NetBackup™ Web UI MySQL Administrator's Guide

Release 10.1.1

VERITAS™

Last updated: 2023-02-14

Legal Notice

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054

http://www.veritas.com

Technical Support

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

https://www.veritas.com/support

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

https://my.veritas.com

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

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

Documentation

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

https://sort.veritas.com/documents

Documentation feedback

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

NB.docs@veritas.com

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

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

Veritas Services and Operations Readiness Tools (SORT)

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

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

Contents

Chapter 1

Overview

This chapter includes the following topics:

Overview of NetBackup for MySQL

NetBackup uses role-based access control to grant access to the web UI. Access control is accomplished through roles.

Table 1-1 Steps to configure and protect MySQL assets

Step	Action	Description
Step 1	■ Open a web browser and go to the URL ■ Enter your credentials and click **Sign in**. ■ On the left, click **Security** > **RBAC** > **Add**. ■ Select **Default MySQL Administrator** and provide a **Role name**, **Role description**, required permissions and assign a webUI user to this role.	For more information on **Sign In** see the *Sign into the NetBackup web UI* in *NetBackup Web UI Administrator's Guide*. **Note:** To perform the MySQL administrator tasks, the **Default MySQL Administrator** role should have the minimum required RBAC permissions. See See "Default MySQL Administrator" on page 15. for more information.
Step 2	Configure and manage MySQL workload.	See "Configure MySQL instance " on page 9.
Step 3	Add and manage credentials.	See "Manage credentials for an instance" on page 11.

Table 1-1 Steps to configure and protect MySQL assets *(continued)*

Step	Action	Description
Step 4	Configure a MySQL protection plan.	See “Protect MySQL instances and databases” on page 19.
Step 5	Protect MySQL instances and databases.	See “Protect MySQL instances and databases” on page 19.
Step 6	Restore MySQL instances and databases.	See “Restore a MySQL instance and database ” on page 23.

Chapter 2

Managing MySQL instances and databases

This chapter includes the following topics:

Quick configuration checklist to protect MySQL instances and databases

Use NetBackup web UI to protect and restore the instances and databases that are created on the MySQL platform.

The following table describes the high-level steps to protect the MySQL environment.

Table 2-1 Configure and protect MySQL using NetBackup

Step overview	Description and reference
Deploy the NetBackup to protect MySQL instances and databases.	On a very high level to protect MySQL instances and databases you need: ■ NetBackup primary server ■ NetBackup media server ■ NetBackup client on MySQL server
(optional) MySQL installed bin directory path should be added to path environment variable.	Verify if MySQL installation bin path is set in environment variable. For Example: ■ For Windows : PATH = `C:\Program Files\MySQL\MySQL Server 8.0\bin` ■ Linux : `export PATH=$PATH:/var/lib/mysql`
Protecting MySQL instances and databases.	See "Protect MySQL instances and databases" on page 19.

Configure MySQL instance

You can configure MySQL protection using the following environment variables:

- (optional) PATH - Add MySQL bin path to this environment variable for running queries and connecting to databases.
- (optional) LIB_MYSQL_CLIENT - For Windows set this environment variable to provide the location of `libmysql.dll` library. For Linux set this environment variable to provide the location of `libmysqlclient.so` library.
- (optional) LIB_MYSQL_CLIENT_<port> - This environment variable is used for MySQL multi-instance deployment.
  - For Windows set this environment variable to provide the location of respective instances `libmysql.dll` library.
  - For Linux set this environment variable to provide the location of respective instances `libmysqlclient.so` library.
- (optional) MYSQL_SOCKFILE_<port> - This environment variable is used for MySQL multi-instance deployment. For Linux set this environment variable to provide the location of respective instances `mysql.sock` file.
- (optional) MYSQL_BACKUP_DUMP_DIRECTORY - Set this environment variable as temporary backup dump directory for non streaming backup. For example, for Linux, user can set this environment variable to required location using below command:

```
echo "export
MYSQL_BACKUP_DUMP_DIRECTORY=/home/custom_dump_dir_location/" > >
~/.bashrc
```

For Windows, user can create new environment variable and add path of folder location as below:

```
MYSQL_BACKUP_DUMP_DIRECTORY=C:\custom_dump_dir_location
```

- (optional) LVM SNAPSHOT_SIZE - Set this environment variable to provide the snapshot size for LVM backup for Linux operating system only. You can set environment variable of LVM Snapshot size to 500 MB using below command:

```
echo "export LVM_SNAPSHOT_SIZE=500MB" >> ~/.bashrc
```

Note: The default snapshot size is set to 500MB.

Add MySQL instance

You can add MySQL instance and its credentials.

To add MySQL instance and its credentials

1. On the left, click **MySQL** then click the **Instances** tab.
2. Click **Add** to add a MySQL instance and enter the following:
   - **Host**
   - **Instance name**
3. Enter or use the up, down arrow keys to add details of **Port number**.
4. Click **Next**.

   Note: You will be redirected to the **Permissions** page and you can also manage credentials of the created instance.

5. Click **Finish**.

   Note: If you click **Previous**, instance created will not get saved.

Assign permissions to MySQL instance

You can assign permissions to an instance added.

To assign permissions to the MySQL instance use the following steps:

1. Click **Add** to add permissions to this instance.
2. Select role and permissions.
3. Click **Save** > **Finish**.

Inline actions on MySQL instance

You can run the following inline actions on a MySQL instance:

- **Recover**: Recovers the MySQL instance.
- **Manage credentials**: Manages the instance credentials.
- **Deactivate**: Deactivates the MySQL instance.
- **Remove**: Removes the MySQL instance.

Actions on multiple MySQL instances

You can select one or more MySQL instances and perform the following actions:

- **Deactivate**: Deactivates the MySQL instances.
- **Manage credentials**: Manages the credentials of the MySQL instances.
- **Remove**: Removes the selected MySQL instances.

Manage credentials for an instance

You can add or update credentials for instances. When you add an instance, you can choose not to include the credentials at the time of its entry.

To add credentials for an instance at the time of its entry into the repository

1. Select **Manage credentials**.
2. In the **Manage credentials** screen, select one of the appropriate methods:
   - **Select from existing credentials**.
   - **Add credentials**.
3. Click **Next**.

Discover MySQL databases

You can discover MySQL databases.

To discover MySQL databases

1. On the left, click **MySQL** then click the **Database** tab.
2. Click **Discover** to discover a MySQL database.
3. Select the required instance from the list of instances for which you need to discover the databases.
4. Click **Discover**.

Remove MySQL instances

Use this procedure to remove MySQL instances.

To remove MySQL instances:

1. On the left, click **MySQL**, then click the **Instances** tab.

   Note: The tab lists the names of instances that you have access to.

2. Select the MySQL instance from the list of instances that you have access to.
3. Select **Actions > Remove** or select **Remove** from top bar.

   Note: If you remove an instance, all databases that are associated with the removed MySQL instance will also get removed.

4. If you are sure that you want to remove the MySQL instance, click **Remove**.

   Note: Manually remove the **Instances** and the associated databases which are deleted from the MySQL Server.

Change the autodiscovery frequency of MySQL assets

Automatic discovery of MySQL assets occurs at regular intervals. The default frequency is every 8 hours. Use this procedure to change the autodiscovery frequency.

To change the frequency of autodiscovery of MySQL assets:

1. On the left, click **Workloads > MySQL**.
2. On the right, click **MySQL settings > Autodiscovery**.
3. Select **Frequency > Edit**.
4. Enter the number of hours or use the up or down arrows to choose how often you want NetBackup to perform autodiscovery of MySQL assets. Then click **Save**.

Note: The range from which you may choose is 1 hour to 24 hours. To set the autodiscovery frequency in minutes or seconds or to disable autodiscovery, you must use the MySQL autodiscovery API.

Chapter 3

Managing MySQL environment credentials

This chapter includes the following topics:

Add new MySQL credentials

You can add new credential to an instance at the time of its creation. See “Manage credentials for an instance” on page 11.

To add new MySQL credentials

1. On left, click **Credential management**.
2. On **Named credentials** tab, click **Add**.
3. Provide **Credential name**, **Tag** and **Description**.
4. Click **Next**.
5. Select **MySQL server** from **Category** dropdown.
6. Enter **Instance username** and **Instance user password** and click **Next**.
7. On the **Permissions** page, click **Add**.
8. Select role and permissions.

9 Click **Save** > **Next**.

10 Review and click **Finish**.

Note: You can **Edit** or **Delete** the added credentials.

Default MySQL Administrator

This role has all the permissions that are necessary to manage MySQL and to back up those assets with protection plans.

Table 3-1 RBAC permissions for Default MySQL Administrator role

Type	Permissions
Global permissions > NetBackup management	
Access hosts	View, Create, Delete
Agentless hosts	View
Host Properties	View
Media Server	View
NetBackup hosts	View, Create, Update
NetBackup backup images	View, View Contents
Jobs	View
Resource limits	View, Create, Update, Delete
Trusted primary servers	View
Global permissions > Storage	
Storage servers	View, Create, Update, Delete
Disk volumes	View, Create, Update
Storage units	View, Create, Update, Delete
Assets	
MySQL assets	Full permissions
Protection plans	Full permissions
Credentials	Full permissions

Validate credentials of MySQL instance

To validate MySQL instance credentials

You can validate a specific or multiple instance's credentials.

1. On the left, click **Workloads** > **MySQL**, then click the **Instances** tab.
2. Locate and select one or more MySQL instances.
3. Click **Manage Credentials** > **Select from existing credentials**.
4. Click **Next** and select the credentials that you want to use for this instance.
5. Click **Next** > **Close**.

Note: NetBackup verifies the current credentials for the selected MySQL instance.

If the credentials are not valid, NetBackup indicates **Invalid** under **Credentials**.

View the credential name that is applied to an asset

You can view the named credential that is configured for an asset type. If the credentials are not configured for a particular asset, this field is blank.

To view credentials for MySQL

1. On the left, select **Workloads** > **MySQL**.
2. On the MySQL **Instances** tab, scroll right to locate the **Credential name** column.

Edit or delete a named credential

You can edit the properties for a named credential or delete a named credential from the **Credential management**.

Edit a named credential

You can edit a named credential when you want to change the credential **Tag**, **Description**, **Category**, authentication details, or permissions. You cannot change the credential name.

To edit a named credential

1. On the left, click **Credential management**.
2. Click **Edit** and update the credential as needed.

   Note: When you update MySQL instances, this action automatically starts the discovery of the MySQL instance.

3. Review the changes and click **Finish**.

Delete a named credential

You can delete a named credential that you no longer need to use.

Warning: Apply another credential to any asset that uses the credential you want to delete or else backup and restore may fail for those assets.

To delete a named credential

1. On the left, click **Credential management**.
2. On the **Named credentials** tab, locate and click on the credential that you want to delete.
3. Click **Delete**.
4. If you are sure that you want to delete, click **Delete**.

Chapter 4

Protecting MySQL instances and databases

This chapter includes the following topics:

Things to know before you protect MySQL instances and databases

Protection plans can be used to predefine backup policies which are then used by others to protect their data. The following table describes the permissions with which MySQL database non-root user must be created:

Table 4-1 User Privileges

User	Privileges
Instance Superuser or Administrator	Select, Insert, Update, Create, Drop, Reload, Shutdown, File, Index, Alter, Super, Lock Tables, Create View, Show View, Trigger, Process, System_User, Create Routine, Delete, Event, Alter Routine

To set the database user privileges, run the following command at MySQL command line:

```
GRANT SELECT, INSERT, UPDATE, CREATE, DROP, RELOAD, SHUTDOWN, FILE,
INDEX, ALTER, SUPER, LOCK TABLES, CREATE VIEW, SHOW VIEW, TRIGGER,
PROCESS, SYSTEM_USER, CREATE ROUTINE, DELETE, EVENT, ALTER ROUTINE
ON *.* TO 'USER'@'localhost' IDENTIFIED BY 'PASSWORD'
```

Protect MySQL instances and databases

Use the following procedure for subscribing a protection plan to MySQL instance or database. When you subscribe an asset to a protection plan, you assign predefined backup settings to the asset.

Note: The RBAC role assigned to user must have access to the assets that you want to manage and also to the protection plans you want to use.

To protect MySQL instance or database:

1. On the left pane, click **MySQL**.
2. On the Instances tab or Databases tab, click the box for the instance or the database and click **Add protection**.
3. Select a protection plan and click **Next**.
4. You can edit one or more of the following settings:
   - **Schedules and retention**
     Change when backups occur and the backup start window.
   - **Backup options**
     Adjust the **Database options** like **Job limit** and **Backup method**.
5. Click **Protect**.

Customize protection settings for the MySQL assets

To customize protection settings for the MySQL assets

You can customize certain settings for a protection plan, including schedules.

1. On the left, select **Workloads > MySQL**.
2. Click on the instance whose protection is to be customized.

   Note: This action allows custom protection for the asset and removes it from the original protection plan. Any future changes to the original plan are not applied to the asset. The customization operation cannot be reversed.

3. Click **Customize protection** > **Continue**.
4. You can edit one or more of the following settings:
   - **Schedules and retention**
   - **Backup options**
5. Click **Protect**.

Remove protection from MySQL instances

You can unsubscribe MySQL instances from a protection plan. When the asset is unsubscribed, backups are no longer performed.

Note: When you unsubscribe an asset from a protection plan, there is a possibility that the asset displays **Classic policy** in the **Protected By** column on the web UI. This situation can happen when an asset is subscribed to a protection plan and a backup runs for that asset. Such assets get unsubscribed from the protection plan. The web UI then displays **Classic policy**, that may or may not have an active policy protecting the asset.

To remove protection from a MySQL instance

1. On the left, click **MySQL**.
2. On the **Instances** tab, select the instance.
3. Click the instance name.
4. Click **Remove protection** > **Yes**.

   Under **MySQL**, the asset is now listed as **Not protected**.

View the protection status of MySQL instance

You can view the protection plans that are used to protect MySQL instance.

To view the protection status of MySQL instance

1. On the left, click **MySQL**.
2. On the **Instances** tab, select the instance. The **Protection** tab shows the details of the asset subscription plans.

   Note: If the asset has been backed up, but status indicates that it has not, you will get an error. See “Error while backup of MySQL asset” on page 33.

3. If the asset is not protected, click **Add protection** to select a protection plan.

Chapter 5

Restoring MySQL instances and databases

This chapter includes the following topics:

Things to know before you restore the MySQL instances and databases

Ensure that the restore server that is added to the Netbackup environment should have MySQL footprint on it.

About the pre-restore check

The pre-restore check verifies the following:

- Availability of the MySQL environment.
- Available space with the storage.

- In mysqldump backup, while performing restore and recovery on MySQL server instance must be up and running and data directory must not be empty.
- (For Windows) ICACLS windows command-line utility packages must be installed and installed path must be a part of environment path variable.

Restore a MySQL instance and database

You can restore a MySQL instance or database either to an original backup location or to an alternate location. You can choose to recover from the default copy of the instance or database. The default copy is also known as the primary copy.

To restore a MySQL instance

1. On the left, click **Workloads** > **MySQL**.
2. On the **Instances** tab, select the instance that you want to recover.
3. Click **Recover** from the top bar.
4. On the **Recovery points** tab, select the date with available backup.

   Note: In the calendar view, dates with available backups are indicated with a green dot.

5. From the listed **Backup images/ Recovery points**, select the desired image or recovery point.

   Note: The backup images or recovery points are listed in rows with the respective backup timestamp.

6. Click **Actions** > **Perform complete instance recovery**.
7. Click the search icon in **Host** field, select the desired host and click **Save**.
   - If the recovery is to alternate an host, then select the corresponding valid credentials from the displayed list.

   For more information, See "Restore target options" on page 26.
8. Select the appropriate instance directory path from one of the following options:
   - **Restore everything to original location**: Files are restored to the location where they were originally backed up from.
   - **Restore everything to a different location**: Files are restored to alternative location that you can specify. The folder structure of the restored data within

the alternate location will be the same as that of the original data that is same folder and sub-folder setup.

- **Directory for restore** – This is MySQL data directory. MySQL full backup data would be restored to the specified path.
- **Binary log directory for restore** – MySQL bin log files will be restored in this directory. MySQL incremental backup data would be restored to the specified path.

For more information, See "Restore target options" on page 26.

9 Click **Next** and follow the instructions prompted.

10 On the **Recovery source** tab, review the storage details.

11 Click **Next**.

12 On the Recovery points tab, select the **Restore** or **Restore and recovery** option to perform instances and database restore and recovery:

- **Restore** – Will restore the instances.
- **Restore and recovery** – Will recover the instances.

Note: For LVM and VSS if **Restore and Recovery** option is selected, then contents of target data directory would be deleted by recovery operation.

---

13 Click **Next**.

14 On the **Review** tab, review the details and click **Start recovery**.

To restore a MySQL database

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

2 On the **Databases** tab, select the database that you want to recover.

3 Click **Recover** from the top bar.

4 On the **Recovery points** tab, select the date with available backup.

Note: In the calendar view, dates with available backups are indicated with a green dot.

---

5 From the listed **Backup images/ Recovery points**, select the desired image or recovery point.

Note: The backup images or recovery points are listed in rows with the respective backup timestamp.

---

6 Click **Actions** > **Perform complete database recovery**.

7 Click the search icon in **Host** field, select the desired host and click **Save**.

- If the recovery is to alternate an host, then select the corresponding valid credentials from the displayed list.

For more information, See "Restore target options" on page 26.

8 Select the appropriate **Database directory paths** from one of the following options:

- **Restore everything to original location**: Files are restored to the location where they were originally backed up from.
- **Restore everything to a different location**: Files are restored to alternative location that you can specify. The folder structure of the restored data within the alternate location will be the same as that of the original data that is same folder and sub-folder setup.
  - **Directory for restore** – This is MySQL data directory. MySQL full backup data would be restored to the specified path.
  - **Binary log directory for restore** – MySQL bin log files would be restored to this directory. MySQL incremental backup data would be restored to the specified path.

For more information, See "Restore target options" on page 26.

9 Click **Next** and follow the instructions prompted.

10 On the **Recovery source** tab, review the storage details.

11 Click **Next**.

12 On the Recovery points tab, select the **Restore** or **Restore and recovery** option to perform instances and database restore and recovery:

- **Restore** – Will restore the database.
- **Restore and recovery** – Will recover the database.

13 Click **Next**.

14 On the **Review** tab, review the details and click **Start recovery**.

Restore target options

Table 5-1 Restore target options

Step overview	Description and reference
Host	■ Host field is pre-populated with the source MySQL client stored during last successful discovery for respective instance. ■ If you want to perform a restore on another NetBackup client, click search and select the required client from the list. **Note:** Ensure that you select clients with homogenous platforms. ■ If search option is unavailable, manually enter **Host**.
Instance directory paths	■ **Change staging location on client**: If you want to provide a different staging location other than the default staging location, enter the desired path. Staging location path must have only ASCII characters. **Note:** Default staging location is user's home directory. ■ **Instance directory paths** : Based on your requirement, select one of the following appropriate Instance directory paths between: ■ **Restore everything to original directory** ■ **Restore everything to different directory** Provide different directory path to restore.

Pre-restore checks for MySQL

Table 5-2 Pre-restore checks

Validation	Description and reference	Input Source
Restore client space	Checks for the required space on restore location.	Restore client
Target client connectivity	Checks if target client is accessible from restore client.	Target client and Target client name

Table 5-2 Pre-restore checks *(continued)*

Validation	Description and reference	Input Source
Target client alternate location on a local disk	Checks if target client alternate location is not a network path.	Target client alternate location
Target client location space	Checks if the required space is available on target client alternate location. **Note:** Required space is total size of selected file with space required for restore and space needed for logs and other files.	Target client alternate location
Target client alternate location permissions	Checks if provided user is an owner and has RBAC permissions on target client alternate location.	Target Target client alternate location
Target client default alternate location path	Checks if provided target client alternate location path contains valid characters. Non-ASCII characters are not supported in target client alternate location path.	Target client alternate location
Target client operating system	Checks if target client has a supported OS.	General

Table 5-3 Permissions for all MySQL assets

Operation	Description	Additional required operations	Additional optional operations
Restore	Restore backup images of MySQL asset. This permission is required on MySQL.	**Global > NetBackup management > NetBackup backup images > View** **Global > NetBackup management > NetBackup backup images > View contents** **Global > NetBackup management > NetBackup hosts > View** **Assets > MySQL assets > Restore**	**Assets > MySQL Assets > Restore to alternate location**

Steps to perform recovery after restore operation

The procedure to perform post recovery is as follows for various platforms:

For Windows (VSS):

1. Go to **Control Panel > System and Security > Administrative Tools >Services**.
2. Select MySQL service and stop it.
3. Delete or move everything from MySQL data directory.

   Note: Post restore, change the attributes of the restored data directory and files by using the following command.

   ```
   attrib -S <restore_path>/*.* /S /D
   ```

4. Copy all the contents of restored data directory to MySQL data directory.

5 Delete all the temporary files from data directory.

For example:

```
C:\ProgramData\MySQL\MySQL Server 8.0\Data\#innodb_temp.
```

Delete `undo_00x` files from following data directory:

```
\ProgramData\MySQL\MySQL Server 8.0\
```

For example:

```
C:\ProgramData\MySQL\MySQL Server 8.0\undo_001
```

6 Start MySQL service.

For Linux (LVM):

1 Stop MySQL services.

2 Copy all the contents of restored data directory to MySQL data directory.

3 Change ownership of MySQL data directory.

For example:

```
chown -R mysql:mysql <full/path/of/MySQL/Data/Dir>
```

4 Start the MySQL service.

Recovery Steps for backup done by mysqldump utility

Recover MySQL database using the mysqldump utility.

Recover single MySQL database

Below are examples of Netbackup commands which are used mostly in Windows and Linux platform.

- For Windows:

```
mysql --host=localhost --user=root --port=3306 -p netbackup <
C:\mysql_dump_testjune13\mysqlBackup_Dump_xxx.sqlx
```

- For Linux:

```
mysql --host=localhost --user=root --port=3306 -p netbackup <
/home/Single_DB_Recovery/mysqlBackup_Dump_1656932006.sqlx
```

Recover MySQL instance

The example commands below will create a single dump file containing all the databases.

- For Windows:

```
mysql --host=localhost --user=root --port=3306 -p <
C:\mysql_dump_testjune13\mysqlBackup_Dump_xxx.sqlx
```

- For Linux:

```
mysql --host=localhost --user=root --port=3306 -p <
/home/Entire_Instance_Recovery/mysqlBackup_Dump_1656933025.sqlx
```

Limitations

- Cross-platform recovery of individual files is not supported. The restore client must be the same platform as the instances that you want to restore. Windows instances can be restored using Windows operating systems and Linux instances can be restored only using Linux operating systems.
- For client platform and file system support and limitations, see https://www.veritas.com/content/support/en_US/doc/NB_70_80_VE .
- If a backup and a restore occur simultaneously on the same database, one or both jobs can have unexpected results.

  Note: If a backup or a restore exits with a non-zero NetBackup status code, one possible cause is simultaneous jobs occurring on the same instance.

- Restore job fails, if NetBackup does not have sufficient privileges or if there is insufficient space in the client memory.
- NetBackup does not support non-ASCII characters in target client location path.

Chapter 6

Troubleshooting MySQL operations

This chapter includes the following topics:

Troubleshooting tips for NetBackup for MySQL

For more information about MySQL troubleshooting, check the following details:

- For discovery failures:
  - Check the `ncfnbcs` log.
- For backup job failures:
  - Check the `bprd`, `bprm`, `bphdb` and `nbmysql` logs.
- For restore job failures:
  - Check the `bprd`, `bprm` and `tar` logs.

Error during MySQL credential addition

Table 6-1 Error during MySQL credential addition

Error message or cause	Explanation and recommended action
Credential validation failed. Provide correct host name.	The host's name is not valid NetBackup client. Ensure that the hostname is registered client of NetBackup and it is whitelisted.

Error during the MySQL instances and databases discovery phase

The following table describes the problem that might occur when you try to discover MySQL database.

Table 6-2 Error run into during the MySQL instance and database discovery phase

Error message or cause	Explanation and recommended action
The MySQL assets are not discovered after the correct MySQL instances credentials are added.	Run discover database and retry the database discovery manually. ■ Ensure that the update permission assigned to the logged in web UI user. ■ Contact Veritas Technical support and share `nbwebservice` logs from NetBackup master server and ncfnbcs logs from NetBackup client.

Error during the MySQL Protection Plan Creation

The following table describes the problem that might occur while creating protection plan for MySQL workload.

Table 6-3 Error during the MySQL Protection Plan Creation

Error message or cause	Explanation and recommended action
A plan with this name already exists.	Protection plan with same name is already present. ■ Please create protection plan with another name.
Storage disk pool is not present	Before adding protection, we need to add storage Unit. ■ Please add Storage Unit from **Storage Configuration** >**Add**.

Error while subscribing protection plan to MySQL asset

The following table describes the problem that might occur during subscribing protection plan to a MySQL asset.

Table 6-4 Error while subscribing protection plan to a MySQL asset

Error message or cause	Explanation and recommended action
This subscription must be reset to protection plan defaults before it can be customized.	If subscription has been already modified, the below warning message will be displayed. ■ User can reset subscription using 'Restore original settings' button and then try to customize subscription again.
Storage disk pool is not present	Before adding protection, we need to add storage Unit. ■ Please add Storage Unit from **Storage Configuration** >**Add**.

Error while removing MySQL asset

Table 6-5 Error while removing MySQL Asset

Error message or cause	Explanation and recommended action
Removed 0 of 1 instance.	If protection plan is attached to MySQL asset, then we cannot delete such an asset. ■ First unsubscribe protection plan from asset and then delete the asset.

Error while backup of MySQL asset

The following table describes the problem that might occur when you back up MySQL asset. Backup jobs fail with error code 6.

Table 6-6 Error while backing up MySQL assets

Error message or cause	Explanation and recommended action
The backup failed to back up the requested files	Verify the MySQL service is up and running on client. ■ Contact Veritas Technical support and share `bphdb` and `nbmysql` logs from backup client.

Error while restoring MySQL asset image

The following table describes the problem that might occur when you restore MySQL asset.

Table 6-7 Error while restore of MySQL asset image

Error message or cause	Explanation and recommended action
Unable to change the Host while modifying the restore target/destination.	If you cannot see the list of the host, you might not have access to NetBackup Host in RBAC. ■ Contact the NetBackup security administrator to resolve this issue.
Restore failed with below error: Restore initiated from XBSA Failed to query the object... 17	If the database user provided for restore operation is different from the backup operation database user. The permission of file differs in the NetBackup file system and hence restore fails. ■ Use the same database user for restore which was used while taking backup of asset, so that file system permissions will be available to the restore user as well.
Restore Image not found at alternate location on recovery host.	No image was found on recovery host alternate location ■ Contact Veritas Technical support and share `tar` log from the recovery host.

Chapter 7

API for MySQL instances and databases

This chapter includes the following topics:

Using APIs to manage, protect or restore MySQL

This topic lists the APIs to manage, protect or restore the MySQL instances and databases. Only the important variables and options are mentioned in this topic.

Following sections are part of this topic:

- See the section called “Add a MySQL instance” on page 36.
- See the section called “MySQL Discovery API” on page 36.
- See the section called “Create a MySQL Protection Plan” on page 37.
- See the section called “MySQL Recovery point Service API ” on page 37.
- See the section called “Restore the MySQL instance and database at the original location ” on page 38.
- See the section called “Restore the MySQL instance and database to an alternate location ” on page 38.

For detailed information on the APIs, use these references:

- All the NetBackup APIs are listed at the following location:
  Services and Operations Readiness Tools (SORT) > Knowledge Base > Documents

Add a MySQL instance

Table 7-1 Add a MySQL instance

API	Important variables and options

`POST /netbackup/asset-service/queries`
 	
`GET /netbackup/asset-service/queries/{aqcId}`
 	
`GET /netbackup/asset-service/workloads /mysql/assets`
	■ `clientName` is the name of the MySQL instance.
	■ `sqlHostName` is hostname of a NetBackup client.
	■ `credentialName` are credentials associated with MySQL instance.
	Note: The credential must exist with `credentialName` mentioned.
	■ `port` is port number of MySQL instance.

MySQL Discovery API

Table 7-2 Discover the MySQL asset for given client

API	Important variables and options

`POST /netbackup/admin/discovery /workloads/mysql/start`
 	
`POST /netbackup/admin/discovery/workloads /mysql/stop`
 	
`GET /netbackup/admin/discovery/workloads /mysql/status`
 	
`POST /netbackup/admin/discovery/workloads /mysql/allclientsdiscovery`
	■ `serverName` is used to identify instance or database
	■ `discoveryHost` is hostname where discovery needs to be triggered
	■ `allclientsdiscovery` triggers discovery for all the clients host associated with the master.

Create a MySQL Protection Plan

Table 7-3 Create a MySQL Protection Plan

API	Important variables and options

`POST /netbackup/servicecatalog/slos`
 	
`POST /netbackup/servicecatalog/slos/{sloId} /subscriptions`
 	
`POST /netbackup/servicecatalog/slos/{sloId} /backup-now`
	■ `policyType` is `DataStore`.
	■ `Add scheduleName` can have values like FULL_AUTO or INCR_AUTO for adding MySQL instance.
	■ `keyword` can have the following values to back up an instance or database using different backup options:
	• pg_dump
	• pg_basebackup
	• Snapshot
	■ `sloId` is the identifier to protection plan
	■ `selectionId` is the AssetId which needs to be subscribed with given slold

After you create a protection plan, other processes like creating the schedule for the policy or triggering the policy backup remain the same.

MySQL Recovery point Service API

Table 7-4 MySQL asset backup instances available for recovery

API	Important variables and options

`GET /netbackup/recovery-point-service /workloads/mysql/recovery-points`
 	
`GET /netbackup/recovery-point-service /workloads/mysql/recovery-points /{backupId}`
 	
`GET /netbackup/wui/workloads/mysql /recovery-point-calendar-summary`
	■ `backupId` is identifier that was used at the time of backup.
	■ `assetId` is identifier that was used to identify instance or database.
	■ `client hostname` is name of backup client.

Restore the MySQL instance and database at the original location

Table 7-5 Restore the MySQL instance and database at the original location

API	Important variables and options

`POST /netbackup/recovery/workloads/mysql/scenarios/instance-complete-recovery/recover`
 	
`POST /netbackup/recovery/workloads/mysql/scenarios/database-complete-recovery/recover`
	■ `backupId` is identifier that was used at the time of backup.
	■ `assetId` is identifier that was used to identify instance or database.
	■ `Client` is server that is to be used as the MySQL recovery host to perform this recovery. Set the following value:
	
	`renameAllFilesToSameLocation`

Restore the MySQL instance and database to an alternate location

Table 7-6 Restore the MySQL instance and database to an alternate location

API	Important variables and options

`POST /netbackup/recovery/workloads/mysql/scenarios/instance-complete-recovery/recover`
 	
`POST /netbackup/recovery/workloads/mysql/scenarios/database-complete-recovery/recover`
	■ `backupId` is identifier that was used at the time of backup.
	■ `assetId` is identifier that was used to identify instance or database.
	■ `Client` is server that is to be used as the MySQL recovery host to perform this recovery. Set the following value:
	
	`renameEachFileToDifferentLocation`