

---

# **bits4docs Documentation**

**GlobalTech Translations**

**Aug 20, 2022**

## Contents

<b>1</b>	<b>Migrating Joomla from shared hosting to an unmanaged VPS</b>	<b>2</b>
1.1	Prerequisites . . . . .	3
1.2	Performing the backup on your shared hosting account . . . . .	4
1.3	Preparing your new VPS server to host your website . . . . .	8
1.3.1	Option 1: Adding an entry to your hosts file . . . . .	8
1.3.1.1	Clearing the DNS cache after updating your hosts file . . . . .	9
1.3.2	Option 2: Adjusting the DNS records . . . . .	9
1.3.3	Uploading the required files to your VPS server . . . . .	11
1.3.4	Creating an empty MySQL database . . . . .	12
1.3.5	Importing the SQL dump into your new database . . . . .	12
1.3.6	Setting up a virtual host on your VPS . . . . .	12
1.4	Restoring your Joomla website on the VPS . . . . .	14
1.5	Installing Let's Encrypt certificates with Certbot . . . . .	19
<b>2</b>	<b>Hardening Apache on an unmanaged VPS</b>	<b>21</b>
2.1	Prerequisites . . . . .	22
2.2	Running Apache with an unprivileged user . . . . .	22
2.3	Hiding your operating system and Apache version . . . . .	22
2.4	Disabling open directory listings . . . . .	24
2.5	Installing a web application firewall . . . . .	25

You will find here FOSS tutorials about the following topics:

- Technical communication
- Software documentation
- Front/backend web development

**About the author:**

Fayçal Alami-Hassani - @GlobalTech Translations<sup>1</sup> - @gnufcl@fosstodon.org<sup>2</sup>

- Technical communicator, translator and interpreter
- Markup: reStructuredText, Markdown, DocBook, XML
- Web development: HTML, CSS, PHP, MySQL, JavaScript, jQuery
- Text editors: Nano, Atom, Sublime Text
- Version control: Git, CVS
- OS: Debian, Fedora
- Now learning: Docs-as-Code based on Antora<sup>3</sup> with AsciiDoc<sup>4</sup>
- Currently reading:



Web Security for Developers - Real Threats, Practical Defense - Malcolm McDonald - No Starch Press - ISBN: 978-1-59327-994-3



Eloquent JavaScript - A Modern Introduction to Programming - Marijn Haverbeke - No Starch Press - ISBN: 978-1-59327950-9

---

**Note:** This project is under active development. If you have any questions, please send an email to: info[@]globaltech-translations[.]com - PGP KeyID: 0x52D6AF10

---

---

<sup>1</sup> <https://globaltech-translations.com>

<sup>2</sup> <https://fosstodon.org/@gnufcl>

<sup>3</sup> <https://antora.org/>

<sup>4</sup> <https://asciidoc-py.github.io/>

## Migrating Joomla from shared hosting to an unmanaged VPS

Published on May 11, 2022 by Fayçal Alami-Hassani @gnufcl@fosstodon.org<sup>5</sup>



Fig. 1: Picture by Stéphane Wootha Richard under CC BY-SA 4.0<sup>6</sup> License

Migrating a website from a shared hosting plan to an unmanaged VPS might seem a daunting task at the beginning. Depending on your acquaintance with server administration, it may take you a few days to several weeks to gather all the relevant information and organize the required steps in a structured, logical way to set up a custom migration project.

There are plenty of resources about the topic, both online and offline. However, most of them only describe a tiny part of the entire process.

The biggest challenges that you might face when migrating from shared hosting to an unmanaged VPS are related to the following topics:

<sup>5</sup> <https://fosstodon.org/@gnufcl>

<sup>6</sup> <https://creativecommons.org/licenses/by-sa/4.0/deed.en>

- Software stack to use, i.e. operating system, database technology, web server, and programming language.
- Backup, data transfer, and restore strategy.
- Choosing between server administration with the Linux command line (CLI) or GUI-based hosting panels such as cPanel, Plesk, and Froxlor.
- User permission management.
- DNS settings and configuration.
- Testing the website before the actual domain transfer.
- SSL certificate management.

In this tutorial, you will learn how to migrate a Joomla website from a shared hosting provider to an unmanaged VPS server.

---

**Note:** When we talk about virtual private servers (VPS), we distinguish between two main categories: *managed* vs. *unmanaged* VPS. Put in a few words, An unmanaged VPS plan gives you full control over the technical aspects of your server management and administration. There is no technical support to resolve issues and you are responsible for maintaining your server, including the installation of patches and updates, among other things.

---

## 1.1 Prerequisites

This tutorial assumes that you already have the following:

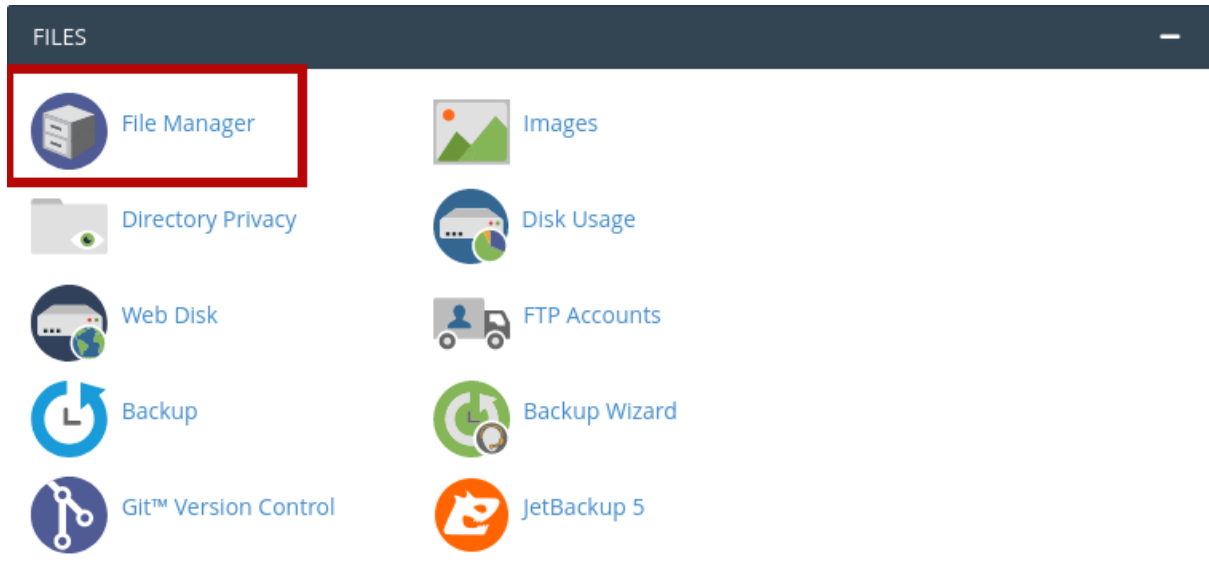
- LAMP stack installed on your VPS server. LAMP is short for “Linux + Apache + MySQL + PHP”.
- Non-root user with sudo privileges on your VPS server.
- SSH access to your VPS server.

Before you follow this guide, make sure to configure your environment accordingly.

## 1.2 Performing the backup on your shared hosting account

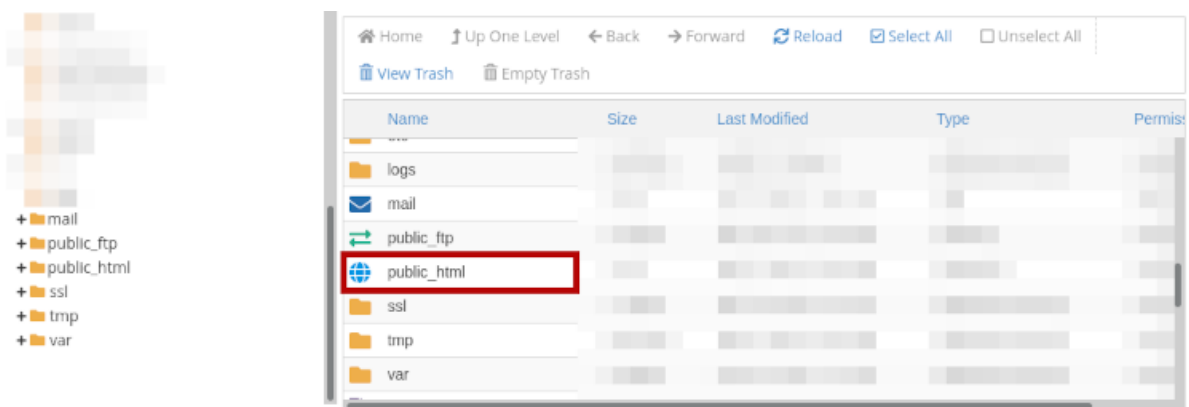
To backup your website files and the corresponding database, follow these steps:

1. Login to your shared hosting account.
2. In your cPanel, go to **File Manager** > **public\_html**.

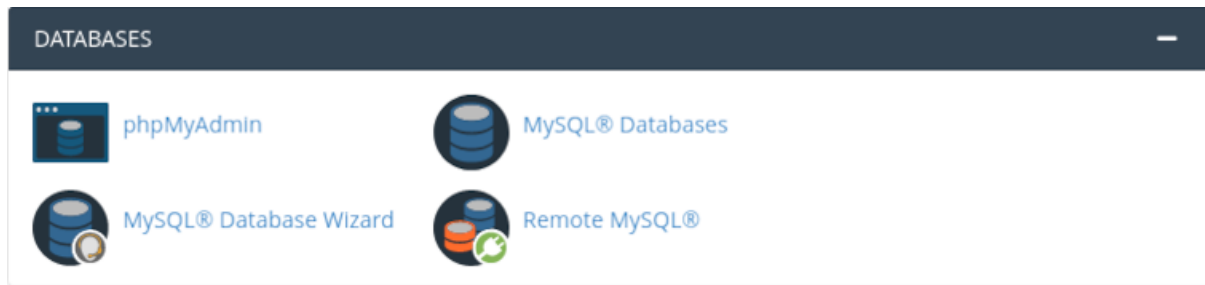


3. Inside the **public\_html** directory, check your database login credentials in the file 'configuration.php'. You should look for the following entries:

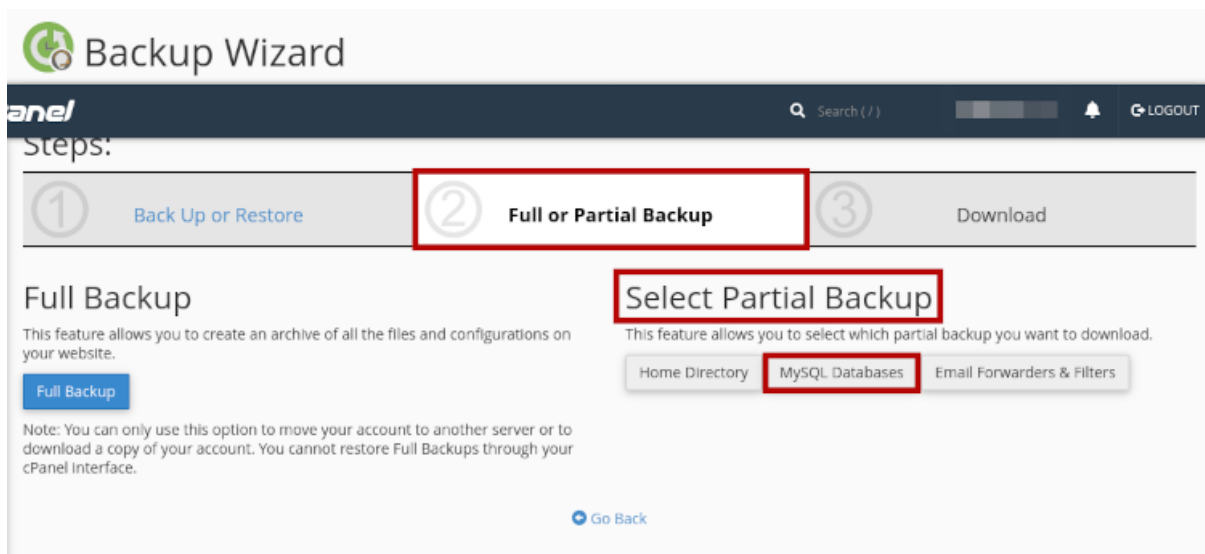
- public \$user: Database user name
- public \$password: Database password
- public \$db: Database name
- public \$dbprefix: Database prefix



4. In the cPanel main menu, go to **Backup** > **Databases** > **MySQL® Database Wizard**.



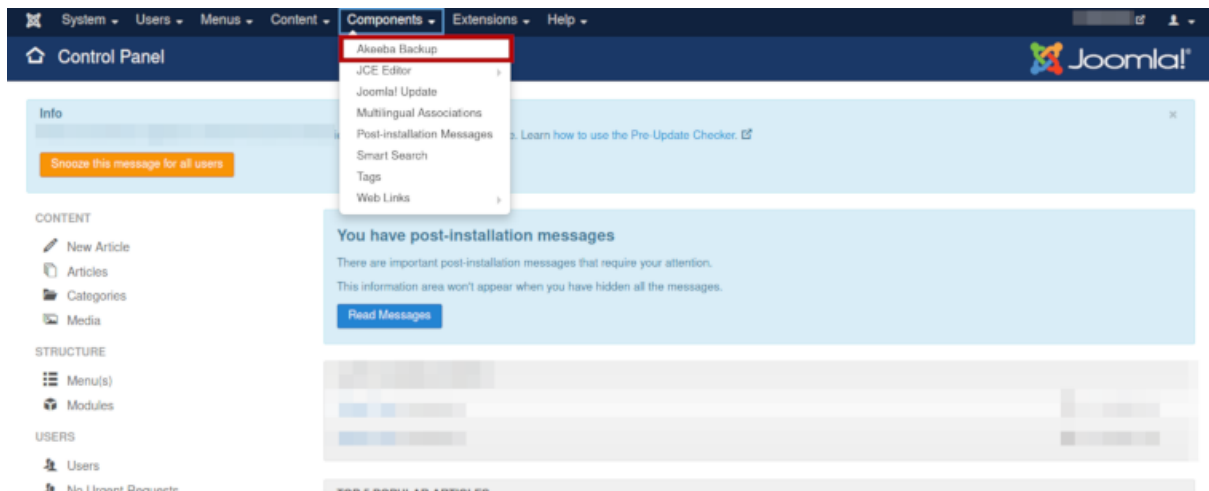
The Backup Wizard provides multiple options. Select **Full or Partial Backup** > **Select Partial Backup** > **MySQL Databases**. This will allow you to download a backup of the MySQL database(s) of your Joomla website to your local machine.



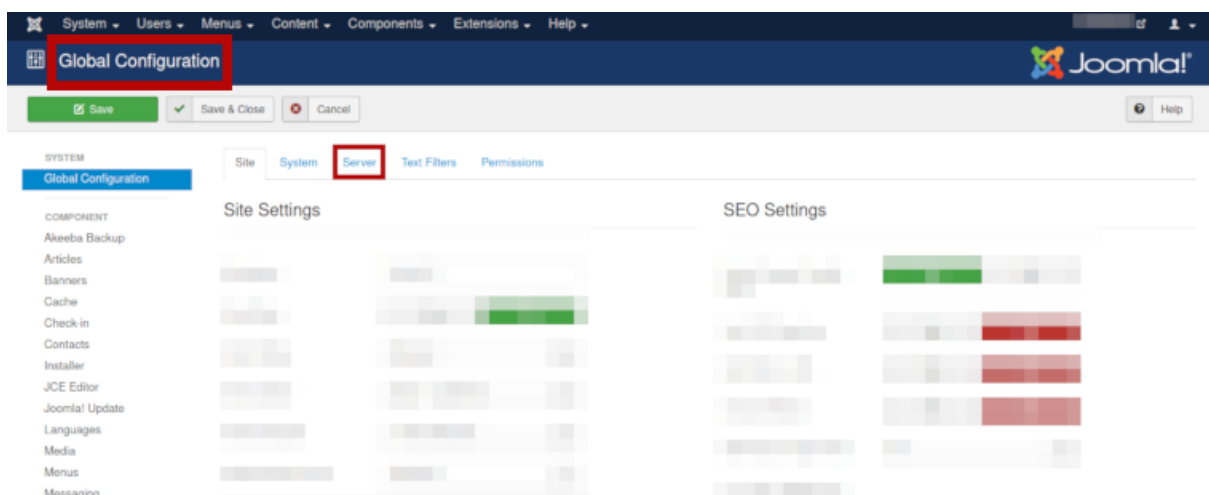
5. Login to the backend of your Joomla website.



6. To backup the website files, we will use an extension called [Akeeba Backup](https://www.akeeba.com/products/akeeba-backup.html)<sup>7</sup>.



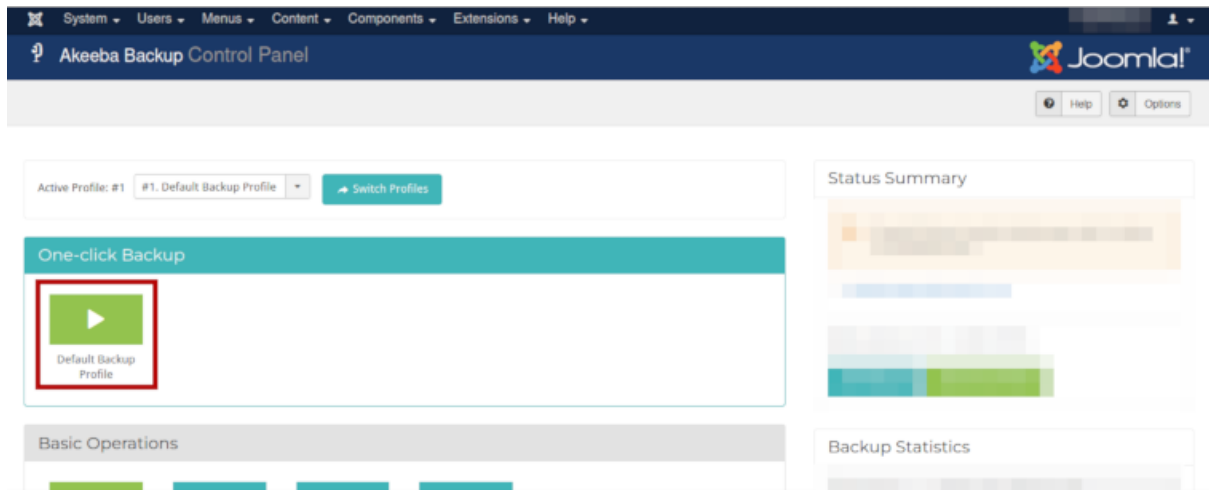
7. Before making a backup with akeeba, make sure to disable SSL. To do so, navigate to **System > Global Configuration > Server > Force HTTPS**. Select the option **None** from the drop-down menu.



<sup>7</sup> <https://www.akeeba.com/products/akeeba-backup.html>



8. Next, go to **Components > Akeeba Backup > One-Click Backup > Default Backup Profile**



9. Once the backup process has completed, click on the “i” button below the green “Download” button on the right to display your “Backup Archive Information”.

**Note:** An Akeeba backup file has a .jpa extension.

<input type="checkbox"/>	ID	Frozen	Description	Profile	Duration	Status	Size	Manage & Download
<input type="checkbox"/>	19			#1. Default Backup Profile Full site backup				<a href="#">Download</a> <a href="#">View Log</a>
<input type="checkbox"/>	18			#1. Default Backup Profile Full site backup				<a href="#">Download</a> <a href="#">View Log</a>
<input type="checkbox"/>	17			#1. Default Backup Profile				<a href="#">Download</a> <a href="#">View Log</a>

10. In your shared hosting account, navigate to the location of your Akeeba backup file through **File Manager > public\_html > path-to-akeeba-backup**. Download the .jpa file to your local machine.
11. Now that you have downloaded your backup file, you need to re-enable SSL for your entire website. On your shared hosting account, navigate to the folder `public_html` and open the file “configuration.php”.
- Search for the entry `public $force_ssl` and switch the value from 0 to 2:

```
public $force_ssl = 2
```

12. Save your changes and return to your Joomla Backend. Go to **System > Global Configuration > Server**.
13. Navigate to the option **Force HTTPS** and select **Entire Site** from the drop-down menu.
14. Download the [Akeeba Kickstart Core](https://www.akeeba.com/products/akeeba-kickstart.html)<sup>8</sup> by clicking on the button **Download Core**.
15. In the next page that will open, click on the green button **Download Core v.xxx**, where xxx refers to the current version number. This will download a .zip file containing the file `kickstart.php`. We will place this php file in the root of our site to restore the Joomla backup.

<sup>8</sup> <https://www.akeeba.com/products/akeeba-kickstart.html>

## 1.3 Preparing your new VPS server to host your website



Fig. 2: “Spacedog Repairman” by Katharsisdrill<sup>Page 8, 9</sup> under CC BY 4.0<sup>10</sup> License

You need to test your website on the new VPS before performing the actual domain transfer from your shared hosting to the new VPS.

### 1.3.1 Option 1: Adding an entry to your hosts file

On linux systems, the `/etc/hosts` file maps hostnames to IP addresses.

To edit the hosts file on your system, type the following command:

```
$ sudo nano /etc/hosts
```

Add the following lines to the bottom of the hosts file:

```
1 IP_address_of_your_VPS    domainname.com
2 IP_address_of_your_VPS    www.domainname.com
```

Replace `domainname.com` by your actual domain name, then press `Ctrl + O` to save your changes and `Ctrl + X` to close the nano editor.

<sup>9</sup> <https://katharsisdrill.art>

<sup>10</sup> <https://creativecommons.org/licenses/by/4.0/>

### 1.3.1.1 Clearing the DNS cache after updating your hosts file

**Note:** On some Linux systems, you may need to flush the DNS cache in order to update the domain resolution to the new IP address. On Debian-based distros, caching DNS queries is performed with the `systemd-resolved` daemon.

To find out if `systemd-resolved` is running on your system, type the following command in your terminal:

```
$ sudo systemctl is-active systemd-resolved
```

If the output shows the status **active**, it means that the daemon is up and running.

To clear the DNS cache, run the following command:

```
$ sudo systemd-resolve --flush-caches
```

You can now check the cache size with the command:

```
$ sudo systemd-resolve --statistics
```

The entry `Current Cache Size: 0` will appear in the output if the DNS cache has been cleared successfully.

## 1.3.2 Option 2: Adjusting the DNS records

For testing purposes, you can create a DNS Zone for your domain on the new VPS server. The DNS Zone section allows you to configure your domain for the different services that you intend to provide.

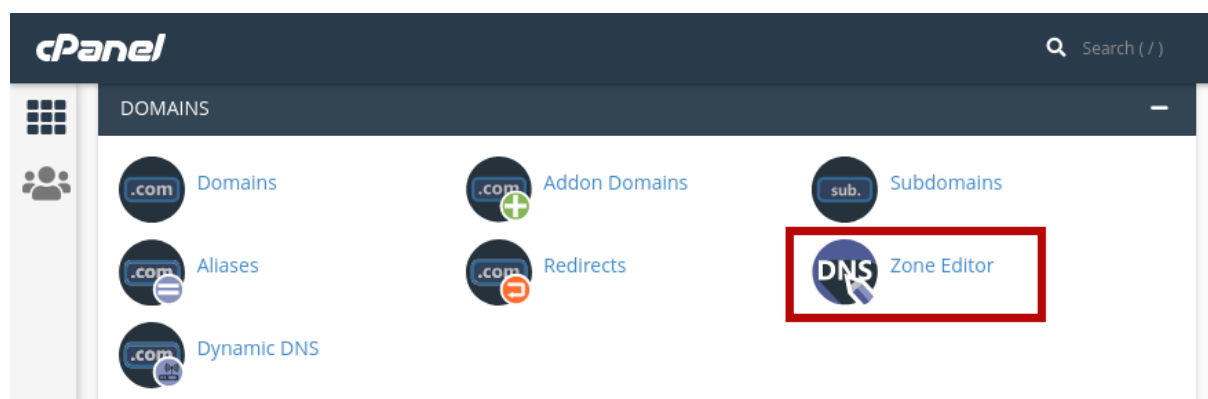
Suppose that you already have a domain that is registered with another service provider. To avoid any service interruptions before transferring your domain to a new provider, you can add a DNS zone before you begin the domain name transfer process.

**Warning:** Make sure to configure the DNS servers accordingly to take the DNS zone into account.

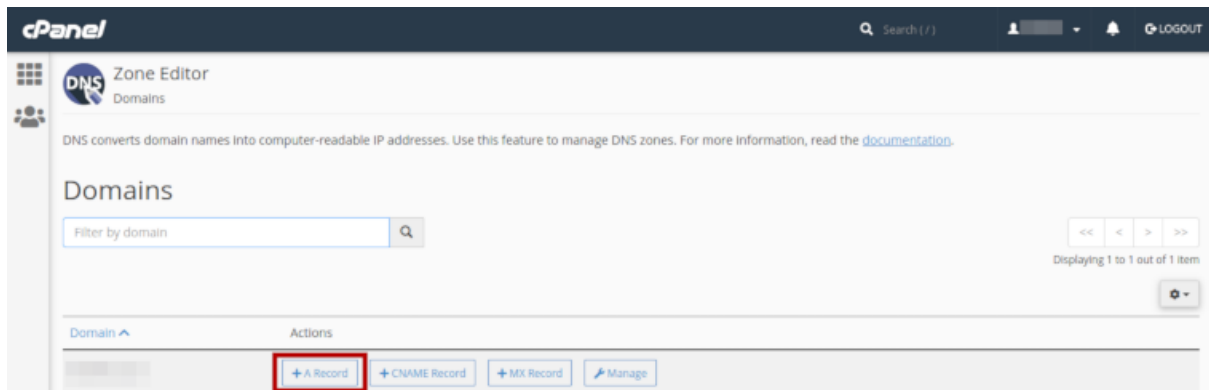
Adding a DNS Zone generally involves the following steps:

- Entering a domain name in the DNS Zone section
- Choosing whether you want to enable minimal records, the default is No
- Checking the pricing details
- Confirming the Special Terms for the Webdomain and the General terms of service

1. Login to your shared hosting account.
2. In your cPanel, go to **Domains > Zone Editor**.



3. In your **Zone Editor**, go to **Actions**, then select the tab **+A Record**. A new window with the title **Add an A Record for “yourdomain.com”** will open.



4. In the **Name** field, enter your fully-qualified domain name (FQDN) by appending a dot at the end of your domain name: `joomla-domain.com.`

The screenshot shows a dialog box titled 'Add an A Record for "example. .com."'. It has two input fields: 'Name' and 'Address'. The 'Name' field is highlighted with a red box and contains the text 'example. .com.'. The 'Address' field is empty. At the bottom right, there are two buttons: 'Add An A Record' and 'Cancel'.

5. In the **Address** field, enter the IP address of your new Virtual Private Server (VPS). Remember that you want the DNS server from your shared hosting plan to point to your new VPS. By doing so, you can test if everything is working fine before requesting a domain transfer.

### 1.3.3 Uploading the required files to your VPS server

To restore the Joomla website on your new VPS server, you will need these three files:

1. The Akeeba backup file with the `.jpa` extension
2. The SQL dump file that we have generated with the Backup Wizard in cPanel
3. The `kickstart.php` file that we have extracted from the Akeeba Kickstart Core

To upload each of these files to your VPS server via ssh, use the `scp` command as shown below:

```
$ scp -P PORT-NUMBER /PATH/TO/FILE USER@IP-ADDRESS:PATH/TO/DESIRED/DESTINATION
```

Replace the parameters of the `scp` command by their actual values, i.e.:

Parameter	Description
PORT-NUMBER	the port number you are using to connect to your VPS server through ssh. The default port number for ssh connections is 22, but you can set a different port number for your ssh connection.
/PATH/TO/FILE	the path to the file that you want to upload to your VPS server
USER	The active ssh user. You will find all your ssh credentials in the corresponding section on your customer page. If still doubt, contact your VPS provider.
IP-ADDRESS	The IP address of your VPS server

### 1.3.4 Creating an empty MySQL database

In the section *Performing the backup on your shared hosting account*, you made a backup of your MySQL database. You will now create an empty database on your VPS to import the SQL dump file.

Login to MySQL by typing the following command in your VPS terminal:

```
$ mysql -u root -p
```

Once you enter your password, you will get access to the MySQL shell prompt. Now, you will create a new database with the following command:

```
mysql> CREATE DATABASE new_database;
```

**Note:** You can replace the value *new\_database* by a name that suits your needs. When choosing a name for your MySQL database, follow these naming convention rules:

- Use lowercase
- Use only alphabetical characters
- Do not use numeric characters
- Avoid using prefixes
- Give your database a self-explanatory name

If everything went fine, the shell prompt will display the following output:

```
1 Output
2 Query OK, 1 row affected (0.00 sec)
```

### 1.3.5 Importing the SQL dump into your new database

We will now assign a user *bob* to our newly created database by typing the command below. Make sure to change the username *bob* and the default password to a strong password of your own:

```
mysql> CREATE USER 'bob'@'localhost' IDENTIFIED BY 'password';
```

Use the key combination `Ctrl + D` to leave the MySQL shell prompt.

In the VPS terminal, you can now import the SQL dump file with the following command:

```
$ mysql -u 'username' -p 'new_database' < 'data-dump.sql'
```

### 1.3.6 Setting up a virtual host on your VPS

At the beginning of this guide, we mentioned in the *Requirements* section that we will use Apache as a web server in our stack. Apache allows you to configure multiple virtual hosts, making it possible to host more than one domain on a single server.

In our particular scenario, this means that we can host all the following domains on our VPS, as long we have sufficient storage, RAM, CPU, and IOPS resources:

- techwriting-website.com
- webdev-website.net
- infosec-website.org
- etc.

1. Before you set up a virtual host, make sure that Apache is up and running on your VPS. To do so, type the following command:

```
$ sudo systemctl start apache2
```

2. To start the Apache2 server automatically on boot, use the following command:

```
$ sudo systemctl enable apache2
```

3. From now on, you will have to create a dedicated folder under `/var/www` for each new domain that you want to host on your VPS. For instance, to create the domain that will host your Joomla backup on the new VPS, type the following command:

```
$ sudo mkdir /var/www/joomla-domain
```

Replace the parameter `joomla-domain` by the actual domain name that you are using for your Joomla website.

4. Assign ownership of the newly created directory with the `$USER` environment variable by using the command below. The `$USER` environment variable is identical to the `$LOGNAME` environment variable, which represents the currently logged-in user:

```
$ sudo chown -R $USER:$USER /var/www/joomla-domain
```

5. Make sure that you granted the correct web root permissions by typing the command below. The folder's owner should have **read/write/execute** permissions, while group and others should only have **read/execute** privileges.

```
$ sudo chmod -R 755 /var/www/joomla-domain
```

---

**Note:** The default permissions on a web server are 755 for directories and 644 for files.

---

6. In order for Apache to serve your content, you need to create an “Apache virtual host configuration file”. To do so, we will create a new empty file with the nano editor:

```
$ sudo nano /etc/apache2/sites-available/joomla-domain.conf
```

Put the following directives inside the configuration file:

```
1 <VirtualHost *:80>
2 ServerAdmin webadmin@localhost
3 ServerName joomla-domain
4 ServerAlias www.joomla-domain
5 DocumentRoot /var/www/joomla-domain
6 ErrorLog ${APACHE_LOG_DIR}/error.log
7 CustomLog ${APACHE_LOG_DIR}/access.log combined
8 </VirtualHost>
```

---

**Note:** The email provided in the field `ServerAdmin`<sup>[2]</sup> is a placeholder. Make sure to use a working email address where the administrator of your Joomla domain can receive notifications. Also replace the parameters `joomla-domain`<sup>[3]</sup> and `www.joomla-domain`<sup>[4]</sup> by the actual domain name of your Joomla website.

---

Once you have entered the relevant information, press `Ctrl + O` to save your changes and `Ctrl + X` to close the nano editor.

7. We will now use a sample `index.html` file to check if our virtual host is working properly. To do so, we will create a new empty file with the nano editor:

```
$ sudo nano /var/www/joomla-domain/index.html
```

Add the following lines in the empty file:

```

1 <html>
2   <head>
3     <title>Welcome to my joomla-domain</title>
4   </head>
5   <body>
6     <h1>The joomla-domain virtual host is up and running</h1>
7   </body>
8 </html>

```

8. **a2ensite** is a script that allows you to enable a specific site within the Apache2 configuration. This is achieved by creating symlinks (short for symbolic links) within the `/etc/apache2/sites-enabled` directory.

We will use **a2ensite** to enable our newly created site on the VPS. To do so, type the command:

```
$ sudo a2ensite joomla-domain.conf
```

9. In the same manner that **a2ensite** adds symbolic links to enable a specific site, **a2dissite** removes symbolic links to disable a site.

In our particular case, we will use **a2dissite** to disable the default configuration file called `000-default.conf`.

This default file is a fallback for all the requests that do not specify a configuration file.

To disable the default configuration file, type the following command:

```
$ sudo a2dissite 000-default.conf
```

10. Make sure that your configuration does not contain any errors by running the following command:

```
$ sudo apache2ctl configtest
```

If everything is fine, you should get the following output:

```

1 Output
2 Syntax OK

```

11. Each time you modify the Apache configuration, you need to restart the Apache service. Use the following command to restart Apache:

```
$ sudo systemctl restart apache2
```

12. To check that the web server is serving your content now, go to `http://joomla-domain` in your browser. You should see the following output:

**The joomla-domain virtual host is up and running**

## 1.4 Restoring your Joomla website on the VPS

To restore your Joomla website on the VPS server, you first have to move the file `kickstart.php` and your Akeeba backup file `backup-file.jpa` to the root of your site on the VPS, i.e. inside the folder `/var/www/joomla-domain`.

1. If you have not already placed both files in the root of your Joomla site, open the terminal, then navigate to the folder containing both files. Next, type the following commands:

```

1 $ sudo mv kickstart.php /var/www/joomla-domain
2 $ sudo mv backup-file.jpa /var/www/joomla-domain

```

Replace the parameter `backup-file.jpa` by the actual backup file name.



2. In your browser, type the following address:

`http://joomla-domain/kickstart.php`

3. The welcome screen of Akeeba Kickstart appears. Press the button **Click here or press ESC to close this message** on the bottom left.

### Things you should know about Akeeba Kickstart

1. Kickstart is not an installer. It is an archive extraction tool. The actual installer was put inside the archive file at backup time.
2. Kickstart is bound by your server's configuration. As such, it may not work at all.
3. You should download and upload your archive files using FTP in Binary transfer mode. Any other method could lead to a corrupt backup archive and restoration failure.
4. Post-restoration site load errors are usually caused by .htaccess or php.ini directives. You should understand that blank pages, 404 and 500 errors can usually be worked around by editing the aforementioned files. It is not our job to mess with your configuration files, because this could be dangerous for your site.
5. Kickstart overwrites files without a warning. If you are not sure that you are OK with that do not continue.
6. Trying to restore to the temporary URL of a cPanel host (e.g. `http://1.2.3.4/~username`) will lead to restoration failure and your site will appear to be not working. This is normal and it's just how your server and CMS software work.
7. You are supposed to read the documentation before using this software. Most issues can be avoided, or easily worked around, by understanding how this software works.
8. This text does not imply that there is a problem detected. It is standard text displayed every time you launch Kickstart.

[Click here or press ESC to close this message](#)

4. The graphical interface of the **Akeeba archive extraction tool** will appear on your browser screen.

## Akeeba Kickstart Core

Want some help to use this tool? Read this first: [Quick Start Guide](#)

### 1 Select a backup archive

Archive directory:  Reload

Archive file:

Archive Password (for JPS files)

Start

### 2 Select an extraction method

Write to files: Hybrid (use FTP only if needed)

Ignore most errors ☐

5. Scroll to the bottom of the screen, then click on the **Start** green button under the section **Extract files**.

Restore file permissions ☐

Applies the file permissions (but NOT file ownership) which was stored at backup time. Only works with JPA and JPS archives. Does not work on Windows (PHP does not offer such a feature).

Files to extract

Enter a file path such as images/cat.png or shell pattern such as images/\* .png on each line. Only files matching this list will be written to disk. Leave empty to extract everything (default).

**4 Extract files**

**Start**

Copyright © 2008-2022 Nicholas K. Dionysopoulos / Akeeba Ltd. All legal rights reserved.  
This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

6. The extraction progress window will appear. Once the files are extracted, click on the green button **Run the Installer** under **Restoration and Cleanup**

**Akeeba Kickstart Core**

**5 Extracting**

Do not close this window while the extraction is in progress

7. The site restoration script of Akeeba Backup will perform a pre-installation check. This allows you to take the necessary actions to correct any possible issues. If everything is fine, press the button → **Next** on the top right side of the screen.

**Akeeba Backup Site Restoration Script** **Start over** **Check again** **→ Next**

**No idea what you are supposed to do? Don't panic!** **Read the documentation page** **Watch the tutorial video**

**Pre-installation > Database Restoration > Site Setup > Finished**

**Pre-installation check**

If any of these items is not supported (marked as No) then please take actions to correct them. Failure to do so could lead to your site not functioning correctly.

Setting	Current
PHP Version >=	✓ Yes
Magic Quotes GPC	✓ Yes
Register Globals	✓ Yes
Zlib Compression Support	✓ Yes

**Recommended settings**

These settings are recommended for PHP in order to ensure full compatibility with your site's software. However, your site should still operate if your settings do not quite match the recommended configuration.

Setting	Recommended	Current
Safe Mode	Off	✓ Off
Display Errors	Off	✓ Off
File Uploads	On	✓ On
Magic Quotes Runtime	Off	✓ Off

8. In the screen that appears, enter the *credentials* for the MySQL database that you have created. Once you have entered all the required information, click on the button → **Next** on the top right side of the screen.

The screenshot shows the 'Akeeba Backup Site Restoration Script' interface. At the top, there are navigation buttons: 'Previous', 'Skip Restoration', and 'Next'. Below this is a help bar with an information icon and the text 'No idea what you are supposed to do? Don't panic! Read the documentation page'. A breadcrumb trail indicates the current step: 'Pre-installation > Database Restoration > Site Setup > Finished'. The main heading is 'Restoration of site's main database'. The interface is divided into two panels: 'Connection information' and 'Advanced options'. The 'Connection information' panel contains fields for 'Database type' (a dropdown menu), 'Database server host name', 'User name', 'Password' (masked with dots), 'Database name', and 'Database table name prefix'. The 'Advanced options' panel includes a 'With existing tables' dropdown set to 'Drop', and several checkboxes: 'Suppress foreign key checks' (checked), 'No auto value on zero' (checked), 'Use REPLACE instead of INSERT' (unchecked), 'Force UTF-8 collation on database' (unchecked), 'Force UTF-8 collation on tables' (unchecked), and 'Allow UTF8MB4 auto-detection' (unchecked). A 'Show / hide help' button is located in the top right corner of the main content area.

9. A **Database Restoration Progress Bar** will appear. If the restoration was successful, you will see the message: **The database restoration was successful.**

This screenshot shows the same interface as the previous one, but with a modal dialog box overlaid. The dialog is titled 'Database restoration' and features a green checkmark icon followed by the message 'The database restoration was successful'. Below the message, it says 'Click the button below to proceed to the next step' and includes a green button labeled 'Next step'. The background interface is dimmed, showing the same 'Connection information' and 'Advanced options' panels as before.

10. In the screen that appears, enter the site parameters such as “Site name” and “Live site URL”. Once you have entered all the required information, click on the button → **Next** on the top right side of the screen.

The screenshot shows the 'Akeeba Backup Site Restoration Script' interface. At the top, there are 'Previous' and 'Next' buttons. Below the header, a message says 'No idea what you are supposed to do? Don't panic! Read the documentation page'. A breadcrumb trail indicates the current step: 'Pre-installation > Database Restoration > Site Setup > Finished'. A 'Show / hide help' button is also present.

The main configuration area is divided into two panels:

- Site Parameters:** This panel contains several input fields: 'Site name', 'Site e-mail address', 'Site e-mail sender name', 'Live site URL', 'Force SSL' (set to 'None'), 'Cookie domain', and 'Cookie path'. At the bottom, there is a 'Turn on mail sending' toggle with 'No' and 'Yes' options.
- Server-specific configuration files:** This panel contains an information box stating: 'Files which modify the way your server behaves when serving your site may cause site loading issues when restoring to a new host. Use the options below to reset them to Joomla! defaults.' Below this, there are three checkboxes:
  - ☐ Remove .user.ini and / or php.ini files from the main site directories
  - ☒ Replace main .htaccess file with default
  - ☐ Delete the .htaccess and .htpasswd files in the administrator directory

11. If the restoration process has completed successfully, you will see the screen below. You can now visit you site's frontend or login to the site's backend.

The screenshot shows the 'Akeeba Kickstart Professional' interface after a successful restoration. The main heading is '6 Restoration and Clean Up'. Below this, there are two prominent green buttons: 'Visit your site's frontend' and 'Visit your site's backend'. A link is provided: 'Something not working after the restoration? Click here for troubleshooting instructions.' At the bottom, there is a footer with copyright information: 'Copyright © 2008–2019 Nicholas K. Dionysopoulos / Akeeba Backup. All legal rights reserved. This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. Design credits: Internet Inspired, heavily modified by AkeebaBackup.com'.

## 1.5 Installing Let's Encrypt certificates with Certbot

Now that you have restored your Joomla website, remember that you had to *disable SSL* before making the backup with Akeeba. To protect your website, you can install TLS/SSL certificates from Let's Encrypt.

Let's Encrypt is a non-profit and open certificate authority managed by the [Internet Security Research Group](https://www.abetterinternet.org/)<sup>11</sup>, a public-benefit corporation based in California.

To issue the TLS/SSL certificates and install them automatically on the web server, we are going to use Certbot, an open-source software developed by the [Electronic Frontier Foundation](https://www.eff.org/)<sup>12</sup>.



Fig. 3: Picture by the Electronic Frontier Foundation under [CC BY 2.0](https://creativecommons.org/licenses/by/2.0/)<sup>Page 19, 13</sup> License

---

**Note:** Before you follow the instructions below, make sure HTTPS traffic is allowed by your firewall. The default port number for HTTPS traffic is 443.

---

1. In your terminal, run the following command to install Certbot with the plugin that allows the integration with the Apache web server:

```
$ sudo apt install certbot python3-certbot-apache
```

2. Press `Y`, then `Enter` to run the installation.
3. To issue a certificate and reconfigure apache automatically, run the command:

```
$ sudo certbot --apache
```

4. Carefully read the questions that will appear on your terminal. Provide a valid email address.
5. Agree to the “Terms of Service” by pressing `A` (short for Agree).
6. Choose whether you want to share your email address with the Electronic Frontier Foundation by pressing `Y` to confirm or `N` to refuse.

---

<sup>11</sup> <https://www.abetterinternet.org/>

<sup>12</sup> <https://www.eff.org/>

<sup>13</sup> <https://creativecommons.org/licenses/by/2.0/>

7. You will then get the output shown below. Indicate the domains that you want to enable HTTPS for by selecting the appropriate listed numbers:

```

Plugins selected: Authenticator apache, Installer apache

Which names would you like to activate HTTPS for?
-----
↪ -----
1: joomla-domain.com
2: www.joomla-domain.com
-----
↪ -----
Select the appropriate numbers separated by commas and/or spaces, or ↪
↪ leave input
blank to select all options shown (Enter 'c' to cancel): c
Please specify --domains, or --installer that will help in domain ↪
↪ names autodiscovery, or --cert-name for an existing certificate name.

```

8. In the next prompt that appears, choose whether or not you want to force redirecting HTTP to HTTPS traffic.

```

Please choose whether or not to redirect HTTP traffic to HTTPS, ↪
↪ removing HTTP access.
-----
↪ -----
1: No redirect - Make no further changes to the webserver ↪
↪ configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. ↪
↪ Choose this for
new sites, or if you're confident your site works on HTTPS. You can ↪
↪ undo this
change by editing your web server's configuration.
-----
↪ -----
Select the appropriate number [1-2] then [enter] (press 'c' to cancel):

```

9. Once you have answered all the questions, Certbot will start the installation.

10. If the installation was successful, you will get the following output:

```

-----
↪ -----
Congratulations! You have successfully enabled https://www.joomla-
↪ domain.com

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=www.joomla-domain.com
-----
↪ -----

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/www.joomla-domain.com/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/www.joomla-domain.com/privkey.pem
  Your cert will expire on 2022-10-25. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot again
  with the "certonly" option. To non-interactively renew *all* of
  your certificates, run "certbot renew"
- If you like Certbot, please consider supporting our work by:

    Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
    Donating to EFF:                   https://eff.org/donate-le

```

Certbot has now installed your TLS/SSL certificate and configured Apache accordingly.

## Hardening Apache on an unmanaged VPS

Published on May 22, 2022 by Fayçal Alami-Hassani @gnufcl@fosstodon.org<sup>14</sup>



Fig. 1: Picture by J4747<sup>Page 21, 15</sup> under CC BY<sup>16</sup> License

This guide is a follow-up to the post published last week about *Migrating Joomla from shared hosting to an unmanaged VPS*. In this tutorial, you will learn how to implement a set of security measures to protect your Apache web server on Debian-based Linux systems.

---

<sup>14</sup> <https://fosstodon.org/@gnufcl>

<sup>15</sup> <https://www.blendswap.com/profile/636833>

<sup>16</sup> <https://creativecommons.org/licenses/by/4.0/>

## 2.1 Prerequisites

This tutorial assumes that you already have the following:

- LAMP stack installed on your VPS server. LAMP is short for “Linux + Apache + MySQL + PHP”.
- Non-root user with sudo privileges on your VPS server.
- SSH access to your VPS server.

Before you follow this guide, make sure to configure your environment accordingly.

## 2.2 Running Apache with an unprivileged user

The “principle of least privilege” is a security best practice in the context of server administration. According to this principle, you should grant users only the strict minimum of permissions they need to perform their tasks. Therefore, running your Apache web server with a non-root user helps you prevent abusive access to the system.

To manage user and group privileges on your server, you need to access the `envvars` file. This file contains the environment variables of your Apache web server.

1. First, type the following command in your terminal:

```
$ sudo nano /etc/apache2/envvars
```

2. In the file that opens, navigate to the following lines:

```
1 # Since there is no sane way to get the parsed apache2 config in scripts, some
2 # settings are defined via environment variables and then used in apache2ctl,
3 # /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.
4 export APACHE_RUN_USER=apache
5 export APACHE_RUN_GROUP=apache
```

3. Set the variables `APACHE_RUN_USER` and `APACHE_RUN_GROUP` to a non-root user and group, respectively.

The table below illustrates some possible values for a non-root `user` and `group`:

↓ Value Pair / Environment Variable →	APACHE_RUN_USER	APACHE_RUN_GROUP
VALUE PAIR 1	apache	apache
VALUE PAIR 2	nobody	nogroup
VALUE PAIR 3	www-data	www-data

## 2.3 Hiding your operating system and Apache version

Each time a user connects to your website, your server sends so-called *response headers* to the user’s browser. Response headers are HTTP headers containing metadata that is not related to the main message being exchanged between the client and your server.

In the default configuration, your web server exposes sensitive information about your infrastructure such as the operating system (OS) and the Apache version installed on the server.

You can check this by using a tool such as `cURL`<sup>17</sup>. In your terminal, type the following command:

```
$ curl -IL your-domain-name
```

<sup>17</sup> <https://curl.se/>



You should then get an output like the one below. Note that the details of your OS and Apache version appear on lines number 3 and 9 under the Server entry:

```

1 HTTP/1.1 301 Moved Permanently
2 Date: Sun, 22 May 2022 19:57:25 GMT
3 Server: Apache/2.4.25 (Debian)
4 Location: https://your-domain-name/
5 Content-Type: text/html; charset=iso-8859-1
6
7 HTTP/1.1 200 OK
8 Date: Sun, 22 May 2022 19:57:25 GMT
9 Server: Apache/2.4.25 (Debian)
10 Expires: Wed, 17 Aug 2005 00:00:00 GMT
11 Last-Modified: Sun, 22 May 2022 19:57:25 GMT
12 Cache-Control: xxx, xxx, xxx-xxx, xxxxxxxx, xxxxxxxx
13 Pragma: no-cache
14 X-Content-Type-Options: nosniff
15 X-Frame-Options: sameorigin
16 Content-Type: text/html; charset=utf-8

```

1. To hide your operating system and Apache version, you need to adjust the settings of your `apache2.conf` file. To revert your settings to their initial state in case of a faulty configuration, you should first make a backup of this file with the following command:

```
$ sudo cp /etc/apache2/apache2.conf /etc/apache2/apache2_bak.conf
```

2. Next, open the `apache2.conf` file by typing this command in your terminal:

```
$ sudo nano /etc/apache2/apache2.conf
```

3. In the file that opens, scroll down to the bottom and add the following two lines:

```
ServerTokens Prod
ServerSignature Off
```

4. Press `Ctrl + O` to save your changes and `Ctrl + X` to close the nano editor.
5. Run the following command to restart Apache:

```
$ sudo systemctl restart apache2
```

6. In your terminal, retype the following command:

```
$ curl -IL your-domain-name
```

7. The new output should now look like this:

```

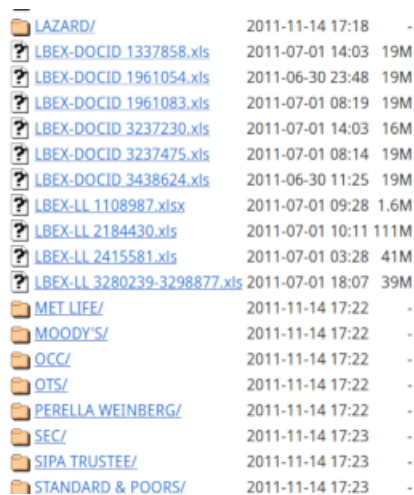
1 HTTP/1.1 301 Moved Permanently
2 Date: Sun, 22 May 2022 19:57:25 GMT
3 Server: Apache
4 Location: https://your-domain-name/
5 Content-Type: text/html; charset=iso-8859-1
6
7 HTTP/1.1 200 OK
8 Date: Sun, 22 May 2022 19:57:25 GMT
9 Server: Apache
10 Expires: Wed, 17 Aug 2005 00:00:00 GMT
11 Last-Modified: Sun, 22 May 2022 19:57:25 GMT
12 Cache-Control: xxx, xxx, xxx-xxx, xxxxxxxx, xxxxxxxx
13 Pragma: no-cache
14 X-Content-Type-Options: nosniff
15 X-Frame-Options: sameorigin
16 Content-Type: text/html; charset=utf-8

```

Note that your OS and Apache version details have disappeared from lines 3 and 9. The `Server` entry only shows **Apache** without any further details.

## 2.4 Disabling open directory listings

If a directory inside your filesystem lacks an index file such as `index.html` or `index.php`, the web server automatically generates a listing of that particular directory. When this feature is enabled, intruders and eavesdroppers can explore the content of your folders to spot any existing vulnerabilities.



LAZARD/	2011-11-14 17:18	-
LBEX-DOCID 1337858.xls	2011-07-01 14:03	19M
LBEX-DOCID 1961054.xls	2011-06-30 23:48	19M
LBEX-DOCID 1961083.xls	2011-07-01 08:19	19M
LBEX-DOCID 3237230.xls	2011-07-01 14:03	16M
LBEX-DOCID 3237475.xls	2011-07-01 08:14	19M
LBEX-DOCID 3438624.xls	2011-06-30 11:25	19M
LBEX-LL 1108987.xlsx	2011-07-01 09:28	1.6M
LBEX-LL 2184430.xls	2011-07-01 10:11	111M
LBEX-LL 2415581.xls	2011-07-01 03:28	41M
LBEX-LL 3280239-3298877.xls	2011-07-01 18:07	39M
MET LIFE/	2011-11-14 17:22	-
MOODY'S/	2011-11-14 17:22	-
OCC/	2011-11-14 17:22	-
OTS/	2011-11-14 17:22	-
PERELLA WEINBERG/	2011-11-14 17:22	-
SEC/	2011-11-14 17:23	-
SIPA TRUSTEE/	2011-11-14 17:23	-
STANDARD & POORS/	2011-11-14 17:23	-

Fig. 2: An example of an open directory listing

To protect your directory content against curious eyes, you need to modify the configuration of your `apache2.conf` file.

1. Open the file with your nano editor by typing the following command:

```
$ sudo nano /etc/apache2/apache2.conf
```

2. In the file that opens, scroll down to the following directive block:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

3. Add a minus sign “-” before the keywords `Indexes` and `FollowSymLinks` to prevent Apache from generating open directory listings and following symbolic links. The result should look like this:

```
<Directory /var/www/>
    Options -Indexes -FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

4. Press `Ctrl + O` to save your changes and `Ctrl + X` to close the nano editor.
5. Restart Apache with the following command:

```
$ sudo systemctl restart apache2
```

## 2.5 Installing a web application firewall

A web application firewall (WAF) protects your applications from malicious attacks by scanning and filtering HTTP traffic. ModSecurity is an open source WAF that provides multiple security features including monitoring, logging, and real-time traffic inspection. You can install ModSecurity on your Apache web server with the free module `mod_security2`.

1. To install `mod_security2` on Debian-based distributions, type the following command in your terminal:

```
$ sudo apt install lib-apache2-mod-security2
```

2. Check if `mod_security2` is up and running on your system by running the command:

```
$ sudo apachectl -M | grep --color security
```

You should get the following output:

```
$ security2_module (shared)
```

---

**Note:** When you install `mod_security2` for the first time, ModSecurity runs in detection-only mode. That is, it detects and logs suspicious activity, and no more than that. To block unwanted traffic, you need to modify the default ModSecurity configuration file: `modsecurity.conf-recommended`.

---

3. Rename the file `modsecurity.conf-recommended` to `modsecurity.conf` by typing the command below:

```
$ mv /etc/modsecurity/modsecurity.conf{-recommended, }
```

4. Open the new file with your nano editor:

```
$ nano /etc/modsecurity/modsecurity.conf
```

5. In the file that opens, navigate to the directive:

```
SecRuleEngine DetectionOnly
```

6. Replace the value **DetectionOnly** by the new value **On**.
7. Press `Ctrl + O` to save your changes and `Ctrl + X` to close the nano editor.
8. Restart Apache with the following command:

```
$ sudo systemctl restart apache2
```

From now on, ModSecurity will also block unwanted traffic.