Holland Documentation

Release 1.0.4

Andrew Garner

July 23, 2010

CONTENTS

CHAPTER

ONE

TABLE OF CONTENTS

1.1 Introduction to Holland

Holland is an Open Source backup framework originally developed by Rackspace and written in Python. It's goal is to help facilitate backing up databases with greater configurability, consistency, and ease. Holland currently focuses on MySQL, however future development will include other database platforms and even non-database related applications. Because of it's plugin structure, Holland can be used to backup anything you want by whatever means you want.

Plugins are provided as Python "eggs" - zip files with Python modules and extra metadata information.

1.1.1 Dependencies

The core Holland framework has the following dependencies (available on any remotely modern Linux distribution):

- Python >= 2.3
- pkg_resources
- python-setuptools
- MySQLdb

Additionally, the Maatkit plugin requires:

- Maatkit
- Perl-TermReadKey

For Red-Hat Enterprise Linux 5, all dependencies, except for Maatkti, are available directly from the base channels. Red-Hat Enterprise Linux 4, EPEL is required for python-setuptools.

Note that other providers may have additional dependency requirements.

1.2 Holland Command-Line Reference

Here are the commands available from the 'holland' command-line tool:

1.2.1 help (h)

Usage: `holland help <command>`

Provides basic information about the provided command. If no command is provided, it displays global help instead.

1.2.2 backup (bk)

Usage: `holland backup [backup-set1, backup-set2, ..., backup-setN]`

Runs the backup operation. If no backup-sets are specified, all active backup-sets (those defined in the 'backupsets' variable in holland.conf) are backed up.

One or more backup-sets can be specified directly, in which case only those backup-sets are backed up.

Additional Command Line Arguments:

`--dry-run` (`-n`): Can be used here to simulate, but not actually run, a backup. This should be used when troubleshooting a particular error before trying to run a real backup.

`--no-lock` (`-f`): Normally, only one instance of Holland can run at any given time using lock-files. Using this flag causes the lock-files to be ignored. This has some very clear use-cases but otherwise be mindful of using this setting as it can cause backups to fail in some cases.

`--abort-immediately`: abort on the first backup-set that fails (assuming multiple backupsets were specified)

Examples:

`# holland bk --dry-run weekly`: Attempts a dry-run of the weekly backup-set.

`# holland bk --no-lock --abort-immediately`: Attempts a backup of all the default backup-sets ignoring locks and aborting immediately if one of the backup-sets fails.

1.2.3 list-backups (lb)

Usage: `holland list-backups`

Provides extended information about available backups.

1.2.4 list-plugins (lp)

Usage: `holland list-plugins`

Lists all the available (installed) plugins available to Holland.

1.2.5 mk-config (mc)

Usage: `holland mk-config <provider>`

Generates a template backup-set for a particular provider (such as mysqldump). By default, the output is sent to standard out but can be copied to a file, either by using the `--file`, `--edit`, or `-name` options (see below).

Additional Command Line Arguments:

`--edit`: Load the file into the system text-editor for further modifications.

`--file=FILE` (`-f`): Write the output directly to provided file.

`--name=NAME`: Creates a backup-set usable in Holland, which basically means that a file is created of the provided name under the backup-set directory.

`--provider`: Indicates that the default provider configuration should be outputted instead. This is really only used when creating a provider config specifically - it should not be used for backup-sets.

Examples:

`# holland mk-config mysql-lvm > mysql-lvm.conf`: Output the default configuration for MySQL-LVM backups and write the contents out to mysql-lvm.conf in the current working directory.

`# holland mc mysqldump --name=Bob --edit`: Create a backup-set using the mysqldump provider named Bob and allow interactive editing of the backup-set before saving the file.

1.2.6 purge (pg)

Usage: `holland purge <backup-set>/<backup-id>`

Purges old backups by specifying the backup-set name and set-id.

For example: `# holland purge mybackups/20090502_155438`: Purge one of the backups taken on May 2nd, 2009 from the mybackups backup-set.

1.3 Usage and Implementation Overview

Because Holland is very pluggable, it may first seem a bit confusing when it comes to configuring Holland to do something useful. Out of the box, Holland is designed to backup MySQL databases using the mysqldump provider. This is the simplest setup, and may be sufficient for most people. However, others may wish to have more fine-grained control over their backups and/or use another method other than mysqldump.

For instance, one can configure a backup set to backup certain databases using Maatkit, others using the mysqlhotcopy (not to be confused with the utility that comes with MySQL of the same name), etc. All this is done by a mix of providers and backup-sets.

1.3.1 Backup-Sets

Each backup-set implements exactly one provider and will inherit the default values of that provider. These values can be overridden to adjust the behavior of the backup set. This includes defining what databases or tables to include (or exclude) in the backup, the type of compression used (if any), what locking method to use, among other things.

1.3.2 Providers

Providers essentially provide a backup service for use in a backup set. As of Holland 0.5, there are 5 providers:

- mysqldump

  Uses the `mysqldump` utility to backup databases.

- MySQL Hot-Copy

  Backs up the actual files that comprise one's MyISAM tables. While this does require locking the tables involved in the backup, this can often result in a much faster backup over using `mysqldump`. It does NOT handle transactional tables (such as InnoDB). Note that this provider has not been extensively tested and is more of a proof of concept.

- Maatkit

Uses the `mk-parallel-dump` utility which, under certain circumstances, is faster than a standard `mysqldump`. Note that this provider is really more of a proof of concept. It should normally not be used for production cases.

- Example

  This is used solely as a template for designing providers. It otherwise does nothing.

- MySQL + LVM

  Backup MySQL databases using LVM snapshots which allows for near lockless or fully lockless (when transactional engines are used) backups.

There are also plans to eventually implement the following providers:

- MySQL Cluster

  MySQL Cluster already offers a simple way of producing lockless backups, however these methods do not often include table-creation data, routines, etc. In addition, backups from the Cluster are stored on each of the Data Nodes.

  Therefore, this provider would facilitate backing up and aggregating the Cluster backups into one centralized location, as well as backing up the periphery data necessary should a full restore be necessary.

As Holland is a framework, it can actually backup most anything as long as there is a provider for it. This includes things that have nothing to do with databases. The idea is to present an easy to use and clear method of backing up and restoring backups no matter the source.

1.4 Holland Config Files

By default, Holland's configuration files reside in /etc/holland. The main configuration file is holland.conf, however there are a number of other configuration files for configuring default settings for providers and for configuring backup sets.

Each configuration file has one ore more sections, defined by square brackets Underneath each section, one or more configuration option can be specified. These options are in a standard "option = value" format. Comments are prefixed by the # sign.

Note that many settings have default values and, as a result, can either be commented out or omitted entirely.

1.4.1 holland.conf - main config

The main configuration file (usually /etc/holland/holland.conf) defines both global settings as well as the active backup sets. It is divided into two sections *[holland]* and *[logging]*.

[holland]

plugin-dirs
: Defines where the plugins can be found. This can be a comma-separated list but usually does not need to be modified.

backup_directory
: Top-level directory where backups are held.

backupsets
A comma-separated list of all the backup sets Holland should backup. Each backup set is defined in `/etc/holland/backupsets/<name>.conf` by default.

umask
Sets the umask of the resulting backup files.

path
Defines a path for holland and its spawned processes

[logging]

filename
The log file itself.

level
Sets the verbosity of Holland's logging process. Available options are `debug`, `info`, `warning`, `error`, and `critical`

1.4.2 Provider Configs

These files control the global settings / defaults for the providers used by the backup-sets. Many of these global settings can be overridden if defined in a backup-set. Note that each provider's configuration file should begin with `[provider-name]`.

Example Provider Configuration [example]

There are currently no configuration options for the example provider.

mysqldump Provider Configuration [mysqldump]

mysql-binpath = /path/to/mysql/bin

Defines the location of the MySQL binary utilities. If not provided, Holland will use whatever is in the path.

lock-method = flush-lock | lock-tables | single-transaction | auto-detect | none

Defines which lock method to use. By default, auto-detect will be used.

- flush-lock

  flush-lock will place a global lock on all tables involved in the backup regardless of whether or not they are in the backup-set. If file-per-database is enabled, then flush-lock will lock all tables for every database being backed up. In other words, this option may not make much sense when using file-per-database.

- lock-tables

  lock-tables will lock all tables involved in the backup. If file-per-database is enabled, then lock-tables will only lock all the tables associated with that database.

- single-transaction

Forces the use of `--single-transaction` which enabled semi-transparent backups of transactional tables. Forcing this can cause inconsistencies with non-transactional tables, however. While non-transactional tables will still lock, they will only lock when they are actually being backed up. **Use this setting with extreme caution when backing non-transactional tables.**

- auto-detect

  Let Holland decide which option to use by checking to see if a database or backup-set only contains transactional tables. If so, `--single-transaction` will be used. Otherwise, `--lock-tables` will be used.

- none

  Does absolutely no explicit locking when backing up the databases or backup-set. This should only be used when backing up a slave and only after the slave has been turned off (ie, this can be used with the **stop-slave** option).

dump-routines = yes | no

Whether or not to backup routines in the backup set directly. Routines are stored in the 'mysql' database, but it can sometimes be convenient to include them in a backup-set directly.

dump-events = yes | no

Whether or not to dump events explicitly. Like routines, events are stored in the 'mysql' database. Nonetheless, it can sometimes be convenient to include them in the backup-set directly.

Note: This feature requires MySQL 5.1 or later.

stop-slave = yes | no

This is useful only when running Holland on a MySQL slave. Instructs Holland to suspend slave services on the server prior to running the backup. Suspending the slave does not change the backups, but does prevent the slave from spooling up relay logs. The default is not to suspend the slave (if applicable).

bin-log-position = yes | no

Record the binary log name and position at the time of the backup.

Note that if both 'stop-slave' and 'bin-log-position' are enabled, Holland will grab the master binary log name and position at the time of the backup which can be useful in using the backup to create slaves or for point in time recovery using the master's binary log. This information is found within the 'backup.conf' file located in the backup-set destination directory (/var/spool/holland/<backup-set>/<backup> by default). For example:

```
[mysql:replication]
slave_master_log_pos = 4512
slave_master_log_file = 260792-mmm-agent1-bin-log.000001
```

flush-logs = yes | no

Whether or not to run FLUSH LOGS in MySQL with the backup. When FLUSH LOGS is actually executed depends on which if database filtering is being used and whether or not file-per-database is enabled. Generally speaking, it does not make sense to use flush-logs with file-per-database since the binary logs will not be consistent with the backup.

file-per-database = yes | no

Whether or not to split up each database into its own file. Note that it can be more consistent an efficient to backup all databases into one file, however this means that restore a single database can be difficult if multiple databases are defined in the backup set.

additional-options = <mysqldump arguments>

Can optionally specify additional options directly to `mysqldump` if there is no native Holland option for it.

Database and Table filtering

Database and Table filtering **databases** = <glob>

exclude-databases = <glob>

tables = <glob>

exclude-tables = <glob>

The above options accepts GLOBs in comma-separated lists. Multiple filtering options can be specified. When filtering on tables, be sure to include both the database and table name.

Be careful with quotes. Normally these are not needed, but when quotes are necessary, be sure to only quote each filtering statement, as opposed to putting quotes around all statements.

Below are a few examples of how these can be applied:

Default (backup everything):

```
databases = *
tables = *
```

Using database inclusion and exclusions:

```
databases = drupal*, smf_forum,
exclude-databases = drupal5
```

Including Tables:

```
tables = phpBB.sucks, drupal6.node*, smf_forum.*
```

Excluding Tables:

```
exclude-tables = mydb.uselesstable1, x_cart.*, *.sessions
```

[compression]

[compression] Specify various compression settings, such as compression utility, compression level, etc.

method = gzip | pigz | bzip | lzop | lzma

Define which compression method to use. Note that `lzop` and `lzma` may not be available on every system and may need to be compiled / installed.

inline = yes | no

Whether or not to pipe the output of mysqldump into the compression utility. Enabling this is recommended since it usually only marginally impacts performance, particularly when using a lower compression level.

level = 0-9

Specify the compression ratio. The lower the number, the lower the compression ratio, but the faster the backup will take. Generally, setting the lever to 1 or 2 results in favorable compression of textual data and is noticeably faster than the higher levels. Setting the level to 0 effectively disables compression.

bin-path = <full path to utility>

This only needs to be defined if the compression utility is not in the usual places or not in the system path.

[mysql:client]

MySQL connection info [mysql:client] These are optional and, if left undefined, Holland will try to login using the standard .my.cnf conventions.

user = <user>

The user to connect to MySQL as.

password = <password>

The password for the MySQL user

socket = <socket>

The socket file to connect to MySQL with.

host = <host>

This would be used for connecting to MySQL remotely.

port = <port>

Used if MySQL is running on a port other than 3306.

MySQL Hot Copy Provider Configuration [mysqlhotcopy]

Reminder: This provider is really only for MyISAM tables. It does NOT handle transactional tables (such as InnoDB).

mysql-binpath = /path/to/mysql/bin

Defines the location of the MySQL binary utilities. If not provided, Holland will use whatever is in the path.

lock-method = lock-tables | flush-lock

Defines the lock method (`flush-lock` or `lock-tables`).

- flush-lock

  Will issue a `FLUSH TABLES WITH READ LOCK` prior to the backup. This is basically a global lock which will block writes to the database for the duration of the backup.

- lock-tables

  Issues a lock-tables for each database, or for the entire backup set.

If this option is not specified, `lock-tables` is used.

partial-indexes = yes | no

If set to true, Holland will only backup the first 2k of a .MYI file, which can save quite a bit of space. This requires repairing the tables to rebuild the remainder of the index on a restore, however.

archive-method = dir | zip | tar

Which method to use when archiving the files. `dir` creates a standard directory `zip` creates a ZIP file `tar` creates a tar.gz file

stop-slave = yes | no Whether to stop the slave before commencing with the backup

bin-log-position = yes | no Whether to record the binary log name and position at the time of the backup.

Database and Table filtering

Database and Table filtering **databases** = <glob>

exclude-databases = <glob>

tables = <glob>

exclude-tables = <glob>

The above options accepts GLOBs in comma-separated lists. Multiple filtering options can be specified. When filtering on tables, be sure to include both the database and table name.

Be careful with quotes. Normally these are not needed, but when quotes are necessary, be sure to only quote each filtering statement, as opposed to putting quotes around all statements.

Below are a few examples of how these can be applied:

Default (backup everything):

```
databases = *
tables = *
```

Using database inclusion and exclusions:

```
databases = drupal*, smf_forum,
exclude-databases = drupal5
```

Including Tables:

```
tables = phpBB.sucks, drupal6.node*, smf_forum.*
```

Excluding Tables:

```
exclude-tables = mydb.uselesstable1, x_cart.*, *.sessions
```

[mysql:client]

MySQL connection info [mysql:client] These are optional and, if left undefined, Holland will try to login using the standard .my.cnf conventions.

user = <user>

The user to connect to MySQL as.

password = <password>

The password for the MySQL user

socket = <socket>

The socket file to connect to MySQL with.

host = <host>

This would be used for connecting to MySQL remotely.

port = <port>

Used if MySQL is running on a port other than 3306.

Maatkit Provider Configuration [maatkit]

Some of these options relate directly to the Maatkit `mk-parallel-dump` utility. For more information on these options and the utility itself, check out the Maatkit `mk-parallel-dump` documentation. Note that this provider is a proof of concept and should normally not be used in production.

lock-method = flush-lock | lock-tables

Defines the lock method (`flush-lock` or `lock-tables`).

- flush-lock

  Will issue a `FLUSH TABLES WITH READ LOCK` prior to the backup. This is basically a global lock which will block writes to the database for the duration of the backup.

- lock-tables

  Issues a lock-tables for each database, or for the entire backup set.

If this option is not specified, `flush-lock` is used.

biggestfirst = yes | no

Backup the biggest tables first.

binlogpos = yes | no

Record the binary log name and position at the time of the backup.

charset = [character-set]

Sets the default character set.

chunksize = # [M | G | k]

Specifies the number of rows or size that each backup file will be. Specifying a G, M or k after the number will cause Holland to split the chunks by size. Otherwise, it will split by rows.

ignoreengine = [engine1, engine2, ..., engineN]

Skips tables which match any of the provided engines. This can be useful in cases where one might not always want to backup certain table-types. One example would be tables which use the ARCHIVE engine, since these tables may not need to be backed up near as often.

numthread = #

If defined, it will create the provided number of threads for the backup operation. If left blank, it will count the number of times 'processor' appears in /proc/cpuinfo and use that.

stopslave = yes | no

This is useful only when running Holland on a MySQL slave. Instructs Holland to suspend slave services on the server prior to running the backup. Suspending the slave does not change the backups, but does prevent the slave from spooling up relay logs. The default is not to suspend the slave (if applicable).

flushlog = yes | no

If enabled, runs a FLUSH LOGS prior to getting the binary log name and position. If left undefined, the default is not to run a FLUSH LOGS. Note if using this with setperdb, be aware that FLUSH LOGS can get run quite often, which can cover up error messages and create a large number of binary logs needlessly.

gzip = yes | no

Whether or not to compress the backups using gzip. Currently gzip is the only option because it is handled by 'mk-parallel-dump' and not by Holland.

setperdb = yes | no

Whether or not to backup each database into its own file.

Database and table filtering

Maatkit's table filtering works slightly different from the other providers as it uses the same syntax that the *mk-parallel-dump* utility uses.

databases = [db1, db2, ..., dbN]

If undefined, will dump all databases.

ignoredb = [db1, db2, ..., dbN]

Ignore one or more databases from being backed up.

dbregex = [Perl regex]

Regular expression matching on databases using the Perl regular expression syntax.

tables = [table1, table2, ..., tableN]

List specific tables to backup. Note that it is possible to specify the database and table using dotted notation (database.table).

tblregex = [Perl regex]

Regular expression matching on tables using the Perl regular expression syntax.

ignoretbl = [table1, table2, ..., tableN]

A list of tables to exclude from the backup. Like the **tables** option, use of the database.table syntax is allowed if preferred.

[mysql:client]

MySQL connection info [mysql:client] These are optional and, if left undefined, Holland will try to login using the standard .my.cnf conventions.

user = <user>

The user to connect to MySQL as.

password = <password>

The password for the MySQL user

socket = <socket>

The socket file to connect to MySQL with.

host = <host>

This would be used for connecting to MySQL remotely.

port = <port>

Used if MySQL is running on a port other than 3306.

MySQL LVM Provider Configuration [mysql-lvm]

snapshot-size = <size-in-MB>

The size of the snapshot itself. By default it is 20% of the size of the MySQL LVM mount or the remaining free-space in the Volume Group (if there is less than 20% available) up to 15GB. If snapshot-size is defined, the number represents the size of the snapshot in megabytes.

snapshot-name = <name>

The name of the snapshot, the default being the name of the MySQL LVM volume + "_snapshot" (ie Storage-MySQL_snapshot)

snapshot-mountpoint = <path>

Where to mount the snapshot. By default a randomly generated directory under /tmp is used.

innodb-recovery = yes | no

Whether or not to run an InnoDB recovery operation. This avoids needing to do so during a restore, though will make the backup process itself take longer.

lock-tables = yes | no

Whether or not to run a FLUSH TABLES WITH READ LOCK to grab various bits of information (such as the binary log name and position). Disabling this requires that binary logging is disabled and InnoDB is being used exclusively. Otherwise, it is possible that the backup could contain crashed tables.

extra-flush-tables = yes | no

Whether or not to run a FLUSH TABLES before running the full FLUSH TABLES WITH READ LOCK. Should make the FLUSH TABLES WITH READ LOCK operation a bit faster.

[compression]

[compression] Specify various compression settings, such as compression utility, compression level, etc.

method = gzip | pigz | bzip | lzop | lzma

Define which compression method to use. Note that `lzop` and `lzma` may not be available on every system and may need to be compiled / installed.

inline = yes | no

Whether or not to pipe the output of mysqldump into the compression utility. Enabling this is recommended since it usually only marginally impacts performance, particularly when using a lower compression level.

level = 0-9

Specify the compression ratio. The lower the number, the lower the compression ratio, but the faster the backup will take. Generally, setting the lever to 1 or 2 results in favorable compression of textual data and is noticeably faster than the higher levels. Setting the level to 0 effectively disables compression.

bin-path = <full path to utility>

This only needs to be defined if the compression utility is not in the usual places or not in the system path.

[mysql:client]

MySQL connection info [mysql:client] These are optional and, if left undefined, Holland will try to login using the standard .my.cnf conventions.

user = <user>

The user to connect to MySQL as.

password = <password>

The password for the MySQL user

socket = <socket>

The socket file to connect to MySQL with.

host = <host>

This would be used for connecting to MySQL remotely.

port = <port>

Used if MySQL is running on a port other than 3306.

1.4.3 Backup-Set Configs

Backup-Set configuration files largely inherit the configuration options of the specified provider. To define a provider for the backup set, you must put the following at the top of the backup set configuration file:

```
[holland:backup]
plugin = <plugin>
backups-to-keep = #
estimated-size-factor = #
```

plugin = <plugin>

> This is the name of the provider that will be used for the backup-set. This is required in order for the backup-set to function.

backups-to-keep = #

> Specifies the number of backups to keep for a backup-set.

estimated-size-factor = #

> Specifies the scale factor when Holland decides if there is enough free space to perform a backup. The default is 1.0 and this number is multiplied against what each individual plugin reports its estimated backup size when Holland is verifying sufficient free space for the backupset.

Backup-Set files are defined in the "backupsets" directory which is, by default, `/etc/holland/backupsets`. The name of the backup-set is defined by its configuration filename and can really be most anything. That means backup-sets can be organized in any arbitrary way, although backup set files must end in .conf. The file extension is not part of the name of the backup-set.

As noted above, in order for a backup-set to be active, it must be listed in the *backupsets* variable.

Backups are placed under the directory defined in the *backup_directory* section of the main configuration file. Each backup resides under a directory corresponding to the backup-set name followed by a date-encoded directory.

CHAPTER

TWO

INDICES AND TABLES

- *Index*
- *Module Index*
- *Search Page*