

How to back up and restore

Overview

This How To covers the individual backup and restore processes for GroundWork Monitor databases and configuration settings including the Foundation database, JBoss framework, GroundWork Insight Reports, and Configuration (monarch). Additionally, the operational databases used by GroundWork Monitor are GWCollageDB (Foundation) which contains state and event data, and the Dashboard database which contains historical information used by Insight and Availability Reports.

CONTENTS

RELATED RESOURCES

- [System Maintenance How To's](#)

WAS THIS PAGE HELPFUL?

- [Leave Feedback](#)



Important Notes

- **Backup process** - The process of making a backup PostgreSQL dump can take several minutes, and during the time of backup the database may be unavailable for regular updates by the monitoring engine. Therefore it is not practical to perform such backups except when the monitoring system has little or no activity or when it can be guaranteed to be inactive. You can mitigate the effects of the interaction between backup and system use, by stopping GroundWork services before making the backup, and restarting them afterward, the `gwcollagedb` database is not open for updates. Feeders are quiescent and all messages that come in during the backup will be held in unprocessed log files and picked up afterward. The choice to do this should be based on experience with your installation. Periodically review the log files, `/usr/local/groundwork/foundation/container/logs`, for the time when backups occur to see if connections to PostgreSQL are refused.

An alternative is to use transaction logging on PostgreSQL. Transaction logging means the database engine writes every transaction to a log file as the transactions are completed. The value of this is that long transactions, like the backup operation, can coexist with other activities, and that recovery of the database is possible up to a moment in time, useful in case of system failure. The combination of the most recent full system backup (a postgresQL dump) and any log files produced since that backup can produce a clean recovery.

The consequence to customers is that a system failure which results in data loss on the monitoring server may compromise recovery of the running image and PostgreSQL databases. If in place recovery fails then the most recent backup image will be used. The loss of monitoring data (status and availability) and configuration changes to Monarch, users, dashboards, and the portal will be for the period between the most recent backup and the time that recovery is complete.

- **Notifying Users** - Before starting with any system maintenance you will need to notify GroundWork Monitor system users that the system will be unavailable for a short time and that they will need to log out otherwise they will receive error messages and will lose their current session. You can schedule a maintenance window to notify all system users.
- **Command not found** - If you received a message similar to command not found or no such file or directory you may need to set the environment variables for the shell by running the script `source /usr/local/groundwork/scripts/setenv.sh`. This works in each shell session and only in shells similar to the Bourne shell, like bash. It will not work in either csh or tcsh.
- **Required password** - You will need to have the PostgreSQL postgres user password available to perform various commands. This is the same database password that was used for the GroundWork Monitor installation or upgrade.
- **Example file name** - Throughout this document we refer to a `<timestamp>` format. For example `2017-01-06_14-06-10` could be the name of a backup directory or part of a backup file name representing the date and time of a backup.

1.0 Foundation Database (`gwcollagedb`)

This section covers backing up and restoring the Foundation database `gwcollagedb` and steps to delete the entire database and reset the initial settings.

To back up the Foundation database

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services with the following command:

```
/etc/init.d/groundwork stop gwservices
```

3. Create a backup sub-directory in the `groundwork` subdirectory:

**Backup directory**

This step is a one time action. Skip this step if you have already created a backup directory.

```
cd /usr/local/groundwork
```

```
mkdir backup
```

```
chown nagios:nagios backup
```

4. Change your working directory to backup with the following command:

```
cd /usr/local/groundwork/backup
```

5. Issue the following command to create a Foundation backup tar file:



The ``date +%F_%H.%M.%S`` in the command below will generate the timestamp dynamically. Back ticks (```) are used and not single quotes (`'`).

```
pg_dump -f gwcollagedb-`date +%F_%H.%M.%S`.sql.tar -F t -c -E LATIN1 gwcollagedb
```

6. Next, restart GroundWork services with the command:

```
/etc/init.d/groundwork start gwservices
```

To restore the Foundation database

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services with the command:

```
/etc/init.d/groundwork stop gwservices
```

3. Change your working directory to backup with the following command:

```
cd /usr/local/groundwork/backup
```

4. You should see a backup file in the following format:

```
gwcollagedb-<timestamp>.sql.tar
```

5. Restore the old database by entering the following command, (replacing `<timestamp>` with the actual timestamp):

```
pg_restore -d gwcollagedb -F t -c gwcollagedb-<timestamp>.sql.tar
```

6. Restart GroundWork services with the following command:

```
/etc/init.d/groundwork start gwservices
```

To delete and reset the Foundation database

To delete the entire Foundation database and reset the initial settings, use the following procedure.



CAUTION!

Before starting this procedure, please contact GroundWork Support regarding deleting and resetting the Foundation database and about resetting the Archive database.

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services with the following command:

```
/etc/init.d/groundwork stop gwservices
```

3. Set the environment variables for the shell:

```
source /usr/local/groundwork/scripts/setenv.sh
```

4. Change your working directory to `postgresql` with the following command. This directory contains all the database create SQL scripts:

```
cd /usr/local/groundwork/core/databases/postgresql
```

5. Drop and re-create the current `gwcollagedb` database by executing the following script:

```
psql -f create-fresh-gwcollagedb.sql postgres
```

6. Load the content of `GWCollageDB.sql` into the `gwcollagedb` database with the following command:

```
psql -f GWCollageDB.sql gwcollagedb postgres
```

7. Load all the seed scripts:

```
psql -f postgres-xtra-functions.sql gwcollagedb postgres
```

```
psql -f GWCollage-Console.sql gwcollagedb postgres
```

```
psql -f GWCollage-Metadata.sql gwcollagedb postgres
```

```
psql -f GWCollage-State.sql gwcollagedb postgres
```

```
psql -f GWCollage-Version.sql gwcollagedb postgres
```

```
psql -f GWCollage_PerformanceLabelData.sql gwcollagedb postgres
```

```
psql -f foundation-base-data.sql gwcollagedb postgres
```

```
psql -f nagios-properties.sql gwcollagedb postgres
```

```
psql -f system-properties.sql gwcollagedb postgres
```

```
psql -f snmp-properties.sql gwcollagedb postgres
```

```
psql -f syslog-seed.sql gwcollagedb postgres
```

- Restart GroundWork services with the command:

```
/etc/init.d/groundwork start gwservices
```

2.0 Archive Database (archive_gwcollagedb)

This section covers backing up and restoring the Archive database archive_gwcollagedb.

To back up the Archive database

- Open a terminal session and login as root user.
- Make sure no one is logged into the system, then stop GroundWork services:

```
/etc/init.d/groundwork stop gwservices
```

- Create a backup sub-directory:



Backup directory

This step is a one time action. Skip this step if you have already created a backup directory.

```
cd /usr/local/groundwork
```

```
mkdir backup
```

```
chown nagios:nagios backup
```

- Change your working directory to backup:

```
cd /usr/local/groundwork/backup
```

- Issue the following command to create a Archive backup tar file:



The ``date +%F_%H.%M.%S`` in the command below will generate the timestamp dynamically. Back ticks (```) are used and not single quotes (`'`).

```
pg_dump -f archive_gwcollagedb-`date +%F_%H.%M.%S`.sql.tar -F t -c -E LATIN1 archive_gwcollagedb
```

- Next, restart GroundWork services:

```
/etc/init.d/groundwork start gwservices
```

To restore the Archive database

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services:

```
/etc/init.d/groundwork stop gwservices
```

3. Change your working directory to backup:

```
cd /usr/local/groundwork/backup
```

4. Listing the files, you should see a backup file in the following format:

```
archive_gwcollagedb-<timestamp>.sql.tar
```

5. Restore the old database by entering the following command, (replacing `<timestamp>` with the actual timestamp):

```
pg_restore -d archive_gwcollagedb -F t -c archive_gwcollagedb-<timestamp>.sql.tar
```

6. Restart GroundWork services:

```
/etc/init.d/groundwork start gwservices
```

3.0 JBoss (`jboss-idm` and `jboss-jcr`)

This section covers the backup and restore procedure for the JBoss Portal databases `jboss-idm` and `jboss-jcr` which contain information for system users, roles, applications and permissions.

To back up the JBoss databases

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services and the Apache `httpd` daemon with the following commands:

```
/etc/init.d/groundwork stop gwservices
```

```
/etc/init.d/groundwork stop apache
```

3. Change your working directory to backup with the following command:

```
cd /usr/local/groundwork/backup
```

4. Issue the following commands to create backup tar files:



The ``date +%F_%H.%M.%S`` in the command below will generate the datestamp dynamically. Back ticks (```) are used and not single quotes (`'`).

```
pg_dump -f jboss-idm-`date +%F_%H.%M.%S`.sql.tar -F t -c -E LATIN1 jboss-idm
```

```
pg_dump -f jboss-jcr-`date +%F_%H.%M.%S`.sql.tar -F t -c -E LATIN1 jboss-jcr
```

5. Restart GroundWork services and the Apache `httpd` daemon with the following commands:

```
/etc/init.d/groundwork start gwservices
```

```
/etc/init.d/groundwork start apache
```

To restore the JBoss databases

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services and the Apache `httpd` daemon with the following commands:

```
/etc/init.d/groundwork stop gwservices
```

```
/etc/init.d/groundwork stop apache
```

3. Change your working directory to backup with the command:

```
cd /usr/local/groundwork/backup
```

4. You should see backup files in the following format:

```
jboss-idm-<timestamp>.sql.tar
```

```
jboss-jcr-<timestamp>.sql.tar
```

5. Restore the old databases by entering the following commands, (replacing `<timestamp>` with the actual timestamp):

```
pg_restore -d jboss-jcr -F t -c jboss-jcr-<timestamp>.sql.tar
```

```
pg_restore -d jboss-idm -F t -c jboss-idm-<timestamp>.sql.tar
```

6. Restart GroundWork services and the Apache `httpd` daemon with the following commands:

```
/etc/init.d/groundwork start gwservices
```

```
/etc/init.d/groundwork start apache
```

4.0 Insight Reports (dashboard)

This section covers backing up and restoring the Insight Reports database `dashboard` and steps to delete the entire dashboard database and reset the initial settings.

To back up the Insight Reports database

You can manually back up the dashboard database at a command line with the following steps:

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services and the Apache `httpd` daemon with the following commands:

```
/etc/init.d/groundwork stop gwservices
```

```
/etc/init.d/groundwork stop apache
```

3. Change your working directory to backup with the following command:

```
cd /usr/local/groundwork/backup
```

4. Issue the following command to create a backup tar file:



The ``date +%F_%H.%M.%S`` in the command below will generate the timestamp dynamically. Back ticks (```) are used and not single quotes (`'`).

```
pg_dump -f dashboard-`date +%F_%H.%M.%S`.sql.tar -F t -c -E LATIN1 dashboard
```

5. Restart GroundWork services and the Apache httpd daemon with the following commands:

```
/etc/init.d/groundwork start gwservices
```

```
/etc/init.d/groundwork start apache
```

To restore the Insight Reports database

To restore a database that was previously backed up, use the following procedure.

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services and the Apache httpd daemon with the following commands:

```
/etc/init.d/groundwork stop gwservices
```

```
/etc/init.d/groundwork stop apache
```

3. Change your working directory to backup with the following command:

```
cd /usr/local/groundwork/backup
```

4. Restore the old database by entering the following command, (replacing `<timestamp>` with the actual timestamp):

```
pg_restore -d dashboard -F t -c dashboard-<timestamp>.sql.tar
```

5. Restart GroundWork services and the Apache httpd daemon with the following commands:

```
/etc/init.d/groundwork start gwservices
```

```
/etc/init.d/groundwork start apache
```

To delete and reset the Insight Reports database

To delete the entire dashboard database and reset the initial settings, use the following procedure.



WARNING!

Performing this procedure will cause data loss including all historical data used for Insight Reports (Alerts, Notifications, and Outages).

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services and the Apache `httpd` daemon with the following commands:

```
/etc/init.d/groundwork stop gwservices
```

```
/etc/init.d/groundwork stop apache
```

3. Change the working directory with the following command. This directory contains all the database create SQL scripts:

```
cd /usr/local/groundwork/core/databases/postgresql
```

4. Drop and re-create the current dashboard database by executing the following script:

```
psql -f create-fresh-insightreport.sql postgres
```

5. Load the content of `dashboard-db.sql` into the dashboard database with the command:

```
psql -f dashboard-db.sql dashboard postgres
```

6. Restart GroundWork services and the Apache `httpd` daemon with the following commands:

```
/etc/init.d/groundwork start gwservices
```

```
/etc/init.d/groundwork start apache
```

5.0 Configuration Database (monarch)

A backup of the configuration database is auto-created after a user performs the commit process within the user interface and a configuration backup can also be manually submitted. This section focuses on the manual back up and how to restore the PostgreSQL configuration database `monarch` (via the user interface and at the command line).

To back up manually

1. Go to *Configuration > Control* and select *Backup and restore*.
2. Before you backup, you have the option to add an *Annotation* which can be referenced when restoring a backup.
3. Next, you have the option to *Lock* the backup which allows it to be saved *indefinitely*.
 - Unlocked backups are subject to being automatically deleted when too many accumulate. The number of old unlocked backups to be retained is set by the *Configuration > Control > Setup > Max unlocked backups* option.
 - A locked backup can be set when a backup is first created, and unlocked thereafter.
 - Unlocked backups may also be manually deleted.
4. Select *Back up* to overwrite the active *Nagios* configuration, restart *Nagios*, and create a backup. Select *Abort* if you do not want to follow through with the commit process.

Figure: Backup



- You should see a *Backup complete* message at the top of the screen indicating the location of where the files are backed up, and the backup instance added under *Manage existing backups* at the bottom of the screen.



To restore and manage existing backups

In the same screen where manual backups are performed you can also manage existing backups including restoring, locking, unlocking and deleting. When performing a restore make sure to take your time and read the various explanations as restoring a backup will destroy your existing setup.

- Open the *Backup and Restore* screen (*Configuration > Control > Backup and restore*) where you will find a list of available backups.
- To manage an existing backup, click on the box next to the corresponding *Backup Date/Time*.
- Then select the desired option (*Restore, Lock, Unlock, or Delete*).

Figure: Select option for existing backup



WARNING!
If restoring a backup, be very sure this is what you want to do as doing so will first destroy your existing setup.

- After selecting the *Restore* button you will be provided a *Restore command* to be run at the command line.

Figure: Restore selected backup



To back up the configuration database from the command line

This section provides steps to backup and restore the configuration database from the command line using the `pg_dump` and `pg_restore` commands. However, it is recommended that you use the **Backup and Restore from the User Interface** process from above. The UI process allows for backups to be annotated for future reference which describes why a back up was completed, and you are also able to lock a backup which helps in managing specific restores. The following process does not offer these options. You can manually back up the monarch database at a command line with the following steps:

- Open a terminal session and login as `root` user, then change to `nagios` user (`su - nagios`).
- Make sure no one is logged into the system, then stop GroundWork services and the Apache `httpd` daemon with the following commands:

```
/etc/init.d/groundwork stop gwservices
```

```
/etc/init.d/groundwork stop apache
```

- Change your working directory to backup with the following command:

```
cd /usr/local/groundwork/core/monarch/backup
```

4. Create a subdirectory using the current timestamp, this will hold the backup tar file:

 The ``date +%F_%H.%M.%S`` in the command below will generate the datestamp dynamically. Back ticks (```) are used and not single quotes (`'`).

```
mkdir `date +%F_%H.%M.%S`
```

5. Change to the new directory, (replacing `<timestamp>` with the actual timestamp):

```
cd <timestamp>
```

6. Issue the following command to create a backup tar file, (replacing `<timestamp>` with the actual timestamp text):

 You may need to set the shell's PATH variable to include the location where the `pg_dump` command resides, see reference tips above)

```
pg_dump -f monarch-<timestamp>.sql.tar -F t -c -E LATIN1 monarch
```

7. Restart GroundWork services and the Apache httpd daemon with the following commands:

```
s/etc/init.d/groundwork start gwservices
```

```
/etc/init.d/groundwork start apache
```

To restore the configuration database from the command line

1. Open a terminal session and login as `root` user, then change to `nagios` user (`su - nagios`).
2. Make sure no one is logged into the system, then stop GroundWork services and the Apache httpd daemon with the following commands:

```
/etc/init.d/groundwork stop gwservices
```

```
/etc/init.d/groundwork stop apache
```

3. Change the working directory to the directory where the monarch backup file resides, (replacing `<timestamp>` with the actual timestamp):

```
cd /usr/local/groundwork/core/monarch/backup/<timestamp>
```

4. List the subdirectory contents and you should see a backup file in the following format:

```
monarch-<timestamp>.sql.tar
```

5. Restore the old database by entering the following command at the system prompt, (replacing `<timestamp>` with the actual timestamp):

```
pg_restore -d monarch -F t -c monarch-<timestamp>.sql.tar
```

6. Restart GroundWork services and the Apache httpd daemon with the following commands:

```
/etc/init.d/groundwork start gwservices
```

```
/etc/init.d/groundwork start apache
```

7. To confirm the restored configuration, in GroundWork Monitor commit the restored monarch database: *Configuration > Control > Pre flight test*. And then, *Commit > Commit*.

6.0 SLA Report Database (slareport)

This section covers backing up and restoring the SLA Report database `slareport`.

To back up the SLA Report database

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services:

```
/etc/init.d/groundwork stop gwservices
```

3. Create a backup sub-directory:



Backup directory

This step is a one time action. Skip this step if you have already created a backup directory.

```
cd /usr/local/groundwork
```

```
mkdir backup
```

```
chown nagios:nagios backup
```

4. Change your working directory to backup:

```
cd /usr/local/groundwork/backup
```

5. Issue the following command to create a SLA Report backup tar file:



The ``date +%F_%H.%M.%S`` in the command below will generate the timestamp dynamically. Back ticks (```) are used and not single quotes (`'`).

```
pg_dump -f slareport-`date +%F_%H.%M.%S`.sql.tar -F t -c -E LATIN1 slareport
```

6. Next, restart GroundWork services:

```
/etc/init.d/groundwork start gwservices
```

To restore the SLA database

1. Open a terminal session and login as `root` user.
2. Make sure no one is logged into the system, then stop GroundWork services:

```
/etc/init.d/groundwork stop gwservices
```

3. Change your working directory to backup:

```
cd /usr/local/groundwork/backup
```

4. Listing the files, you should see a backup file in the following format:

```
slareport-<timestamp>.sql.tar
```

5. Restore the old database by entering the following command, (replacing <timestamp> with the actual timestamp):

```
pg_restore -d slareport -F t -c slareport-<timestamp>.sql.tar
```

6. Restart GroundWork services:

```
/etc/init.d/groundwork start gwservices
```

7.0 Data Files Backup and Restore

To back up data files

These are RRD files with value for long-term analysis. These files are updated in operational mode frequently by the monitoring process. A successful backup means capturing the changes without getting partially written files. The decision to capture these must be driven by requirement.

The GroundWork Monitor controlled active or passive checks produce RRD files; In order to back them up successfully we recommend integrating the backup with the file based performance data processing (for Nagios). This requires at the least halting the generating process Nagios and CloudHub. Following this, perform a backup. Then restart the appropriate daemon or cron job.

To restore data files

As with making the backup, care must be exercised to avoid collision between the generating process and the restore process. Stop the daemons, perform the restore, start the daemons.

8.0 Automating the Backup Process

This section covers automating the backup process. Depending on the size of the database(s) you may choose to make the backups more frequently.

You can run a cron job to perform a backup of each of the PostgreSQL databases using the syntax in the example replacing the <database name> variable for each database. This produces a point in time image of each database in a single backup file with a timestamp.



- Note the section ``/bin/date -Iminutes`` is back ticks (```) not single quotes (`'`). The back ticks embed the date within the file name.
- The "find" statement at the end will purge any backups older than 5 days

```
30 */23 * * * /usr/local/groundwork/postgresql/bin/pg_dump -f /backup/postgres/<data base>`/bin/date -Iminutes`.sql.tar -F t -c -E LATIN1 <data base>; /usr/bin/find /backup/postgres -mtime +5 -exec /bin/rm -rf {} \;
```

In the normal course of operation when using Monarch, always select to make a backup before performing a Commit. The backups created are functionally equivalent to that produced in the cron jobs above and have the advantage of spanning all changes ever made (assuming you keep the entire list of backups).

The location of the target directory in the examples is an important choice. It should be a share on a resource that is:

- Not associated with the machine or machines being backed up

- An offsite copy should be considered, produced from the dump files by writing to removable media (/backup/postgres in these examples)
- Regularly scheduled rotation of the offsite media should be set up in accordance with your company's data retention and recovery policy guidelines

Foundation database

Syntax for the Foundation database gwcollagedb cron automation:

```
30 */23 * * * /usr/local/groundwork/postgresql/bin/pg_dump -f /backup/postgres/gwcollagedb_`/bin/date
-Iminutes`.sql.tar -F t -c -E LATIN1 gwcollagedb; /usr/bin/find /backup/postgres -mtime +5 -exec
/bin/rm -rf {} \;
```

JBoss databases

Syntax for the JBoss databases jboss-idm and jboss-jcr cron automation:

```
30 */23 * * * /usr/local/groundwork/postgresql/bin/pg_dump -f /backup/postgres/jboss-idm_`/bin/date
-Iminutes`.sql.tar -F t -c -E LATIN1 jboss-idm; /usr/bin/find /backup/postgres -mtime +5 -exec /bin/rm
-rf {} \;
```

```
30 */23 * * * /usr/local/groundwork/postgresql/bin/pg_dump -f /backup/postgres/jboss-jcr_`/bin/date
-Iminutes`.sql.tar -F t -c -E LATIN1 jboss-jcr; /usr/bin/find /backup/postgres -mtime +5 -exec /bin/rm
-rf {} \;
```

Insight Reports database

Syntax for the Insight Reports database dashboard cron automation:

```
30 */23 * * * /usr/local/groundwork/postgresql/bin/pg_dump -f /backup/postgres/dashboard_`/bin/date
-Iminutes`.sql.tar -F t -c -E LATIN1 dashboard; /usr/bin/find /backup/postgres -mtime +5 -exec /bin/rm
-rf {} \;
```

Configuration database

Syntax for the Configuration database monarch cron automation:

```
30 */23 * * * /usr/local/groundwork/postgresql/bin/pg_dump -f /backup/postgres/monarch_`/bin/date
-Iminutes`.sql.tar -F t -c -E LATIN1 monarch; /usr/bin/find /backup/postgres -mtime +5 -exec /bin/rm
-rf {} \;
```