

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И
МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ
ФЕДЕРАЦИИ

«Сибирский государственный университет
телекоммуникаций и информатики»
(СибГУТИ)

ОТЧЕТ
по дисциплине
«*WEB-технологии*»

по теме:
НАСТРОЙКА СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ WordPress

Студент:
Группы ИКС-432

А.А. Пастухов

Преподаватель:

А.В. Андреев

Новосибирск 2025

Содержание

ВВЕДЕНИЕ	3
1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	4
2. ПРАКТИЧЕСКАЯ ЧАСТЬ.....	6
ЗАКЛЮЧЕНИЕ	12

ВВЕДЕНИЕ

В данной практической работе рассматривается процесс развертывания и настройки системы управления контентом (CMS) WordPress, а также сервера для безопасной передачи паролей и конфиденциальной информации – PrivateBin.

Работа состоит из двух частей:

1. **Установка и настройка WordPress** – включает подготовку виртуальной машины, развертывание LAMP-стека, создание базы данных, загрузку и конфигурацию WordPress.
2. **Развертывание PrivateBin** – предполагает настройку HTTPS, установку PrivateBin и проверку его работы в связке с почтовым сервером.

Цель работы – приобрести навыки работы с веб-серверами, CMS и инструментами безопасного обмена данными в Linux-среде

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

WordPress - это наиболее популярная система управления контентом (CMS), используемая для создания и управления веб-сайтами. Будучи открытым программным обеспечением, WordPress предоставляет гибкость и масштабируемость, что делает его подходящим решением как для простых блогов, так и для сложных корпоративных сайтов.

Frontend (пользовательская часть) — отображает контент.

- Backend (админ-панель) — управление содержимым, настройками, плагинами.
- База данных (MySQL/MariaDB) — хранит статьи, пользователей, настройки.
- Веб-сервер (Apache/Nginx) — обрабатывает HTTP-запросы.

PrivateBin - это open-source решение для безопасного обмена зашифрованными сообщениями.

Основные характеристики:

- Шифрование на стороне клиента (AES-256)
- Самоуничтожение сообщений после прочтения
- Отсутствие метаданных на сервере
- Поддержка различных форматов (текст, код, markdown)

Принцип работы:

1. Отправитель создает сообщение в браузере
2. Сообщение шифруется локально
3. На сервер передается только зашифрованный текст
4. Получатель получает ключ дешифровки отдельно
5. После прочтения сообщение уничтожается

Для обеспечения безопасности приложений необходимо:

- Использовать HTTPS (SSL/TLS)
- Регулярно обновлять ПО
- Ограничивать права доступа
- Настраивать брандмауэры

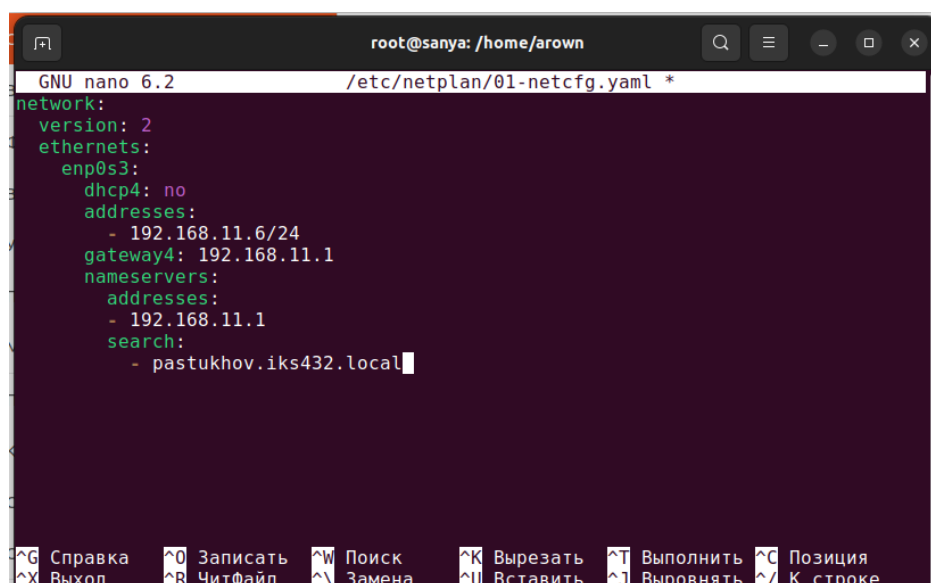
- Вести мониторинг логов

Использование виртуальных машин позволяет:

- Изолировать среды разработки
- Тестировать конфигурации
- Моделировать производственные среды
- Обеспечивать воспроизводимость результатов

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

Была выполнена подготовка виртуальной среды путем клонирования нового экземпляра виртуальной машины из золотого образа. Сетевой адаптер виртуальной машины был настроен на работу во внутренней сети, что обеспечивает изолированность данной виртуальной машины от внешних сетевых подключений, ограничивая ее взаимодействие только другими машинами в рамках внутренней сети. Для назначения статического IP-адреса был отредактирован конфигурационный файл сети, расположенный по пути `/etc/netplan/01-netcfg.yaml`, куда внесены соответствующие изменения, подробно отображенные на прилагаемом рисунке 1.



```
root@sanya: /home/arown
GNU nano 6.2 /etc/netplan/01-netcfg.yaml *
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.11.6/24
      gateway4: 192.168.11.1
      nameservers:
        addresses:
          - 192.168.11.1
      search:
        - pastukhov.iks432.local
```

Рисунок 1 – настройка netplan

После внесения изменений файл был сохранен, и конфигурация была применена с помощью команды: `netplan apply`

Для изменения имени хоста на сервере wordpress была выполнена команда: `hostnamectl set-hostname wordpress`. Далее было отредактировано имя сервера в файле `/etc/hosts`:

```
root@sanya: /home/arown
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 pastdesk.myguest.virtualbox.org pastdesk

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^M Замена ^U Вставить ^J Выводить ^_ К строке
```

Рисунок 2 – настройка hosts

Для добавления прямой и обратной записи в домен pastukhov.iks432.local выполнено подключение к серверу gate2, который является DNS-сервером. Были отредактированы файлы конфигурации DNS. Записи прямой зоны представлены на рисунке 3, записи обратной зоны представлены на рисунке 4:

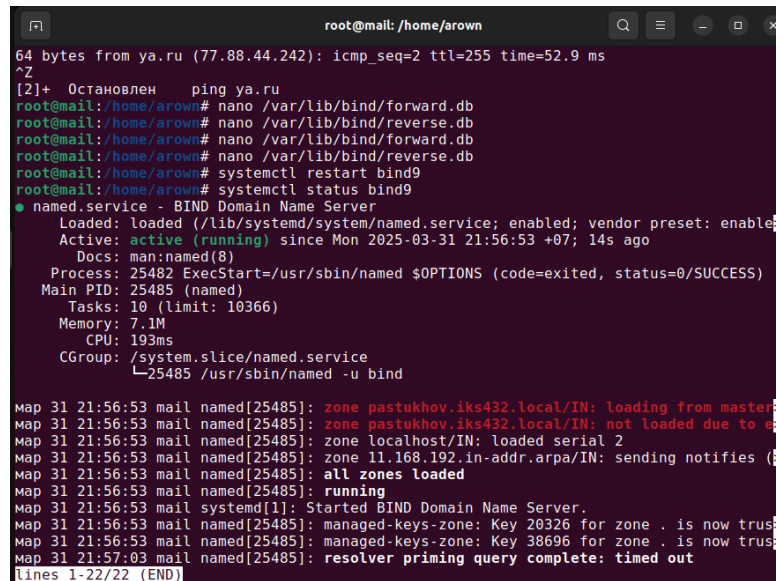
```
root@mail: /home/arown
GNU nano 6.2 /var/lib/bind/forward.db
$ORIGIN .
$TTL 86400 ; 1 day
pastukhov.iks432.local IN SOA gateway.pastukhov.iks432.local.
admin.pastukhov.iks432.local. (
    20110111 ;Serial
    10800 ;Refresh
    3600 ;Retry
    604800 ;Expirt
    86400 ;Minimum TTL
)
NS gateway.pastukhov.iks432.local.
$ORIGIN pastukhov.iks432.local.
gateway A 192.168.11.1
localhost A 127.0.0.1
$TTL 3600 ; 1 hour
mail A 192.168.11.5
pasdesk A 192.168.11.11
DHCID ( AAIBuModHxJgbqMZOXFzf3971I04SzaTkJ0z4+LIIdyZ
v4I= ) ; 2 1 32
$TTL 86400 ; 1 dayyy
mail A 192.168.11.1
localhost A 127.0.0.1
$TTL 3600 ; 1 hour
UB-DESKTOP A 192.168.11.51
DHCID ( AAEb3yU0xpsKa00rWrPu+6HP2GC1fD6u6FVIZ0yoS9wq
lps= ) ; 1 1 32
wordpress IN A 192.168.11.6
seafille A 192.168.11.4
Записано 28 строк
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^M Замена ^U Вставить ^J Выводить ^_ К строке
```

Рисунок 3 – записи forward.db

```
root@mail: /home/arown
GNU nano 6.2 /var/lib/bind/reverse.db
$TTL 86400 ; 1 day
11.168.192.in-addr.arpa. IN SOA pastukhov.iks432.local. pastukhov.iks432.local (
    20110114 ;Serial
    10800 ;Refresh
    3600 ;Retry
    604800 ;Expire
    3600 ;Minimum
)
NS mail.pastukhov.iks432.local.
$ORIGIN 16.168.192.in-addr.arpa.
PTR mail.pastukhov.iks432.local.
$TTL 3600 ; 1 hour
11 PTR pastdesk.pastukhov.iks432.local.
5 PTR mail.pastukhov.iks432.local.
5 PTR wordpress.pastukhov.iks432.local.
Записано 14 строк
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^M Замена ^U Вставить ^J Выводить ^_ К строке
```

Рисунок 4 – записи reverse.db

После внесения изменений DNS-сервер был перезапущен: `sudo systemctl restart bind9` и проверяем корректность работы сервера.

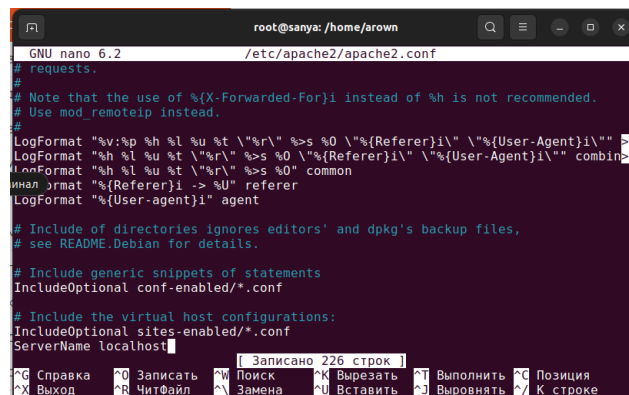


```
root@mail: /home/arown
64 bytes from ya.ru (77.88.44.242): icmp_seq=2 ttl=255 time=52.9 ms
^Z
[2]+  Остановлен  ping ya.ru
root@mail:/home/arown# nano /var/lib/bind/forward.db
root@mail:/home/arown# nano /var/lib/bind/reverse.db
root@mail:/home/arown# nano /var/lib/bind/forward.db
root@mail:/home/arown# nano /var/lib/bind/reverse.db
root@mail:/home/arown# systemctl restart bind9
root@mail:/home/arown# systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enable>
   Active: active (running) since Mon 2025-03-31 21:56:53 +07; 14s ago
     Docs: man:named(8)
  Process: 25482 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 25485 (named)
      Tasks: 10 (limit: 10366)
     Memory: 7.1M
        CPU: 193ms
    CGroup: /system.slice/named.service
            └─25485 /usr/sbin/named -u bind

мар 31 21:56:53 mail named[25485]: zone pastukhov.iks432.local/IN: loading from master>
мар 31 21:56:53 mail named[25485]: zone pastukhov.iks432.local/IN: not loaded due to e>
мар 31 21:56:53 mail named[25485]: zone localhost/IN: loaded serial 2
мар 31 21:56:53 mail named[25485]: zone 11.168.192.in-addr.arpa/IN: sending notifie>
мар 31 21:56:53 mail named[25485]: all zones loaded
мар 31 21:56:53 mail named[25485]: running
мар 31 21:56:53 mail systemd[1]: Started BIND Domain Name Server.
мар 31 21:56:53 mail named[25485]: managed-keys-zone: Key 20326 for zone . is now trus>
мар 31 21:56:53 mail named[25485]: managed-keys-zone: Key 38696 for zone . is now trus>
мар 31 21:57:03 mail named[25485]: resolver priming query complete: timed out
lines 1-22/22 (END)
```

Рисунок 5 – состояние bind9

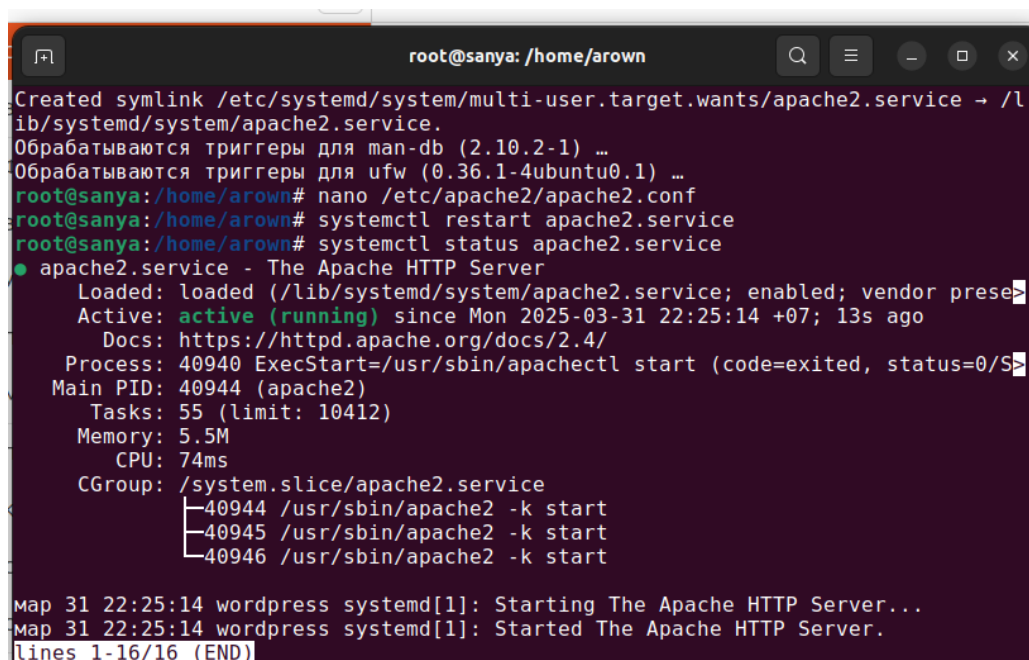
Перед установкой LAMP были обновлены списки пакетов: `sudo apt-get update`. Устанавливаем LAMP (Linux-Apache-MySQL-PHP) сервер на базе Ubuntu, в который входят Apache 2, PHP 5, и MySQL 5. После установки всех конфигураций, открывает файл командой “`nano /etc/apache2/apache2.conf`”. И в конец всего содержимого впишем без каких-либо изменений:



```
root@sanya: /home/arown
GNU nano 6.2 /etc/apache2/apache2.conf
# requests.
#
# Note that the use of %[X-Forwarded-For]i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\""
LogFormat "%h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\" combin>
LogFormat "%h %l %u %t \"%r\" %>s %0" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
#
# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.
#
# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf
#
# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf
ServerName localhost
```

Рисунок 6 – редакция /etc/apache2/apache2.conf

Сохраним. Перезапустим Apache 2 командой: “systemctl restart apache2.service”



```
root@sanya: /home/arown
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service -> /l
ib/systemd/system/apache2.service.
Обрабатываются триггеры для man-db (2.10.2-1) ...
Обрабатываются триггеры для ufw (0.36.1-4ubuntu0.1) ...
root@sanya:/home/arown# nano /etc/apache2/apache2.conf
root@sanya:/home/arown# systemctl restart apache2.service
root@sanya:/home/arown# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Mon 2025-03-31 22:25:14 +07; 13s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 40940 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S
 Main PID: 40944 (apache2)
    Tasks: 55 (limit: 10412)
   Memory: 5.5M
      CPU: 74ms
   CGroup: /system.slice/apache2.service
           └─40944 /usr/sbin/apache2 -k start
             └─40945 /usr/sbin/apache2 -k start
               └─40946 /usr/sbin/apache2 -k start

мар 31 22:25:14 wordpress systemd[1]: Starting The Apache HTTP Server...
мар 31 22:25:14 wordpress systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

Рисунок 7 – статус apache2 service

Изменим права на содержимое каталога www:

1. cd /var
2. chown -R root:root www
3. chmod -R 755 /var/www

Для управления и хранения информации о сайтах и пользователях WordPress использует
реляционную базу данных.

У нас установлен MySQL, который и выполняет эту функцию, но нужно создать базу

данных и пользователя, с которыми будет работать WordPress.

Для начала залогиньтесь в root-аккаунт (административный аккаунт) MySQL при помощи следующей команды: “mysql -u root -password”

Создаём отдельную базу данных wordpress, которой WordPress может управлять. Все операторы MySQL должны заканчиваться точкой с запятой (;), поэтому в случае возникновения проблем прежде всего проверьте этот момент.

Затем создадим отдельный пользовательский аккаунт MySQL с именем past и паролем qqqrwww98, который мы будем использовать исключительно для работы с новой базой данных.

Теперь, когда мы конфигурировали наше приложение, нам нужно скопировать его в корень документа Apache, где он будет предоставлен посетителям вебсайта. Один из самых простых и надежных способов переноса файлов из каталога в каталог – при помощи команды rsync. При использовании такого способа сохраняются разрешения и поддерживается целостность данных. В руководстве по Ubuntu LAMP корень документа расположен по адресу: /var/www/html/.

Чтобы перенести туда файлы WordPress, введите: “rsync -avP ~/wordpress/ /var/www/html/”

```
wordpress/wp-includes/class-wp-recovery-mode-link-service.php
wordpress/wp-includes/class-wp-embed-controller.php
wordpress/wp-includes/class-wp-block-supports.php
wordpress/wp-includes/PHPMailer/PHPMailer.php
wordpress/wp-includes/PHPMailer/PHPMailer.php
wordpress/wp-includes/PHPMailer/Exception.php
wordpress/wp-includes/PHPMailer/SMTP.php
wordpress/wp-includes/class-wp-recovery-mode.php
wordpress/wp-includes/class-wp-simplepie-sanitize-kses.php
wordpress/wp-includes/class-wp-date-query.php
wordpress/wp-includes/class-walker-comment.php
wordpress/wp-includes/nav-menu.php
wordpress/wp-includes/class-wp-paused-extensions-storage.php
wordpress/wp-includes/class-wp-block-metadata-registry.php
wordpress/wp-includes/class-wp-dependencies.php
wordpress/wp-signup.php
wordpress/wp-links-opml.php
root@wordpress:~# cd ~/wordpress
root@wordpress:~/wordpress# cp wp-config-sample.php wp-config.php
root@wordpress:~/wordpress# nano wp-config.php
root@wordpress:~/wordpress# rsync -avP ~/wordpress/ /var/www/html/
sending incremental file list
./
index.php 405 100% 0,00kB/s 0:00:00 (xfr#1, tr-chk=1020/1022)
license.txt 19,915 100% 18,99MB/s 0:00:00 (xfr#2, tr-chk=1019/1022)
readme.html 7,409 100% 7,07MB/s 0:00:00 (xfr#3, tr-chk=1018/1022)
wp-activate.php 7,587 100% 7,04MB/s 0:00:00 (xfr#4, tr-chk=1017/1022)
wp-blog-header.php 351 100% 342,77kB/s 0:00:00 (xfr#5, tr-chk=1016/1022)
wp-comments-post.php 2,323 100% 2,22MB/s 0:00:00 (xfr#6, tr-chk=1015/1022)
wp-config-sample.php 2,323 100% 2,22MB/s 0:00:00 (xfr#7, tr-chk=1014/1022)
```

Рисунок 8 - копирование в корень документа

Также удалите файл index.html из этой же директории командой: “rm /var/www/html/index.html”. Можно проверить состояние сервиса Apache

```
Country Name (2 letter code) [AU]:String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:
String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:^C
root@wordpress:~/wordpress# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mpm for ssl:
Module mpm already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@wordpress:~/wordpress# sudo systemctl restart apache2
root@wordpress:~/wordpress# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Sat 2025-03-29 19:13:18 +07; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 6224 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU
   Main PID: 6229 (apache2)
      Tasks: 55 (limit: 9438)
     Memory: 6.6M
        CPU: 87ms
   CGroup: /system.slice/apache2.service
           └─6229 /usr/sbin/apache2 -k start
             6230 /usr/sbin/apache2 -k start
             6231 /usr/sbin/apache2 -k start

мар 29 19:13:18 wordpress systemd[1]: Starting The Apache HTTP Server...
мар 29 19:13:18 wordpress apachectl[6228]: AH00112: Warning: DocumentRoot [/var
мар 29 19:13:18 wordpress systemd[1]: Started The Apache HTTP Server.
root@wordpress:~/wordpress#
```

Рисунок 10 – статус apache2 service

Заключение

В ходе выполнения практической работы №8 была успешно развернута система управления контентом WordPress и сервер для безопасной передачи данных PrivateBin на двух виртуальных машинах Ubuntu Server 22.04. Для WordPress предварительно настроен LAMP-стек (Apache, MySQL, PHP), создана база данных с пользователем author, после чего файлы CMS были загружены и перемещены в /var/www/html/. Настройка подключения к БД выполнена через wp-config.php, а завершающий этап установки проведен через веб-интерфейс. Для PrivateBin установлены Apache и PHP, создан самоподписанный SSL-сертификат для HTTPS.