



Server Management



om:

<https://pandorafms.com/manual/!776/>

permanent link:

https://pandorafms.com/manual/!776/en/documentation/pandorafms/installation/06_server_management

2024/06/10 14:34



Server Management

Pandora FMS Server maintenance

Manual startup/shutdown for Pandora FMS servers

Bear in mind that if for maintenance reasons you want to manually stop Pandora FMS server, verify whether a service Watchdog is running. [Check this section](#) for more details.

To start or stop manually Pandora FMS server execute the following in a shell console:

- Stop the *daemon* :

```
/etc/init.d/pandora_server stop
```

- Start the daemon:

```
/etc/init.d/pandora_server start
```

- Restart the daemon:

```
/etc/init.d/pandora_server restart
```

Version NG 756 or later.

From version NG 756 onwards the previous instructions will also start `pandora_ha` service.

To start and/or stop manually only Pandora FMS server, you have to execute the following in a shell console:

- Stop the daemon:

```
/etc/init.d/pandora_server stop-server
```

- Start the daemon:

```
/etc/init.d/pandora_server start-server
```

- Restart the daemon:

```
/etc/init.d/pandora_server restart-server
```

You may monitor the status of `pandora_ha` through `systemd` with:

```
systemctl status pandora_ha.service
```

Database management

Pandora FMS infrastructure does not need external maintenance, but it is very important to purge old data, and maintain the database as much compacted as possible. There is an essential tool for the proper functioning of Pandora FMS which is in charge of carrying out these tasks.

In Pandora FMS Open version its location is:

```
/usr/share/pandora_server/util/pandora_db.pl
```

E Or here in case of Pandora FMS Enterprise version:

```
/usr/bin/pandora_db
```

This tool, hereinafter `pandora_db.pl` is included in Pandora FMS server package, so it must be executed from a system where there is a Pandora FMS server installed. If there are two systems, one for the console and one for the server, `pandora_db` must be executed where Pandora FMS server is hosted.

`pandora_db` is a key tool for Pandora FMS to work properly, and that is why its execution is programmed in system cron tasks with an hourly interval. Its execution is configured within the file:

```
/etc/cron.hourly/pandora_db
```

This tool performs all database maintenance tasks automatically:

- It deletes old data.
- It compresses existing data, interpolating them at several intervals, so that graphics are the same but the space needed to store them is much smaller (this is one reason why Pandora FMS is able to process such an amount of information).
- It checks the consistency of the database for non-existing modules, or modules that are not used because they can not be started (these modules appear in the tactical view as uninitialized modules).
- It deletes the daily agent contact information. Pandora FMS does not need more than 24hr agent contact history, and if it builds up, it slows down database access.
- **E** In the enterprise version, it moves all old data to the standby history database.

As mentioned before, `pandora_db` execution is configured in system `cron` tasks and although this execution is automatically included in Pandora FMS server installation, it is convenient to check it. Therefore, `/etc/cron.hourly/pandora_db` must exist and contain the following.

In Pandora FMS Open version:

```
"/usr/share/pandora_server/util/pandora_db.pl"  
"/etc/pandora/pandora_server.conf">/dev/null 2>&1
```

Or in Pandora FMS Enterprise version:

```
"/usr/bin/pandora_db" "/etc/pandora/pandora_server.conf">/dev/null 2>&1
```

It is equally important to check permissions and the file's owner. The appropriate file permissions would be 755, which can be granted by executing:

```
chmod 755 /etc/cron.hourly/pandora_db
```

Regarding the owner, it must be root both for the user and group, which is set executing:

```
chown root:root /etc/cron.hourly/pandora_db
```

Maintenance tool manual execution

If necessary, it is possible to launch `pandora_db` execution manually as it was exposed in the previous section. Execute from a shell the following.

In Pandora FMS Open version:

```
/usr/share/pandora_server/util/pandora_db.pl /etc/pandora/pandora_server.conf
```

 Or in Pandora FMS Enterprise version:

```
/usr/bin/pandora_db /etc/pandora/pandora_server.conf
```

It should show an output similar to this one:

```
Pandora FMS DB Tool v7.0NG.757 Build 210915

This program is Free Software, licensed under the terms of GPL License v2
You can download latest versions and documentation at official web

DB Tool now initialized and running (PURGE=45 days, COMPACT=0 days, STEP=1) .

[*] Pandora FMS Enterprise module not available.

Starting at 2021-12-16 12:18:42
12:18:42 [PURGE] Deleting old data from tagente_datos. 100%
12:18:42 [PURGE] Deleting old export data from tserver_export_data
12:18:42 [PURGE] Deleting old session data from tsessions_php
12:18:42 [PURGE] No data in tagente_datos_string.
12:18:42 [PURGE] Deleting old event data at tevento table (More than 7 days).
12:18:42 [PURGE] Deleting old audit data (More than 15 days).
12:18:42 [PURGE] Deleting old SNMP traps (More than 7 days).
12:18:42 [PURGE] Deleting old GIS data (More than 7 days).
12:18:42 [PURGE] Deleting pending delete modules (data table).
12:18:42 [PURGE] Deleting pending delete modules (status, module table).
12:18:42 [PURGE] Deleting old access data (More than 24hr)
12:18:44 [PURGE] Agent access deletion progress %100
12:18:44 [PURGE] Delete contents in report that have some deleted modules.
12:18:44 [PURGE] Delete contents in report that have some deleted agents.
12:18:44 [PURGE] Delete empty contents in report (like SLA or Exception).
12:18:44 [PURGE] Delete autodisabled agents where last contact is bigger than 30 days.
12:18:44 [PURGE] Deleting old netflow data.
12:18:44 [!] Cannot execute /usr/bin/nfexpire, skipping.
12:18:44 [PURGE] Deleting old log data.
12:18:44 [PURGE] Deleting log data older than 90 days.
12:18:44 [PURGE] Deleting old special days.
12:18:44 [PURGE] Deleting old tgraph_source data.
12:18:44 [PURGE] Deleting old network matrix data.
12:18:44 [PURGE] Deleting old messages.
12:18:44 [PURGE] Deleting old cache data.
12:18:44 [CHECKDB] Ignoring not-init data.
12:18:44 [CHECKDB] Checking database consistency (Missing status).
12:18:44 [CHECKDB] Checking database consistency (Missing module).
12:18:44 [CHECKDB] Updating empty aliases.
12:18:44 [INTEGRITY] Cleaning up group stats.
12:18:44 [INTEGRITY] Deleting orphan alerts.
12:18:44 [INTEGRITY] Deleting orphan modules.
12:18:44 [FORGOTTEN DISCOVERY TASKS] Check for discovery tasks bound to inactive server
12:18:44 [FORGOTTEN DISCOVERY TASKS] There are not defined discovery tasks. Skipping.
Ending at 2021-12-16 12:18:44
# euclides root ~
```

This may take hours in overloaded systems, so it is recommended to execute the process in the background.

To execute manually the maintenance tool and leave it in the background, execute.

In Pandora FMS Open version:

```
nohup /usr/share/pandora_server/util/pandora_db.pl
/etc/pandora/pandora_server.conf
```

✘ Or in Pandora FMS Enterprise version:

```
nohup /usr/bin/pandora_db /etc/pandora/pandora_server.conf
```

The process will take some time until it loads completely in the background. Then, you may close the *shell* console window, while the process will still be executed.

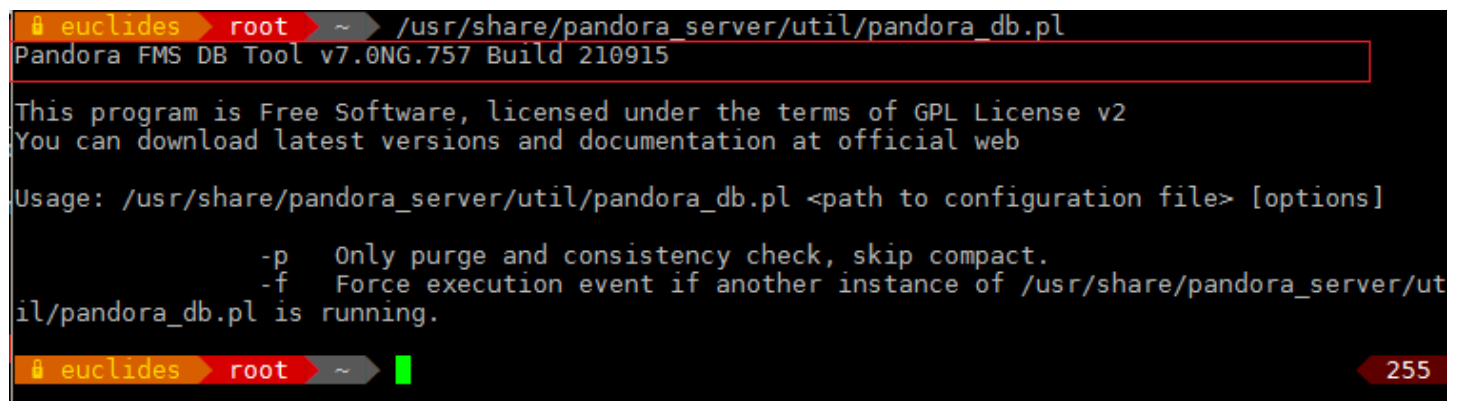
In some installations the tool directory could change. The most common one is:

```
/usr/share/pandora_server/util/
```

In Pandora FMS previous versions, it could be found at:

```
/usr/share/pandora/util/
```

It is very important to make sure you are using the updated tool version, and not that of a previous version. If you execute the program without arguments, it will show the tool version at the head of the message and it must match the version installed on the server.



```
euclides root ~ /usr/share/pandora_server/util/pandora_db.pl
Pandora FMS DB Tool v7.0NG.757 Build 210915

This program is Free Software, licensed under the terms of GPL License v2
You can download latest versions and documentation at official web

Usage: /usr/share/pandora_server/util/pandora_db.pl <path to configuration file> [options]
        -p  Only purge and consistency check, skip compact.
        -f  Force execution event if another instance of /usr/share/pandora_server/ut
il/pandora_db.pl is running.

euclides root ~ 255
```

Database Backup

It is important to remember that this only backs up or restores the database, *without including other files*, nor the server configuration.

The command `mysqldump` will do a complete database dump of the table structure as well as its contents. This is done by doing a full database dump, both of the table structure as well as their contents. This command has several options to do a backup, and its most basic usage is enough to dump from the same system where the database is hosted. Indicate the database name from which the backup will be done and the credentials to access it:

```
mysqldump -u <user> -p <data_base>
```

For instance, to backup “pandora” database and dump the result to an .sql file execute:

```
mysqldump -u root -p pandora> /backup/pandoradb_backup.sql
```

From a backup done that way, you may fully restore the database. Login in MySQL, proceed to create the database that will be restored and then load the backup in that database. Following the previous example:

```
[root@pandorafms ~]# mysql -u root -p
mysql> create database pandora;
mysql> use pandora;
mysql> source /backup/padnoradb_backup.sql
```

Finally, it is necessary to set configured user permissions again both in Pandora FMS Console and server so they have total access to the database (replace mypassword by your password):

```
grant all privileges on pandora.* to pandora@localhost identified by
'mypassword';
```

Pandora FMS complete backup and restore

There is a script in Pandora FMS server distribution that is useful to backup and restore Pandora FMS completely. This script is intended for backup and restoration in systems where the server and console are located in the same machine.

If there are several components in your environment, then you should use the tool with the most adequate parameters for its use or modify them so they are tailored to your needs.

This script needs to be executed as root. You are the only responsible for said key. It is located at:

```
/usr/share/pandora_server/util/pandora_backup.sh
```

If it is executed without parameters, it will show the help menu:


```

euclides root ~ /usr/share/pandora_server/util/pandora_backup.sh
Pandora FMS Command line backup tool. http://www.pandorafms.org
(c) 2009-2015 Sancho Lerena <slerena@gmail.com>, Artica Soluciones Tecnologicas

Syntax:
  -c Path to Pandora FMS console, p.e: /srv/www/htdocs/pandora_console
  -d Destination path for backup file. p.e: /tmp
  -s Source filename for backup restore. p.e: /tmp/pandorafms
  -f Restore also files
  -q Quiet. No output message (used for scripts/cron)
  -b No database backup/restore

Please BE SURE TO USE RESTORE (-s) option. This will OVERWRITE ALL your
PandoraFMS install, including files, configuration and data. Please backup first!

euclides root ~ █

```

This script is designed to back up and restore the following components:

- Server configuration file(s).
- Files waiting for execution, and also agent remote configuration files.
- Complete DB.
- Complete WEB Console.

Backup source and destination options

This script obtains access credentials to the database directly from the WEB console configuration. That is why the full path with the `-c` must be sent to the WEB console. That same parameter is also used to back up the web console itself.

The backup's destination is specified with the `-d` parameter. The compressed backup file will be lodged in that path, with a name similar to `pandorafms_backup_xxxxxxx.tar.gz`.

That way the following command can perform a full environment backup:

```
/usr/share/pandora_server/util/pandora_backup.sh -c
/var/www/html/pandora_console/ -d /tmp/
```

With the `-b` parameter the database backup can be prevented:

```
/usr/share/pandora_server/util/pandora_backup.sh -c
/var/www/html/pandora_console/ -d /tmp/ -b
```

Database restore

To restore the database with the script, replace the parameter `-d` by `-s`, indicating in this particular case the path to the backup:

```
/usr/share/pandora_server/util/pandora_backup.sh -c  
/var/www/html/pandora_console/ -s  
/tmp/pandorafms_backup_2021-12-16-16-11-50.tar.gz
```

This is the default restoration, with no directories included.

Database and directory restore

The `-f` option allows restoring backup files (overwriting the current ones). Since overwriting current configuration files could have serious consequences, use `-f` if you want to do a backup recovery and you want it to restore all Pandora FMS files (console and server).

```
/usr/share/pandora_server/util/pandora_backup.sh -c  
/var/www/html/pandora_console/ -s  
/tmp/pandorafms_backup_2021-12-16-16-11-50.tar.gz -f
```

Remember restoring user permissions to the database with the `grant` command.

File Restoration (not database)

Same as with the previous option, it is possible to restore only the files, without including the database. To do it, add the `-b` option to the previous execution.

```
/usr/share/pandora_server/util/pandora_backup.sh -c  
/var/www/html/pandora_console/ -s  
/tmp/pandorafms_backup_2021-12-16-16-11-50.tar.gz -f -b
```

Usage examples

Create backup

To do a full Pandora FMs backup, execute:

```
/usr/share/pandora_server/util/pandora_backup.sh -c  
/var/www/html/pandora_console/ -d /tmp/
```

It will return something similar to this:

```
Backup completed and placed in  
/tmp//pandorafms_backup_2021-12-16-16-11-50.tar.gz
```

This means that the backup is at, in this case:

```
/tmp/pandorafms_backup_2021-12-16-16-11-50.tar.gz
```

Restore backup

To restore the backup automatically, you need a console with the authentication credentials on the database correctly defined.

Execute:

```
/usr/share/pandora_server/util/pandora_backup.sh -c  
/var/www/html/pandora_console/ -s  
/tmp/pandorafms_backup_2021-12-16-16-11-50.tar.gz -f
```

It will give back something similar to:

```
Detected Pandora FMS backup at  
/tmp/pandorafms_backup_2021-12-16-16-11-50.tar.gz, please wait...  
Dropping current database  
Restoring backup database  
Restoring files and configuration  
Done. Backup in /tmp/pandorafms_backup_2021-12-16-16-11-50.tar.gz restored
```

Restore backup in case of console loss

If you have lost the Pandora FMS console, but you have a backup generated by this tool, then first restore the console directory. So, decompress manually its backup:

```
cd /tmp  
tar zxvf pandorafms_backup_2021-12-16-16-11-50.tar.gz
```

This will decompress your WEB console full directory in /tmp. In case of the generated backup of the previous example, it creates a directory named:

```
/tmp/var/www/html/pandora_console
```

Copy the content of all that directory to to your web publication directory, which may vary depending on your distribution:

```
cp -R /tmp/var/www/html/pandora_console /var/www/html
```

Then restore the backup as usual.

Watchdog implementation for Pandora FMS

In the code repository, there is a small script that is used as Watchdog. This script performs a Pandora FMS status monitoring.

Thus you may perform a recovery operation (trying to start Pandora FMS), and if that fails, a warning must be issued. This tool is at:

```
/usr/share/pandora_server/util/pandora_watchdog.sh
```

Script for alert generation

The `pandora_watchdog.sh` script looks for a file in `/usr/bin/pandora_alert` with the instructions to generate the alert. This file must be created defining the code to be executed when the watchdog cannot lift the Pandora FMS server. In this example, in addition to warning through a SMS, it stops the Tentacle server:

```
#!/bin/bash
sendsms +34458474843 "The Pandora FMS has suffered some issue and it cannot be
started."
/etc/init.d/tentacle_serverd stop
```

Grant execution permissions to this script:

```
chmod 750 /usr/bin/pandora_alert
```

Watchdog startup

To launch the Watchdog and leave it workin in the background, execute the following:

```
nohup /usr/share/pandora_server/util/pandora_watchdog.sh &
```

When starting a Watchdog, bear in mind that if for maintenance reasons you wish to manually stop Pandora FMS server, you must also previously have stopped the Watchdog process to prevent it from trying to start the service automatically continuously.

History database



E A history database is a database where old module data are transferred to make the main Pandora FMS database more responsive for everyday operations. Those data will still be available for Pandora FMS console when viewing reports, module charts etc.

Setting up a historical database

Version NG 754 and later versions **E**.

E To configure a historical database, it is necessary to have a new server where to host it (different from the one of the main database). Once you have that server where MySQL is installed, follow these steps:

In Pandora FMS console, go to ►Setup → Setup → Historical database to access [connection configuration](#) (Configure connection target).

Enable historical database

Configure connection target

Host	<input type="text"/>
Port	<input type="text" value="3306"/>
Database name	<input type="text" value="pandora"/>
Database user	<input type="text" value="pandora"/>
Database password	<input type="password"/>

Fill in the following fields:

- Host: Host name (web link or IP address) of the history database.

- Port: Connection port of the history database. Standard value: 3306.
- Database name: Name of the history database. Default value: pandora.
- Database user: History database user. Default value: pandora.
- Database password: Password of the history database. This will allow to connect to the history database. Next fill in the custom parameters (Customize settings):

Customize settings

Active to historical settings

Advanced options

Data days old to keep in active database ⓘ

15

Transference block size (Step) ⓘ

1500

Delay between transfereces (seconds)

1

Historical data settings

Maximum historical data age (days)

180

Maximum historical string data age (days)

180

Automatic partition of big tables. ⓘ

Historical events settings

 Enable historical events

Events days old to keep in active database ⓘ

6

Maximum historical events age (days) ⓘ

180

Historical trap settings

 Enable historical traps

Days old to keep in active database ⓘ

6

Maximum historical traps age (days) ⓘ

180

Update 

The last fields of this form *Data days old to keep in active database*, *Transference block size (Step)* and *Delay between transferences (seconds)* will define the way in which the data will be sent to the historical database, that is, the oldest data those with more than *n days (Data days)* will be moved to the history database in blocks of made of *n rows (Step)* waiting *n seconds (Delay)* between one block and the next to avoid overloads.

In the same screen, it is also possible to decide whether to send the events with more than *n days* to the history database (*Events days old to keep in active database*), although it must be taken into account that including the events will considerably increase the growth rate of the historical database, and that these will only be consulted when generating reports, not in the event view.

E The historical database is a feature of the Enterprise version that uses the `/usr/bin/pandora_db` binary to transfer data. In addition, you must configure the settings [for the maintenance of the history database](#).

History database for NG 753 and earlier versions

These steps are only for version 753 and previous versions. From version 754 the only thing necessary is to configure the connection values explained in the previous section and Pandora FMS will take care of the whole installation process and the [maintenance of the historical database](#).

E To configure a historical database, it is necessary to have a new server where to host it (different from the one of the main database). Once you have that server where MySQL is installed, follow these steps:

- Create the new historical database:

```
[root@pandorafms ~]# mysql -u root -p
mysql> create database pandora_history;
```

- Create Pandora FMS database schema. You may use the script `/var/www/html/pandora_console/pandoradb.sql` included in Pandora FMS console, copy it to the historical database server:

```
cat pandoradb.sql | mysql -u root -p -D pandora_history
```

- Grant permissions to a user (user and password) which will be used from the Pandora FMS server and console to send and check historical data:

For MySQL 5.7

```
GRANT ALL PRIVILEGES ON pandora_history.* TO 'user'@'%' IDENTIFIED BY 'password';
```


For MySQL 8:

- \$DBROOTPASS: MySQL 8 root user password.
- \$DBUSER: MySQL 8 user name.
- \$DBPASS: MySQL 8 user password.
- \$DBPORT: Connection port for MySQL 8.
- \$DBHOST: IP address or FQDN of the MySQL server 8.
- \$DBNAME: Name of the database.

To use these environment variables just define them before running *grants* in the terminal, enter the following:

```
env TZ='Europe/Madrid' \  
DBHOST='127.0.0.1' \  
DBNAME='pandora' \  
DBUSER='pandora' \  
DBPASS='pandora' \  
DBPORT='3306' \  
DBROOTPASS='pandora'
```

Enter in the terminal:

```
systemctl restart mysql
```

Enter in the terminal:

```
export MYSQL_PWD=$DBROOTPASS  
echo "CREATE USER \"\$DBUSER\"@'%' IDENTIFIED BY \"\$DBPASS\";" | mysql -uroot -  
P\$DBPORT -h\$DBHOST  
echo "ALTER USER \"\$DBUSER\"@'%' IDENTIFIED WITH mysql_native_password BY  
\"\$DBPASS\"" | mysql -uroot -P\$DBPORT -h\$DBHOST  
echo "GRANT ALL PRIVILEGES ON \$DBNAME.* TO \"\$DBUSER\"@'%' " | mysql -uroot -  
P\$DBPORT -h\$DBHOST  
  
export MYSQL_PWD=$DBPASS
```

The history database (DB) can contain *all the system data* (without limit), but if you want to delete data or compact them from the history database, you will need to use specific data in the database that the `pandora_db` script takes into account when executing from the node.

Remember that all these steps are only for version 753 and earlier versions.

First, enter some data from the history database in `tconfig` table. Use this SQL queries to create a minimal configuration and configure the performance of `pandora_db` when executed against the history database. First you need to connect to your database using the command line (CLI) of

MySQL.

This is an example, replace values according to your criteria (but leave `history_db_enabled` at zero):

```
INSERT INTO `tconfig` VALUES (1,'days_purge','180');
INSERT INTO `tconfig` VALUES (2,'history_db_enabled','0');
INSERT INTO `tconfig` VALUES (3,'days_compact','120');
INSERT INTO `tconfig` VALUES (4,'step_compact','1');
INSERT INTO `tconfig` VALUES (5,'event_purge','180');
INSERT INTO `tconfig` VALUES (6,'string_purge','180');
INSERT INTO `tconfig` VALUES (7,'MR','<actual MR value>');
```

- Take special care to replace `< actual MR value >` with the current MR value that you currently have in the main Pandora FMS server.
- This example is for the history database to store for 6 months (180 days) from the execution date (`days_purge`).
- Compact data more than 120 days old (`days_compact`).
- If you have 1 month worth of data in your main database, you will have a total of 6 months of data, because the last month has no data in the history database, but it has them in the main database (`event_purge` y `string_purge`).
- *Use the values that you think convenient* (except for MR and `history_db_enabled`). The limit in the storage of the history database only depends on the free space of the machine. Just remember that the history database must be in a physical server independent from the main database and Pandora FMS.

In versions prior to Pandora FMS 753 you must execute the script `pandora_db` in the history server itself using the data specified previously in the database and additionally create the configuration file as explained below, in order to use the maintenance script as if we were using a normal database.

In versions prior to Pandora FMS 753, create an additional `pandora_server.conf` file. Use this small version to create your own and name it `/etc/pandora/pandora_server_history_db.conf` :

```
dbengine mysql
dbname pandora_history
dbuser user
dbpass password
dbhost 192.168.70.140
log_file /var/log/pandora/pandora_db_history.log
```

Remember that all these steps are only for version 753 and earlier versions.

Now you can execute `pandora_db` tool against the history database configuration and program it for its periodical execution:

```
/usr/share/pandora_server/util/pandora_db.pl  
/etc/pandora/pandora_server_history_db.conf
```

This process **SHOULD NOT** affect your main operation because it is running against a different database in a different server.

[Go back to Pandora FMS documentation index](#)