

# Installing eZ Publish on a Linux-UNIX based system

The requirements for doing a normal installation must be met, read the ["Requirements for doing a normal installation"](#) section first!

This section will guide you through the following steps:

- 1 Setting up a database
  - 1.1 MySQL
  - 1.2 PostgreSQL
- 2 Downloading eZ Publish
- 3 Unpacking eZ Publish
- 4 Setting up folder permission
- 5 Updating Zetacomponents
- 6 Link assets
- 7 Initiating the setup wizard
  - 7.1 Document root example
  - 7.2 Home directory example
  - 7.3 Virtual host example
  - 7.4 Defining the desired environment (optional)

## Setting up a database

A database must be created before running the setup wizard. The following text explains how to set up a database using either MySQL or PostgreSQL.

### MySQL

1. Log in as the root user (or any other MySQL user that has the `CREATE`, `CREATE USER` and `GRANT OPTION` privileges):

```
$ mysql --host=<mysql_host> --port=<port> -u <mysql_user>
-p<mysql_password>
```

Note that if MySQL is installed on the same server, the "--host" parameter can be omitted. If the "--port" parameter is omitted, the default port for MySQL traffic will be used (port 3306).

The MySQL client should display a "mysql>" prompt.

2. Create a new database:

```
mysql> CREATE DATABASE <database> CHARACTER SET utf8;
```

3. Grant access permissions:

```
mysql> GRANT ALL ON <database>.* TO <user>@<ezp_host> IDENTIFIED BY
'<password>';
```

Note that if the specified user account does not exist, it will be created.

Reference	Description
<mysql_host>	The hostname of the MySQL database server.
<port>	The port number that will be used to connect to the MySQL database server.
<mysql_user>	The MySQL user (if no user is set up, use "root").
<mysql_password>	The password that belongs to the <mysql_user>.
<database>	The name of the database, for example "my_new_database".
<user>	The username that will be used to access the database.
<ezp_host>	The hostname of the server on which eZ Publish will be running. (may be "localhost" if MySQL is installed on the same server).
<password>	The password you wish to set in order to limit access to the database.

## PostgreSQL

1. Log in as the postgres user (or any other PostgreSQL user that has sufficient [privileges](#) to create roles and databases):

```
$ psql -h <psql_host> -p <port> -U <psql_user> -W
```

Note that if PostgreSQL is installed on the same server, the "-h" parameter can be omitted. If the "-p" parameter is omitted, the default port for PostgreSQL traffic will be used (in most cases, port 5432).

The PostgreSQL client will ask you to specify the password that belongs to the <psql\_user>. If the password is correct, the client should display a "<psql\_user>=#" prompt.

2. Create a new database:

```
postgres=# CREATE DATABASE <database> ENCODING='utf8';
```

3. Create a new user:

```
postgres=# CREATE USER <user> PASSWORD '<password>';
```

4. Grant access permissions:

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE <database> TO <user>;
```

Import the "pgcrypto" module into the new database:

```
postgres=# \c <database>
<database>=# \i '<path_to_pgcrypto>'
```

Reference	Description
<psql_host>	The hostname of the PostgreSQL database server.
<port>	The port number that will be used to connect to the PostgreSQL database server.

<psql_user>	The PostgreSQL user (if no user is set up, use "postgres").
<database>	The name of the database, for example "my_new_database".
<user>	The username that will be used to access the database.
<password>	The password you wish to set in order to limit access to the database.
<path_to_pgcrypto>	The path to the "pgcrypto.sql" file, for example "/usr/share/pgsql/contrib/pgcrypto.sql".

**Note for version 9.1+ of PostgreSQL users:**The following changes might be necessary for these users:

```
postgres=# \c <database>
<database>=# CREATE EXTENSION pgcrypto;
```

## Downloading eZ Publish

The latest community version of eZ Publish can be downloaded from <http://share.ez.no/download-develop/downloads>, while enterprise version is available in your [support portal](#) or via partner portal.

## Unpacking eZ Publish

Use your favorite tool to unpack the downloaded eZ Publish distribution to a web-served directory (a directory that is reachable using a web browser), or in case of virtual host mode it can be any folder. The following example shows how to do this using the tar utility (to unpack a tar.gz file, assuming that the "tar" and the "gzip" utilities are installed on the system):

```
$ tar zxvf ezpublish-<version_number>-gpl.tar.gz -C <web_served_directory>
```

Reference	Description
<version_number>	The version number of eZ Publish that was downloaded.
<web_served_directory>	Full path to a directory that is served by the web server. This can be the path to the document root of the web server, or a personal web-directory (usually called "public_html" or "www", and located inside a user's home directory).

The extraction utility will unpack eZ Publish into a sub-directory called "ezpublish-<version\_number>". Feel free to rename this directory to something more meaningful, for example "my\_site".

## Setting up folder permission

### Important

In the 3 first folder permission setup options, always ensure to run application scripts and the web server with the appropriate UNIX user (must be same as you setup rights for below).

As for Apache you can control which user to use in your Apache configuration. Using [PHP-FPM](#) or [SuExec module](#), you can even specify a user per virtual host.

Several cache, log and config folders must be writable both by the web server and the command line user, use *one* of the following alternatives to do this:

- Using ACL on a system that supports `chmod +a`

These shell commands will give proper permission to the web server and command line users:

```
$ cd /<ezp5-root>/
$ sudo chmod +a "www-data allow
delete,write,append,file_inherit,directory_inherit" \
  ezpublish/{cache,logs,config}
ezpublish_legacy/{design,extension,settings,var} web
$ sudo chmod +a "`whoami` allow
delete,write,append,file_inherit,directory_inherit" \
  ezpublish/{cache,logs,config}
ezpublish_legacy/{design,extension,settings,var} web
```

- **Using ACL on a system that does not support chmod +a**

Some systems don't support chmod +a, but do support another utility called setfacl. You may need to enable ACL support on your partition and install setfacl before using it (as is the case with Ubuntu), like so:

```
$ cd /<ezp5-root>/
$ sudo setfacl -R -m u:www-data:rwx -m u:www-data:rwx \
  ezpublish/{cache,logs,config}
ezpublish_legacy/{design,extension,settings,var} web
$ sudo setfacl -dR -m u:www-data:rwx -m u:`whoami`:rwx \
  ezpublish/{cache,logs,config}
ezpublish_legacy/{design,extension,settings,var} web
```

- **Using chown on systems that don't support ACL**

Some systems don't support ACL at all. You will either need to set your web server's user as the owner of the required directories.

```
$ cd /<ezp5-root>/
$ sudo chown -R www-data:www-data ezpublish/{cache,logs,config}
ezpublish_legacy/{design,extension,settings,var} web
$ sudo find
{ezpublish/{cache,logs,config},ezpublish_legacy/{design,extension,set
tings,var},web} -type d | sudo xargs chmod -R 775
$ sudo find
{ezpublish/{cache,logs,config},ezpublish_legacy/{design,extension,set
tings,var},web} -type f | sudo xargs chmod -R 664
```

- **Using chmod**

If you can't use ACL and aren't allowed to change owner, you can use chmod, making the files writable by everybody. Note that this method really isn't recommended as it allows any user to do anything.

```
$ cd /<ezp5-root>/
$ sudo find
{ezpublish/{cache,logs,config},ezpublish_legacy/{design,extension,settings,var},web} -type d | sudo xargs chmod -R 777
$ sudo find
{ezpublish/{cache,logs,config},ezpublish_legacy/{design,extension,settings,var},web} -type f | sudo xargs chmod -R 666
```

## Updating Zetacomponents

As of eZ Publish 5.2, if you will use eZ Publish on legacy mode the Zetacomponents configuration must be updated. For that you will need to install composer, which you can do by running one of the following command from you eZ Publish root folder:

If you **have curl** installed:

```
curl -sS https://getcomposer.org/installer | php
```

If you **don't have curl** installed:

```
php -r "eval('?'>'.file_get_contents('https://getcomposer.org/installer'))';"
```

After composer installed run the following command **from the `ezpublish_legacy/` folder**:

```
php composer.phar install
```

**Version >= 5.2**  
Composer will need GD to run.

## Link assets

The download file should already have generated these assets for you as of 5.2 / 2013.07, but in case of issues on Windows with symlinks within the download file, make sure to run these commands.

To be able to run eZ Publish 5.2 correctly, assets need to be exposed in the public "web" folder.

The following commands will first symlink eZ Publish 5.2 assets in "Bundles" and the second will symlink assets (design files like images, scripts and css, and files in var folder) from eZ Publish Legacy

```
cd /<ezp5-root>
php ezpublish/console assets:install --symlink web
php ezpublish/console ezpublish:legacy:assets_install --symlink web
php ezpublish/console assetic:dump --env=prod web
```

**Note:** In both cases "web" is the default folder and can be skipped from the command. Further information about alternative options

is available with `-h` just like it is with `"php ezpublish/console -h"`.

#### Warning regarding APC

If you are planning to use APC to speed up your site, please be sure to check the available notes [here](#), before entering the setup wizard chapter.

## Initiating the setup wizard

The setup wizard can be started using a web browser immediately after the previous steps (described in this section) are completed. It will be automatically run the first time someone tries to access/browse the `/ezsetup` url. Let's assume that we are using a server with the host name `"www.example.com"` and that after unpacking, the eZ Publish directory was renamed to `"my_site"`.

### Document root example

If eZ Publish was unpacked into a directory called `"my_site"` under the document root, the setup wizard can be initiated by browsing the following URL: `http://www.example.com/my_site/web/index.php/ezsetup`.

### Home directory example

If eZ Publish was unpacked to a web-served directory located inside the home directory of a user with the user name `"peter"`, (usually called `"public_html"`, `"www"`, `"http"`, `"html"` or `"web"`), the setup wizard can be initiated by browsing the following URL: `http://www.example.com/~peter/my_site/web/index.php/ezsetup`.

### Virtual host example

If you have setup the host name `www.example.com` to point to the web folder of eZ Publish using [rewrite](#) rules, then you should access it at: `http://www.example.com/ezsetup`.

Refer to the ["The setup wizard"](#) section for a detailed description of the web based setup wizard.

## Defining the desired environment (optional)

This is a procedure to be done for the case you need to set up several multiple environments for development purposes.

Environment configuration is a new feature introduced as of eZ Publish 5.2 and eZ Publish Community Project 2013.06.

You can configure several environments, from production, development or staging, even if for each one of them you need require using different configurations sets.

In the `VirtualHost` example in the [Virtual host setup](#) chapter the required `VirtualHost` configurations have been already included. You can switch to the desired environment by setting the `ENVIRONMENT` environment variable to `"prod"`, `"dev"` or other custom value, as you can see in the following example:

```
# Environment.
# Possible values: "prod" and "dev" out-of-the-box, other values
possible with proper configuration (described below)
# Defaults to "prod" if omitted
SetEnv ENVIRONMENT "prod"
```

If you want to use a custom environment (something else than `"prod"` and `"dev"`) the next step is to create the dedicated configuration files for your environment:

- `ezpublish/config/config_<env_name>.yml`

- `ezpublish/config/ezpublish_<env_name>.yml`

The name used as `<env_name>` will be the one that can be used as value of the `ENVIRONMENT` environment variable.

Those files must import the main configuration file, just like the default `config_dev.yml` already does. Here's an example:

```
imports:
  - { resource: config.yml }
```

This allows you to override settings defined in the main configuration file, depending on your environment (like the DB settings or any other setting you may want to override).