GLPI Installation

GLPI Project, Teclib'

Oct 13, 2025

CONTENTS

1 Prerequisites **3**
- 1.1 Web server . . . 3
  - 1.1.1 Apache configuration . . . 3
  - 1.1.2 Nginx configuration . . . 4
  - 1.1.3 lighttpd configuration . . . 5
  - 1.1.4 IIS configuration . . . 5
- 1.2 PHP . . . 6
  - 1.2.1 Mandatory extensions . . . 6
  - 1.2.2 Optional extensions . . . 6
  - 1.2.3 Security configuration for sessions . . . 7
- 1.3 Database . . . 7

2 Install GLPI **9**
- 2.1 Choose a version . . . 9
- 2.2 Download . . . 9
- 2.3 Installation . . . 9
- 2.4 Files and directories locations . . . 10
  - 2.4.1 Plugins files locations . . . 11

3 Install wizard **13**
- 3.1 Choose lang (Select your language) . . . 13
- 3.2 License . . . 13
- 3.3 Install / Update . . . 14
  - 3.3.1 Environment checks . . . 14
  - 3.3.2 Database connection . . . 15
  - 3.3.3 Database choice . . . 16
  - 3.3.4 Database initialization . . . 17
  - 3.3.5 Telemetry informations . . . 17
  - 3.3.6 End of installation . . . 18

4 Timezones **19**
- 4.1 Non windows users . . . 19
- 4.2 Windows users . . . 19
- 4.3 Grant access . . . 19

5 Update **21**

6 Command line tools **23**
- 6.1 Console options . . . 23
- 6.2 Additional install and update tools . . . 24
  - 6.2.1 Check requirements . . . 24

6.2.2 Enable/Disable maintenance . . . 24
6.3 Install . . . 24
6.4 Database connection configuration . . . 24
6.5 Update . . . 25
6.6 Security key . . . 25
6.7 Various tools . . . 26
6.7.1 Database schema check . . . 26
6.7.2 LDAP synchonization . . . 26
6.7.3 Task unlock . . . 27
6.8 Suspend/Resume plugins execution . . . 27
6.9 Plugins tools . . . 28
6.10 Migration tools . . . 28
6.10.1 From MyISAM to InnoDB . . . 28
6.10.2 Missing timestamps builder . . . 28
6.10.3 Use timestamp data type . . . 28
6.10.4 Migrate Domains plugin . . . 29
6.10.5 Migrate Racks plugin . . . 29

7 Advanced configuration **31**
7.1 SSL connection to database . . . 31

This documentation presents GLPI installation instructions.

GLPI (Gestion Libre de Parc Informatique) is a free (as in "free speech" not as in "free beer"!) asset and helpdesk management solution accessible from a web browser built to manage all you asset management issues, from hardware components and software inventories management to user helpdesk management.

CHAPTER

ONE

PREREQUISITES

GLPI is a Web application that will need:

- a webserver;
- PHP;
- a database.

1.1 Web server

GLPI requires a web server that supports PHP, like:

- Apache 2 (or more recent);
- Nginx;
- lighttpd;
- Microsoft IIS.

1.1.1 Apache configuration

Here is a virtual host configuration example for `Apache 2` web server.

> **Warning**
>
> The following configuration is only suitable for GLPI version 10.0.7 or later.

```
<VirtualHost *:80>
    ServerName glpi.localhost

    DocumentRoot /var/www/glpi/public

    # If you want to place GLPI in a subfolder of your site (e.g. your virtual host is␣
↪serving multiple applications),
    # you can use an Alias directive. If you do this, the DocumentRoot directive MUST␣
↪NOT target the GLPI directory itself.
    # DocumentRoot /var/www/html
    # Alias "/glpi" "/var/www/glpi/public"

    <Directory /var/www/glpi/public>
        Require all granted
```

(continues on next page)

(continued from previous page)

```
        RewriteEngine On

        # Ensure authorization headers are passed to PHP.
        # Some Apache configurations may filter them and break usage of API, CalDAV, ...
        RewriteCond %{HTTP:Authorization} ^(.+)$
        RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

        # Redirect all requests to GLPI router, unless file exists.
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>
</VirtualHost>
```

Note

If you cannot add specific rewrite rules to the `Apache` configuration (e.g. you are using a shared hosting), you can use a `.htaccess` file. Still, you need to define the `DocumentRoot` to be the `/public` directory of GLPI.

```
# /var/www/glpi/public/.htaccess
RewriteBase /
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php [QSA,L]
```

1.1.2 Nginx configuration

Here is a configuration example for `Nginx` web server using `php-fpm`.

Warning

The following configuration is only suitable for GLPI version 10.0.7 or later.

```
server {
    listen 80;
    listen [::]:80;

    server_name glpi.localhost;

    root /var/www/glpi/public;

    location / {
        try_files $uri /index.php$is_args$args;
    }

    location ~ ^/index\.php$ {
        # the following line needs to be adapted, as it changes depending on OS␣
↪distributions and PHP versions
```

(continues on next page)

(continued from previous page)

```
        fastcgi_pass unix:/run/php/php-fpm.sock;

        fastcgi_split_path_info ^(.+\.php)(/.*)$;
        include fastcgi_params;

        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}
```

1.1.3 lighttpd configuration

Here is a virtual host configuration example for `lighttpd` web server.

> **Warning**
>
> The following configuration is only suitable for GLPI version 10.0.7 or later.

```
$HTTP["host"] =~ "glpi.localhost" {
    server.document-root = "/var/www/glpi/public/"

    url.rewrite-if-not-file = ( "" => "/index.php${url.path}${qsa}" )
}
```

1.1.4 IIS configuration

Here is a `web.config` configuration file example for `Microsoft IIS`. The physical path of GLPI web site must point to the `public` directory of GLPI (e.g. `D:\glpi\public`), and the `web.config` file must be placed inside this directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <rewrite>
            <rules>
                <rule name="Rewrite to GLPI" stopProcessing="true">
                    <match url="^(.*)$" />
                    <conditions>
                        <add input="{REQUEST_FILENAME}" matchType="IsFile" ignoreCase=
↪"false" negate="true" />
                    </conditions>
                    <action type="Rewrite" url="index.php" appendQueryString="true" />
                </rule>
          </rules>
        </rewrite>
    </system.webServer>
</configuration>
```

> **Warning**
>
> The URL Rewrite module is required.

1.2 PHP

Table 1: PHP Compatibility Matrix

GLPI version	Minimum PHP version
10.0.X	7.4
11.0.X	8.2

Note

GLPI compatibility with new PHP versions is validated shortly after their release. We therefore recommend using the most recent version, for better performances.

1.2.1 Mandatory extensions

Following PHP extensions are required for the app to work properly:

- `dom`, `fileinfo`, `filter`, `libxml`, `simplexml`, `tokenizer`, `xmlreader`, `xmlwriter`: these PHP extensions are enable by default and are used for various operations;
- `curl`: used for remote access to resources (inventory agent requests, marketplace, RSS feeds, ...);
- `gd`: used for images handling;
- `intl`: used for internationalization;
- `mysqli`: used for database connection;
- `session`: used for sessions support;
- `zlib`: used for handling of compressed communication with inventory agents, installation of gzip packages from marketplace and PDF generation.

For GLPI 11.0, these additional PHP extensions are mandatory:

- `bcmath`: used for QRCodes generation;
- `mbstring`: used for multibyte chars support and charset conversion;
- `openssl`: used for handling of encrypted communication with inventory agents and OAuth 2.0 authentication.

1.2.2 Optional extensions

Note

Even if those extensions are not mandatory, we advise you to install them anyways.

Following PHP extensions are required for some extra features of GLPI:

- `bz2`, `Phar`, `zip`: enable support of most common packages formats in marketplace;
- `exif`: enhance security on images validation;
- `ldap`: enable usage of authentication through remote LDAP server;
- `openssl`: enable email sending using SSL/TLS;
- `Zend OPcache`: enhance PHP engine performances.

Following PHP extensions are emulated in GLPI and can be enabled to slightly improve performances: * `ctype`; * `iconv`; * `sodium`.

1.2.3 Security configuration for sessions

To enhance security, it is recommended to configure PHP sessions with the following settings:

- `session.cookie_secure`: should be set to `on` when GLPI can be accessed on **only** HTTPS protocol;
- `session.cookie_httponly`: should be set to `on` to prevent client-side scripts from accessing cookie values;
- `session.cookie_samesite`: should be set, at least, to `Lax`, to prevent cookies from being sent cross-origin (across domains) POST requests.

> **Note**
>
> Refer to PHP documentation for more information about session configuration.

1.3 Database

In order to work, GLPI requires a database server.

> **Warning**
>
> Currently, only MySQL and MariaDB database servers are supported by GLPI.

Table 2: Database server compatibility matrix

GLPI version	Database server	Minimum version
10.0.X	MySQL	5.7
10.0.X	MariaDB	10.2
11.0.X	MySQL	8.0
11.0.X	MariaDB	10.6

> **Note**
>
> GLPI compatibility with new LTS database servers versions is validated shortly after their release. We therefore recommend using a recent version, for better performances.

(cc) BY-NC-ND

CHAPTER

TWO

INSTALL GLPI

Proceed as follow:

1. *Configure your webserver*,
2. Choose a version,
3. Download the archive,
4. Install :)

2.1 Choose a version

> **Note**
>
> It is hightly recommended you choose the latest stable release for a production usage.

GLPI follows a semantic versioning scheme, on 3 digits. The first one is the major release, the second the minor and the third the fix release.

Major releases may come with important incompatibilities as well as new features; minor versions may bring new features as well, but stay perfectly compatible inside a major version.

Fixes releases will only fix reported issues without adding anything new.

2.2 Download

> **Warning**
>
> On GitHub, there are always two archives named *Source code* which should not be used.

Go to the *download* section of the GLPI website (or get archive directly from Github release) and choose the `glpi-{version}.tgz` archive.

2.3 Installation

GLPI installation itself is composed of three steps:

1. Uncompress the archive in your website;
2. Give your webserver write access to the `files` and `config` directories;

3. Launch *installation wizard* (or use the *command line installation script*).

Once these three steps have been completed the application is ready to be used.

If you need to set advanced configuration, like SSL database connection parameters, please refer to *advanced configuration*.

2.4 Files and directories locations

As long as the web server is configured properly (see *web server configuration*), sensible files should not be exposed publicly.

GLPI stores some data in the `files` directory, the database access configuration is stored in the `config` directory, etc. To ease the GLPI maintenance, the location of GLPI storage directories can be customized.

There are a few configuration directives you may use to achieve that:

- `GLPI_CONFIG_DIR`: set path to the configuration directory;
- `GLPI_VAR_DIR` : set path to the `files` directory;
- `GLPI_LOG_DIR` : set path to logs files.

> **Note**
>
> There are many other configuration directives available, the ones we talked about are the main to move all stored files outside the GLPI source code.

Directories choice is entirely up to you; the following example will follow the FHS recommendations.

Our GLPI instance will be installed in `/var/www/glpi`, a specific virtual host in the web server configuration will reflect this path.

GLPI configuration will be stored in `/etc/glpi`, just copy the contents of the `config` directory to this place. GLPI requires read rights on this directory to work; and write rights during the installation process.

GLPI data will be stored in `/var/lib/glpi/files`, just copy the contents of the `files` directory to this place. GLPI requires read and write rights on this directory.

GLPI logs files will be stored in `/var/log/glpi`, there is nothing to copy here, just create the directory. GLPI requires read and write access on this directory.

Following this instructions, we'll create a `inc/downstream.php` file into GLPI directory with the following contents:

```php
<?php
define('GLPI_CONFIG_DIR', '/etc/glpi/');

if (file_exists(GLPI_CONFIG_DIR . '/local_define.php')) {
   require_once GLPI_CONFIG_DIR . '/local_define.php';
}
```

> **Warning**
>
> GLPI packages will certainly provide a `inc/downstream.php` file. This one must not be edited!
>
> GLPI looks for a *local_define.php* file in its own *config* directory. If you want to use one from new config directory, you have to load it.

Then, create a file in `/etc/glpi/local_define.php` with the following contents:

```
<?php
define('GLPI_VAR_DIR', '/var/lib/glpi/files');
define('GLPI_LOG_DIR', '/var/log/glpi');
```

Note

`GLPI_VAR_DIR` permits to change the storage path of all the GLPI files, but you can adapt the storage path for each kind of files.

```
<?php
define('GLPI_VAR_DIR',        '/var/lib/glpi/files');
define('GLPI_DOC_DIR',        GLPI_VAR_DIR);                  // Path for documents␣
↪storage
define('GLPI_CACHE_DIR',      GLPI_VAR_DIR . '/_cache');      // Path for cache␣
↪storage
define('GLPI_CRON_DIR',       GLPI_VAR_DIR . '/_cron');       // Path for cron storage
define('GLPI_GRAPH_DIR',      GLPI_VAR_DIR . '/_graphs');     // Path for graph␣
↪storage
define('GLPI_LOCAL_I18N_DIR', GLPI_VAR_DIR . '/_locales');    // Path for local i18n␣
↪files
define('GLPI_LOCK_DIR',       GLPI_VAR_DIR . '/_lock');       // Path for lock files␣
↪storage (used by cron)
define('GLPI_LOG_DIR',        GLPI_VAR_DIR . '/_log');        // Path for log storage
define('GLPI_PICTURE_DIR',    GLPI_VAR_DIR . '/_pictures');   // Path for picture␣
↪storage
define('GLPI_PLUGIN_DOC_DIR', GLPI_VAR_DIR . '/_plugins');    // Path for plugins␣
↪documents storage
define('GLPI_RSS_DIR',        GLPI_VAR_DIR . '/_rss');        // Path for RSS feeds␣
↪storage
define('GLPI_SESSION_DIR',    GLPI_VAR_DIR . '/_sessions');   // Path for sessions␣
↪files storage
define('GLPI_TMP_DIR',        GLPI_VAR_DIR . '/_tmp');        // Path for temporary␣
↪files storage
define('GLPI_UPLOAD_DIR',     GLPI_VAR_DIR . '/_uploads');    // Path for upload␣
↪storage
define('GLPI_INVENTORY_DIR',  GLPI_VAR_DIR . '/_inventories');// Path for inventory␣
↪files storage
define('GLPI_THEMES_DIR',     GLPI_VAR_DIR . '/_themes');     // Path for custom␣
↪themes storage
```

2.4.1 Plugins files locations

Added in version 11.0.0.

Plugins files location can be configured using the `GLPI_MARKETPLACE_DIR` configuration directive.

To store the plugins in the `/var/lib/glpi/plugins` directory, just copy the contents of the `marketplace` and `plugins` directories to this place. GLPI requires read and write rights on this directory.

Then, in the `/etc/glpi/local_define.php` file, add the following contents:

```
define('GLPI_MARKETPLACE_DIR', '/var/lib/glpi/plugins');
```

(cc) BY-NC-ND

CHAPTER

THREE

INSTALL WIZARD

To begin installation process, point your browser to the GLPI main address: https://\{adresse_glpi\}/

When GLPI is not installed; a step-by-step installation process begins.

3.1 Choose lang (Select your language)

The first step will let you choose the installation language. Select your lang, and click validate.

3.2 License

Usage of GLPI is subject to GNU license approval. Once licensing terms read and accepted, just validate the form.

If you do not agree with licensing terms, it is not possible to continue installation process.

3.3 Install / Update

This screen allows to choose between a fresh GLPI installation or an update.

Click on install.

3.3.1 Environment checks

This step will check if prerequisites are met. If they're not, it is not possible to continue and an explicit error message will tell you about what is wrong and what to do before trying again.

Some prerequisites are optionals, it will be possible to continue installation event if they're not met.

3.3.2 Database connection

Database connection parameters are asked.

- *MySQL server*: enter the path to your MySQL server, *localhost* or *mysql.domaine.tld* as example;
- *MySQL user*: enter user name that is allowed to connect to the Database;
- *MySQL password*: enter user's password.

Once all fields are properly filled, validate the form.

A first database connection is then established. If parameters are invalid, an error message will be displayed, and you'll have to fix parameters and try again.

3.3.3 Database choice

Once connection to the database server is OK, you have to create or choose the database you want for your GLPI and init it.

There are 2 ways to go:

- use an existing database

  Select this database in the displayed list. Validate to use.

> **Warning**
>
> Selected database contents will be destroyed on installation.

- Create a new database

  Choose *Create a new database*, enter the database name in the relevant field and then validate to create the base.

> **Warning**
>
> SQL user must be able to create new database for this option to work.

3.3.4 Database initialization

This step initializes the database with default values.

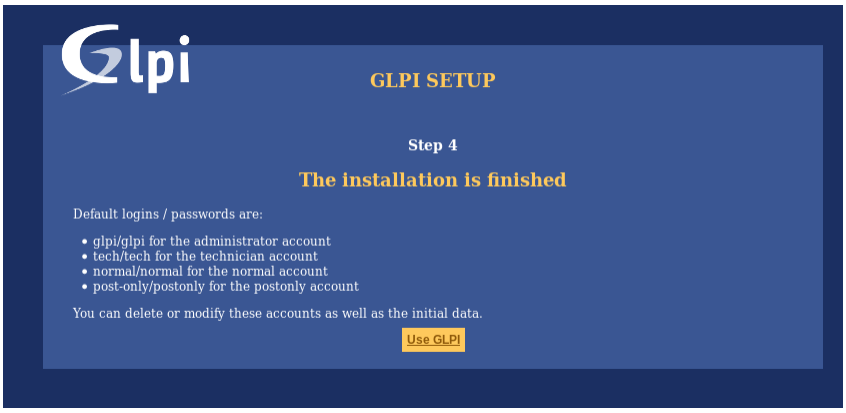
If there is any error; pay attention to the displayed informations.

3.3.5 Telemetry informations

GLPI will ask you to share some Telemetry informations and to register. This is not mandatory.

3.3.6 End of installation

This step presents a summary of the installation and give created users list. Please pay attention to those informations and validate to go to the app.



Note

Default user accounts are:

- *glpi/glpi* admin account,
- *tech/tech* technical account,
- *normal/normal* "normal" account,
- *post-only/postonly* post-only account.

Warning

For obvious security concerns, you'll have to delete or edit those accounts.

Before removing the `glpi` account, please make sure you have created another user with `super-admin` profile.

(cc) BY-NC-ND

CHAPTER

FOUR

TIMEZONES

In order to get timezones working on a MariaDB/MySQL instance, you will have to initialize Timezones data and grant GLPI database user read ACL on their table.

> **Warning**
>
> Enabling timezone support on your MySQL instance may affect other database in the same instance; be carefull!

> **Warning**
>
> Currently, MySQL, and MariaDB (prior to 11.5), have a maximum date limited to 2038-01-19 on fields relying on `timestamp` type!
>
> MariaDB 11.5 onwards is limited to 2106.

4.1 Non windows users

On most systems, you'll have to initialize timezones data from your system's timezones:

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -p -u root mysql
```

You may want to check MariaDB documentation about mysql_tzinfo_to_sql and your system documentation to know where data are stored (if not in `/usr/share/zoneinfo`).

Do not forget to restart the database server once command is successfull.

4.2 Windows users

Windows does not provide timezones informations, you'll have to download and intialize data yourself.

See MariaDB documentation about timezones.

4.3 Grant access

Removed in version 10.0.20: Granting access to the `mysql.time_zone_name` is no longer required to support timezones usage.

> **Warning**
>
> Be carefull not to give your GLPI database user too large access. System tables should **never** grant access to app users.

In order to list possible timezones, your GLPI database user must have read access on `mysql.time_zone_name` table. Assuming your user is `glpi@localhost`, you should run something like:

```
GRANT SELECT ON `mysql`.`time_zone_name` TO 'glpi'@'localhost';
```

(cc) BY-NC-ND

CHAPTER

FIVE

UPDATE

Note

As for every update process, you have to backup some data before processing any upgrade:

- **backup your database**;
- backup your `config` directory, especially for your GLPI key files (`config/glpi.key`, `config/glpicrypt.key`, `config/oauth.pem`, `config/oauth.pub`) which are randomly generated;
- backup your `files` directory, it contains users and plugins generated files, like uploaded documents;
- backup your `marketplace` and `plugins` directories.

Here are the steps to update GLPI:

- Download latest GLPI version.
- Ensure the target directory is empty and extract files there.
- Restore the previously backed up `config`, `files`, `marketplace` and `plugins` directories.
- Then open the GLPI instance URI in your browser, or (recommended) use the `php bin/console db:update` *command line tool*.

Warning

As soon as a new version of GLPI files is detected, you will not be able to use the application until the update process has been done.

Warning

You should not try to restore a database backup on a non empty database (say, a database that has been partially migrated for any reason).

Make sure your database is empty before restoring your backup and try to update, and repeat on fail.

Note

Before GLPI 11.0, all the plugins were disabled during the update process.

Since GLPI 11.0, as soon as a new version of GLPI files is detected, the plugins execution is suspended. Their execution will be resumed after the update only in case of a minor/bugfixes update. Otherwise, you will have to do it manually using the `php bin/console plugin:resume_execution` command line tool or from the plugins administration page.

Note

You can use the `php bin/console db:check` *command line tool* before executing the update command. This will allow you to check the integrity of your database, and to identify changes to your database that could compromise the update.

(cc) BY-NC-ND

CHAPTER

SIX

COMMAND LINE TOOLS

Since GLPI 9.2.2, command line tools are provided as supported scripts and are available from the `scripts` directory of the archive. On previous versions, those scripts were present in the `tools` directory that is not official and therefore not in the release archive.

Since GLPI 9.4.0, command line tools are being centralized in a console application (`bin/console`). Calling `php bin/console` from GLPI directory displays the list of available commands.

> **Note**
>
> If APCu is installed on your system, it may fail from command line since default configuration disables it from command-line. To change that, set `apc.enable_cli` to `on` in your APCu configuration file.

> **Warning**
>
> When using cli tools, please check the system user you are currently logged in with, and permissions on files and directories. With a wrong user, logs, cache and other files may be created with rights that would not allow your webserver to read or write on thos files!

6.1 Console options

For every console command, following options are available:

- `--config-dir=CONFIG-DIR` path of configuration directory to use, relative to current working directory (required only if a custom path is used)
- `-h`, `--help` displays command help
- `--lang=LANG` output language code (default value is existing GLPI "language" configuration or "en_GB")
- `-n`, `--no-interaction` disable command interactive questions
- `--no-plugins` disable GLPI plugins during command execution
- `-q`, `--quiet` disable command output
- `-v|vv|vvv`, `--verbose=VERBOSE` verbosity level: 1 for normal output, 2 for more verbose output and 3 for debug

6.2 Additional install and update tools

6.2.1 Check requirements

Before installing or upgrading, requirements are automatically checked; but you can run them separately and see state for all of them using the `php bin/console glpi:system:check_requirements` command.

6.2.2 Enable/Disable maintenance

GLPI provides a maintenance mode that can be activated prior to an update, and deactivated after all has been checked.

Just use the `glpi:maintenance:enable` and `glpi:maintenance:disable` commands.

6.3 Install

The `php bin/console db:install` has been made to install GLPI database in CLI mode.

Possible options for this command are:

- `-r`, `--reconfigure` to enable overriding of any existing DB configuration file
- `-f`, `--force` to force execution of installation even if database is not empty
- `-L`, `--default-language=DEFAULT_LANGUAGE` default language of GLPI (*en_GB* per default)
- `-H`, `--db-host=DB_HOST` host name (*localhost* per default)
- `-P`, `--db-port=DB_PORT` database port (default MySQL port if option is not defined)
- `-d`, `--db-name=DB_NAME` database name
- `-u`, `--db-user=DB_USER` database user name
- `-p`, `--db-password=DB_PASSWORD` database user's password (use it without value to be prompted for password)

If mandatory options are not specified in the command call, the console will ask for them.

Database connection parameters may be omitted if a configuration file already exists.

See also *console options*.

6.4 Database connection configuration

> Added in version 9.5.0.

The `php bin/console db:configure` has been made to define database connection parameters in CLI mode.

Possible options for this command are:

- `-r`, `--reconfigure` to enable overriding of any existing DB configuration file
- `-H`, `--db-host=DB_HOST` host name (*localhost* per default)
- `-P`, `--db-port=DB_PORT` database port (default MySQL port if option is not defined)
- `-d`, `--db-name=DB_NAME` database name
- `-u`, `--db-user=DB_USER` database user name
- `-p`, `--db-password=DB_PASSWORD` database user's password (use it without value to be prompted for password)

If mandatory options are not specified in the command call, the console will ask for them.

See also *console options*.

6.5 Update

The `php bin/console db:update` has been made to update GLPI database in CLI mode from a previously installed version.

There is no required arguments, just run the command so it updates your database automatically.

> **Warning**
>
> Do not forget to backup your database before any update try!

> **Warning**
>
> Since GLPI 10.0.2, *db:check_schema_integrity* is executed before performing the update. If an error is detected, the command will ask you if you want to continue (unless `--no-interaction` is used). You can bypass this *db:check_schema_integrity* by using the option `-s`, `--skip-db-checks`.

Possible options for this command are:

- `-u`, `--allow-unstable` allow update to an unstable version (use it with cautions)
- `-f`, `--force` force execution of update from v-1 version of GLPI even if schema did not changed
- `-s`, `--skip-db-checks` do not check database schema integrity before performing the update
- `--enable-telemetry` allow usage statistics sending to Telemetry service (https://telemetry.glpi-project.org)
- `--no-telemetry` disallow usage statistics sending to Telemetry service (https://telemetry.glpi-project.org)

See also *console options*.

6.6 Security key

Added in version 9.4.6.

> **Note**
>
> GLPI key file is available for GLPI >= 9.4.6 but is not mandatory. As of GLPI 9.5, using the key file will be mandatory.

In order to store some sensitive data, GLPI relies on a homemade encryption/decryption tool, which uses a key to:

- encrypt data before storing them in the database,
- decrypt data that has been retrieved from the database.

The `php bin/console glpi:security:change_key` command allows to change the key, if it has been compromised for example. By default, command will:

- generate a new key and and store it in the key file,
- update all configured fields (for core and compatible plugins) to use the new key,
- update all configuration entries listed (for core and compatible plugins) to use the new key.

6.7 Various tools

6.7.1 Database schema check

The `php bin/console db:check_schema_integrity` command can be used to check if your database schema differs from expected one.

Possible options for this command are:

- `--strict`: Strict comparison of definitions
- `--check-all-migrations`: Check tokens related to all databases migrations.
- `--check-innodb-migration`: Check tokens related to migration from "MyISAM" to "InnoDB".
- `--check-timestamps-migration`: Check tokens related to migration from "datetime" to "timestamp".
- `--check-utf8mb4-migration`: Check tokens related to migration from "utf8" to "utf8mb4".
- `--check-dynamic-row-format-migration`: Check tokens related to "DYNAMIC" row format migration.
- `--check-unsigned-keys-migration`: Check tokens related to migration from signed to unsigned integers in primary/foreign keys.
- `-p`, `--plugin`: Plugin to check. If option is not used, checks will be done on GLPI core database tables.

If you have any diff, output will looks like :

```
$ php bin/console glpi:database:check_schema_integrity
Table schema differs for table "glpi_rulecriterias".
--- Original
+++ New
@@ @@
 create table `glpi_rulecriterias` (
   `id` int(11) not null auto_increment
   `rules_id` int(11) not null default '0'
   `criteria` varchar(255) default null
   `condition` int(11) not null default '0'
-  `pattern` text default null
+  `pattern` text
   primary key (`id`)
```

Compared to the GLPI installation file:

- a line that starts with - means that something is missing in your database
- a line that starts with + means that there is something extra in your database

You can also have a message like `Unknown table "glpi_tablename" has been found in database.`, this indicates that this table doesn't exist in the installation file of the current GLPI schema:

- either it's a table that you have voluntarily created for your needs, you can ignore this message
- either it's an old GLPI table which is no longer useful, you can delete it (taking care to make a backup before)

6.7.2 LDAP synchonization

The `bin/console glpi:ldap:synchronize_users` command can be used to synchronize users against LDAP server informations.

Possible options for this command are:

- `-c`, `--only-create-new` only create new users

- `-u`, `--only-update-existing` only update existing users
- `-s`, `--ldap-server-id[=LDAP-SERVER-ID]` synchronize only users attached to this LDAP server (multiple values allowed)
- `-f`, `--ldap-filter[=LDAP-FILTER]` filter to apply on LDAP search
- `--begin-date[=BEGIN-DATE]` begin date to apply in "modifyTimestamp" filter
- `--end-date[=END-DATE]` end date to apply in "modifyTimestamp" filter
- `-d`, `--deleted-user-strategy[=DELETED-USER-STRATEGY]` force strategy used for deleted users:
  - 0: Preserve
  - 1: Put in trashbin
  - 2: Withdraw dynamic authorizations and groups
  - 3: Disable
  - 4: Disable + Withdraw dynamic authorizations and groups

See http://php.net/manual/en/datetime.formats.php for supported date formats in `--begin-date` and `--end-date` options.

See also *console options*.

6.7.3 Task unlock

The `php bin/console task:unlock` command can be used to unlock stucked cron tasks.

> **Warning**
>
> Keep in mind that no task should be stucked except in case of a bug or a system failure (database failure during cron execution for example).

Possible options for this command are:

- `-a`, `--all` unlock all tasks
- `-c`, `--cycle[=CYCLE]` execution time (in cycles) from which the task is considered as stuck (delay = task frequency * cycle)
- `-d`, `--delay[=DELAY]` execution time (in seconds) from which the task is considered as stuck (default: 1800)
- `-t`, `--task[=TASK]` `itemtype::name` of task to unlock (e.g: `MailCollector::mailgate`)

See also *console options*.

6.8 Suspend/Resume plugins execution

Added in version 11.0.

The `php bin/console plugin:suspend_execution` and the `php bin/console plugin:resume_execution` commands can be used to suspend or resume the plugins execution.

6.9 Plugins tools

Added in version 9.5.

Some command line tolls are also available to manage plugins from command line:

- `glpi:plugin:install`
- `glpi:plugin:activate`
- `glpi:plugin:deactivate`

In order to install `MyGreatPlugin`; you should end with something like:

```
$ ./bin/console glpi:plugin:install MyGreatPlugin
$ ./bin/console glpi:plugin:activate MyGreatPlugin
```

Each of those plugin commands can take a plugin name as argument, or the `--all` flag to be ran on all plugins.

6.10 Migration tools

6.10.1 From MyISAM to InnoDB

Added in version 9.3.0.

Since version 9.3.0, GLPI uses the `InnoDB` engine instead of previously used `MyISAM` engine.

The `php bin/console glpi:migration:myisam_to_innodb` command can be used to migrate exiting tables to `InnoDB` engine.

6.10.2 Missing timestamps builder

Added in version 9.1.0.

Prior to GLPI 9.1.0, fields corresponding to creation and modification dates were not existing.

The `php bin/console glpi:migration:build_missing_timestamps` command can be used to rebuild missing values using available logs.

6.10.3 Use timestamp data type

Added in version 9.5.0.

Many date fields were using the `DATETIME` type, but this does not allow to rely on timezones. Timezone support requires all fields to use `TIMESTAMP` data type, but this query can be very long and therefore is not included in the standard update process.

Using the `glpi:migration:timestamps` command will change those fields to the correct data type, but read *documentation on timezones* before.

> **Warning**
>
> Ensure to backup your database before!

6.10.4 Migrate Domains plugin

Added in version 9.5.0.

Domains in GLPI have evolved from a simple dropdown to a more complex object, including records management among others. Therefore, the Domains plugins feature are now included in core.

To migrate your plugin data; use the `glpi:migration:domains_plugin_to_core` command. Presence of the plugin is mandatory so checks can be run, you can use the `--without-plugin` switch but this is not recommended. If you were using an older version of the plugin than the one required, you can use the `--update-plugin` flag.

At the end, all domains types, domains and item relations will be migrated in core tables.

6.10.5 Migrate Racks plugin

Added in version 9.5.0.

Since GLPI 9.3.0, data center infrastructure management is available as a core feature. A migration script from Racks plugin was provided inside the `scripts` directory. Since GLPI 9.5.0, this migration script has been refactored and moved inside the CLI console.

To migrate your plugin data; use the `glpi:migration:racks_plugin_to_core` command. Presence of the plugin is mandatory so checks can be run, you can use the `--without-plugin` switch but this is not recommended. If you were using an older version of the plugin than the one required, you can use the `--update-plugin` flag.

(cc) BY-NC-ND

CHAPTER

SEVEN

ADVANCED CONFIGURATION

7.1 SSL connection to database

Added in version 9.5.0.

Once installation is done, you can update the `config/config_db.php` to define SSL connection parameters. Available parameters corresponds to parameters used by mysqli::ssl_set():

- `$dbssl` defines if connection should use SSL (*false* per default)
- `$dbsslkey` path name to the key file (*null* per default)
- `$dbsslcert` path name to the certificate file (*null* per default)
- `$dbsslca` path name to the certificate authority file (*null* per default)
- `$dbsslcapath` pathname to a directory that contains trusted SSL CA certificates in PEM format (*null* per default)
- `$dbsslcacipher` list of allowable ciphers to use for SSL encryption (*null* per default)

Warning

For now it is not possible to define SSL connection parameters prior or during the installation process. It has to be done once installation has been done.

(cc) BY-NC-ND

(cc) BY-NC-ND