Backing up and Recovering Uptime Infrastructure Monitor

Contents

Overview

Performing regular backups of your Uptime Infrastructure Monitor DataStore is a highly recommended practice. The DataStore is the Uptime Infrastructure Monitor backbone and holds all configuration information and historical performance data. This article outlines five common methods for backing up your DataStore. It is recommended to choose at least one of these methods to back up your Uptime DataStore. For added safety, you can implement more than one of the options below in case one of the methods should falter. Only one of the options below can be used to restore a backup; the methods cannot be merged when restoring the backup.

> **Note**
>
> These backup methods are intended for a single Monitoring Station. Never attempt to merge two or more Monitoring Station configurations in a single database.

In addition to backing up the Uptime DataStore, it is also important to back up the Uptime configuration files and any customized files. Be sure to follow the recommendations in the Other Files and Directories to Backup section at the bottom of this article.

Option 1: Creating a tar or zip Archive

One backup option is to regularly create tar or zip archives of your `/datastore` directory. This is the most straightforward method but requires Uptime Infrastructure Monitor to be stopped during the backup period.

To use this method, simply include the `/datastore` directory in the tar/zip archive. If you need to recover your DataStore from a tar/zip archive, ensure that all Uptime Infrastructure Monitor services are stopped and that you delete the existing `/datastore` files before extracting your archive.

1. **Stop the Uptime Infrastructure Monitor services. -** Review Starting (or restarting) and Stopping Uptime Infrastructure Monitor

1. **Archive the datastore directory.**

   On a Windows system:

   a. Locate the Uptime Infrastructure Monitor install directory (default `C:\Program Files\uptime software\uptime`).
   b. Archive the `\datastore` directory (using an archiving tool such as Winzip).
   c. Move the zip archive to another system or drive.

   On a Unix system, enter the following commands:

   ```
   # cd /usr/local/uptime OR cd /opt/uptime
   (depending on the OS)
   # tar -cvf uptime_backup.tar datastore
   # gzip uptime_backup.tar
   ```

2. **Start the Uptime Infrastructure Monitor services**. **-** Review Starting (or restarting) and Stopping Uptime Infrastructure Monitor

Option 2: Using the mysqldump Tool

mysqldump is a free utility included with the standard Uptime Infrastructure Monitor MySQL database. This tool will export DataStore contents into a human readable .sql file that can later be used to recreate your DataStore. The commands below detail how to export both configuration and performance data using mysqldump.

> **Note**
>
> The following sample commands are provided with the assumption that you are executing them from the Uptime Infrastructure Monitor installation directory and that the default database access options have not been changed. If any parameters have changed, refer to your *uptime .conf* file for the correct port, user, password and database access information.

The standard format for the mysqldump command is as follows:

```
mysqldump --single-transaction -u[username] -p[password] -P[port #] --protocol=tcp [dbname]
```

By adding > *mybackup.sql* to the commands below, all mysqldump data will be directed to the *mybackup.sql* file. The *mybackup.sql* file name should be changed to a date-stamped file name for easy reference.

Exporting Your Entire DataStore

```
mysql/bin/mysqldump --single-transaction -uuptime -puptime -P3308 --protocol=TCP uptime > mybackup.sql
```

> **Note**
>
> The `[dbname]` variable may be `uptime_v4` if your database was created in Uptime Infrastructure Monitor 4.

Exporting Only Your Configuration Information

In version Uptime Infrastructure Monitor 7.2 or later there is the feature to generate a Problem Report with Configuration via the command line. One can schedule the execution of the command line Problem Report with a cron job or scheduled task.

```
prexport [-p] [-cN]
Options:
-p to dump last hour of data
-c dump historical configuration; N is number of months defaults to 3 months
```

Alternatively, here are the configuration tables:

```
mysql/bin/mysqldump -uuptime -puptime -P3308 --protocol=tcp
--ignore-table=uptime.erdc_decimal_data
--ignore-table=uptime.erdc_int_data
--ignore-table=uptime.erdc_string_data
--ignore-table=uptime.ranged_object_value
--ignore-table=uptime.performance_aggregate
--ignore-table=uptime.performance_cpu
--ignore-table=uptime.performance_disk
--ignore-table=uptime.performance_esx3_workload
--ignore-table=uptime.performance_fscap
--ignore-table=uptime.performance_lpar_workload
--ignore-table=uptime.performance_network
--ignore-table=uptime.performance_nrm
--ignore-table=uptime.performance_psinfo
--ignore-table=uptime.performance_sample
--ignore-table=uptime.performance_vxvol
--ignore-table=uptime.performance_who
--ignore-table=uptime.archive_delenda
uptime > mybackup.sql
```

Exporting Only Your Historical Performance Data

```
mysql/bin/mysqldump -uuptime -puptime -P3308 --protocol=tcp uptime
performance_aggregate
performance_cpu
performance_disk
performance_esx3_workload
performance_fscap
performance_lpar_workload
performance_network
performance_nrm
performance_psinfo
performance_sample
performance_vxvol
performance_who
erdc_decimal_data
erdc_int_data
erdc_string_data > mybackup.sql
```

Importing Your Backup Data

> ⚠ Before importing data, you must stop the Uptime Infrastructure Monitor services (see steps outlined in the Creating a tar or zip Archive section).

To import your backup data, run the following command:

```
mysql/bin/mysql -q -f -u uptime -puptime -P3308 --protocol=tcp uptime < mybackup.sql
```

This process will attempt to insert any non-duplicate data that is found in your *mybackup.sql* file. If you need to rebuild your database from scratch, run the *resetdb* utility before importing your backup file. This utility will erase ALL data in your existing DataStore; be absolutely sure that a full backup recovery is your best option before running this command.

```
resetdb really --nodata
```

Option 3: MySQL Replication

MySQL replication is the most complex backup method but is the most powerful for quick recovery. MySQL's built-in replication feature will maintain a completely up-to-date copy of your DataStore on another database instance (on the local system or a secondary server). This copy can be quickly set up to act as the primary DataStore in the event of a failure, or can be easily copied from the replication server to the primary server in the event of an outage.

Information on starting replication can be found at:

- MySQL Online Documentation: How to Set Up Replication.
- Uptime Infrastructure Monitor KB article: Setting up a Reporting Instance.

Option 4: Oracle Data Pump

If your DataStore is running on Oracle, you can use the Data Pump utility to export data from an Oracle database. Refer to the Oracle database utilities page for more information:

- Oracle Database Utilities

Option 5:SQL Server Backup

If your DataStore is running on Microsoft SQL Server, you can use the SQL Backup tool to export data. Refer to the Microsoft Developer Network for more information:

- SQL Server Backup

Other Files and Directories to Backup

The following files and directories listed in this section are not part of the DataStore but should also be backed up on a regular basis as well.

- <uptime_dir>/license.dat
- <uptime_dir>/UptimeDataCollector.ini (Windows OS)
- <uptime_dir>/uptime.conf
- <uptime_dir>/uptime.jcnf (Linux OS)
- <uptime_dir>/apache/conf/httpd.conf
- <uptime_dir>/apache/conf/php.ini (Linux OS)
- <uptime_dir>/apache/php/php.ini (Windows OS)
- <uptime_dir>/dashboards
- <uptime_dir>/gadgets
- <uptime_dir>/plugins
- <uptime_dir>/scripts
- <uptime_dir>/xml

If you have configured SSL browsing in Apache, don't forget to back up the cert files in the <uptime_dir>/apache/conf directory.

If you have added MIB files to the <uptime_dir>/mibs directory, you will want to back those custom files up as well.