Restore MySQL databases with innodb_force_recovery

Why use the innodb_force_recovery option?

A higher rate of database restore failures have been observed in MySQL 5.7 and higher versions, after innodb_file_per_table became the default tablespace option. This is due to the data consistency standards that InnoDB adheres to. In short, when an attempt is made to restore a database, the database files are first used to start a temporary MySQL instance. This instance can fail to start when only a subset of databases are restored from the original instance. InnoDB attempts to locate the other database tablespaces referenced in the logs, and fails to find them if they were not included in the restore. Upon failing to find the referenced database tablespaces that were not included in the restore, the temporary instance fails to start.

The database restore failure may log an error similar to the following :

"Failed to start temporary database instance"

Fortunately, MySQL has provided the innodb_force_recovery option, which instructs InnoDB to start up in a recovery mode, allowing various portions of the InnoDB start-up process to be skipped.

> ⚠ NOTE
> Due to the destructive nature of higher innodb_force_recovery levels, please exercise caution when selecting the level. We recommend starting with the lowest value of 1 and increasing the value only if required.

How to Restore MySQL databases with innodb_force_recovery option

1. Follow the steps in this article to select your databases and begin the restore.

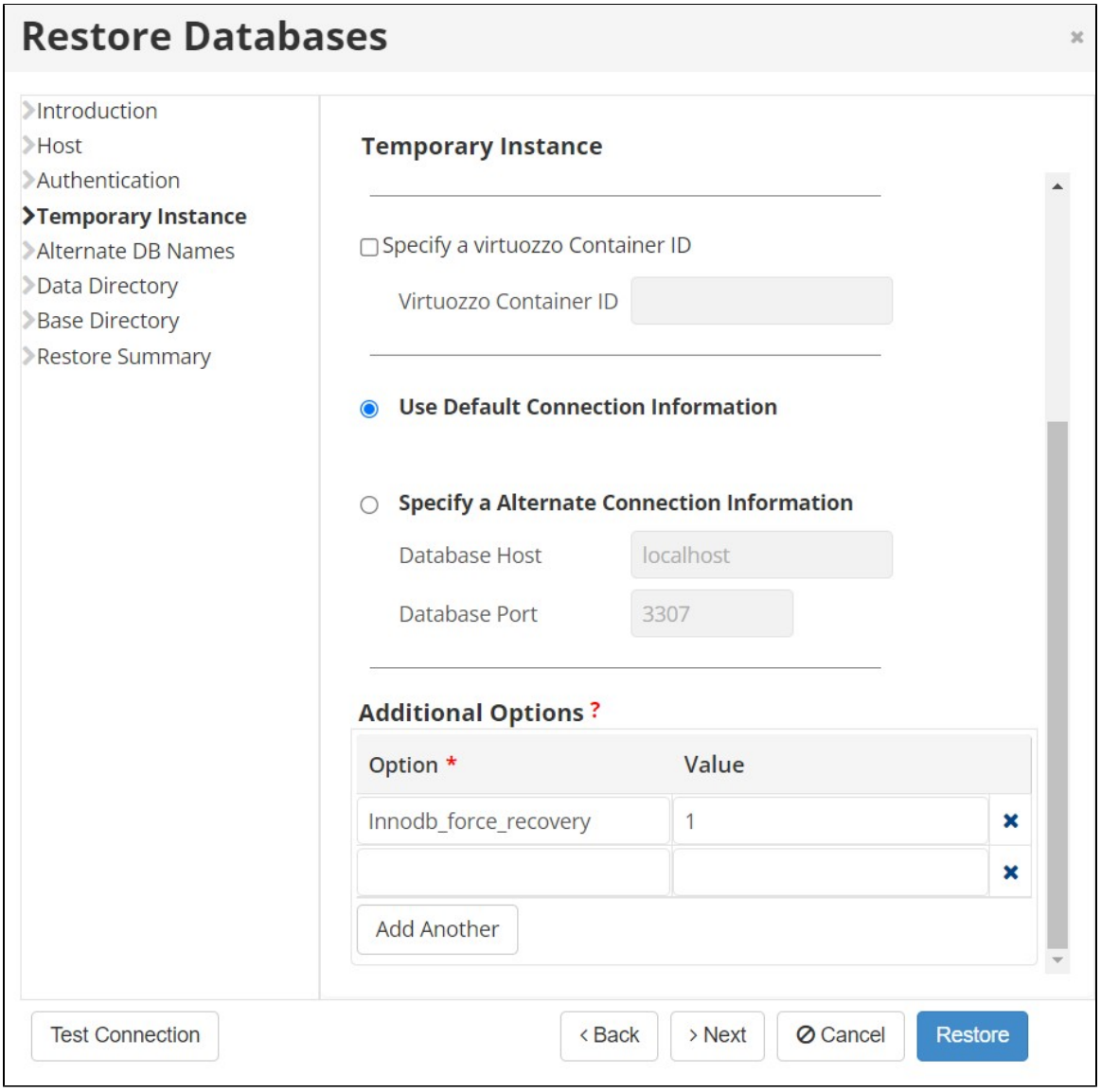
2. From the "Temporary Instance" page of the database restore wizard, locate the "Additional Options" section at the bottom.

3. In the "Option" column, enter innodb_force_recovery

4. In the "Value" column, enter a value between 1 and 6

5. Click "Add Another" to save your entry and add a new line

The finished entry should resemble this screenshot :



6. Select "Restore" to begin the restore immediately, or "Next" to continue forward to more options.