



Expert Cloud Consulting

Enhance Optimise & Scale

ASCP GPUonCLOUD Pvt Ltd

“Expert Cloud Consulting” -

**Documentation for Installing Wordpress,
Owncloud, Osticket with LAMP stack on Ubuntu
VPS** [Title,18, Arial]

14.April.2025 [Subtitle,14, Arial]

version 1.0

—

Contributed by Yogiraj Deshpande [Normal text,14, Arial]

Approved by Akshay Shinde(In Review)

Expert Cloud Consulting

Office #811, Gera Imperium Rise,

Hinjewadi Phase-II Rd, Pune, India – 411057

“Expert Cloud Consulting”

Documentation for Installing Wordpress, Owncloud, Osticket with LAMP stack on Ubuntu VPS [Title,18, Arial]

1.0 Document Overview:

This document outlines the procedure for installing and configuring WordPress, ownCloud, and osTicket on an Ubuntu 22.04 VPS using the LAMP stack (Linux, Apache, MariaDB, PHP). These applications serve different purposes: WordPress is a content management system, ownCloud provides file sharing and collaboration, and osTicket is an open-source support ticket system.

This setup includes multiple PHP versions (PHP 7.4 & PHP 8.1) to ensure compatibility with the applications, separate database configurations for security and manageability, and Apache configurations to handle each application independently.

2.0 Objective:

- Deploy a scalable and secure environment on Ubuntu VPS for hosting multiple PHP-based web applications.
- Use multiple PHP versions to meet each application's version requirement.
- Ensure clean separation of data by using individual databases for each application.

3.0 : Prerequisites:

The following environment is required to perform the installation:

- Ubuntu 22.04 VPS
- Root or sudo access
- Static public IP address
- Basic knowledge of Linux and Apache



4.0 LAMP Stack Installation

4.1: What is the LAMP Stack?

LAMP stands for **L**inux, **A**pache, **M**ySQL (or MariaDB), and **P**HP. It's a popular open-source web server stack used to host dynamic websites and web applications.

- Linux: Operating System (Ubuntu in our case).
- Apache: Web server that serves HTTP content.
- MariaDB: Drop-in replacement for MySQL, used to manage application data.
- PHP: Server-side scripting language to process dynamic content

4.2 Install Apache, MariaDB, PHP 7.4 and 8.1

```
sudo apt update && sudo apt upgrade -y
sudo apt install apache2 mariadb-server software-properties-common -y
```

Add PHP repository and install multiple versions:

```
sudo add-apt-repository ppa:ondrej/php -y
sudo apt update

# Install PHP 7.4
sudo apt install php7.4 php7.4-fpm php7.4-mysql php7.4-cli php7.4-xml
php7.4-curl php7.4-mbstring php7.4-zip -y

# Install PHP 8.1
sudo apt install php8.1 php8.1-fpm php8.1-mysql php8.1-cli php8.1-xml
php8.1-curl php8.1-mbstring php8.1-zip -y
```

Enable Apache Modules for PHP-FPM Supports:

```
sudo a2enmod proxy_fcgi setenvif
sudo a2enmod php8.1
```

- **Why Use Two PHP Versions?**

Some applications have strict version compatibility requirements. For instance, ownCloud may not support PHP 8.1, whereas WordPress and osTicket do. Multi-PHP support allows each app to use its compatible version without conflicts.

4.3: Setup MariaDB (MySQL-compatible) and Databases:

Each application requires its own database for storing configurations, user data, and content.

```
sudo mysql_secure_installation
```

Log into MySQL and execute:

```
// WordPress
CREATE DATABASE wordpress_db;
CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'wp_pass';
GRANT ALL PRIVILEGES ON wordpress_db.* TO 'wp_user'@'localhost';

// ownCloud
CREATE DATABASE owncloud_db;
CREATE USER 'oc_user'@'localhost' IDENTIFIED BY 'oc_pass';
GRANT ALL PRIVILEGES ON owncloud_db.* TO 'oc_user'@'localhost';

// osTicket
CREATE DATABASE osticket_db;
CREATE USER 'os_user'@'localhost' IDENTIFIED BY 'os_pass';
GRANT ALL PRIVILEGES ON osticket_db.* TO 'os_user'@'localhost';

FLUSH PRIVILEGES;
```

- **Why Do We Need Separate Databases?**

Isolating each application's database enhances security, avoids data conflicts, and simplifies backups and migrations.

4.4: Download and Extract Applications

Navigate to the web root:

```
cd /var/www/html
```

WordPress:

```
sudo wget https://wordpress.org/latest.tar.gz
sudo tar -xzf latest.tar.gz
sudo rm latest.tar.gz
```

OwnCloud:

```
sudo wget
https://download.owncloud.org/community/owncloud-10.15.2.zip
sudo apt install unzip -y
sudo unzip owncloud-10.15.2.zip
```

```
sudo rm owncloud-10.15.2.zip
```

OsTicket:

```
sudo wget
https://github.com/osTicket/osTicket/releases/download/v1.18.1/osTicket-v1.18.1.zip
sudo unzip osTicket-v1.18.1.zip
sudo rm osTicket-v1.18.1.zip
sudo mv upload osticket
sudo cp osticket/include/ost-sampleconfig.php
osticket/include/ost-config.php
sudo chown -R www-data:www-data osticket
```

- **Why This Step?**

We're fetching the latest versions of these applications directly from their official sources to ensure we're installing clean, up-to-date packages.

4.5: Configure Apache for Multi-PHP Support

- **Why Apache Configuration?**

Apache must know which PHP version to use for each directory. With PHP-FPM and the FilesMatch directive, we assign specific PHP socket handlers to each app.

Edit default Apache site configuration:

```
sudo vim /etc/apache2/sites-available/000-default.conf
```

Add the following inside `<VirtualHost*:80>` block:

```
DocumentRoot /var/www/html

# WordPress (PHP 8.1)
<Directory /var/www/html/wordpress>

    Options +FollowSymlinks
    AllowOverride All
    Require all granted

    <FilesMatch "\.php$">
        SetHandler
        "proxy:unix:/run/php/php8.1-fpm.sock|fcgi://localhost"
    </FilesMatch>

</Directory>
```



```
# ownCloud (PHP 7.4)
<Directory /var/www/html/owncloud>
    Options +FollowSymlinks
    AllowOverride All
    Require all granted
    <FilesMatch "\.php$">
        SetHandler
"proxy:unix:/run/php/php7.4-fpm.sock|fcgi://localhost"
    </FilesMatch>
</Directory>

# osTicket (PHP 8.1)
<Directory /var/www/html/osticket>
    Options +FollowSymlinks
    AllowOverride All
    Require all granted
    <FilesMatch "\.php$">
        SetHandler
"proxy:unix:/run/php/php8.1-fpm.sock|fcgi://localhost"
    </FilesMatch>
</Directory>
```

Enable site and reload Apache:

```
sudo a2ensite 000-default.conf
sudo systemctl reload apache2
```

4.6: Access Applications in Browser

Visit the following URLs in your browser to complete each app's installation:

- WordPress: <http://your-ip/wordpress>
- OwnCloud: <http://your-ip/owncloud>
- OsTicket: <http://your-ip/osticket>

Use the previously created database credentials during setup.

4.7: Troubleshooting Tips

404 Not Found?

Check directory permissions and ensure the applications were extracted to the correct paths.

Incorrect PHP Version?

Check `<FilesMatch>` and ensure the correct FPM socket is assigned.

Check Apache Error Logs:

```
sudo tail -f /var/log/apache2/error.log
```

