are running using service account `mongod`.

Prerequisites and the best practices for protecting MongoDB

Prerequisites

- For sharded MongoDB clusters, `mongos` and `mongod` processes must be running on the application server that is specified as the client in the backup policy.
- Only RHEL and SUSE platforms are supported for MongoDB clusters and backup hosts.
- The NetBackup for MongoDB plug-in requires that the NetBackup master server, media server, and the backup host are on NetBackup version 8.2 or later.
- Verify that NetBackup supports the MongoDB version that you have. For more information, refer to the Software Compatibility List.
- NetBackup supports the MongoDB clusters that are configured or installed on RHEL and SUSE operating systems.
- NetBackup supports the following MongoDB configurations:
  - Sharded MongoDB cluster (MongoDB cluster with configuration server and shards)
  - Replica set MongoDB cluster

  - Standalone MongoDB without replica sets
- NetBackup supports the following authentication types for MongoDB:
  - No authentication
  - Simple authentication
  - Certificate-based authentication
- NetBackup supports the following file systems for backup and restore:
  - XFS
  - ext4
- Install OpenSSH packages on all the MongoDB nodes. Enable SSH on all the MongoDB nodes.
- NetBackup supports the MongoDB clusters that are configured with the **WiredTiger** storage engine.
- NetBackup protects the MongoDB clusters that are configured or installed locally using the `.tar` files or installed using the MongoDB official repositories.
- NetBackup supports the Differential Incremental backup for MongoDB along with a Full Backup. Cumulative Incremental backups are not supported currently.
- NetBackup recommends that you have at least three configuration servers in your sharded MongoDB environment to support high availability of the backups.
- Do not install the MongoDB plug-in on a server that also has the MongoDB application. A server that has the MongoDB application cannot be used as a backup host.
- Ensure that the local time on the MongoDB server and the backup host are synchronized with the NTP server.
- For sharded MongoDB clusters, the query router role must be present on the config server.
- For the MongoDB cluster that has a SUSE operating system, on all the MongoDB nodes set the `PasswordAuthentication` field to **Yes** in the `/etc/ssh/sshd_config` file.

  After you update the file, restart `sshd`.

  Ensure that all the clusters support the same hash key algorithm (RSA).
- Ensure that the host user credentials that are configured using the `tpconfig` command are of the host user account that is used to configure the MongoDB cluster (MongoDB daemon's host user account that is either root or non-root). For more details, See "Using a non-root user as a host user" on page 38.

- To protect MongoDB version 4.2, upgrade the NetBackup backup hosts that you have defined in the backup policy to version 9.0 along with NetBackup master and media server. Backup host versions earlier than 9.0 can affect the MongoDB 4.2 cluster.
- Ensure that the `ss` command is available on the MongoDB nodes. The `ss` command is required to identify the MongoDB clusters that are created using an externally sourced configuration file.
- To protect MongoDB version 4.2, upgrade the NetBackup backup hosts that you have defined in the backup policy to version 9.0 along with NetBackup master and media server. Backup host versions earlier than 9.0 can affect the MongoDB 4.2 cluster.

Best practices for communication between MongoDB and NetBackup

- If you use a NetBackup client as a backup host, ensure to add the following value in the `bp.conf` file of the NetBackup master server:

  `APP_PROXY_SERVER=NBU_CLIENT_FQDN`
- If the MongoDB host user does not have root permissions, ensure that the user has access to all the temporary paths to copy the thin client (`mdbserver`), logs, snapshots, etc. Add the non-root user to the `sudoers` file in the operating system.
- If you install the MongoDB using the `.tar` file or to a non-default location, add the path of the MongoDB bin folder in the `bashrc` file of the operating system to ensure that you can run the MongoDB commands from the CLI.
- If your MongoDB server uses the SUSE 12.3 operating system, ensure that you can connect to `mongod` and `mongos` process with the `--host <`*`FQDN`*`>` option. For more information, refer to the MongoDB Administrator's Guide.
- When you use the `-host_password` option with the `tpconfig` command and `mongodb.conf HostPassword`, ensure that the password:
  - Does not exceed 63 characters
  - Contains one or more alphanumeric characters: a-z, A-Z, 0-9
  - Contains one or more of the following characters: `-` (hyphen), `_` (underscore), `,`(comma), `.` (period), `?` (question mark)
- When you define the paths for logs, thin clients (`mdbserver`), snapshots, or anything else in the `mongodb.conf` file, ensure that the host user in the credentials file has valid permissions to access these paths.
- To enable SSH, add the following entry in the `sudoers` file:

  `Default <`*`host_user`*`> !requiretty`

Best practices for protecting MongoDB using NetBackup

- Ensure that the MongoDB limits and thresholds are as per the official MongoDB guidelines.
- Ensure that the host name is consistently used in the `tpconfig` command, during the policy configuration, and in the `mongodb.conf` file. For example, if you use the FQDN, use it for all host name instances instead of short names.
- Ensure that `application_server` matches with the host name that is used in the MongoDB environment and verified using the `db.hostInfo()` command.
  For example, the host name value that displayed by `db.hostInfo()`:
  `"hostname" : "<hostname_value>:<port>"`
- Ensure that there are no JSON format errors or typos in the `mongodb.conf` file before you run a backup or restore job.
- Ensure that the path of the security certificates that are added in the `mongod.conf` file and used with the `tpconfig` command are the same for all the MongoDB nodes.
- For simple authentication, configure the same user who is part of the root group from the admin database for every MongoDB node.
- If you use the `mongod.conf` or the `mongos.conf` file to start the MongoDB processes, run the `mongod` file using the absolute system path on the MongoDB cluster. For example, use the following command:
  `mongod --config /home/user1/mongod.conf`
- NetBackup recommends that you run a full backup after making any configuration changes in the MongoDB instance. If an incremental backup is scheduled to run after you make the configuration changes, then run a full backup manually before the incremental backup.
  For examples, when you modify the MongoDB Feature Compatibility Version (FCV), MongoDB version, authentication type, topology (addition of new shards or removal of existing shards), storage parameters, etc. then run a full backup.

Chapter 2

Verify the pre-requisites for the MongoDB plug-in for NetBackup

This chapter includes the following topics:

- Operating system and platform compatibility
- Prerequisites for configuring the MongoDB plug-in

Operating system and platform compatibility

With this release, RHEL and SUSE platforms are supported for MongoDB clusters and NetBackup backup hosts.

For more information, see the:

- NetBackup Database and Application Agent Compatibility List
  NetBackup Master Compatibility List.

Prerequisites for configuring the MongoDB plug-in

Consider the following when you configure NetBackup for MongoDB:

Prerequisites:

- Add the MongoDB thin client package that is part of `vxupdate_nb_version SJA` to the package repository on the NetBackup primary server.

  Note: The version number must be the same as the backup host.

To add the package, run the `nbrepo` command on the NetBackup primary server:

```
./nbrepo -add vxupdate_nb_version_suse_x64.sja
./nbrepo -add vxupdate_nb_version_redhat_x64.sja
```

For a MongoDB host with CentOS operating system, add the Linux RHEL VxUpdate package of the NetBackup version of the backup host in the package repository on the NetBackup primary server.

Note: If the package is not added, the MongoDB backups can fail with error - 6729: "Unable to download the thin client from the package repository."

- Use consistent conventions for host names of backup hosts, media servers, and primary server. For example, if you are using the host name as **MongoDB.veritas.com** (FQDN format) use the same everywhere, specially while running the `tpconfig` command.
- Ensure that the backup host can communicate with all the MongoDB nodes.
- Ensure that the `bindIp` setting in the configuration file of `mongod` instance on the MongoDB hosts has value `0.0.0.0`.

Best practices:

- Add the entries of all the nodes of the MongoDB cluster to the`/etc/hosts` file on all the backup hosts. You must add the host name in FQDN format.
  Or
  Add the appropriate DNS entries in the `/etc/resolv.conf` file.

Chapter 3

Configuring NetBackup for MongoDB

This chapter includes the following topics:

- About the MongoDB configuration tool
- Prerequisites for manually creating the mongodb.conf file
- Configuring backup options for MongoDB using the mongodb.conf file
- Obtaining the RSA key of the MongoDB nodes
- Adding MongoDB credentials in NetBackup
- Using a non-root user as a host user
- Managing backup hosts

About the MongoDB configuration tool

NetBackup provides a command line based configuration tool that enables you to accurately capture and update the information that is required to protect the MongoDB.

You can use the MongoDB configuration tool to generate the following files automatically:

- The credentials file that configures the MongoDB cluster topology and credentials for NetBackup.
  For more information about the credential configuration file and the manual method to create it, refer to the following topic:
  See “Adding MongoDB credentials in NetBackup” on page 34.

- The MongoDB configuration file that configures the global NetBackup parameters for the MongoDB cluster.
  For more information about the MongoDB configuration file and the manual method to create it, refer to the following topic:
  See "Configuring backup options for MongoDB using the mongodb.conf file " on page 25.

Note: You can create the two files manually, but you must ensure that the formatting and the parameters are correct.

You can access the MongoDB configuration tool using the `tpconfig` command line on the NetBackup master server. The path to access the `tpconfig` command is `/usr/openv/volmgr/bin/`.

- On a Windows master server run the `tpconfig -mongo_configuration`command to activate Mongo configuration interface.
- On a Linux and Solaris master server, run `./tpconfig` and `tpconfig` and select the fourth option for MongoDB configuration.

See "Sample MongodB configuration utility workflow to add and update MongodB credentials" on page 83.

For more information about the `tpconfig` command, refer to the *NetBackup Commands Reference Guide*.

Whitelist the `mongodb.conf` file path in `bp.conf` using the `bpcd_allowed_path` option. For more information, See "Including the configuration file path on NetBackup master server allowed list" on page 32.

Adding MongoDB credentials during recovery to an alternate MongoDB cluster

To recover to an alternate MongoDB cluster, use the configuration tool to add credentials of the alternate cluster in the existing cluster credentials.

Sharded MongoDB cluster

1. Use the configuration tool to update credentials of the existing cluster.
2. Add new configuration server using the **Add new secondary config server** option and save.
3. Add shards of the new cluster using the **Add new shard host server** option and save.
4. Initiate the alternate recovery job.

Replica Set MongoDB

1. Use the configuration tool to update credentials of the existing replica set.
2. Add new primary server using the **Add secondary server** option and save.
3. Add all of the secondary servers using the **Add secondary server** option and save.
4. Initiate the alternate recovery job.

Note: If you are using the credentials file, you can manually update the file and upload the file using the `tpconfig` command.

Prerequisites for manually creating the mongodb.conf file

Note: If you use the MongoDB configuration tool, these manual steps are not required.

- If you do not specify any values in the `mongodb.conf` file for MongoDB cluster ports and paths to deploy the thin clients, create snapshots, or logs, the default values are considered.
- The minimum value of the `max_streams` field is 32. If `max_streams` is not defined, the default value is 32 parallel data streams per backup host.
  For the `max_streams` field in the `mongodb.conf` file, the value of the backup host takes priority over the `global_default` value.
  For example, the value 32 takes priority over the value 34 and the job runs 32 streams during a backup in this scenario:

  ```
  "max_streams": {
                "global_default": 34,
                "Backup_Host":32}
  ```

  The backup streams are distributed across the backup hosts as defined in the backup policy and not as per the backup hosts that are defined in the `max_streams` option in the `mongodb.conf` file.
- For a sharded MongoDB environment, ensure that the `mongodb.conf` file has the latest primary config server and secondary config server.

- Ensure that the folders or directories that are mentioned in the `mongodb.conf` file are available on the MongoDB cluster. For example, the folders or directories for `snapshot_mount_path`, `oplog_location`, `logdir`, etc.
- Give the host user access to the port and the port range that is specified in the `mongodb.conf` file.

Configuring backup options for MongoDB using the mongodb.conf file

Note: If you use the MongoDB configuration tool, these manual steps are not required.

NetBackup uses the default options to back up MongoDB data. To specify custom options to use during a backup operation, you must create a `mongodb.conf` file in the `/usr/openv/var/global/` directory on a UNIX and `<Install_Dir>\NetBackup\var\global\` on a Windows master server.

Caution: The file name `mongodb.conf` is case-sensitive.

It is not mandatory to specify all the options in the `mongodb.conf` file. NetBackup uses the default values for the options that do not have custom values.

Ensure that the `mongodb.conf` file uses JSON format and whitelist the file path in `bp.conf` using the `bpcd_allowed_path` option. For more information, See "Including the configuration file path on NetBackup master server allowed list" on page 32.

Backup options in the mongodb.conf file

You can specify the following backup options and their values in the `mongodb.conf` file:

Caution: The options in the file are case-sensitive.

Options	Details
application_servers	Fully Qualified Domain Name (FQDN), or hostname, or short name and the port number of the primary configuration server and mongod and mongos port in the following format: clientFQDN_OR_hostname_OR_shortname:portnumber Ensure that application_server matches with the hostname value that is used in the MongoDB environment and verified using the db.hostInfo() command. For example, the hostname value that is displayed by db.hostInfo(): "hostname" : "<*hostname_value*>:<*port*>" **Warning:** Do not enter the node that acts an Arbiter node for MongoDB.
alternate_config_server	Fully Qualified Domain Name (FQDN), or hostname, or short name and the port number of the secondary or the alternate configuration server. You can add only one alternate configuration server for one cluster. Use the following format for the value: clientFQDN_OR_hostname_OR_shortname:portnumber Ensure that alternate_config_server matches with the hostname value that is used in the MongoDB environment and verified using the db.hostInfo() command. For example, the hostname value that is displayed by db.hostInfo(): "hostname" : "<*hostname_value*>:<*port*>" If a connection to the primary configuration server fails, the first, active alternate configuration server is used. For sharded MongoDB clusters, both the mongod and mongos processes must be running on the alternate config server. You must enter the value of the alternate_config_server separately for every application_servers entry.
cleanup_time_in_min	Specify the time in minutes to clean up the stale snapshots or oplogstore that are created because of canceled jobs. The value must be an integer.

Options	Details
`free_space_percentage_snapshot`	Specifies the percentage of the free space on a volume group that can be used to create a snapshot. This option is used only in case of full backups. The default value (if not specified) is 20%. The value must be between 0 and 100. Do not use the percentage symbol (%). For example, run the `vgdisplay` command to verify the value for the `"Free PE / Size"` field. The `free_space_percentage_snapshot` value is the percentage of `Free PE / Size` of the volume group where the data path resides. Adjust the `free_space_percentage_snapshot` value based on the data change rate of the MongoDB instance during the backup operation and the free space that is available on the volume group. For example, when: ■ Data change rate is 250 MB ■ Volume group has 1 GB `Free PE / Size` ■ The data change rate is 25% of Free PE / Size Then specify the minimum value of `free_space_percentage_snapshot` as 25%. Keeping the free space snapshot percentage value too low can result in snapshot (and subsequent backup) failure. Keeping the free space snapshot percentage value too high can reduce the amount of available space on the volume group. For more information and standard practices, refer to the Linux man page for the `lvcreate` command.
`data_channel_tls`	Use this parameter to disable or enable the data channel encryption between the MongoDB cluster and the backup host. For example, use `"data_channel_tls": false` to disable the data channel encryption. By default, all the traffic between the NetBackup backup host and the thin client (`mdbserver`) is over a TLS channel. You can disable this TLS for data movement from thin client (`mdbserver`) to the backup host to improve the performance. **Note:** Control data and sensitive data such as credentials are still transferred over the TLS channel when this option is disabled.

Options	Details
`logdir`	Location where the thin client (`mdbserver`) logs are generated on the MongoDB nodes. Default location is `/tmp`. If the directory path is mentioned, but the directory does not exist on the server, NetBackup creates a directory.
`loglevel`	Specify the logging level. Default value is 3. Refer to the following options for logging level values: ■ `ESERROR = 1` ■ `ESWARN = 2` ■ `ESINFO = 3` ■ `ESDEBUG = 4` ■ `ESTRACE = 5` ■ `ESCRITICAL = 6`
`max_log_mbsize`	Specify the maximum file size in MB for the NetBackup thin client log file. The default size is 10 MB. A new log file is created every day or if the existing log file exceeds the maximum size. The log file creation does not affect an ongoing job and the log roll-over happens during the next job that is performed by `mdbserver`. The logs are cleaned up after 30 days.

Options	Details

`max_streams`	**Note:** This parameter is applicable only for sharded MongoDB clusters.
	
	Defines the number of parallel data streams per backup host. The minimum value is 32.
	
	If `max_streams` is not defined, the default value is 32 parallel data streams per backup host.
	
	Add the following entry to the mongodb.conf file:
	
	`max_streams:`
	`{`
	`"global_default":<set_value>,`
	`"<backup_host>":<set_value>`
	`}`
	
	Where:
	
	■ `global_default`
	Default upper limit of the parallel data streams for all the backup hosts.
	■ *`backup_host`*
	Set the upper limit of the parallel data streams for a specified backup host.
	The *`backup_host`* must be the same that is specified in the backup policy. If you have multiple backup hosts, the entry can be repeated for all backup hosts. If you do not specify a backup host, the `global_default` value is used.
	
	Note: This option sets the upper limit on the number of parallel data streams per backup host. The backup or the recovery job might not use all the available streams.

`mdb_progress_loglevel`	Lets you print the progress logging information about the files that are restored to the activity monitor. Default value is 0 (off). To enable set `"mdb_progress_loglevel": 1`. **Note:** Enabling this option can increase the recovery time.

Options	Details
`mdbserver_location`	Specify a location to copy the thin client (`mdbserver`) binaries on the MongoDB nodes that are required for the MongoDB backup and restore operation. The files are copied to the servers that contain the data that needs to be protected and are removed once the backup operation completes. Default location to copy the files is `/tmp`. **Note:** Do not specify the mount path or the high-level Linux directories because that can cause conflicts in directory permissions. For example, avoid specifying the path as /root, /etc, /usr, /bin, /home, etc.
`mdbserver_port`	Port that is used by the backup host to connect with the NetBackup thin client (`mdbserver`) that is running on the MongoDB node. Default value is "`11000`". This value is a string.
`mdbserver_port_range`	Use this parameter when multiple `mongod` instances are running on a single MongoDB node. This option lets you use the next available port within the range for the backup and restore operation if the existing port is in use. This option lets you run multiple backup jobs simultaneously on different ports by deploying multiple NetBackup thin clients (`mdbserver`). Enter the value as "`mdbserver_port_range`":*`range_value`*, where the `range_value` is an integer to define the range of port numbers that can be used. For example, if you add the range_value as 10 and the mdbserver_port is defined 12000, the port range is used from 12000 to 12009. The default value is 10. Change this value based on the number of `mongod` instances on a MongoDB host that are backed up simultaneously.

Options	Details
`mdbserver_timeout_min`	Defines the time in minutes to wait before a NetBackup thin client (`mdbserver`) process is killed. The default value is 300 (minutes). Set the value higher than 300 minutes if your backup window requires more time. Ideally, `mdbserver` is killed after the plug-in terminates or the backup is complete.
`mongos_port`	Port that the `mongos` process uses for communication. This is a mandatory parameter for sharded MongoDB clusters. You must specify this value for each of the `application_servers` or `alternate_config_server` entry This value is a string.
`oplog_location`	For differential incremental backups, specify a custom directory to store the MongoDB `oplog` file. The location is stored in the backup image. Default location is `/tmp/oplogstore`. Ensure that there is enough free space at this location for the `oplog` data of the incremental backups.
`snapshot_mount_path`	Specify the path on the MongoDB nodes for mounting LVM snapshots during full backups. Default path is `/tmp`.

Note: Ensure that the `HostUser` that you have configured in the MongoDB credential file has the read and write permissions to all the paths mentioned in the `mongodb.conf` file.

If you do not add all the options, an entry is added in the logs about the missing options. The default values are used for the options that are not mentioned and the backup operation proceeds.

Sample of the mongodb.conf file contents

```
{
  "application_servers": {
    "FQDN_primary_configuration_server_1:port": {
      "alternate_config_server": [
```

```
        {
          "hostname:port": "FQDN_alternate_configuration_server_1:26051",
          "mongos_port": "26051"
        }
      ],
      "mongos_port": "26052"
    },
    "FQDN_primary_configuration_server_2:port": {
      "alternate_config_server": [
        {
          "hostname:port": "FQDN_alternate_configuration_server_2:26053",
          "mongos_port": "26053"
        }
      ],
      "mongos_port": "26054"
    }
  },
  "mdbserver_location": "/path/to/store/mdbserver/",
  "logdir": "/path/to/store/logdir/",
  "mdbserver_port": "21020",
  "loglevel": 5,
  "max_log_mbsize": 4,
  "oplog_location": "/path/to/store/oplog/",
  "free_space_percentage_snapshot": "25",
  "mdb_progress_loglevel": 1,
  "snapshot_mount_path": "/path/to/mount/snapshot/",
  "max_streams":
   {
     "global_default":2,
     "FQDN_backup_host_1":1
   }
}
```

Including the configuration file path on NetBackup master server allowed list

After you create the configuration file, you must include the path of the configuration file on the allowed list. So, that NetBackup lets the backup operation to run successfully. Run the allowed list procedure on a NetBackup master server.

Allowlisting is a security practice used for restricting systems from running software or applications unless these have been approved for safe execution.

To place the configuration file path on the allowed list:

Run the following command on the NetBackup master server:

1. For UNIX:

   ```
   bpsetconfig -h masterserver_name
   bpsetconfig bpcd_allowed_path = /usr/openv/var/global/
   ```

   Exit the command line

2. For Windows:

   ```
   bpsetconfig -h masterserver_name
   bpsetconfig bpcd_allowed_path = <install_dir>\NetBackup\var\global\
   ```

   Exit the command line

For more information about the `bpsetconfig` command, refer to the *NetBackup Commands Reference Guide*.

For more information about `bpcd_allowed_path`, see the *Configuration options for NetBackup servers section* in the *NetBackup Administrator's Guide, Volume I*.

Obtaining the RSA key of the MongoDB nodes

Use the following command on the RHEL or SUSE OS that hosts the MongoDB cluster to get the SHA256-based RSA key of every MongoDB node from the MongoDB cluster:

```
cat /etc/ssh/ssh_host_rsa_key.pub |awk '{print $2}' |base64 -d |sha256sum |awk '{print $1}'
```

The output of the commands is the RSA key.

For example:

```
cat /etc/ssh/ssh_host_rsa_key.pub |awk '{print $2}' |base64 -d |sha256sum |awk '{print $1}'
```

Command output:

```
b2352722053ac9f40bc1XXXXXXXXXXXXXXXXXXXXXXXXX419fa241ba9431fd6b9
```

Copy the RSA fingerprint. You need to provide this fingerprint when you add the MongoDB credentials.

Adding MongoDB credentials in NetBackup

Note: If you use the MongoDB configuration tool, these manual steps are not required.

To establish a seamless communication between MongoDB clusters and NetBackup for successful backup and restore operations, you must add and update MongoDB credentials to the NetBackup master server.

About the authentication types for MongoDB that NetBackup supports

NetBackup supports the following authentication types for protecting the MongoDB data:

- No authentication
- Simple - Salted Challenge Response Authentication Mechanism (SCRAM)
- Certificate-based - x.509

Different options are required for each of the authentication types when you add the credentials using the `tpconfig` command.

The following table describes the options that are required for each authentication type:

Table 3-1 Required options for authentication types

Options	Option description	No authentication	Simple authentication	Certificate-based authentication
`AppUserId`	Specifies the user name that is required to log into the application server.			
`AppUserPassword`	Specifies the user password that is required to log into the application server.			

Table 3-1 Required options for authentication types *(continued)*

Options	Option description	No authentication	Simple authentication	Certificate-based authentication
`HostUser`	Specify the host's user ID for SSH implementation. If the host user that you want to use is a non-root user or does not have root permissions for the MongoDB server then: See "Using a non-root user as a host user" on page 38.			
`HostPassword`	Specify the host's user password for SSH implementation.			
`HostRSAKey`	RSA key is required to perform password-less remote operations.			
`ServerPemPath`	Path to the PEM certificate file on the MongoDB node.			
`CAPemPath`	Path to the CA PEM certificate file on the MongoDB node.			
`Passkey`	Password of CA certificate.			
`CADir`	Path to the CA certificate.			
`CARole`	User role that is defined in the CA.			
`CertificateUser`	Specifies the details fo the certificate user.			
`application_server_conf`	Specifies a path to the credential configuration file that contains the authentication type, user details, and the directory paths for the CA security certificates. See "About the credential configuration file" on page 35.			

About the credential configuration file

The credential configuration file is required for authentication. You can create this file at any location and use the path when you add the MongoDB node credentials.

Add the credentials of the MongoDB node using the `tpconfig` command for the application server that is specified in the **Clients** tab in the backup policy.

Multiple MongoDB nodes must be separated using a comma.

- For sharded MongoDB cluster, add all the mongod and mongos ports in the following format:

  `mongod_hostname:mongod_port`

  `mongos_hostname:mongos_port`

- For replica set MongoDB cluster and standalone MongoDB cluster, add the mongod port in the following format:

  `mongod_hostname:mongod_port`

Refer to the following sample credential file that contains all the authentication types:

- No Authentication

```
{
"app.server.com:26050" : {
    "AuthType":"NOAUTH",
    "HostUser":"root",
    "HostPassword":"password”,
    "HostRsaKey":"b2352722053ac9f40bc1XXXXXXXXXXXXXXXXXXXXXXXXX419fa241ba9431fd6b9"
    }
}
```

- Simple Authentication

```
{
"app3.server.com:26051" : {
    "AuthType":"SIMPLEAUTH",
    "HostUser":"root",
    "HostPassword":"password",
    "HostRsaKey":"b2352722053ac9f40bc1XXXXXXXXXXXXXXXXXXXXXXXXX419fa241ba9431fd6b9",
    "AppUserId":"admin",
    "AppUserPassword":"password"
    }
}
```

- Certificate-based authentication

```
{
"app4.server.com:26052" : {
    "AuthType":"CERTAUTH",
    "HostUser":"root",
    "HostPassword":"password",
    "HostRsaKey":"b2352722053ac9f40bc1XXXXXXXXXXXXXXXXXXXXXXXXX419fa241ba9431fd6b9",
    "ServerPemPath":"/root/certs/user1.pem",
    "CAPemPath":"/root/certs/mongo-CA-cert.crt",
    "Passkey":"password",
    "CADir":"/root/",
    "CARole":"root",
    "CertificateUser":"CN=user1,OU=nbu,O=vtas,L=pune,ST=mh,C=in"
    }
}
```

How to add the MongoDB credentials in NetBackup

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

For more information about the `tpconfig` command, see the *NetBackup Commands Reference Guide*.

Before you run the `tpconfig` command, ensure to remove all the earlier entries for the MongoDB nodes.

To run the `tpconfig` command:

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

   On UNIX systems, `/<install_directory>/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 MongoDB credentials is displayed.

To add credentials for all of the authentication types:

1. Run the following command by providing appropriate values for each options to add MongoDB credentials.

   ```
   tpconfig -add -application_server
   app_server_name-mongod_port_number -application_type mongodb
   -requiredport mongod_port_number -application_server_conf
   /<install_directory>/var/global/mongodb_cred_file.conf
   ```

   Where:

   `application_server` is `mongodb_hostname-mongodport`

   `application_server_conf` is a credential file to add support for single or multiple `mongod` per MongoDB cluster

   You can use the `-update` or `-delete` options of the `tpconfig` command to update or delete the MongoDB credentials.

   For more information, See "About the credential configuration file" on page 35.

2. Run the `tpconfig -dappservers` command to verify if the NetBackup master server has the MongoDB credentials added.

Note: The encrypted credential configuration file (name:*appservername-portnumber.conf*) is created on the NetBackup master server at the following location: `/usr/openv/var/global/`.

About the MongoDB roles for protecting the data

For completing NetBackup operations, the application user that you add in NetBackup must have the required roles. The required role must have permissions to access, query, back up, and restore all the databases and manage the cluster. It is recommended that you assign the user a root role.

Note: Ensure that the same user is added for all the MongoDB nodes.

For more information about roles, refer to the MongoDB Manual.

Using a non-root user as a host user

If the host user that you want to use is a non-root user or does not have root permissions for the MongoDB server, then you must complete the following steps:

- Ensure that the host user credentials that are configured using the `tpconfig` command are of the host user account that is used to configure the MongoDB cluster (MongoDB daemon's host user account).
- Give that host user the ownership rights to all the directories that you have mentioned in the `mongodb.conf` file. This step ensures that the backup host can access the directories to copy the required files for backup operations. For example, the user needs rights to `mdbserver_location`, `logdir`, `oplog_location`.
- Ensure that the host user has ownership of the MongoDB database path and its contents. This ownership is required for the backup as well as restore operations.
- On the server where MongoDB is installed, add the host user in the `sudoers` file.
- NetBackup does not support service ids like `mongod` as a host user id.

Managing backup hosts

A backup host acts as a proxy client which hosts all the backup and restore operations for MongoDB clusters. In case of MongoDB plug-in for NetBackup, backup host performs all the backup and restore operations without any separate agent installed on the MongoDB cluster.

The backup host must have a Linux operating system. NetBackup supports only RHEL and SUSE platforms as a backup host.

The backup host can be a NetBackup client or a media server or a master server. Veritas recommends that you have media server as a backup host.

Consider the following before adding a backup host:

- For backup and restore operations, you can add one or more backup hosts.
- A master, media, or client can perform the role of a backup host.
- MongoDB plug-in for NetBackup is installed on all the backup hosts.
- When using multiple backup host, make sure that all backup hosts are communicating with the media server.

You can add a backup host while configuring **BigData** policy using either the NetBackup Administration Console or Command Line Interface.

For more information on how to create a policy, see See "Creating a BigData backup policy" on page 46.

To add a backup host

In the **Backup Selections** tab, click **New** and add the backup host in the following format:

Backup_Host=<FQDN_or_hostname>

For more information on how to create a policy, See "Creating a BigData backup policy" on page 46.

Alternatively, you can also add a backup host using the following command:

For Windows:

```
bpplinclude PolicyName -add "Backup_Host=FQDN_or_hostname"
```

For UNIX:

```
bpplinclude PolicyName -add "Backup_Host=FQDN_or_hostname"
```

For more information about the `bpplinclude` command, refer to the *NetBackup Commands Reference Guide*.

For more information, See "Using NetBackup Command Line Interface (CLI) to create a BigData policy for MongoDB clusters " on page 48.

To remove a backup host

1. In the **Backup Selections** tab, select the backup host that you want to remove.
2. Right click the selected backup host and click **Delete**.

   Alternatively, you can also remove a backup host using the following command:

   For Windows:

   ```
   bpplinclude PolicyName -delete "Backup_Host=FQDN_or_hostname"
   ```

   For UNIX:

   ```
   bpplinclude PolicyName -delete "Backup_Host=FQDN_or_hostname"
   ```

   For more information about the `bpplinclude` command, refer to the *NetBackup Commands Reference Guide*.

Including a NetBackup client on NetBackup master server allowed list

To use the NetBackup client as a backup host, you must include it in the allowed list. Perform the allow list procedure on the NetBackup master server .

Allowlisting is a security practice used for restricting systems from running software or applications unless these have been approved for safe execution.

To place a NetBackup client on NetBackup master server on the allowed list

Run the following command on the NetBackup master server:

- For UNIX

```
bpsetconfig -h masterserver
bpsetconfig> APP_PROXY_SERVER = clientname1.domain.org
bpsetconfig> APP_PROXY_SERVER = clientname2.domain.org
bpsetconfig>
UNIX systems: <ctl+D>
```

- For Windows

```
bpsetconfig -h masterserver
bpsetconfig> APP_PROXY_SERVER = clientname1.domain.org
bpsetconfig> APP_PROXY_SERVER = clientname2.domain.org
bpsetconfig>
Windows systems: <ctl+Z>
```

For more information about the `bpsetconfig` command, refer to the *NetBackup Commands Reference Guide*.

This command sets the *APP_PROXY_SERVER = clientname* entry in the backup configuration (`bp.conf`) file.

For more information about the *APP_PROXY_SERVER = clientname*, refer to the *Configuration options for NetBackup clients* section in *NetBackup Administrator's Guide, Volume I*

Chapter 4

Backing up MongoDB using NetBackup

This chapter includes the following topics:

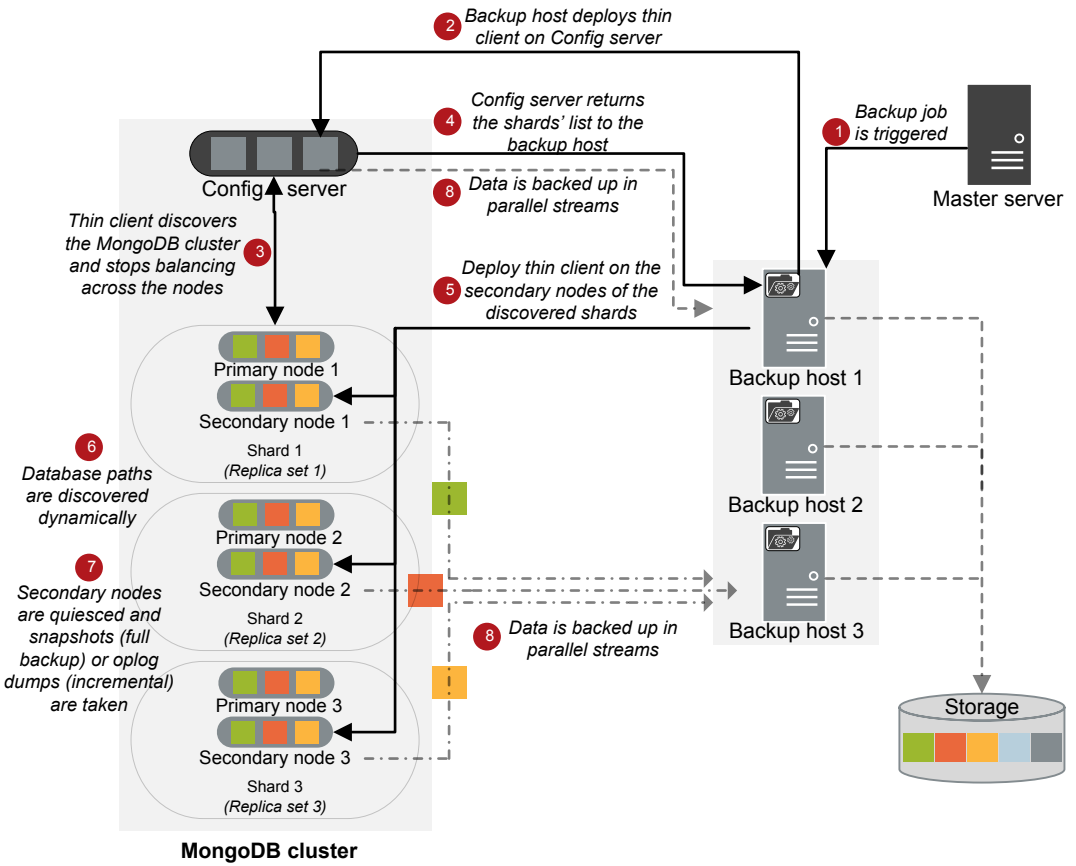
- Backing up MongoDB data
- Prerequisites for backing up a MongoDB cluster
- Configuring NetBackup policies for MongoDB plug-in

Backing up MongoDB data

MongoDB data is backed up in parallel streams wherein MongoDB data nodes stream data blocks simultaneously to multiple backup hosts.

The following diagram provides an overview of the backup flow:

Figure 4-1 Backup flow



As illustrated in the above diagram:

1. A scheduled backup job is triggered from the master server.
2. Backup job for MongoDB data is a compound job. When the backup job is triggered, first a discovery job runs.
3. During discovery, the backup host deploys a transient thin client (`mdbserver`) on the configuration server and obtains the details of the shards in the MongoDB cluster. The thin client also stops the balancing across the nodes in a replica set.
4. After receiving the information about the cluster, the backup host deploys a thin client on the secondary node of a replica set in the MongoDB cluster.

5. The thin client discovers the database paths dynamically, quiesces the secondary nodes, and takes snapshots for full backups and captures `oplog` for incremental backups.
6. Individual child jobs run for each backup stream and data is backed up.
7. Data blocks are streamed simultaneously from different secondary nodes to multiple backup hosts.

Once the backup operation is completed, the thin client is removed from the servers.

The compound backup job is not completed until all the child jobs are completed. After the child jobs are completed, NetBackup cleans all the snapshots from the secondary nodes. Only after the cleanup activity is completed, the compound backup job is completed.

Backing up a MongoDB cluster

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

For overview of the backup process, See “Backing up MongoDB data” on page 42.

Prerequisites for backing up a MongoDB cluster

- NetBackup chooses the node in a MongoDB cluster to take a backup in the following order:
  - Active hidden node
  - Active secondary node
  - Active primary node

  If you want NetBackup to select a particular backup node in the MongoDB cluster, set it as a hidden node.
- Before you run a backup job, ensure that you get a successful ping response from all the MongoDB nodes on the backup host. Check and update the firewall settings so that the backup hosts can communicate with the MongoDB cluster.
- Ensure that the MongoDB cluster that you want to protect lets you take LVM snapshots.
- Logical volume requirement for snapshot:
  - Ensure that the MongoDB database directory is mounted on a logical volume to complete the snapshot operations.
  - Use the `vgdisplay` command to ensure that the free physical extent size is sufficient in the logical volume group to complete the snapshot operations.

- Renaming the volume group or the physical and logical volumes of the LVM for MongoDB database paths causes the backup to fail. If you rename the volume group or the physical and logical volumes of the LVM, ensure that the MongoDB database is mounted on the new path before taking a backup.
- The backup shuts down the balancer on the `mongos` process and blocks all the other operations. Hence, during a backup process, ensure that you do not run any other operation that uses the `mongos` process. For example, import the database.
- Always run a full backup when you make change to the database path, or modify the configuration file for the `mongod` or the `mongos` processes, or change the MongoDB topology.
- If there are multiple MongoDB clients in a single Netbackup backup policy increase the **Client read timeout** parameter for master server, media server, and the client to ensure that the all the backups are successful.
  For more information, refer to the NetBackup™ Administrator's Guide, Volume I and the Timeouts properties section.
- Incremental backup jobs use consistent backup images as a reference for determining the incremental changes. If the previous backup has failed, or was partially successful (failed for one of the nodes), it is skipped completely and a backup image prior to that is considered. In such cases, the backup operation can take longer and the image that is created might be of a larger size.
- The `oplog` file has a capped or rolling cache and you can configure the size of the file. NetBackup uses `oplog` to capture incremental data. `Oplog` roll-over can cause the incremental backups to fail. To prevent this, make sure that `oplog` file size is sufficient to hold the incremental data that is generated between the incremental backups.
- Ensure that the user that you have added using the `tpconfig` command has access to the entire MongoDB cluster and the custom folder paths that are specified in the `mongodb.conf` file.
- For MongoDB 4.2 sharded clusters, ensure that the Feature Compatibility Version (FCV) is same across all the MongoDB cluster nodes. This step ensures data consistency during backups.
- If you are using *oplog retention feature* on replica set, ensure that the scheduled time between incremental backups is less than the minimum oplog retention period. This ensures capturing the correct incremental backup.
- NetBackup supports incremental backups for MongoDB 4.2 sharded clusters only if the Feature Compatibility Version (FCV) is set to 4.0 or earlier. If your MongoDB 4.2 sharded cluster has FCV 4.2, then only full backups are supported.

- NetBackup supports full backup only for mongodb 4.4 version on sharded cluster.

Configuring NetBackup policies for MongoDB plug-in

Backup policies provide the instructions that NetBackup follows to back up clients. For configuring backup policies for MongoDB plug-in, use the **BigData** policy as the **Policy Type**.

You can create **BigData** policy using either the **NetBackup Administration Console** or the **Command Line Interface**.

For more information on how to create a BigData policy, See "Creating a BigData backup policy" on page 46.

Creating a BigData backup policy

Use the BigData policy to backup big data applications such as MongoDB clusters.

A BigData policy differs from other policies in the following respects:

- You must specify **BigData** as the policy type.
- The entries which are provided in the **Clients** tab and the **Backup Selections** differ based on the application that you want to back up.
- In the **Backup Selections** tab, you must specify certain parameters and their appropriate values.

Creating BigData policy using the NetBackup Administration Console

If you prefer using the **NetBackup Administration Console** for creating BigData policy, you can use either of the following methods:

- Creating a BigData policy using the **Policy Configuration Wizard**
- Creating a BigData policy using the NetBackup **Policies** utility

The easiest method to set up a **BigData** policy is to use the **Policy Configuration Wizard**. This wizard guides you through the setup process by automatically choosing the best values for most configurations. Not all policy configuration options are presented through the wizard. For example, **calendar-based** scheduling and the **Data Classification** setting. After the policy is created, modify the policy in the Policies utility to configure the options that are not part of the wizard.

Using the Policy Configuration Wizard to create a BigData policy for MongoDB clusters

Use the following procedure to create a BigData policy with the Policy Configuration Wizard.

To create a BigData policy with the Policy Configuration Wizard

1. In the **NetBackup Administration Console**, in the left pane, click **NetBackup Management**.
2. In the right pane, click **Create a Policy** to begin the **Policy Configuration Wizard**.
3. Select the type of policy to create:
   - **BigData** policy : A policy to backup **MongoDB** data
4. Select the storage unit type for BigData policy.
5. Click **Next** to start the wizard and follow the prompts.

   Click **Help** on any wizard panel for assistance while running the wizard.

Using the NetBackup Policies utility to create a BigData policy for MongoDB clusters

Use the following procedure to create a BigData policy with the NetBackup Policies utility.

To create a BigData policy with the NetBackup Policies utility

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 new schedule.

   You can create a schedule for a **Full Backup** or **Differential Incremental Backup** for your BigData policy. Once you set the schedule, MongoDB data is backed up automatically as per the set schedule without any further user intervention.
7. On the **Clients** tab, based on your MongoDB setup, enter the following value:

- Sharded MongoDB cluster
  The client name as seen in the MongoDB shell and the `mongod` port number of the primary configuration server in the following format:
  `MongoDBNode-portnumber`
- Replica set MongoDB cluster
  The client name as seen in the MongoDB shell and the `mongod` port number of the primary node of the replica set in the following format:
  `MongoDBNode-portnumber`
- Standalone MongoDB setup
  The client name as seen in the MongoDB shell and the `mongod` port number of the standalone node in the following format:
  `MongoDBNode-portnumber`

Warning: Do not enter the node that acts an Arbiter node for MongoDB.

8 On the **Backup Selections** tab, enter the following parameters and their values as shown:

- *Application_Type=mongodb*
  The parameter values are case-sensitive.
- *Backup_Host=FQDN or hostname*
  The backup host must have a Linux operating system. The backup host can be a NetBackup client or a media server.
  You can specify multiple backup hosts.
- Manually add the `ALL_DATABASES` directive.

9 Click **OK** to save the changes.

For more information on using NetBackup for big data applications, refer to the Veritas NetBackup documentation section on the Veritas support page.

Using NetBackup Command Line Interface (CLI) to create a BigData policy for MongoDB clusters

You can also use the CLI method to create a BigData policy for MongoDB.

For more information about the commands, refer to the *NetBackup Commands Reference Guide*.

To create a BigData policy using NetBackup CLI method

1. Log on as an Administrator.
2. Navigate to `/usr/openv/netbackup/bin/admincmd` on UNIX or *`install_path`*`\NetBackup\bin\admincmd\` on Windows.
3. Create a new BigData policy using the default settings.

   ```
   bppolicynew policyname
   ```

4. View the details about the new policy using the `-L` option.

   ```
   bpplinfo policyname -L
   ```

5. Modify and update the policy type as **BigData**.

   ```
   bpplinfo PolicyName -modify -v -M MasterServerName -pt BigData
   ```

6. Specify the *Application_Type* as MongoDB.

   ```
   bpplinclude PolicyName -add "Application_Type=mongodb"
   ```

   Note: The parameter values for *Application_Type=mongodb* are case-sensitive.

7. Specify the backup host on which you want the backup operations to be performed for MongoDB.

   ```
   bpplinclude PolicyName -add "Backup_Host=IP_address or hostname"
   ```

   Note: The backup host must have a Linux operating system. The backup host can be a NetBackup client or a media server or a master server.

8. Specify the MongoDB directory or folder name that you want to backup.

   ```
   bpplinclude PolicyName -add "ALL_DATABASES"
   ```

9. Modify and update the policy storage type for BigData policy.

   ```
   bpplinfo PolicyName -residence STUName -modify
   ```

10. Specify the host name based on your MongoDB setup as follows:

    - Sharded MongoDB cluster
      The client name as seen in the MongoDB shell and the `mongod` port number of the primary configuration server in the following format:
      `MongoDBNode-portnumber`
    - Replica set MongoDB cluster

The client name as seen in the MongoDB shell and the `mongod` port number of the primary node of the replica set in the following format:

`MongoDBNode-portnumber`

- Standalone MongoDB setup
  The client name as seen in the MongoDB shell and the `mongod` port number of the standalone node in the following format:

  `MongoDBNode-portnumber`

```
bpplclients PolicyName -M "MasterServerName" -add
"MongoDB_configserver" "Linux" "RedHat"
```

Warning: Do not enter the node that acts an Arbiter node for MongoDB.

11 Assign a schedule for the created BigData policy as per your requirements.

```
bpplsched PolicyName -add Schedule_Name -cal 0 -rl 0 -st
sched_type -window 0 0
```

Here, *`sched_type`* value can be specified as follows:

Schedule Type	Description
FULL	Full backup
INCR	Differential Incremental backup

The default value for *`sched_type`* is **FULL**.

Once you set the schedule, MongoDB data is backed up automatically as per the set schedule without any further user intervention.

12 Alternatively, you can also perform a manual backup for MongoDB data.

For performing a manual backup operation, execute all the steps from Step 1 to Step 11.

13 For a manual backup operation, navigate to `/usr/openv/netbackup/bin`

Initiate a manual backup operation for an existing BigData policy using the following command:

```
bpbackup -i -p PolicyName -s Schedule_Name -S MasterServerName
-t 44
```

Here, `-p` refers to policy, `-s` refers to schedule, `-S` refers to master server, and `-t 44` refers to BigData policy type.

Chapter 5

Restoring or recovering MongoDB data using NetBackup

This chapter includes the following topics:

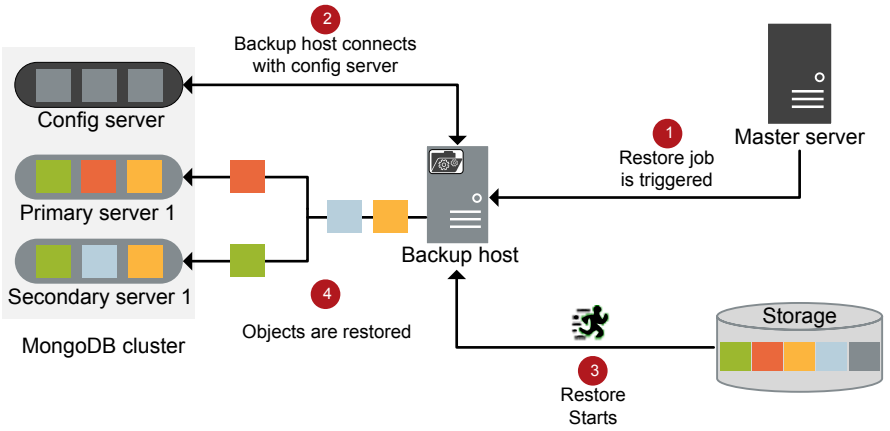
- Restoring MongoDB data
- Prerequisites for MongoDB restore and recovery
- About the restore scenarios for MongoDB database from the BAR interface
- Using the BAR interface to restore the MongoDB data on the same cluster
- Using the BAR interface to restore the MongoDB data on an alternate cluster
- About restoring MongoDB data in a high availability setup on an alternate client
- Recovering a MongoDB database using the command line
- Manual steps after the recovery process

Restoring MongoDB data

For restore you require a backup host. Backup hosts can be either master server, media server, or a NetBackup client.

The following diagram provides an overview of the restore flow.

Figure 5-1 Restore flow



As illustrated in the diagram:

1. The restore job is triggered from the master server.
2. The backup host connects with the config server. Backup host is also the destination client.
3. The actual data restore from the storage media starts.
4. The data blocks are restored on the MongoDB cluster.

Prerequisites for MongoDB restore and recovery

- Ensure that your source MongoDB cluster (during backup) and the target MongoDB cluster (during recovery) have the same:
  - MongoDB version
  - Authentication type
- Ensure that the user that you configure using the `tpconfig` command for restoring the MongoDB data has read, write, and execute permissions for all the files and subfolders on the target MongoDB directory. NetBackup uses this user account to recover and run the MongoDB instance.
- Before you restore the MongoDB data to an alternate cluster, ensure that you add the alternate MongoDB cluster credentials in the source cluster credentials file.

See "Adding MongoDB credentials in NetBackup" on page 34.

- Ensure that the PEM files or security certificates are available on the destination cluster before you start the recovery operation.
- The authentication type of the target cluster during the recovery process must be the same as the authentication type during the backup.
- During the recovery process, ensure that the target MongoDB cluster has sufficient free storage space for restoring the data.
- During recovery, ensure that you select only one full backup image group and its relevant subsequent incremental images. If you select more than one full backup image group, the recovery may fail as the restored data can get corrupted.
- NetBackup for MongoDB plug-in does not support cross-platform file system restore. For example, XFS to ext4 or vice versa is not supported.
- During the restore or recovery, ensure that the `HostUser` value that is defined in the `tpconfig` command is the same as the host user account that is used to configure the MongoDB cluster (MongoDB daemon's host user account).
- Ensure to select a backup host in destination client in the BAR UI before you submit the restore job.
- Point-in-time recovery is valid only for recovery from incremental backups.
- Canceling a parent job in a compound restore job does not cancel the child restore jobs. You must manually cancel the child restore jobs as well.
- After running a restore or a recovery job from the BAR interface, look for the job record and status in the **Task Progress** tab. The job might take some time to appear in the list and the compound job can take time to trigger the parent pre-recovery check. Click **Update Task List** to refresh the task list.
- In a **Restore Only** operation for a sharded cluster, follow the standard **Restore Only** steps:
  Shut down all the MongoDB processes (`mongos` or `mongod`) before starting the restore.
- The MongoDB log file paths remain the same from the original configuration. If you do an alternate restore:
  - Make sure that the same path is available during restore
  - Change the log file path in the configuration file for `mongod` or `mongos` process after a successful recovery.
- The path to the `.pid` files must be available on the destination MongoDB cluster to ensure that the recovery operation is successful.

- In case of a scenario where multiple MongoDB clusters are running on the same server and are backed up using the same or different backup policies, ensure that you select the correct application server that you want to restore.
  For example, if there are multiple clusters with the following configuration:

  ```
  Replica1
  Primary:   host1:26050
  Secondary: host1:26060

  Replica2
  Primary:   host1:26055
  Secondary: host1:26066
  ```

  And you want to recover `Replica1`, ensure that you specify correct application server and its port (`host1-26050`) as the source client in the BAR UI.

- Before you start a recovery operation, ensure that stale `mdbserver` (thin client) processes are not running on your MongoDB instance at the following path:
  /<mdbserver_location>/<Host>-<MongodPort>-<mdbserver_port in range>/mdbserver
  If stale `mdbserver` processes are running on the MongoDB host for the same MongoDB instance that you want to recover, the recovery operation is unable to shutdown the MongoDB instance. This issue causes the recovery job to stop responding.

- Restore and recovery of the MongoDB cluster requires the same security mode that is used at the time of the backup. Ensure that security mode is the same for the original cluster and the target cluster.
  For example, if SSL is used during the backup, then recovery is done using SSL and the target configuration is changed to SSL. Similarly, if TLS is used during the backup, then recovery is done using TLS and the target configuration is changed to TLS.

- Restore and recovery of the MongoDB cluster requires the same value of the Feature Compatibility Version (FCV) that is used at the time of the backup. Ensure that FCV is the same for the original cluster and the target cluster.
  For example, if FCV is 4.2 during the backup, then restore uses FCV 4.2 and the target cluster has FCV 4.2 after the recovery process completes. Similarly, if FCV is 4.0 during the backup, then restore uses FCV 4.0 and the target cluster has FCV 4.0 after the recovery process completes.

About the restore scenarios for MongoDB database from the BAR interface

For restoring MongoDB cluster from the **Backup, Archive, and Restore** interface, the following scenarios are possible:

- **Restore Only** or **Restore and Recover** a MongoDB cluster to the same location or a different location.
- Restore all or specific nodes of a MongoDB cluster to the same or a different location, and then run the manual steps to recover the MongoDB cluster.

Restore destination

The restore destination options are available in the **General** tab after you enter the details in the **Backup, Archive, and Restore** interface and proceed to the **Restore Marked Files** dialog box.

Table 5-1 Destination options for restore

Option from the General tab in the Restore Marked Files dialog box	Scenario
Restore everything to its original location.	Restore the MongoDB database to the same location or cluster.
Restore individual directories and files to different locations.	Restore the MongoDB database to an different location or cluster.

Restore and recovery options for MongoDB

Restore and recovery options are available in the **MongoDB** tab in the **Restore Marked Files** dialog box.

Figure 5-2 Restore and recovery options

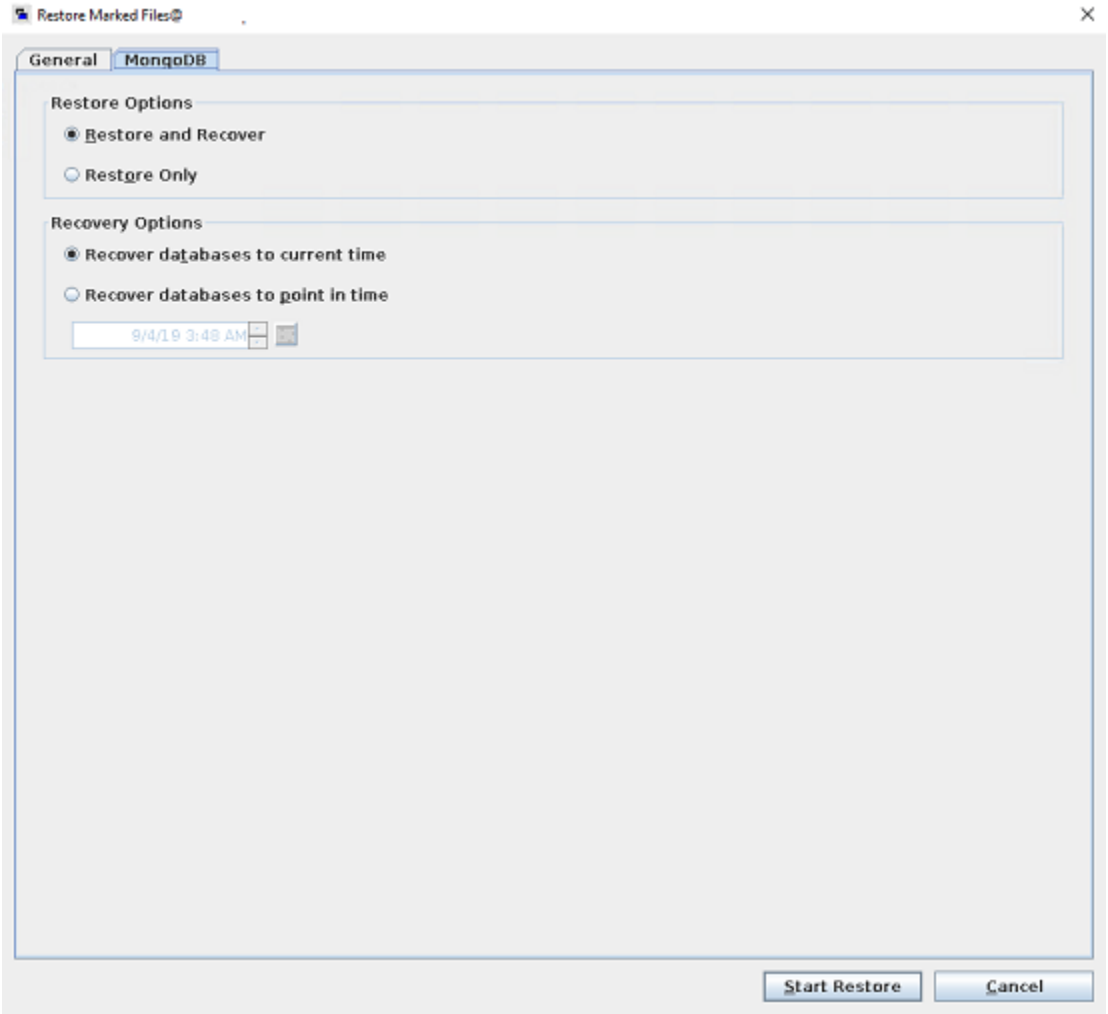


Table 5-2 Restore options for MongoDB

Option from the MongoDB tab in the Restore Marked Files dialog box	Scenarios to select the restore and recovery options
Restore and Recover	Restores and recovers the entire MongoDB cluster with the following recovery options: ■ **Recover databases to current time** Restores and recovers to the latest available backup time. ■ **Recover databases to point in time** Restores and recovers to a specified point-in-time in the selected backup set.
Restore Only	**Warning:** Use this option with caution since this may lead to inconsistent state of the cluster. ■ Restore the specific MongoDB cluster nodes and MongoDB oplogs that are specified in the selection. ■ You can use this option to restore the MongoDB config server without restoring the entire MongoDB cluster. ■ Recover the cluster manually using the command line options in the MongoDB application. Before you use this option, you must shutdown the MongoDB cluster. Stop the `mongos` process and then the `mongod` processes on all the nodes of the configuration server and shards. MongoDB recommends to use the `db.shutdownServer()` command. **Note:** The MongoDB services are not validated on the target cluster nodes.

The **Recovery Options** section is applicable only for the **Restore and Recover** option.

High-level steps involved in the Restore and Recovery process

- Pre-restore
  - The current topology and configuration of the MongoDB cluster is gathered.
  - The following validations are performed:
    - The restore and recovery of a replica set or a standalone MongoDB cluster is targeted to a single node only.
    - Restore and recovery is not targeted to an Aribter node.

- Restore of multiple shard's data is not targeted to a single shard.
- Whether the **Overwrite existing files** option is enabled.

- The target cluster is shutdown and the WiredTigerLogs files that are present in the journal folder at the database path are removed.

- Restore
  The data moves in parallel streams.
- Post-restore or recovery
  - The target cluster's configuration parameters along with the configuration parameters of the source cluster that was backed up are used to start the MongoDB services.
  - The `mongod` services are started on each node on a local host in maintenance mode without enabling the authentication.
    The minimum required operations are run in the maintenance mode on the local host, the service is shutdown and restarted with authentication.
  - Sharding, replication, and authentication is enabled and the MongoDB cluster is initiated.
  - The `oplogs` are replayed on the MongoDB cluster nodes.

Note: The restored node is now the primary node in the replica set.

Only the nodes that are backed up and selected for recovery are recovered. The other cluster members must be added manually to the recovered cluster.

You must select the **Overwrite existing files**.

Using the BAR interface to restore the MongoDB data on the same cluster

This topic describes how to use the NetBackup Admin console's BAR interface to restore MongoDB data on the same cluster.

To use the NetBackup Admin console's BAR interface to perform a restore

1. Open the **Backup, Archive, and Restore** interface.
2. From the File menu (Windows) or Actions menu (UNIX), choose **Specify NetBackup Machines and Policy Type**.
3. On the **Specify NetBackup Machines and Policy Type** wizard, enter the source and destination details for restore.

- Specify the MongoDB application server as the source for which you want to perform the restore operation.
  From the **Source client for restores** list, select the required Application server.
- Specify the backup host as the destination client.
  From the **Destination client for restores** list, select the required backup host. Restore is faster if the backup host is the media server that had backed up the node.
- On the **Specify NetBackup Machines and Policy Type** wizard, enter the policy type details for restore.
  From the **Policy type for restores** list, choose **BigData** as the policy type for restore.
  Click **Ok**.

4. Select the appropriate date range to restore the complete data set.
5. Go to the **Backup History** and select the backup images that you want to restore.
6. In the **Browse** directory, specify the root directory (“/”) as the path to browse.

   In the **Directory Structure** pane, expand the **Directory**.

   All the subsequent files and folders under the directory are displayed in the **Contents of Selected Directory** pane.
7. In the **Contents of Selected Directory** pane, select the check box for the MongoDB MongoDB nodes that you want to restore.
8. In the **Restore Marked Files** dialog box, select the destination for restore as per your requirement.
   - Select **Restore everything to its original location** to restore your files to the same location where you performed your backup.
9. Click **Start Restore**.
10. Verify that the database nodes gets restored and instantiated.

Using the BAR interface to restore the MongoDB data on an alternate cluster

NetBackup supports the following alternate recovery scenarios for MongoDB:

- Redirected restore and recovery to an alternate cluster

- Redirected restore and recovery to an alternate node or port or database path in an existing cluster

Complete the following steps to run alternate recovery for MongoDB:

1. Run the `tpconfig` command to update the original cluster's credentials with the alternate application server's credentials.

   For example, to recover source client Host1-26050 to an alternate application server Host2 that is running on port 28001:

   - Add the credentials of Host2:28001 and its related nodes in the original cluster's credential configuration file. For more information, See "About the credential configuration file" on page 35.
   - Run the update `tpconfig` command for `application_server` that is getting recovered (Host1-26050)

   Here is a sample command:

   ```
   /usr/openv/volmgr/bin/tpconfig -update -application_server
   Host1-26050 -application_type mongodb -requiredport 26050
   -application_server_conf /usr/openv/var/global/credential.conf
   ```

2. Rename the application server and its nodes and set the value for the alternate application server.

   - In the BAR UI select the **General** tab > **Restore individual directories and files to different location** and use **Change Selected Destination(s)** to add the alternate application server and port.
   - To change the folder path, select **Add Destination** and add the new destination path.

   See the section called "Alternate restore from a nested database path" on page 61.

3. Click **Start Restore** to start the recovery operation. You can check the status from the **Activity Monitor**.

Restoring the MongoDB oplog file to an alternate temporary location

You can restore the MongoDB `oplog` files from an incremental backup to an alternate path. The files and their path are seen in the BAR UI.

You must specify the paths during the alternate restore using the **Restore individual directories and files to different locations** option.

If you want to retain the original MongoDB path but change the `oplog` file path, in the **Add Destination** dialog box, specific the source and alternate paths.

For example, Source `/host:port/tmp` and Destination `/host:port/alternate_tmp`.

Alternate restore from a nested database path

For an alternate restore from a nested database path, use the Add Destination dialog box and for every subfolder, add an appropriate target alternate path.

For example, to change the path from `/host:port/usr/mongodb/db1` to `/host:port/alt-dir/dbpath/mydb`:

- Specify the source and the destination path:
  Source `/host:port/usr/mongodb/db1` and Destination `/host:port/alt-dir/dbpath/mydb`
- Specify the source and the destination path for the parent folder:
  Source `/host:port/usr/mongodb` and Destination `/host:port/alt-dir/dbpath`
- Specify the source and the destination path for the base parent folder:
  Source `/host:port/usr` and Destination `/host:port/alt-dir`

Note: When you do an alternate restore to a non-root path, the restore is partially successful if the database path contains multiple subfolders.

In such a scenario, when you do an alternate restore to a different location, you must add an entry for each directory level.

For example:

Source:`/hostname1:port1/Config_Data`

Destination: `/hostname2:port3/mongo_inst2`

Source:`/hostname1:port1/Config_Data/data`

Destination:`/hostname2:port3/mongo_inst2/data`

Source:`/hostname2:port2/Shard1_Primary`

Destination:`/hostname2:port3/mongo_inst2`

Source:`/hostname2:port2/Shard1_Primary/data`

Destination:`/hostname2:port3/mongo_inst2/data`

Restoring a MongoDB cluster where the backups are taken from different MongoDB nodes in the same replica set

You can restore a MongoDB cluster (sharded or replica set) that was backed up from different nodes because of the role switch (between primary and secondary nodes) within a shard or replica set. In such a scenario, the full backup can be taken

from one host and incremental backup is taken from another host in the same shard or replica set.

During restore, you must redirect the restore of these backup images to the same MongoDB host.

For example, to restore backups from `/host1:port1/dbpath` and `/host2:port1/tmp`, specify the following:

Source `/host1:port1/dbpath` and Destination `/althost:port1/dbpath`

Source `/host2:port1/tmp` and Destination `/althost:port1/tmp`

About restoring MongoDB data in a high availability setup on an alternate client

Use this information when the client (`MongoDBnode-port`) that is defined in the **Clients** tab of the backup policy is not available and you want to restore to a different client in the same MongoDB cluster (`MongoDBnode-port`).

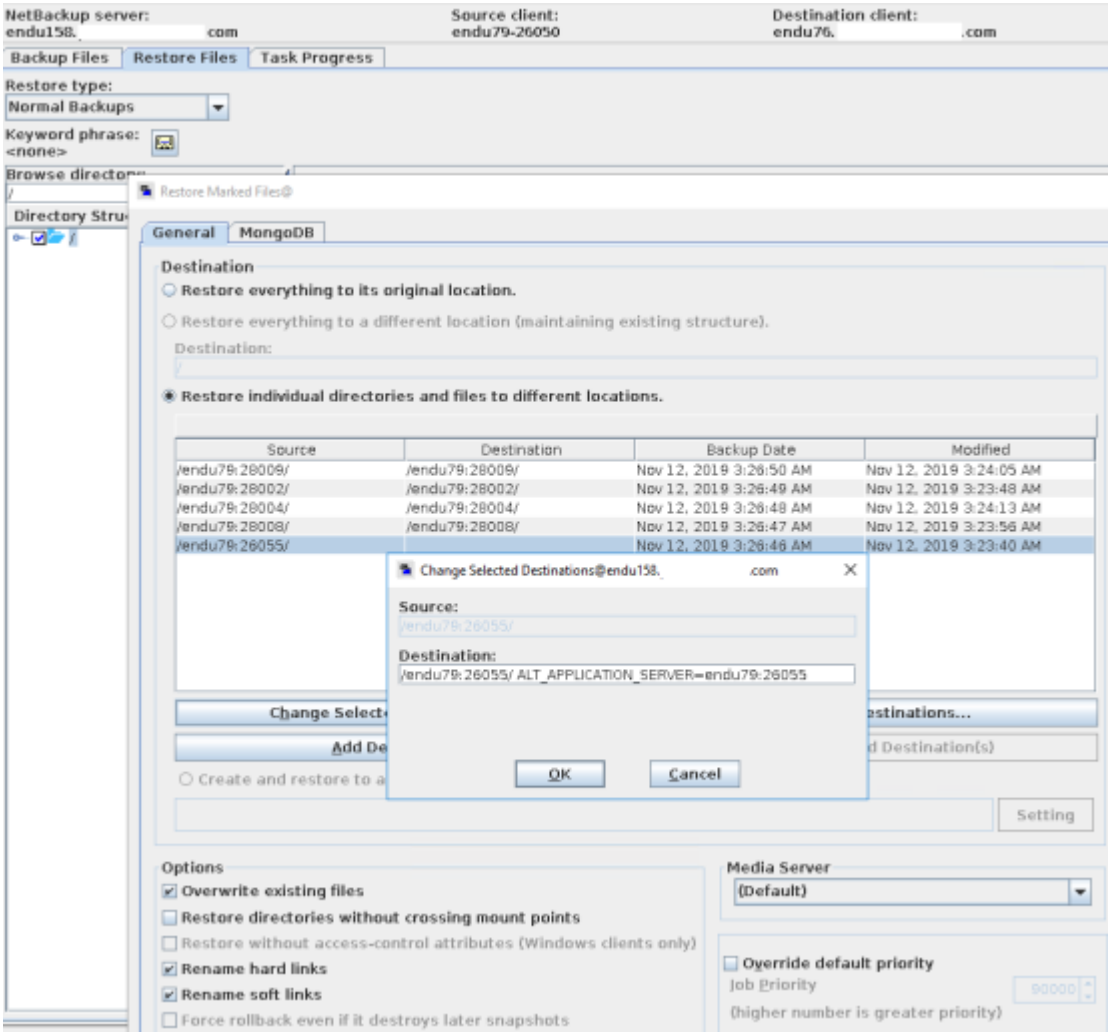
In a high availability setup, you can restore the MongoDB data as follows:

- Sharded MongoDB cluster
  Restore to an alternate config server in the same MongoDB cluster
- Replica set MongoDB cluster
  Restore to an alternate node of the replica set in the same MongoDB cluster

Complete the following steps:

- Rename the application server and its nodes and set the value for the alternate application server.
  In the BAR UI select the **General** tab > **Restore individual directories and files to different location** and use **Change Selected Destination(s)** or **Add Destination** to add the alternate application servers.
- If the `application_server` (`Host1-port1`) is different from the target `application_server` (`Host2:Port2`), the rename entry must contain `ALT_APPLICATION_SERVER=Host2:Port2`.

In the following example, the **Source Client** that is defined in the backup policy is `endu79-26050`, and the backup was done from the MongoDB node `endu79-26055`. In this scenario, as part of restore and recovery, append `endu79:26055` as follows:`ALT_APPLICATION_SERVER=endu79:26055`.



Recovering a MongoDB database using the command line

Use the `bprestore` command to recover a MongoDB database.

For more information about the `bprestore` command, refer to the *NetBackup Commands Guide*.

Use these steps from the command line interface to recover a MongoDB database:

1. Create or modify the rename file if you want to recover the MongoDB database to an alternate location
2. Use the command line to recover the MongoDB database.

   See "Using the command line to recover a MongoDB database" on page 65.

Creating or modifying the rename file

Create or modify the rename file in the any directory for the following scenarios:

- Redirect restore and recover the MongoDB database to an alternate cluster
- Redirect restore and recover the MongoDB database to an alternate node, or port, or database path in an existing cluster

If the rename file is not available, then you must create it and save it as `rename.txt` on the NetBackup master server.

The MongoDB rename file contains the following fields:

```
BIGDATA_MONGODB RestoreOnly
BIGDATA_MONGODB RecoverOnly
BIGDATA_MONGODB PointInTime
change /MongoDBnode_hostname1:port1/db1 to /MongoDBnode_hostname2:port2/db2
```

Sample rename file

For restore only:

```
BIGDATA_MONGODB RestoreOnly YES
BIGDATA_MONGODB RecoverOnly NO
BIGDATA_MONGODB RecoverPointInTime 0
```

For recovery in a specific point in time:

```
BIGDATA_MONGODB RestoreOnly NO
BIGDATA_MONGODB RecoverOnly NO
BIGDATA_MONGODB RecoverPointInTime 1290571200
```

For recovery and changing MongoDB node hostname, port, or database path:

```
BIGDATA_MONGODB RestoreOnly NO
BIGDATA_MONGODB RecoverOnly NO
BIGDATA_MONGODB RecoverPointInTime 0
change /MongoDBnode_hostname1:port1/db1 to /MongoDBnode_hostname2:port2/db2
```

Note: If the `application_server` (`MongoDBnode_hostname1:port1`) is different from the target `application_server` (`MongoDBnode_hostname2:port2`), the rename entry must contain `ALT_APPLICATION_SERVER=MongoDBnode_hostname2:port2`.

After making the required changes in the rename file, you can run the `bprestore` command. For more information, See "Using the command line to recover a MongoDB database" on page 65.

Using the command line to recover a MongoDB database

You can use the `bprestore` command to recover a backed-up MongoDB instance.

For more information about the `bprestore` command, refer to the *NetBackup Commands Reference Guide*.

To recover a MongoDB database

1. On the NetBackup master server, log on as an Administrator or root user based on Windows or UNIX system respectively.

2 Run the following command on the NetBackup master server by providing appropriate values:

```
bprestore -R /usr/openv/tmp/rename.txt -C MongoDBnode-port -s
starttime -e endtime -D backup_host -S master_server -L
path_progress_log -t 44 -p policyname -f
/usr/openv/tmp/filelist.txt
```

Where,

`-C`

Specifies the MongoDB cluster node and port number that you have backed up.

`-D`

Specifies the host name or the FQDN of the backup host.

`-e`

Specifies the end time of the backup window.

`-f`

Specifies a file (`listfile`) that contains a list of files to be restored and can be used instead of the file names option (`filenames`). In `listfile`, list each file path must be on a separate line.

For MongoDB, the file list must contain *<MongoDBnode_hostname>:<port>* of all the MongoDB nodes.

`-p`

Specifies the backup policy.

`-R`

Specifies the directory path to a rename file.

`-s`

Specifies the start time of the backup window.

`-S`

Specifies the host name or FQDN of the master server.

`-t 44`

Specifies BigData as the policy type.

`-L progress_log`

Specifies the name of allowed list file path in which to write progress information.

Manual steps after the recovery process

- After you recover the backup images that were taken from the hidden MongoDB nodes, the hidden nodes become primary nodes. Update all such primary nodes in the shard list and restart the `mongos` process using the following command:

  ```
  db.getSiblingDB('config').shards.updateOne({ "_id" : "shard1" },{
  $set : { "host" :
  "ShardName/repl1.example.net:27018,repl2.example.net:27018,repl3.example.net:27018"
  } })
  ```

- After the recovery process is complete, manually add the secondary nodes to the cluster.

  For more information, refer to the following article:
  add-members-to-the-replica-set

- After the recovery operation, the `mongod` or `mongos` process is started using the configuration files from the `/tmp` location. Ensure that you move the configuration files to a selected location and restart the services from that location.

  Remove the configuration files from the `/tmp` location so that the restore or recovery operations can restore files at the `/tmp` location with the same name for different users. If you do not remove the files, the subsequent recovery operations using a different user fails with error 2850 because the configuration files cannot be restored at the `/tmp` location.

  You can add more MongoDB configuration parameters if there are any changes from the backup data that is restored.

Chapter 6

Troubleshooting

This chapter includes the following topics:

- About NetBackup for MongoDB debug logging
- Known limitations for MongoDB protection using NetBackup

About NetBackup for MongoDB 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.

Additionally, after every job (backup/restore), `mdbserver` log(s) that are created on the MongoDB nodes are copied to the respective backup hosts from where it was processed. These logs are kept in the `nbaapireq_handler` folder so that they can be collected by `nbsu` or `nbcplogs` utilities. To retain uniqueness of log file names collected from different hosts to a single folder, each log file name is prefixed with the hostname. For example, if the log files generated by `mdbserver` on hosts MDBSERVER1 and MDBSERVER2 for a backup job are "root.mdbserver.121219_00001.log", they are copied back to the backup host as MDBSERVER1-root.mdbserver.121219_00001.log and MDBSERVER2-root.mdbserver.121219_00001.log.

The log folders reside on the following directories

- On Windows: `install_path\NetBackup\logs`
- On UNIX or Linux: `/usr/openv/netbackup/logs`

Table 6-1 NetBackup logs related to MongoDB

Log Folder	Messages related to	Logs reside on
`install_path/NetBackup/logs/nbaapidiscv`	BigData framework, discovery, and MongoDB configuration file logs	Backup host
`install_path/NetBackup/logs/bpbrm`	Policy validation, backup, and restore operations	Media server
`install_path/NetBackup/logs/bpbkar`	Backup	Backup host
`install_path/NetBackup/logs/tar`	Restore and MongoDB configuration file	Backup host
`install_path/NetBackup/logs/nbaapireq_handler`	`nbaapireq_handler` and `mdbserver`	Backup host

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

Known limitations for MongoDB protection using NetBackup

The following table lists the known limitations for MongoDB protection using NetBackup:

Table 6-2 Known limitations

Limitation	Workaround
In a sharded MongoDB cluster with high availability that contains multiple `mongos` processes, before starting a restore and recover operation, only the `mongos` process on the restore destination for the Config Server Replica Set (CSRS) image should be running. Manually stop any other `mongos` processes in the cluster before starting a restore and recover operation. After recovery, reconfigure the `mongos` services to point to the recovered cluster. If `mongos` process is not shut down on all nodes except one, the additional `mongos` processes might conflict with the restore and recover operation, causing the data that is restored to be inaccessible via a connection to `mongos`.	In case the `mongos` processes are not shutdown before starting the restore and recovery, then after recovery you must manually shutdown the stale `mongos` processes and then restart all the recovered `mongod` and `mongos` processes under cluster.

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
You must start the MongoDB processes with an absolute path to the configuration files. You must use the absolute paths for the certificate files and the CA file as well. You must specify the absolute paths for the CA file, PEM file and Key Files as well.	N/A
If the authentication type that was present during backup changes and you run a recovery job that requires a different authentication, the recovery process might fail.	Ensure that the authentication type during recovery remains the same as the type used during the backup.
After running a backup if you rename the volume group or the logical volume, the subsequent backup might fail.	N/A
During recovery, ensure that you select only one full backup image and its relevant subsequent incremental images. If you select more than one image, the recovery may fail as the restored data could be corrupted.	N/A
After your recover the MongoDB cluster, the cluster information for only the restored node is available.	After the recovery process is complete, manually add the secondary nodes to the cluster. For more information, refer to the following article: add-members-to-the-replica-set
During the backup process, if the MongoDB import operation is running, it can become unresponsive. Avoid the MongoDB import operation during the backup or restore process.	N/A
During the restore process, “The restore was successfully initiated” popup is displayed, but the restore job does not start. This issue occurs when you enter the Application Server in both the Source Client and Destination Client in the BAR UI.	Ensure that Source Client and Destination Client are entered correctly. The Source Client must be the Application Server and the Destination Client must be the backup host.
If your environment has DNAT, ensure that the backup host or the restore host and all the MongoDB nodes are in the same private network.	N/A
The NetBackup for MongoDB plug-in does not support the command line `bprestore` options `-w` and `-print_jobid`.	N/A

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
MongoDB restores are not supported from the backup hosts. All the restore operations for MongoDB must be initiated from the NetBackup master.	N/A
If your restore job submission is not displaying the restore job, check if your destination node has a MongoDB plug-in installed on it.	N/A
If you restore the MongoDB database to a non-LVM location and then try to take a backup from this non-LVM location, the backup fails.	Restore the data to an LVM location and then try to take a backup of the restored data.
The NetBackup for MongoDB plug-in does not support hard or soft links in the data path folders. Do not add any hard or soft links that point to locations in a different logical volume or a non-logical volume. NetBackup cannot ensure that the data is consistent at the time of backups if you have hard or soft links in the data path folder. During the restore process, the hard or soft links are created as folders and not links.	N/A
When you cancel a child restore job during the MongoDB restore and recovery process, the thin client (`mdbserver`) is not removed immediately. The thin client is removed after the next restore operation.	N/A
MongoDB restore fails and displays error 2850.	Ensure that the destination host and port is valid and has the credentials configured using the `tpconfig` command and the credentials file. For more information, refer to the `tar` logs.

After recovery, the MongoDB shard node fails to restart manually and the following error is seen in the MongoDB logs:
`NoSuchKey: Missing expected field "configsvrConnectionString"`
	On the MongoDB shard where the problem occurs, start MongoDB in the maintenance mode and run the following method on the `system.version` collection in the admin database:
	`use admin`
	`db.system.version.deleteOne ( { _id: "minOpTimeRecovery" } )`

Table 6-2 Known limitations *(continued)*

Limitation	Workaround

In a restore and recover operation containing one or more replica sets, replica set members are restored to the replica set using the default "`cfg.members[#].host`" value provided by `rs.config()`.
 	
If this value was previously updated from the default value, after the restore and recover completes, this value may need to be updated (for example, from shortname to FQDN), to match the original configuration.
	Workaround:
	
	1 Log in to the replica set MongoDB cluster
	
	2 Use the following command to check the configuration:
	
	`rs.conf()`
	
	3 Use the following command to update the configuration for replica set:
	
	`Update configuration for replica set member 0:`
	`cfg = rs.conf();`
	`cfg.members[0].host = '<hostname.domain.com>:`
	`<port-number>';`
	`rs.reconfig(cfg)`
	
	4 Verify the changes using the following command:
	
	`rs.conf()`
	
	5 Repeat the steps for the other replica sets and the members, or just the replica set members.

Backup jobs fail and the following error codes are displayed: ■ (50) client process aborted ■ (1) The requested operation was partially successful ■ (112) no files specified in the file list	Ensure that the backup windows for incremental backups are different for the same MongoDB cluster. The backup windows must not overlap each other for incremental backups for the same MongoDB cluster. Ensure that permissions are in place for the `mdbserver` location, `oplog` location, and snapshot mount location. For more information, See "Using a non-root user as a host user" on page 38. In a sharded MongoDB cluster environment, a 112 error can indicate that the `mongos` process is not running on the client defined in the backup policy. An error 112 can also indicate that same hosts names for multiple backup hosts are added to the BigData policy. Use unique host names for multiple backup hosts that are running the backup operations.

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
After a restore and recovery operation, if you try to stop and restart the `mongod` or `mongos` services (`service mongod stop` or `service mongod restart`), the commands fail. This error occurs when the `mongod` or `mongos` processes are launched as service using the `service` or `systemctl` commands and not using a direct command.	Workaround: Stop the `mongod` or `mongos` services using alternative methods. For example, `mongod -f /etc/mongod.conf --shutdown` or `kill <PID>`. After stopping the services, you can use the `service` or `systemctl` commands again. **Note:** When you stop the services after restore and recovery, the `.pid` or `.sock` files remain when you shutdown the `mongod` or `mongos` processes. You must delete the files if the `mongod` or `mongos` services do not start after shutting them down. The default location of the `.sock` files is `/tmp` The default location of the `.pid` files is `/var/run/mongodb/`
Backup operation fails if a command that generates output in `.bashrc` is added. Backup fails with error 6646 and displays the following error: Error: Unable to communicate with the server.	Ensure that no output is generated by `.bashrc` (`echo` or any other output generating command). The output should not return `STDERR` or `STDOUT` when the shell is non-interactive.
When you select two full backup images and try to restore to a point-in-time image that is between the two full backup images, the latest full backup image is restored.	Workaround: During the restore and recovery operation, do not select more than one full backup image. For an effective point-in-time recovery, ensure that you run differential incremental backups of shorter duration.
Unable to see the restore job progress in the Activity Monitor.	Workaround: For compound restore jobs that use a non-master server as the restore host, you must use the Update Task List button to display the restore job progress in the Activity Monitor.
Backup fails with the following error: (6625) The backup host is either unauthorized to complete the operation or it is unable to establish a connection with the application server.	Workaround: On the server where MongoDB is intalled, ensure that `PasswordAuthentication` is not disabled in `/etc/ssh/sshd_config` file. Run the `sudo service sshd restart` command.

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
Backup fails with the following error: (6646) Unable to communicate with the server.	Workaround: Ensure that the backup host can access the defined port in `mongodb.conf` file or the default `mdbserver_port` (11000). There could be an error while copying the thin client files on the MongoDB server because of the following: ■ Connectivity issues with the MongoDB server ■ User does not have permissions to the location for copying the thin client files
The following error is displayed in the `mdbserver` logs: `error-sudo: sorry, you must have a tty to run sudo`	Workaround: ■ To disable the `requiretty` option globally, in the `sudoers` file, replace `Defaults requiretty` with `Defaults !requiretty`. This action changes the global `sudo` configuration. ■ You can change the `sudo` configuration for the user, group, or command. On the server where MongoDB is installed, add the host user, or group, or command in the `sudoers` file. Add `Defaults` *`/path/to/my/bin`* `!requiretty` Add `Default <`*`host_user`*`> !requiretty`
The `nbaapireq_handler` log folder is not created on a Flex Container, even after running the `mklogdir` command.	Workaround: When a Flex Appliance is upgraded from version 8.1.2 to 8.2 and the Flex media server is used as backup host, then for logging the MongoDB plug-in restore logs create the `nbaapireq_handler` folder in the `/usr/openv/netbackup/logs/` directory.
MongoDB Restore fails with the error 2850	The target database path does not exist and there are insufficient permissions for the non-root user. Workaround: Ensure that the target database path exists and there are sufficient permissions for the non-root user.

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
The snapshot size as described by the `free_space_percentage_snapshot` parameter must be set according the MongoDB cluster size and must be large enough. If these criteria are not met, the backup fails and displays the following error: `invalid command parameter (20)`	Validate the `free_space_percentage_snapshot` value with the MongoDB cluster.
Backup fails with the following error: (13) file read failed for Media	Ensure that the: ■ NetBackup version on the master server is the latest. ■ NetBackup version on the media server is the same as the master server but newer than the NetBackup client version on the backup host. ■ NetBackup client version on the backup host is the same as or older than the media server.
The `mdb_progress_loglevel` parameter is missing from the MongoDB configuration tool.	To modify the `mdb_progress_loglevel` parameter, update the `mongodb.conf` file after it is created by the MongoDB configuration tool. For more information, refer to the MongoDB Administrator's Guide.
Snapshots are not deleted and stale `mdbserver` instances are seen. This scenario might cause `Cannot lstat` errors during backup and partially successful backups.	Change the configuration settings for the following parameters in the `mongodb.conf` file: ■ `cleanup_time_in_min` ■ `mdbserver_timeout_min` Set the values such that the stale snapshots and stale instances of `mdbserver` are cleared before the next full or incremental backup schedule.

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
If the backup host has NetBackup version earlier than 8.3 and master and media server have the latest version of NetBackup, the following invalid error codes can be seen for various scenarios: 13302, 13303, 13304, 13305, 13306, 13307, 13308, 13309, 13310, 13311, 13312, 13313, 13314, 13315	

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
	Workaround: Refer to the following list of corresponding actual error codes if you see the invalid error codes for the actual scenarios and recommended actions: ■ Invalid error code: 13302 Actual error: 6724 Message: Restore node count is invalid. ■ Invalid error code: 13303 Actual error: 6725 Message: Unable to find information about the MongoDB replica set. ■ Invalid error code: 13304 Actual error: 6704 Error: Message: Restoring multiple MongoDB nodes on one replica set is invalid. ■ Invalid error code: 13305 Actual error: 6705 Message: Restoring MongoDB data on an arbiter node is invalid. ■ Invalid error code: 13306 Actual error: 6706 Message: A discovered shard was found in drain state, cannot proceed with backup. ■ Invalid error code: 13307 Actual error: 6707 Message: An unsupported MongoDB storage engine is detected. ■ Invalid error code: 13308 Actual error: 6708 Message: Unable to parse command output ■ Invalid error code: 13309 Actual error: 6709 Message: Unable to run the command. ■ Invalid error code: 13310 Actual error: 6710 Message: Pre-check for recovery has failed as WiredTiger log files are present at the database path. ■ Invalid error code: 13311 Actual error: 6711

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
	Message: Unable to backup MongoDB configuration file. ■ Invalid error code: 13312 Actual error: 6712 Message: Unable to find operation log for previous backup. ■ Invalid error code: 13313 Actual error: 6713 Message: Operations log roll-over detected. ■ Invalid error code: 13314 Actual error: 6714 Message: Error while collection was iterated. ■ Invalid error code: 13315 Actual error: 6715 Message: Operation log verification error. For detailed information and recommended actions, refer to the *NetBackup Status Codes Reference Guide*.
Restore button in the NetBackup BAR UI can get disabled for the imported MongoDB backup images.	Workaround: If you import the images to the same NetBackup master server that was originally used to back them up, use either of the following methods: ■ Perform the restore operation using the `bprestore` command. ■ Restore the catalog backup that enables the restore button in the BAR UI and then restore the images. If you import the images to a different NetBackup master server than the one that was originally used to back them up, use the `bprestore` command to run the restore operation.

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
Recovery operation fails on an alternate, sharded MongoDB cluster. The following error is displayed: Unable to find the configuration parameter. (6661)	

Table 6-2 Known limitations *(continued)*

Limitation	Workaround

	This issue occurs during an alternate cluster recovery because the pre-recovery check is unable to find the `mongos` port for the alternate cluster in the `mongodb.conf` file. This is because of the way the MongoDB configuration tool creates the `mongodb.conf` file when you add the alternate MongDB cluster details using the **Update** option from the tool.
	
	Workaround:
	
	Before you start the recovery process, update the `mongodb.conf` file to separate the alternate cluster from the original cluster.
	
	For example:
	
	Existing `mongodb.conf` file:
	
	`"application_servers":`
	`{`
	`"original.mongodb.cluster.com:26050":`
	`{`
	`"alternate_config_server":`
	`[`
	`{`
	`"hostname:port":`
	`"alt.mongodb.cluster.com:26000",`
	`"mongos_port": "26001"`
	`}`
	`],`
	`"mongos_port": "26051"`
	`}`
	`}`
	
	Suggested updated to the `mongodb.conf` file:
	
	`"application_servers":`
	`{`
	`"original.mongodb.cluster.com:26050":`
	`{`
	`"mongos_port": "26051"`
	`},`
	`"alt.mongodb.cluster.com:26000":`
	`{`
	`"mongos_port": "26001"`

Table 6-2 Known limitations *(continued)*

Limitation	Workaround
	} }
The MUI tool displays the following error: Unable to delete configuration.	Recommended Action: ■ Check that the `<hostname-port>.conf` file still exists in the `/usr/openv/var/global` directory. ■ Refer to the `tpconfig` logs and check for error: Translating EMM_ERROR_MachineNotExist(2000000) to 88 in the Device Config context. Work Around: Delete the `<hostname-port>.conf` file manually from `/usr/openv/var/global`.

Appendix A

Additional information

This appendix includes the following topics:

- Sample MongodB configuration utility workflow to add and update MongodB credentials

Sample MongodB configuration utility workflow to add and update MongodB credentials

Adding MongoDB credentials

```
Device Management Configuration Utility
1) Drive Configuration
2) Robot Configuration
3) Credentials Configuration
4) MongoDB Configuration
5) Print Configuration
6) Help
7) Quit

Enter option :4

MongoDB Application Configuration

1) Configure MongoDB Application Topology & Credentials
2) Configure NetBackup Global Parameters for MongoDB Application
3) Quit

Enter option :1

Configure the MongoDB cluster credentials
1) ADD Credentials
```

```
2) UPDATE Credentials
3) DELETE Credentials
4) Return to previous menu

Select the operation :1

Please select your MongoDB cluster type.
1) Standalone node
2) Sharded Cluster
3) Replica set
4) Return to main menu

Select the type of your MongoDB cluster :3

Select MongoDB host credentials type
1) No Auth
2) Simple Auth
3) Certificate based
4) Return to main menu

Select the authentication type used in the MongoDB cluster :2

Configure Replica Set MongoDB Cluster

Enter the hostname of primary server : host1.fqdn.com
Enter the mongod port of primary server [On the MongoDB Shell, run the
command "rs.status()" for replica set and "sh.status()" for sharded
environment] : 28000
Enter the name of MongoDB host user : root
Enter the password of MongoDB host user :
Enter the RSA key of the MongoDB host [On the MongoDB host, run the
command "cat /etc/ssh/ssh_host_rsa_key.pub  | awk '{print $2}' |  base64
-d| openssl dgst -sha256 | awk '{print $2}'"] : RSA-KEY-OF-THE-HOST
Enter MongoDB database user : mongodb-shell-login-user
Enter MongoDB database user password :

Does this primary server has replicas?(y/n) :y

Enter the hostname of secondary server : host2.fqdn.com
Enter the mongod port of secondary server [On the MongoDB Shell, run the
command "rs.status()" for replica set and "sh.status()" for sharded
environment] : 28001
Enter the name of MongoDB host user : root
Enter the password of MongoDB host user :
```

```
Enter the RSA key of the MongoDB host [On the MongoDB host, run the
command "cat /etc/ssh/ssh_host_rsa_key.pub  | awk '{print $2}' |  base64
-d| openssl dgst -sha256 | awk '{print $2}'"] : RSA-KEY-OF-THE-HOST
Enter MongoDB database user : mongodb-shell-login-user
Enter MongoDB database user password :

Do you have more secondary servers for this primary server? (y/n) :n
```

Summary is displayed after you add the credentials.

```
----------------------REPLICA SET MONGODB CONFIGURATION SUMMARY-------------------------

-------------------------------------------------------------------------------------

        Primary Server :
               Server Hostname        : host1.fqdn.com
               Server Mongod Port     : 28000
               No of Secondary Servers             : 1
               HostUser: root
               HostPassword: ******
               AppUserId: mongodb-shell-login-user
               AppUserPassword: ******
               HostRsaKey: RSA-KEY-OF-THE-HOST

-------------------------------------------------------------------------------------

        Secondary Server number 1:
               Secondary Server Hostname        : host2.fqdn.com
               Secondary Server Mongod Port     : 28001
               HostUser: root
               HostPassword: ******
               AppUserId: mongodb-shell-login-user
               AppUserPassword: ******
               HostRsaKey: RSA-KEY-OF-THE-HOST

-------------------------------------------------------------------------------------
******Please make sure to save this entered config and credentials. If you
don't save it now, you will have to enter it again.******

Do you want to save this cluster configuration and credential info?(y/n) :

Please wait while we save the cluster configuration.
Successfully saved config and credentials for this cluster.
Please use Client name as "host1.fqdn.com-28000" under 'Clients' tab in
```

```
mongodb backup policy.
Press any key to to return to main menu...
```

Updating MongoDB credentials

```
Device Management Configuration Utility
1) Drive Configuration
2) Robot Configuration
3) Credentials Configuration
4) MongoDB Configuration
5) Print Configuration
6) Help
7) Quit

Enter option :4

MongoDB Application Configuration

1) Configure MongoDB Application Topology & Credentials
2) Configure NetBackup Global Parameters for MongoDB Application
3) Quit

Enter option :1

Configure the MongoDB cluster credentials
1) ADD Credentials
2) UPDATE Credentials
3) DELETE Credentials
4) Return to previous menu

Select the operation :2

Please select your MongoDB cluster type.
1) Standalone node
2) Sharded Cluster
3) Replica set
4) Return to main menu

Select the type of your MongoDB cluster :3

Update replica set MongoDB cluster configuration

Enter the hostname of primary server : host1.fqdn.com
Enter the mongod port of primary server [On the MongoDB Shell, run the
command "rs.status()" for replica set and "sh.status()" for
```

```
sharded environment] : 28000
[Note- similar steps can be followed for deleting creds for cluster]

--
Update host1.fqdn.com:28000 replica set MongoDB cluster configuration

        1) Update primary server credentials
        2) Add secondary server
        3) Update secondary server config & credentials
        4) Delete secondary Replica server
        5) Return to previous menu

Enter option: option as applicable
```

Index

A

Adding
    backup host 39
allowed list
    backuphost 40

B

Backup 44
backup 42
BigData policy
    Command Line Interface 48
    NetBackup Administration Console 46
    Policies utility 47
    Policy Configuration Wizard 47

C

compatibility
    supported operating system 20
Creating
    BigData backup policy 46

L

Limitations 15

M

MongoDB credentials
    adding 34

N

NetBackup
    debug logging 69

O

overview
    backup 12
    configuration 12
    deployment 12
    installation 12
    restore 12

P

parallel streaming framework 12
policies
    configuring 46

R

Removing
    backup host 39
restore 51

T

terms 14