# **Пошаговая установка и настройка**

## **Шаг 0: Установка LAMP-стека на сервере**

* Следуйте инструкции по установке LAMP-стека, используя [этот гайд](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-22-04), если ваш сервер не имеет предустановленных Apache, MySQL и PHP.

Обновите пакеты:

sudo apt update

sudo apt upgrade

Установите Apache:

sudo apt install apache2

Настройте фаервол для Apache:

sudo ufw allow 'Apache'

Проверьте статус Apache:

sudo systemctl status apache2

Установите MySQL:

sudo apt install mysql-server

Запустите скрипт безопасности MySQL:

sudo mysql\_secure\_installation

Установите PHP и необходимые модули:

sudo apt install php libapache2-mod-php php-mysql

Перезапустите Apache для применения изменений:

sudo systemctl restart apache2

Проверьте установку PHP:

sudo nano /var/www/html/info.php

Вставьте следующий код:

<?php

phpinfo();

?>

Сохраните и закройте файл. Затем проверьте http://your\_server\_ip/info.php в браузере.

Удалите info.php, чтобы предотвратить несанкционированный доступ:

sudo rm /var/www/html/info.php

Теперь у вас должен быть установленный и сконфигурированный LAMP стек на вашем сервере Ubuntu.

## **Шаг 1: Распаковка и создание базы данных**

* Распакуйте архив в папку веб-сервера с помощью следующей команды:
* unzip your\_archive.zip -d /путь/к/вашему/веб-серверу
* Замените your\_archive.zip на имя вашего архива и /путь/к/вашему/веб-серверу на путь к вашему веб-серверу.
* Создайте базу данных MySQL с использованием следующих команд в терминале:
* mysql -u ваше\_имя\_пользователя -p
* Введите пароль, когда будет запрошено. После этого в MySQL-консоли выполните:
* CREATE DATABASE имя\_вашей\_базы;
* Замените ваше\_имя\_пользователя на ваше имя пользователя MySQL и имя\_вашей\_базы на желаемое имя базы данных.

## **Шаг 2: Редактирование файла конфигурации .env**

* Отредактируйте файл .env с помощью следующей команды:
* nano /путь/к/вашему/проекту/.env
* Замените /путь/к/вашему/проекту на путь к вашему проекту.
* Внесите необходимые изменения в файл .env, например:
* APP\_URL=http://181.41.141.146/
* DB\_DATABASE=1arb
* DB\_USERNAME=andre
* DB\_PASSWORD=5jXK\_XKXKnyc
* Сохраните изменения, нажав Ctrl + X, затем подтвердите сохранение, нажав Y, и нажмите Enter.

## **Шаг 3: Изменение пути в config/app.php**

* Отредактируйте файл config/app.php с помощью следующей команды:

nano /путь/к/вашему/проекту/config/app.php

* Замените /путь/к/вашему/проекту на путь к вашему проекту.
* Измените путь на ваш проект, где это необходимо.
* Сохраните изменения, нажав Ctrl + X, затем подтвердите сохранение, нажав Y, и нажмите Enter.

## **Шаг 4: Настройка прав для папки storage**

* Дайте необходимые права на запись вложенным папкам storage.

## **Шаг 5: Выполнение миграции базы данных**

* В терминале перейдите в папку проекта.
* Выполните команду
* php artisan migrate
* для миграции базы данных.

## **Шаг 6: Настройка планировщика задач**

* В терминале выполните команду
* crontab -e.
* В конце файла добавьте строку:
* \*/20 \* \* \* \* curl http://181.41.141.146/updatestart.
* Эта команда будет проверять работу ботов каждые 20 минут.

## **Шаг 7: Установка библиотеки Binance**

* В терминале выполните команду
* pip3 install python-binance
* для установки библиотеки Binance.

Теперь ваш проект должен быть настроен и готов к использованию. Пожалуйста, удостоверьтесь, что вы выполнили каждый шаг внимательно, чтобы избежать проблем с работой проекта.