Установка и настройка SSH в Ubuntu 22.04

Введение

SSH (англ. Secure Shell — «безопасная оболочка»[1]) — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Схож по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео (например, с веб-камеры)[2]. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов X Window System.

Большинство хостинг-провайдеров за определённую плату предоставляет клиентам доступ к их домашнему каталогу по SSH. Это может быть удобно как для работы в командной строке, так и для удалённого запуска программ (в том числе графических приложений).

В этом материале вы узнаете, как установить и настроить SSH в системе Ubuntu 22.04 LTS. Это руководство также совместимо с системами Ubuntu 20.04 LTS и Ubuntu 18.04 LTS.

Предварительные условия

Сначала войдите в Ubuntu 22.04 через консоль. Затем обновите кэш Apt и обновите текущие пакеты системы с помощью следующей команды:

```
sudo apt update && sudo apt upgrade
```

При появлении запроса нажмите «Y» , чтобы подтвердить установку.

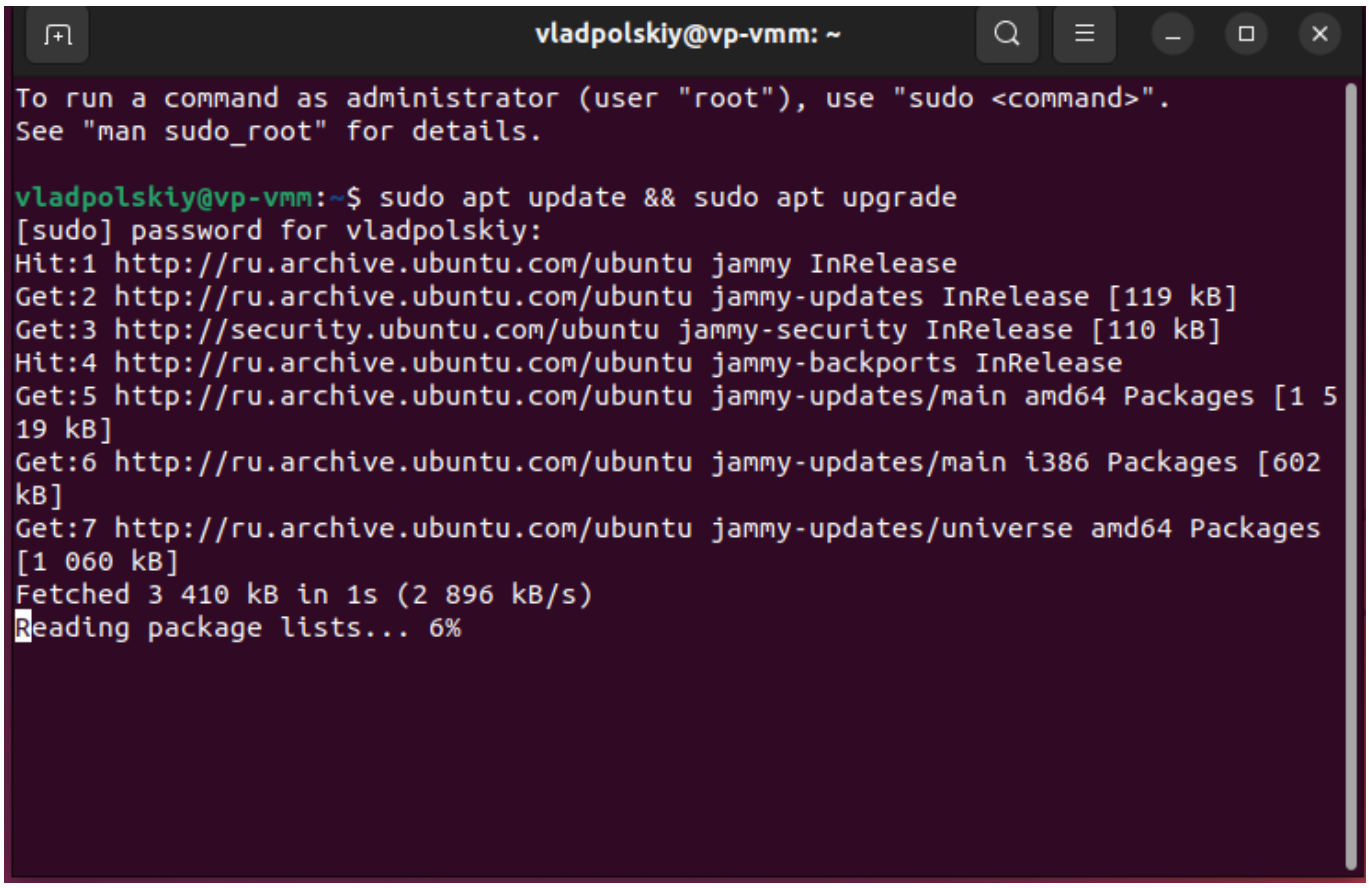
Установите SSH в Ubuntu 22.04

OpenSSH не предустанавливается в системе, поэтому выполним его установку вручную. Чтобы это сделать, в окне терминала вводим:

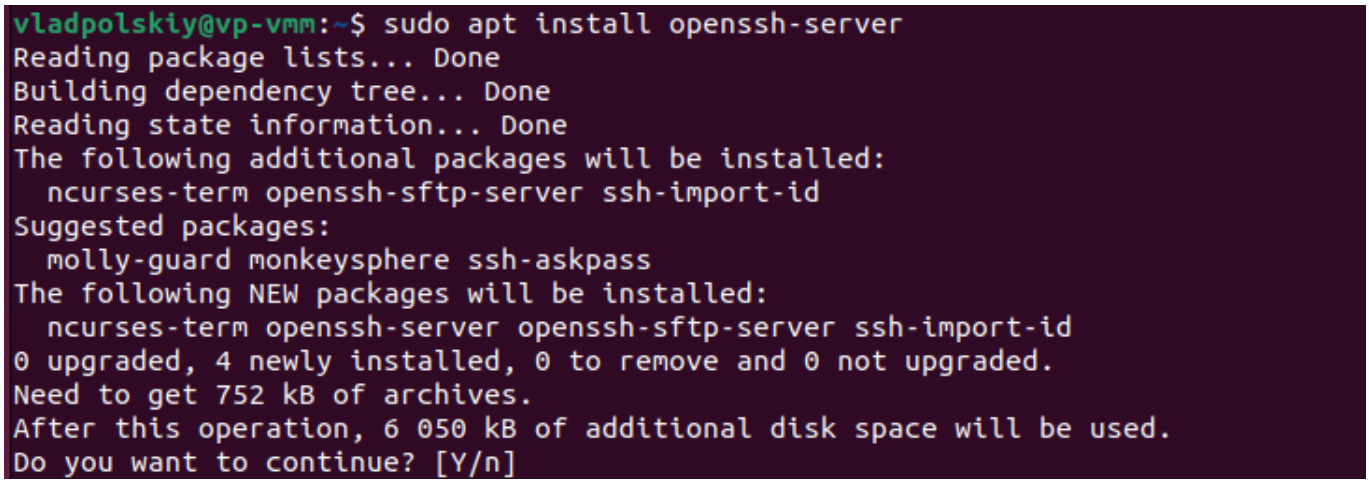
Выполните следующие шаги, чтобы завершить установку SSH в Ubuntu:

Установите несколько зависимостей, необходимых для этого руководства, с помощью приведенной ниже команды:

```
sudo apt install openssh-server
```



Как только команда будет выполнена, начнется установка всех необходимых компонентов в систему, как показано на картинке ниже.



На все предложения системы в момент установки отвечаем утвердительно.

Установка прошла успешно. Теперь перейдем к запуску службы

Запуск SSH

```
sudo systemctl enable --now ssh
```

При успешном запуске вы увидите следующее системное сообщение.

```
vladpolskiy@vp-vmm:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
vladpolskiy@vp-vmm:~$
```

Ключ **–now** в команде выше отвечает за одновременный запуск программы и добавление ее в автозагрузку.

Теперь проверим, что служба включена и успешно функционирует. Для этого вводим:

```
sudo systemctl status ssh
```

В результате система выдаст следующее сообщение:

```
vladpolskiy@vp-vmm:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
vladpolskiy@vp-vmm:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e>
     Active: active (running) since Sun 2024-04-07 21:21:27 MSK; 7min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 2939 (sshd)
      Tasks: 1 (limit: 4597)
     Memory: 1.7M
        CPU: 16ms
     CGroup: /system.slice/ssh.service
             └─2939 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

апр 07 21:21:27 vp-vmm systemd[1]: Starting OpenBSD Secure Shell server...
апр 07 21:21:27 vp-vmm sshd[2939]: Server listening on 0.0.0.0 port 22.
апр 07 21:21:27 vp-vmm sshd[2939]: Server listening on :: port 22.
апр 07 21:21:27 vp-vmm systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (END)
```

Как видно по картинке выше, установленная служба успешно функционирует. Об этом говорит строка Active: active (running).

Если необходимо выключить службу и убрать ее из автозагрузки, в окне терминала введите:

```
sudo systemctl disable ssh
```

Настройка брандмауэра

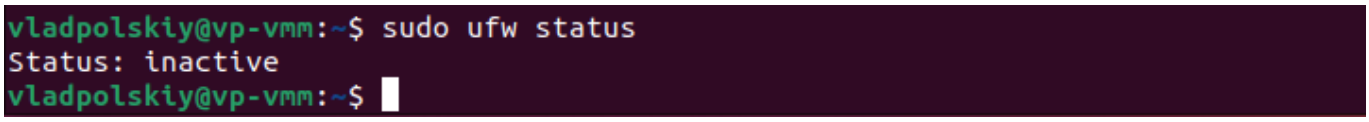
Перед подключением к серверу через протокол SSH проверим состояние брандмауэра. Хотя во время установки службы и происходит настройка файрвола, повторная проверка поможет

убедиться, что он настроен правильно.

В нашем случае установлен интерфейс UFW, поэтому воспользуемся следующей командой:

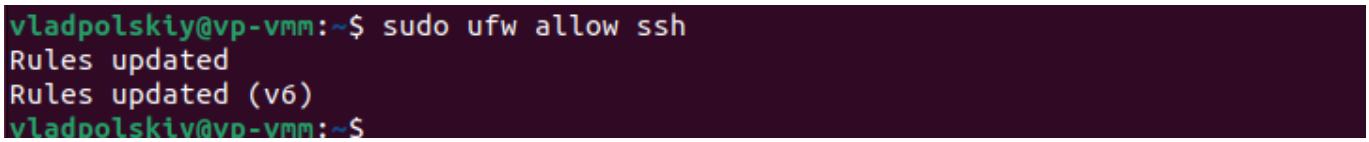
```
sudo ufw status
```

Результат команды представлен на картинке ниже.



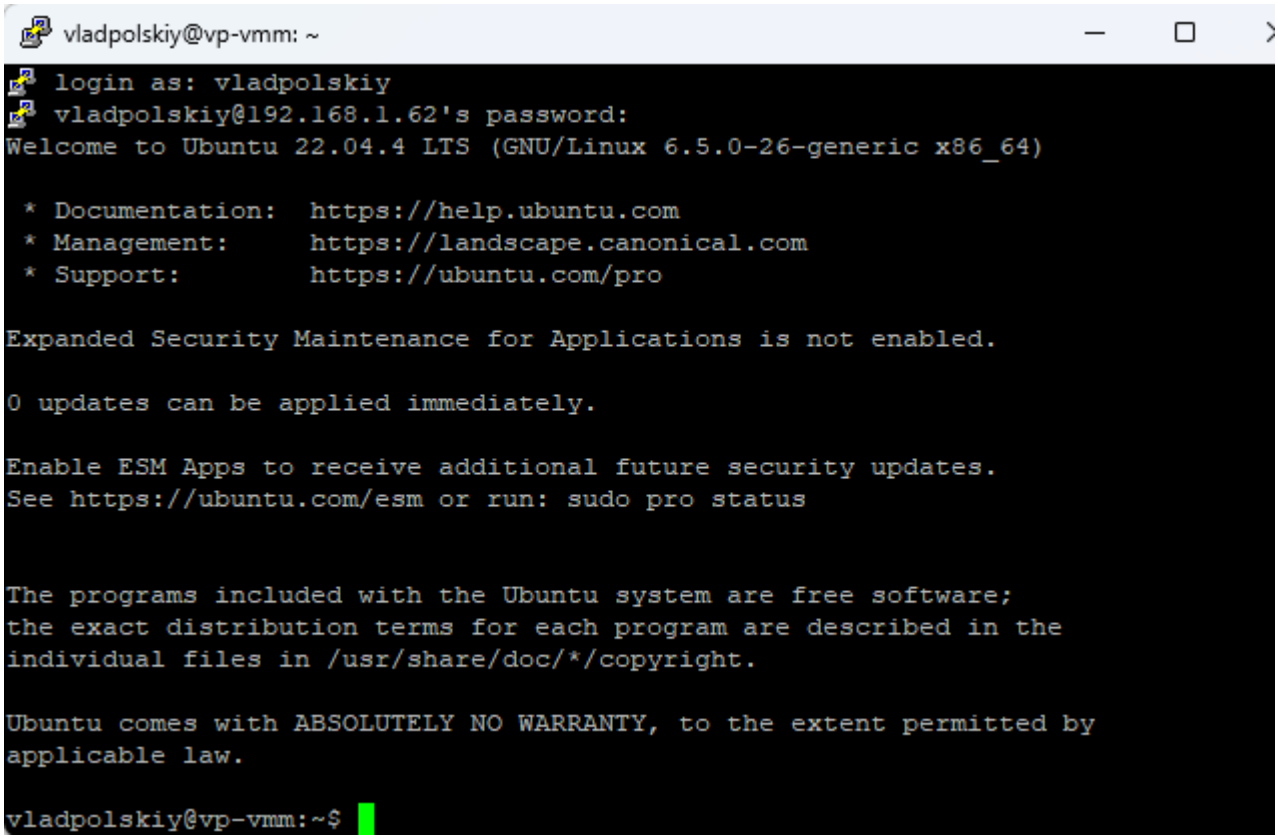
Как видно по картинке, трафик по протоколу SSH разрешен. Если у вас он отсутствует в списке, то необходимо разрешить входящие SSH-соединения. В этом поможет команда:

```
sudo ufw allow ssh
```



Подключение к серверу

После выполнения всех предыдущих шагов можно наконец переходить ко входу на сервер, используя протокол SSH и программу PuTTy.



Для подключения к серверу пользователю понадобятся его IP-адрес или доменное имя.

Настройка конфигурации SSH

Основные настройки OpenSSH-сервера хранятся в главном конфигурационном файле – sshd_config (расположение: /etc/ssh). Перед тем, как приступать к редактированию, необходимо создать резервную копию данного файла:

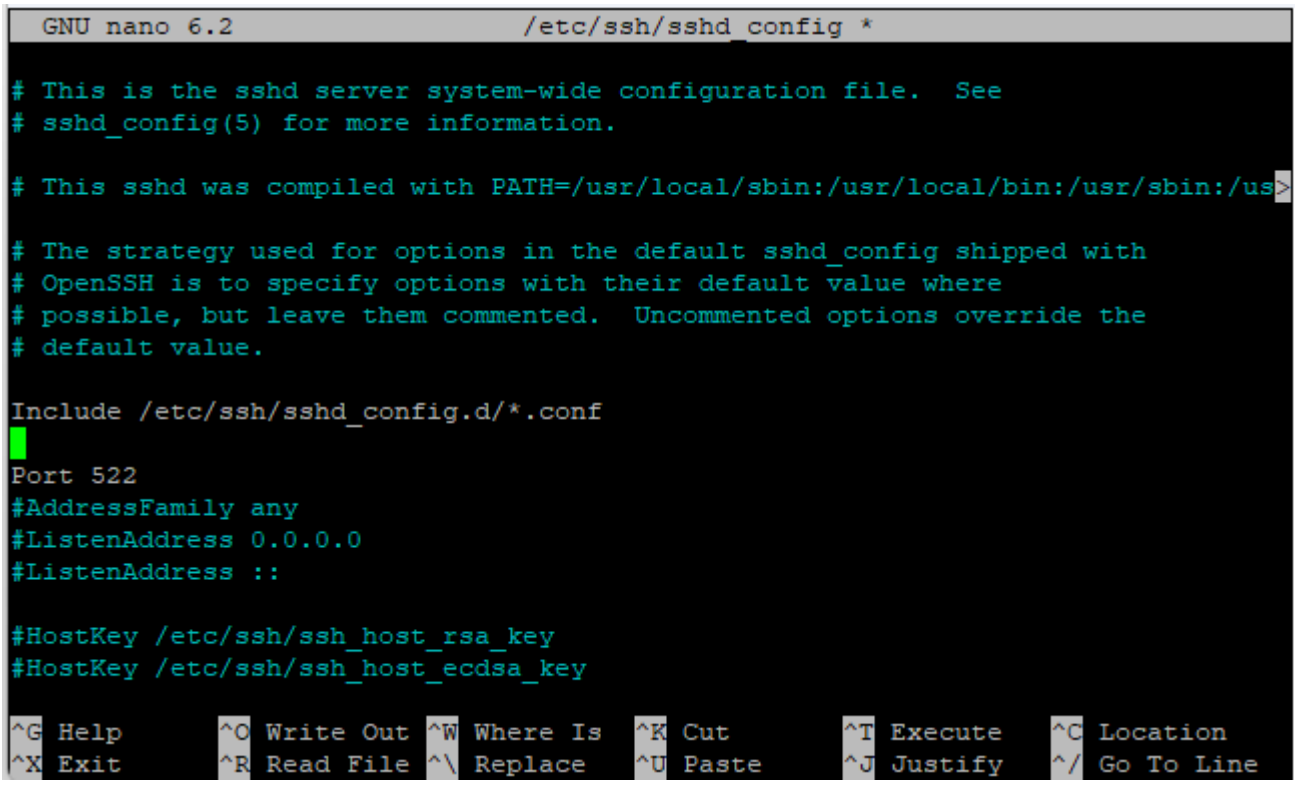
```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.initial
```

Теперь в случае возникновения ошибок после редактирования конфигурационного файла, пользователь без проблем сможет вернуть его к первоначальному виду.

После создания резервной копии можно переходить непосредственно к редактированию конфигурационного файла. Для этого откроем его, используя терминал и редактор Nano:

```
sudo nano /etc/ssh/sshd_config
```

В открывшемся файле сразу изменим значение порта на более безопасное. Лучше всего устанавливать значения из динамического диапазона портов (49152 — 65535), при этом использовать набор отличных друг от друга цифр для дополнительной безопасности. Например, отредактируем значение порта на 49532. Для этого разкомментируем соответствующую строку в файле и изменим значения порта, как показано на картинке ниже.



Кроме данной настройки, необходимо изменить режим аутентификации по паролю на более безопасный по ключу. Для этого раскомментируем соответствующую строку, значение которой должно быть “Yes”, как показано на картинке ниже.

```
  GNU nano 6.2                    /etc/ssh/sshd_config *
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
```

Аутентификацию по ключу разрешена. О том, как сгенерировать и использовать пару SSH-ключей, было подробно рассказано в соответствующей статье.

Теперь запретим вход на сервер под суперпользователем. Для этого также изменим значение соответствующей строки, как показано на рисунке ниже.

```
  GNU nano 6.2                    /etc/ssh/sshd_config *

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
```

Кроме внесенных выше изменений, перечислим ниже основные директивы конфигурационного файла, которые отвечают за безопасность сервера:

- **UseDNS** – отвечает за проверку соответствия имени хоста с его IP -адресом. Значение “Yes” включает этот параметр.
- **PermitEmptyPasswords** – данный параметр запрещает использовать пустые пароли при аутентификации, если в качестве значения указано “No”.

- **MaxAuthTries** – ограничение на количество неудачных попыток подключения к серверу в рамках одного сеанса связи. В качестве значения передается число.
- **AllowUsers и AllowGroups** – данные параметры отвечает за список пользователей и групп соответственно, которым разрешен доступ к серверу.

```
# AllowUsers Пользователь1, пользователь2, пользователь3
# AllowGroups Группа1, группа2, группа3
```

- **Login GraceTime** – параметр, отвечающий за время, предоставляемое для успешной авторизации. Рекомендуем уменьшить значение данного параметра в 4 раза.
- **ClientAliveInterval** – ограничение на время бездействия пользователя. При выходе за указанную границу происходит отключение пользователя.

После внесения всех изменений в главный конфигурационный файл, необходимо их сохранить и закрыть редактор. После перезагружаем службу, чтобы все изменения вступили в силу:

```
sudo systemctl restart ssh
```

Заключение

В данной статье была продемонстрирована подробная инструкция по установке и настройке SSH в Ubuntu 22.04. Также был описан процесс внесения изменений в главный файл конфигурации для повышения безопасности. Благодаря данной инструкции пользователь сможет выполнить безопасное удаленное подключение к серверу и не беспокоиться о потере или краже передаваемых данных.

Удаление PHP (необязательно)

Если какая-либо версия PHP больше не требуется, ее можно удалить из системы. Это освободит дисковое пространство, а также повысит безопасность системы.

Чтобы удалить любую версию PHP, просто введите:

```
sudo apt remove php7.4
```

Также удалите все модули для этой версии с помощью следующей команды:

```
sudo apt remove php7.4-*
```

```
alisa@linux:~$ sudo apt remove php7.4
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libapache2-mod-php7.4 php7.4-cli php7.4-common php7.4-json php7.4-opcache
  php7.4-readline
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  php7.4
0 upgraded, 0 newly installed, 1 to remove and 3 not upgraded.
After this operation, 88.1 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 75852 files and directories currently installed.)
Removing php7.4 (1:7.4.33-8+ubuntu22.04.1+deb.sury.org+1) ...
alisa@linux:~$
```

Проверим, что версия php7.4 удалена

```
sudo update-alternatives --config php
```

```
alisa@linux:~$ update-alternatives --config php
There are 3 choices for the alternative php (providing /usr/bin/php).

  Selection    Path             Priority   Status
------------------------------------------------------------
  0            /usr/bin/php8.3   83        auto mode
* 1            /usr/bin/php8.1   81        manual mode
  2            /usr/bin/php8.2   82        manual mode
  3            /usr/bin/php8.3   83        manual mode

Press <enter> to keep the current choice[*], or type selection number:
```

Ссылки и Дополнения

- Ondrej PPA
- Ссылка на оригинальную статью
- Команда Update-alternatives: подробное руководство для пользователей Linux
- Как переключиться между несколькими версиями PHP в Debian 11.12.10