

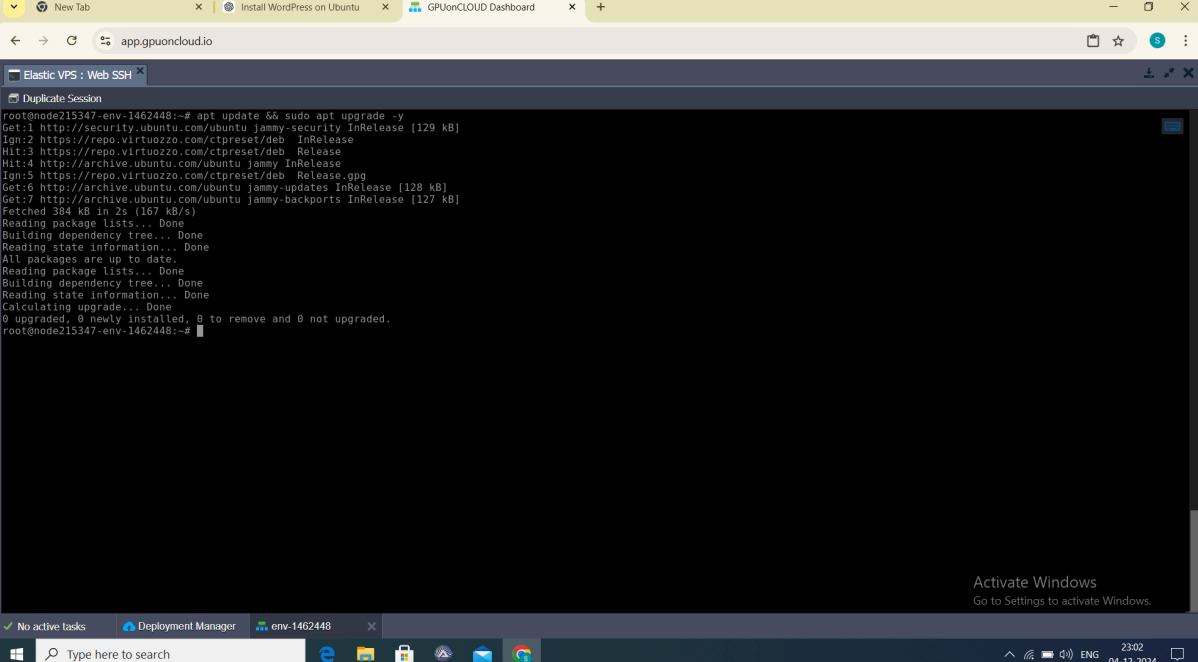
I have created an Ubuntu environment. On this Ubuntu server, we will install wordpress, osticket, and owncloud. Along with that, we will point a domain to the server and enable SSL. So now, we will proceed step by step, starting with wordpress.

## Install WordPress on Ubuntu 22.04

Since Ubuntu uses “apt” (advanced package tool) for package management, we need to update our repository and upgrade our packages before we get started.

-> **sudo -l** ☐ for log-in from root to user

-> **apt update && sudo apt upgrade -y**



```
root@node215347-env-1462448:~# apt update && sudo apt upgrade -y
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Ign:2 https://repo.virtuozzo.com/ctpreset/deb InRelease
Hit:3 https://repo.virtuozzo.com/ctpreset/deb Release
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:5 https://repo.virtuozzo.com/ctpreset/deb Release.gpg
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Fetched 384 kB in 2s (167 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@node215347-env-1462448:~#
```

Next, you'll install Apache server and the dependencies you're going to need.

-> apt install apache2 ghostscript libapache2-mod-php  
mysql-server php php-bcmath php-curl php-imagick php-intl  
php-json php-mbstring php-mysql php-xml php-zip -y

Once Apache and related dependencies are finished installing, navigate to <http://domain> name or the IP address where you installed WordPress to verify that the default Apache page is showing.



Next we're going to make the directory where we'll host the wordpress files

-> mkdir -p /var/www/html

Change the directory's ownership to the user www-data

-> **sudo chown www-data: /var/www/html**

```
Elastic VPS : Web SSH
Duplicate Session
root@node215347-env-1462448:~# chown www-data: /var/www/html
root@node215347-env-1462448:~# curl https://wordpress.org/latest.tar.gz | sudo -u www-data tar zx -C /var/www/html
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload   Total   Spent    Left   Speed
100 25.6M  100 25.6M    0     0  4903k      0  0:00:05  0:00:05 --:--:-- 5761k
root@node215347-env-1462448:~# cd /var/www/html
root@node215347-env-1462448:/var/www/html# ls
index.html  wordpress
root@node215347-env-1462448:/var/www/html#
```

And then curl the zip files from the official wordpress page and unzip them into the directory we created.

-> **curl <https://wordpress.org/latest.tar.gz> | sudo -u www-data tar zx -C /var/www/html**

```
Duplicate Session
root@node215347-env-1462448:~# chown www-data: /var/www/html
root@node215347-env-1462448:~# curl https://wordpress.org/latest.tar.gz | sudo -u www-data tar zx -C /var/www/html
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload   Total   Spent    Left   Speed
100 25.6M  100 25.6M    0     0  4903k      0  0:00:05  0:00:05 --:--:-- 5761k
root@node215347-env-1462448:~# cd /var/www/html
root@node215347-env-1462448:/var/www/html# ls
index.html  wordpress
root@node215347-env-1462448:/var/www/html#
```

When we installed Apache, it created a default configuration in the /etc/apache2/sites-available directory. That's where we want to create our wordpress configuration.

-> **sudo touch /etc/apache2/sites-available/wordpress.conf**

After that we'll use the tee command to drop text into our configuration file. We're only using port 80 for now because it's HTTP; once we install an SSL certificate from Let's Encrypt, it will automatically create a configuration file for HTTPS (port 443) that we can further modify later. For now, copy and paste the following into your terminal:

-> **vim /etc/apache2/sites-available/wordpress.conf**

```

2
3 <VirtualHost *:80>
4
5     DocumentRoot /var/www/html/wordpress
6
7     <Directory /var/www/html/wordpress>
8
9         Options FollowSymLinks
10
11         AllowOverride Limit Options FileInfo
12
13         DirectoryIndex index.php
14
15         Require all granted
16
17     </Directory>
18
19     <Directory /var/www/html/wordpress/wp-content>
20
21         Options FollowSymLinks
22
23         Require all granted
24
25     </Directory>
26 </VirtualHost>
27
28 EOF

```

Now let's enable the wordpress site, disable the default Apache site, and enable rewrites.

-> a2ensite wordpress  
 -> a2dissite 000-default  
 -> a2enmod rewrite

A2ensite and a2dissite stand for "Apache2 enable site" and "Apache2 disable site," respectively; as you've probably guessed, a2enmod means "Apache2 enable modification."

You can use MariaDB here for the database that stores all your posts, but for simplicity we're going to use MySQL (which we installed at the beginning). First you need to create the database, which we'll call "wordpress." Log into your MySQL server.

-> mysql -u root -p

```

Duplicate Session
root@node215347-env-1462448:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 220
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

```

Next you'll create a user and select a password. You can also substitute your own desired username and password in these commands to avoid doing it later.

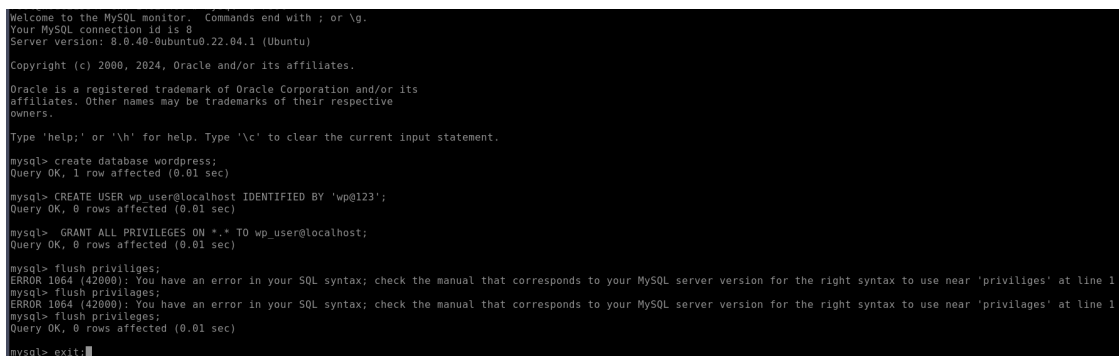
-> **CREATE USER wp\_user@localhost IDENTIFIED BY 'wp@123';**

After creating the user, we need to give the user privileges to the database.

-> **GRANT ALL PRIVILEGES ON \*.\* wp\_user@localhost;**

-> **FLUSH PRIVILEGES;**

-> **EXIT;**



```
welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database wordpress;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER wp_user@localhost IDENTIFIED BY 'wp@123';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO wp_user@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'privileges' at line 1
mysql> flush privileges;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'privileges' at line 1
mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> exit;
```

Now start the MySQL database.

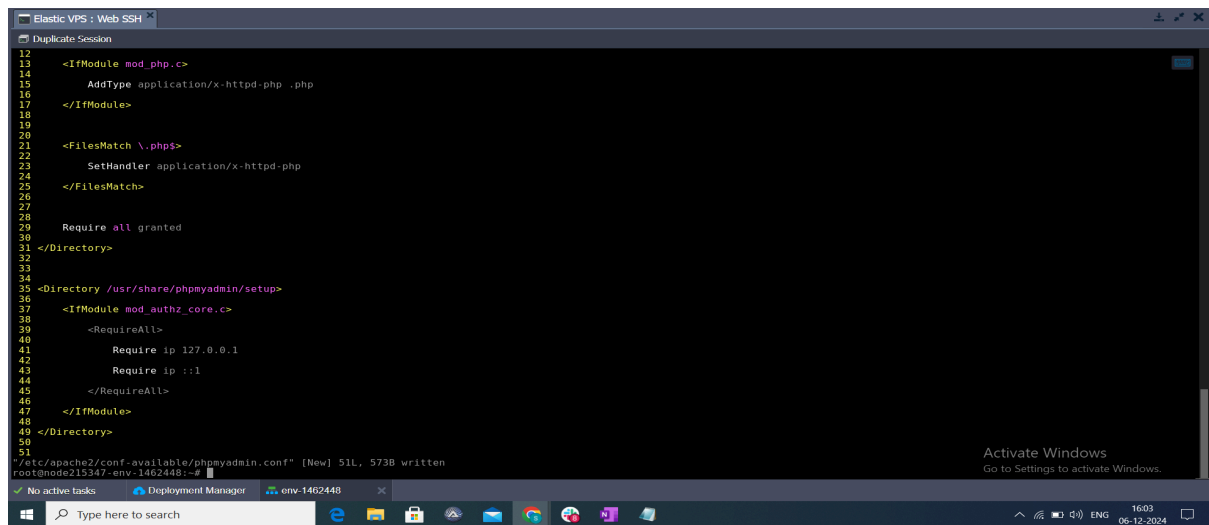
-> **service mysql start**

You can verify that status of the server with

-> **systemctl status mysql**

The WordPress files we unzipped includes a sample configuration that we'll copy into the configuration we'll actually use.

```
-> -u www-data cp  
/var/www/html/wordpress/wp-config-sample.php  
/var/www/html/wordpress/wp-config.php
```



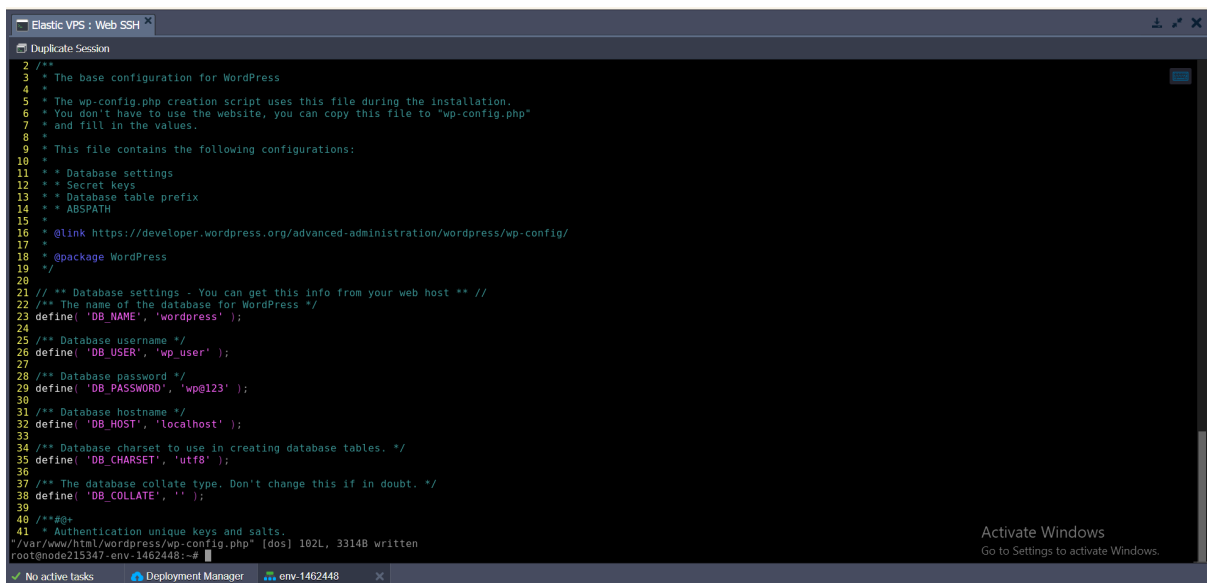
```
12 <IfModule mod_php.c>  
13  
14     AddType application/x-httpd-php .php  
15  
16 </IfModule>  
17  
18  
19  
20 <FilesMatch \.php$>  
21  
22     SetHandler application/x-httpd-php  
23  
24 </FilesMatch>  
25  
26  
27  
28     Require all granted  
29  
30 </Directory>  
31  
32  
33  
34 <Directory /usr/share/phpmyadmin/setup>  
35  
36     <IfModule mod_authz_core.c>  
37  
38         <RequireAll>  
39  
40             Require ip 127.0.0.1  
41  
42             Require ip ::1  
43  
44         </RequireAll>  
45     </IfModule>  
46  
47 </Directory>  
48  
49  
50  
51  
"/etc/apache2/conf-available/phpmyadmin.conf" [New] 51L, 573B written  
root@node215347-env-1462448:~#
```

Now we need to edit the configuration file and connect our database to our WordPress installation. You can do this one of two ways: edit directly with vim, or use sed ("stream edit") to edit the file. You can certainly use vim, but since I have the commands listed below, it'll be faster to copy and paste. Leave database\_name\_here, username\_here, and password\_here alone, but change the db, user, and password name if you've changed them when you created your database user and password;

```
-> -u www-data sed -i 's/wordpress/wordpress/'  
/var/www/html/wordpress/wp-config.php
```

-> -u www-data sed -i 's/wp\_user/admin/'  
/var/www/html/wordpress/wp-config.php

-> -u www-data sed -i 's/wp@123 /defaultpassword/'  
/var/www/html/wordpress/wp-config.php



The screenshot shows a terminal window titled "Elastic VPS : Web SSH". The terminal displays the content of the wp-config.php file, which includes database settings and authentication keys. The database settings are as follows:

```
2 /**
3  * The base configuration for WordPress
4  *
5  * The wp-config.php creation script uses this file during the installation.
6  * You don't have to use the website, you can copy this file to "wp-config.php"
7  * and fill in the values.
8  *
9  * This file contains the following configurations:
10  *
11  * Database settings
12  * Secret keys
13  * Database table prefix
14  * ABSPATH
15  *
16  * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
17  *
18  * @package WordPress
19  */
20
21 // ** Database settings - You can get this info from your web host ** //
22 /** The name of the database for WordPress */
23 define( 'DB_NAME', 'wordpress' );
24
25 /** Database username */
26 define( 'DB_USER', 'wp_user' );
27
28 /** Database password */
29 define( 'DB_PASSWORD', 'wp@123' );
30
31 /** Database hostname */
32 define( 'DB_HOST', 'localhost' );
33
34 /** Database charset to use in creating database tables. */
35 define( 'DB_CHARSET', 'utf8' );
36
37 /** The database collate type. Don't change this if in doubt. */
38 define( 'DB_COLLATE', '' );
39
40 /**#@+
41  * Authentication unique keys and salts.
42  * These keys are used for various WordPress features and must be unique to your installation.
43  * You can get these keys from your web host.
44  */
45 define( 'AUTH_KEY', 'put your unique phrase here' );
46 define( 'SECURE_AUTH_KEY', 'put your unique phrase here' );
47 define( 'LOGGED_IN_KEY', 'put your unique phrase here' );
48 define( 'NONCE_KEY', 'put your unique phrase here' );
49 define( 'AUTH_SALT', 'put your unique phrase here' );
50
51 /**#@-
```

The terminal window also shows the status bar at the bottom with "No active tasks", "Deployment Manager", and "env-1462448".

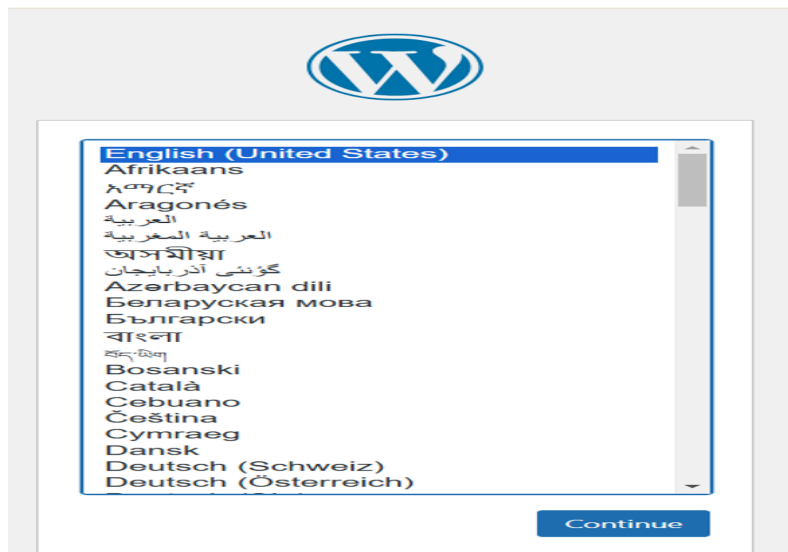
Finally we'll enable a modification for hardening the website later, and restart Apache.

-> phpenmod mbstring

-> a2enmod headers

-> systemctl restart apache2

Navigate to <http://shwp.gpuoncloud.com>



Not Secure:



## Blog

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

December 6, 2024

Once we browse url we got wordpress page like above

**Enable SSL:**

**Install Certbot**

**-> apt install certbot python3-certbot-apache -y**

**Obtain an SSL Certificate** Run the following command to generate and install an SSL certificate:



-> certbot --apache

```
Elastic VPS : Web SSH
Duplicate Session
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.4-April-3-2024.pdf. You must agree in
order to register with the ACME server. Do you agree?
.....
(Y)es/(N)o: y
.....
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
.....
(Y)es/(N)o: ^CExiting due to user request.
root@node215347-env-1462448:~# certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log

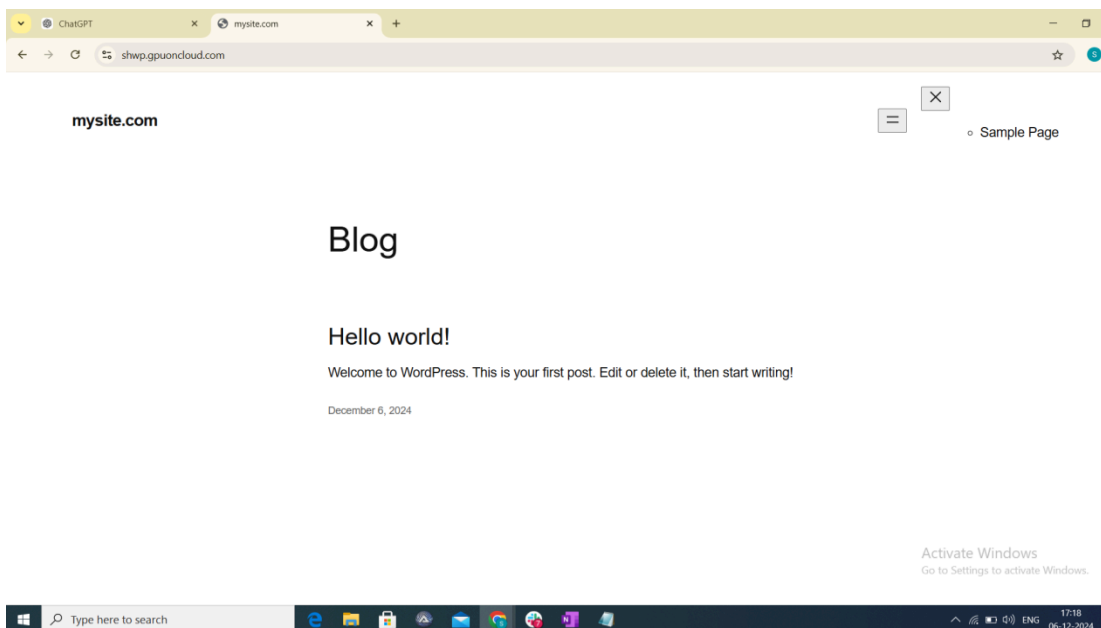
Which names would you like to activate HTTPS for?
.....
1: shwp.gpuoncloud.com
.....
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
Requesting a certificate for shwp.gpuoncloud.com

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/shwp.gpuoncloud.com/fullchain.pem
Key is saved at: /etc/letsencrypt/live/shwp.gpuoncloud.com/privkey.pem
This certificate expires on 2025-03-06.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for shwp.gpuoncloud.com to /etc/apache2/sites-available/wordpress-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://shwp.gpuoncloud.com

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
.....
root@node215347-env-1462448:~#
```

Verify SSL Visit <https://shwp.gpuoncloud.com> to ensure the site is secure.



**2: Installtion of OS-Ticket, also point domain to the server and enablen SSL:**

For OS-Ticket we need to install Apache, PHP with extension as well as mysql, so as I already install this so no required to install again.

## Download osTicket

Navigate to /var/www/html and download the osTicket package.

->cd /var/www/html

->wget

<https://github.com/osTicket/osTicket/releases/download/v1.17.1/osTicket-v1.17.1.zip>

```
Bye
root@node215347-env-1462448:~# cd /var/www/html
root@node215347-env-1462448:/var/www/html# wget https://github.com/osTicket/osTicket/releases/download/v1.17.1/osTicket-v1.17.1.zip
```

## Extract osTicket

Unzip the downloaded osTicket package.

->apt install unzip -y # Ensure unzip is installed

-> unzip osTicket-v1.17.1.zip -d /var/www/html

## Rename the Extracted Directory

Rename the extracted folder to osticket for easier access.

->mv /var/www/html/upload /var/www/html/osticket

```
Elastic VPS : Web SSH x
Duplicate Session
root@node215347-env-1462448:/var/www/html# mv /var/www/html/upload /var/www/html/osticket
root@node215347-env-1462448:/var/www/html# cd /osticket
```

## Set Permissions

Set the appropriate permissions for the osticket directory.

->chown -R www-data:www-data /var/www/html/osticket

->chmod -R 755 /var/www/html/osticket

```
root@node215347:~# cd /osticket; No such file or directory
root@node215347:~# cd /var/www/html; chown -R www-data:www-data /var/www/html/osticket
root@node215347:~# cd /var/www/html; chmod -R 755 /var/www/html/osticket
root@node215347:~# cd /var/www/html; chmod 644 /var/www/html/osticket/include/ost-config.php
chmod: cannot access '/var/www/html/osticket/include/ost-config.php': No such file or directory
root@node215347:~# cd /var/www/html; 
```

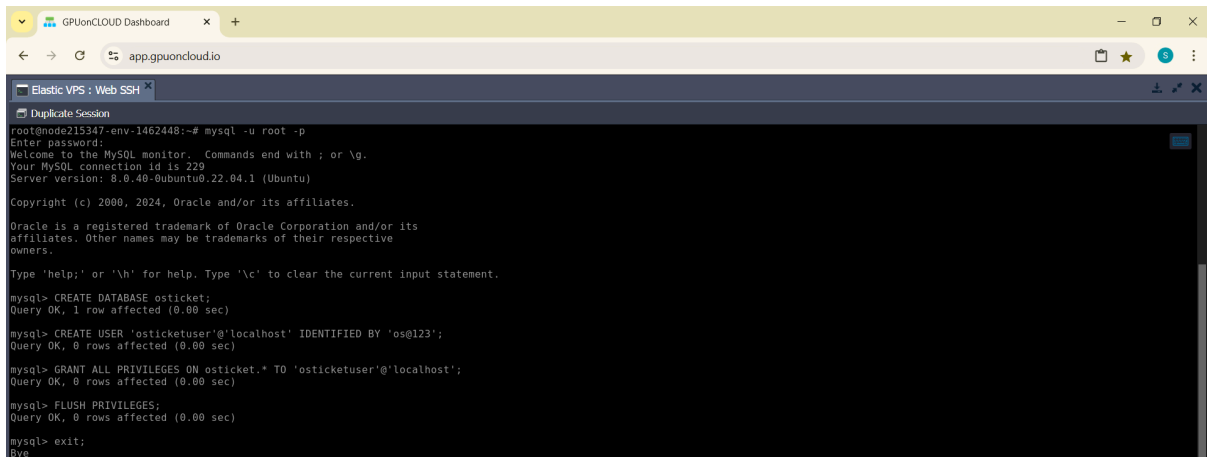
## Create a MySQL Database and User for osTicket

Log in to MySQL and create the database and user.

->mysql -u root -p

Once logged in, execute the following SQL commands:

```
sql
Copy code
CREATE DATABASE osticket;
CREATE USER 'osticketuser'@'localhost' IDENTIFIED BY 'os@123';
GRANT ALL PRIVILEGES ON osticket_db.* TO
'osticketuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

A screenshot of a web browser window displaying a terminal session. The browser's address bar shows 'app.gpuoncloud.io'. The terminal window is titled 'Elastic VPS : Web SSH' and shows a 'Duplicate Session' button. The terminal output shows a user logging in as root on a MySQL server. The user enters a password and is prompted to enter commands. The user then runs several MySQL commands: 'CREATE DATABASE osticket;', 'CREATE USER 'osticketuser'@'localhost' IDENTIFIED BY 'os@123';', 'GRANT ALL PRIVILEGES ON osticket.\* TO 'osticketuser'@'localhost';', and 'FLUSH PRIVILEGES;'. The terminal shows the results of these commands, indicating that the database, user, and privileges were successfully created and flushed. The terminal session ends with 'mysql> exit;' and 'bye'.

To enable the recommended extensions for osTicket on your Ubuntu 22.04 server, follow these steps:

## Step 1: Install Missing PHP Extensions

Run the following command to install the required extensions:

```
->apt install php-gd php-imap php-xml php-mbstring php-json  
php-intl php-apcu php-zip php-bcmath php-curl php-phar  
php-opcache -y
```

## Step 2: Enable Installed Extensions

Ensure that the installed extensions are enabled. Restart the Apache server to load the new extensions:

```
-> systemctl restart apache2
```

## Step 3: Verify Installed Extensions

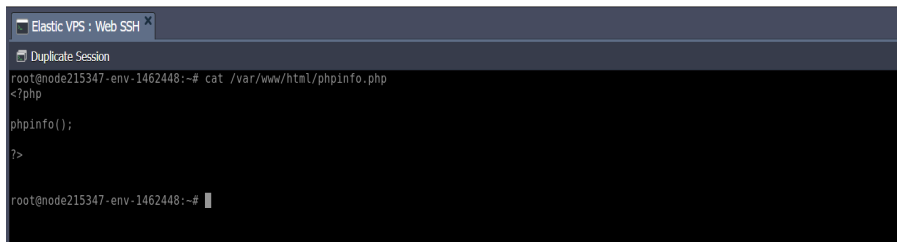
You can verify if the extensions are enabled by creating a PHP info file:

Create the file in the web directory:

```
vim /var/www/html/phpinfo.php
```

Add the following content to the file:

```
<?php
phpinfo();
?>
```

A screenshot of a terminal window titled "Elastic VPS : Web SSH". The terminal shows the command `cat /var/www/html/phpinfo.php` being executed, followed by the output of the PHP script, which is the PHP information page. The prompt is `root@node215347-env-1462448:~#`.

Save the file and access it in your browser:

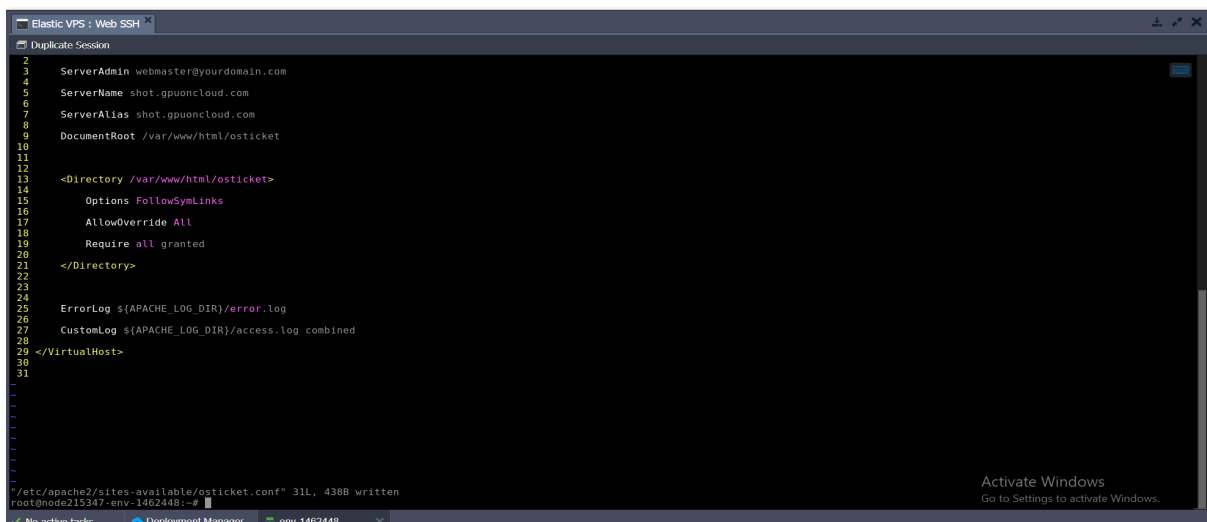
<http://shot.gpuoncloud.com.com/phpinfo.php>

## Configure Apache for osTicket

Create a new Apache configuration file for osTicket.

```
vim /etc/apache2/sites-available/osticket.conf
```

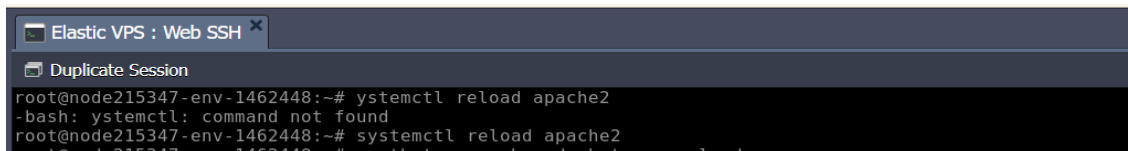
Add the following content:

A screenshot of a terminal window titled "Elastic VPS : Web SSH". The terminal shows the content of the file `/etc/apache2/sites-available/osticket.conf` being displayed. The content includes server configuration details like `ServerAdmin`, `ServerName`, `ServerAlias`, and `DocumentRoot`, as well as a `<Directory>` block for `/var/www/html/osticket` with options like `FollowSymLinks`, `AllowOverride All`, and `Require all granted`. It also shows `ErrorLog` and `CustomLog` settings. The prompt is `root@node215347-env-1462448:~#`. At the bottom, there is a message: `"/etc/apache2/sites-available/osticket.conf" 31L, 438B written`.

## Enable the Apache Configuration

Enable the new site and the required Apache modules.

- > a2ensite osticket.conf
- > a2enmod rewrite
- > systemctl reload apache2

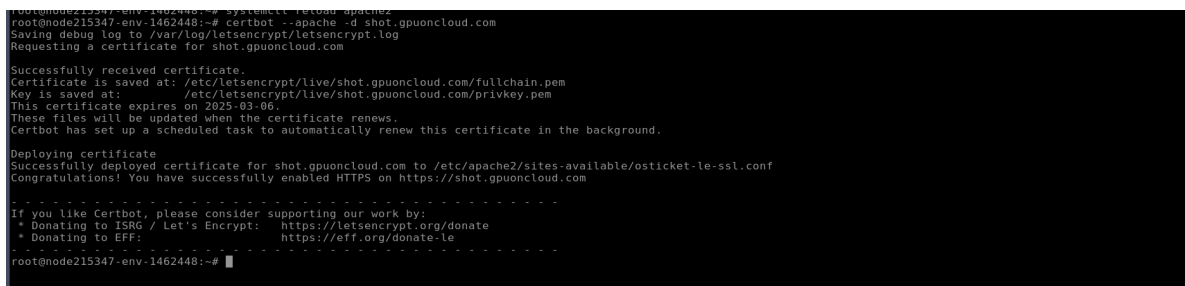


```
Elastic VPS : Web SSH
Duplicate Session
root@node215347-env-1462448:~# systemctl reload apache2
-bash: systemctl: command not found
root@node215347-env-1462448:~# systemctl reload apache2
root@node215347-env-1462448:~# certbot --apache -d shot.gpuoncloud.com
```

## Secure Apache with SSL

To enable SSL, install and run Certbot to get a free SSL certificate.

- > apt install certbot python3-certbot-apache -y
- > certbot --apache



```
root@node215347-env-1462448:~# systemctl reload apache2
root@node215347-env-1462448:~# certbot --apache -d shot.gpuoncloud.com
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for shot.gpuoncloud.com

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/shot.gpuoncloud.com/fullchain.pem
Key is saved at: /etc/letsencrypt/live/shot.gpuoncloud.com/privkey.pem
This certificate expires on 2025-03-06.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for shot.gpuoncloud.com to /etc/apache2/sites-available/osticket-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://shot.gpuoncloud.com

.....
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
.....
root@node215347-env-1462448:~#
```

Follow the prompts to set up SSL. When asked for your email and whether to share data with EFF, answer accordingly. Certbot will automatically configure SSL for your site.

## Step 10: Finalize osTicket Installation

1. Open your web browser and navigate to <https://shot.gpuoncloud.com> or <https://www.shot.gpuoncloud.com>.



Installing osTicket v1.17.1  
Installation Guide — Get Professional Help — Contact Us

## Configuration file missing!

osTicket installer requires ability to write to the configuration file, `include/ost-config.php`. A template copy is located in the include directory (`include/ost-sampleconfig.php`).

**Solution:** Configuration file does NOT exist. Follow steps below to add one.

Rename the sample file `include/ost-sampleconfig.php` to `ost-config.php` and click continue below.

- **CLI:**  
`cp include/ost-sampleconfig.php include/ost-config.php`
- **Windows PowerShell:**  
`Copy-Item -Path include\ost-sampleconfig.php -Destination include\ost-config.php`
- **FTP:**
- **Cpanel:**

If sample config file is missing - please make sure you uploaded all files in 'upload' folder or refer to the [Installation Guide](#)

Continue »

**Need Help?**

If you are looking for a greater level of support, we provide [professional installation services](#) and commercial support with guaranteed response times, and access to the core development team. We can also help customize osTicket or even add new features to the system to meet your unique needs. [Learn More!](#)

### 3: Installtion of owncloud, also point domain to the server and enablen SSL:

For owncloud we need to install Apache, PHP with extension as well as mysql, so as I already install this so no required to install again.

#### **->Install OwnCloud**

#### **Add the OwnCloud Repository:**

First, add the OwnCloud repository to your server:

**->wget -q  
<https://download.owncloud.org/download/repositories/stable/owncloud/owncloud.list> -O /etc/apt/sources.list.d/owncloud.list**

## 1. Import the Repository Key:

```
-> wget -q  
https://download.owncloud.org/download/repositories/stable/owncloud/Release.key -O- | sudo apt-key add -
```

## Update the Package List:

```
-> apt update -y
```

## Install OwnCloud:

```
-> apt install owncloud
```

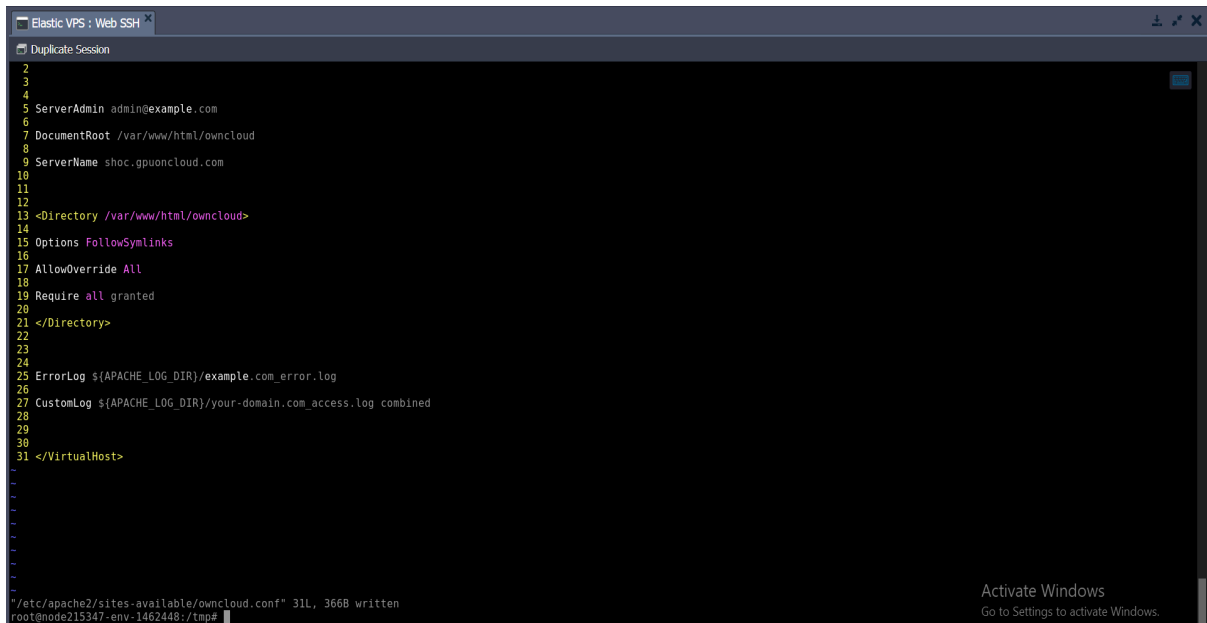
## Configure the Web Server:

OwnCloud comes with Apache2 configuration. To make it work, create an Apache virtual host configuration:

```
-> vim /etc/apache2/sites-available/owncloud.conf
```

Add the following configuration (replace `your_domain.com` with your actual domain):





```
2
3
4
5 ServerAdmin admin@example.com
6
7 DocumentRoot /var/www/html/owncloud
8
9 ServerName shoc.gpuoncloud.com
10
11
12
13 <Directory /var/www/html/owncloud>
14
15 Options FollowSymLinks
16
17 AllowOverride All
18
19 Require all granted
20
21 </Directory>
22
23
24
25 ErrorLog ${APACHE_LOG_DIR}/example.com_error.log
26
27 CustomLog ${APACHE_LOG_DIR}/your-domain.com_access.log combined
28
29
30
31 </VirtualHost>
```

"/etc/apache2/sites-available/owncloud.conf" 31L, 366B written  
root@node215347-env-1462448:/tmp#

Activate Windows  
Go to Settings to activate Windows.

## Configure MySQL Database for OwnCloud

### 1. Log into MySQL:

-> `mysql -u root -p`

### Create a Database and User for OwnCloud:

```
CREATE DATABASE owncloud;
CREATE USER 'os_user'@'localhost' IDENTIFIED BY 'wp@123';
GRANT ALL PRIVILEGES ON owncloud.* TO
'os_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

```
Elastic VPS : Web SSH
Duplicate Session
root@node215347-env-1462448:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE owncloud;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE USER 'os_user'@'localhost' IDENTIFIED BY 'wp@123';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON new_db.* TO 'os_user'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;
Bye
root@node215347-env-1462448:~# cd /tmp
root@node215347-env-1462448:/tmp#
```

## Complete OwnCloud Setup in Web Browser

Open a web browser and go to <http://shoc.gpuoncloud.com>.

Not Secure:

OwnCloud with SSL

ownCloud

Not secure shoc.gpuoncloud.com

ownCloud

Create an admin account

Username

Username

Password

Password

Storage & database

Data folder

/var/www/html/owncloud/data

Configure the database

## Enable SSL for OwnCloud Using Let's Encrypt

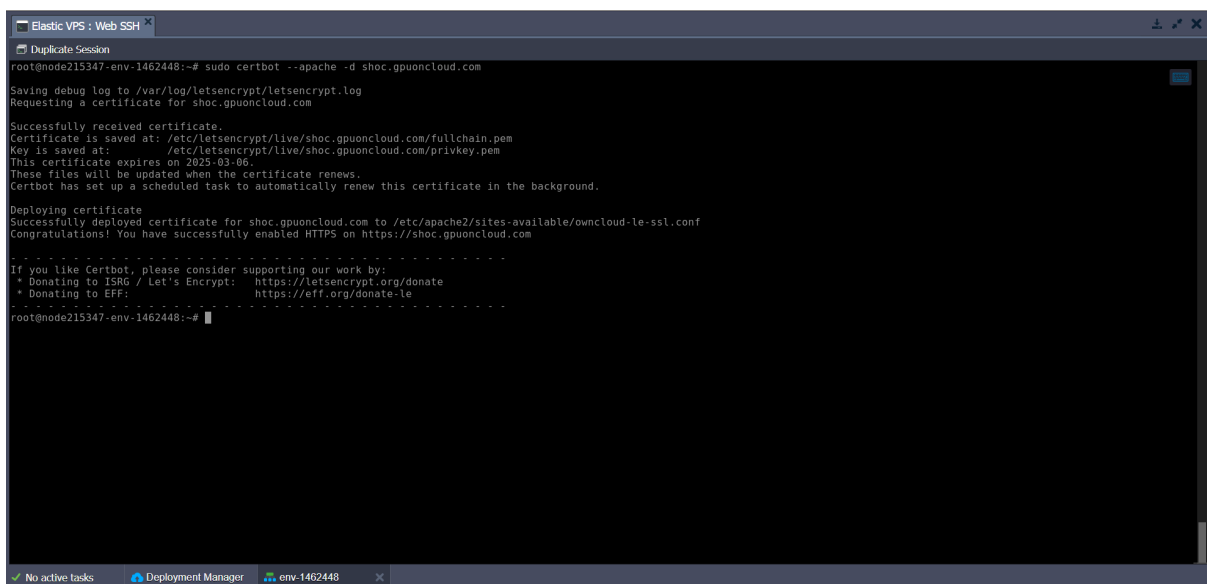
## 1. Install Certbot:

-> apt install certbot python3-certbot-apache

## Obtain an SSL Certificate:

-> certbot --apache -d your\_domain.com

-> certbot --apache



```
root@node215347-env-1462448:~# sudo certbot --apache -d shoc.gpuoncloud.com
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for shoc.gpuoncloud.com

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/shoc.gpuoncloud.com/fullchain.pem
Key is saved at: /etc/letsencrypt/live/shoc.gpuoncloud.com/privkey.pem
This certificate expires on 2025-03-06.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for shoc.gpuoncloud.com to /etc/apache2/sites-available/owncloud-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://shoc.gpuoncloud.com

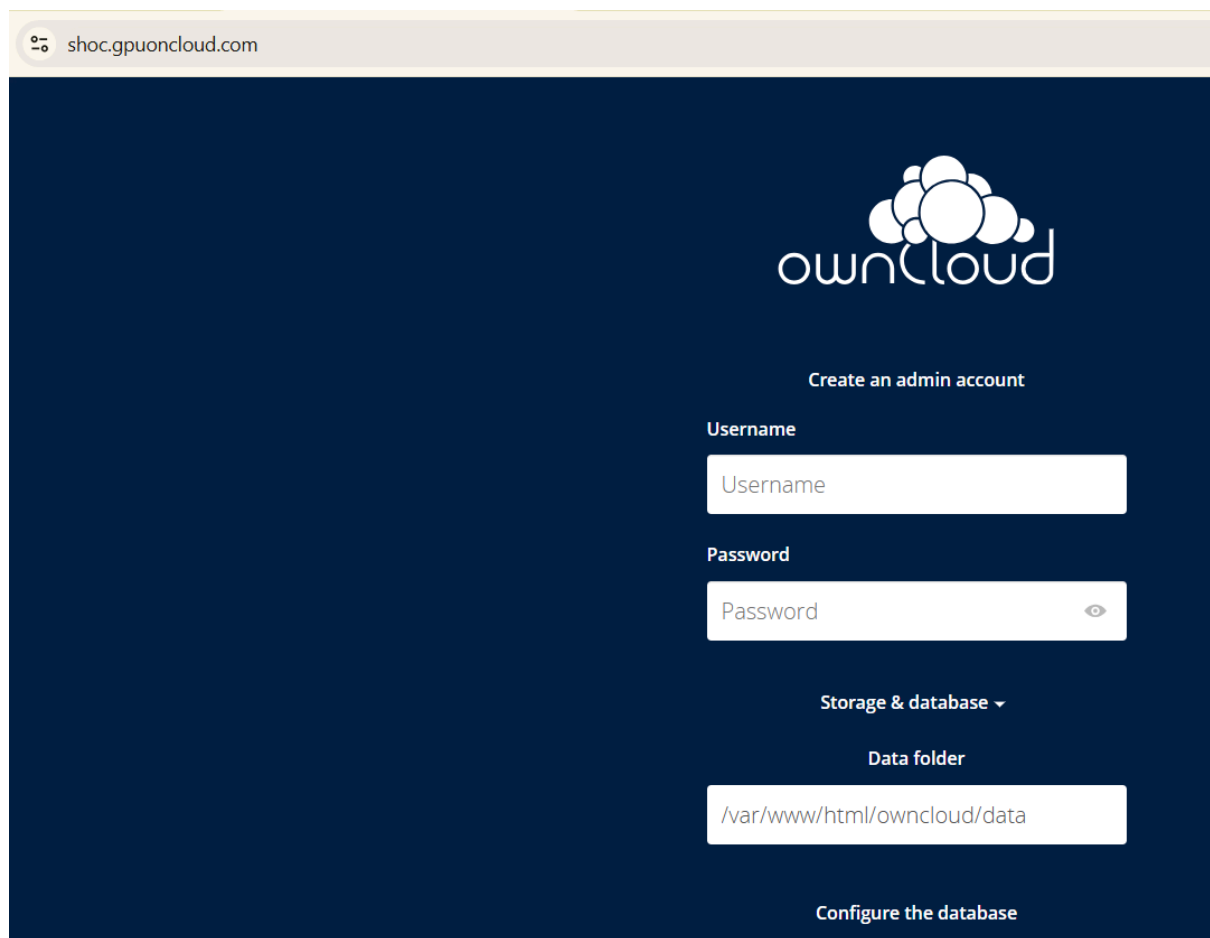
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
 * Donating to EFF: https://eff.org/donate-le

root@node215347-env-1462448:~#
```

## Finalize owncloud Installation:

1. Open your web browser and navigate to <https://shoc.gpuoncloud.com> or <https://www.shoc.gpuoncloud.com>.

This one is secure **with SSL**



The screenshot shows a web browser window with the address bar displaying "shoc.gpuoncloud.com". The main content area has a dark blue background with the OwnCloud logo at the top right. Below the logo, the text "Create an admin account" is centered. The form includes a "Username" label and a text input field containing "Username". Below that is a "Password" label and a password input field containing "Password" with a toggle icon. A "Storage & database" dropdown menu is expanded, showing "Data folder" as the selected option. Below this, a text input field contains the path "/var/www/html/owncloud/data". At the bottom, the text "Configure the database" is visible.

So, Successfully we installed Wordpress, Osticket, and Owncloud applications on Ubuntu.

Created By

Shraddha Chaudhari

