**Отчет по самостоятельной работе №**2

**по дисциплине МДК 01.02 “Технология разработки программного обеспечения”.**

Выполнил: студент

группы 118

Меркушин Евгений Романович

Дата 19.11.2024

**Цель работы**

Цель работы – изучить процесс установки и настройки веб-сервера Apache на ОС Ubuntu для обеспечения работы веб-приложений.

**Основная структура задания**

# Задание 1: Установка Apache

1. **Установите Apache с помощью пакетного менеджера apt**:
2. sudo apt update

sudo apt install apache2

1. **Убедитесь, что веб-сервер запущен**:После установки Apache автоматически запускается. Чтобы проверить его статус, выполните команду:

systemctl status apache2

Вы увидите информацию о статусе службы. Если служба остановлена, запустите её командой:

sudo systemctl start apache2

# Задание 2: Настройка брандмауэра

1. **Откройте доступ к порту 80 с помощью IPTABLES**:Для того чтобы открыть доступ к веб-серверу через порт 80, добавьте правило в iptables:
2. # Разрешить входящие подключения на порт 80
3. sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
4. # Сохранить изменения в правилах

sudo netfilter-persistent save

1. **Убедитесь, что веб-сервер доступен через браузер**:Перейдите в браузере по адресу вашего сервера (например, http://your\_server\_ip/) и убедитесь, что отображается страница по умолчанию Apache.

# Задание 3: Настройка виртуальных хостов

1. **Настройте виртуальные хосты для размещения нескольких сайтов на одном сервере**:Создайте конфигурационные файлы для каждого сайта в директории /etc/apache2/sites-available/.Например, создайте файл site1.conf для первого сайта:
2. <VirtualHost \*:80>
3. ServerName site1.example.com
4. DocumentRoot /var/www/site1/public\_html
6. ErrorLog ${APACHE\_LOG\_DIR}/error.log
7. CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Аналогично создайте конфигурацию для второго сайта (site2.conf):

<VirtualHost \*:80>

ServerName site2.example.com

DocumentRoot /var/www/site2/public\_html

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Затем активируйте эти сайты:

sudo a2ensite site1.conf

sudo a2ensite site2.conf

И перезапустите Apache:

sudo service apache2 restart

1. **Убедитесь, что каждый сайт работает корректно**:Добавьте записи в ваш локальный файл /etc/hosts, чтобы связать доменные имена с IP-адресом сервера:
2. your\_server\_ip site1.example.com

your\_server\_ip site2.example.com

Теперь откройте оба сайта в браузерах по их доменным именам и проверьте, что загружаются правильные страницы.

# Задание 4: SSL-сертификаты

1. **Установите и настройте SSL-сертификаты для обеспечения защищённого соединения**:Установите пакет certbot от Let's Encrypt:

sudo apt install certbot python3-certbot-apache

Запросите сертификат для ваших доменов:

sudo certbot --apache -d site1.example.com -d www.site1.example.com

Certbot предложит вам выбрать опции настройки, следуйте инструкциям.

1. **Проверьте корректность работы HTTPS**:Перейдите по адресам https://site1.example.com и https://www.site1.example.com. Убедитесь, что соединение защищено, а сертификаты валидны.

# Задание 5: Оптимизация Apache

1. **Настройте параметры конфигурации для оптимизации производительности веб-сервера**:Отредактируйте файл /etc/apache2/apache2.conf, добавив или изменив следующие параметры:
   * **MaxKeepAliveRequests**: Максимальное количество запросов, которое может обработать одно соединение.

MaxKeepAliveRequests 100

* + **KeepAliveTimeout**: Время ожидания нового запроса после завершения предыдущего.

KeepAliveTimeout 15

1. Также можно настроить параметры MPM-модуля (worker или event). Например, для worker-модуля:
2. В файле /etc/apache2/mods-available/mpm\_worker.conf:
3. <IfModule mpm\_worker\_module>
4. StartServers 2
5. MinSpareThreads 25
6. MaxSpareThreads 75
7. ThreadLimit 64
8. ThreadsPerChild 50
9. MaxRequestWorkers 150
10. MaxConnectionsPerChild 5000
11. </IfModule>
12. Не забудьте сохранить изменения и перезагрузить Apache:
13. sudo service apache2 reload
14. **Проверьте производительность веб-приложения**:Используйте инструменты для тестирования нагрузки, такие как ab (Apache Benchmark), чтобы протестировать производительность:

ab -n 1000 -c 100 http://your\_server\_ip/

Это отправит 1000 запросов с одновременной нагрузкой 100 пользователей. Анализируйте результаты, чтобы оценить время ответа и количество обработанных запросов в секунду.

**Выводы работы**

Эти шаги помогут установить, настроить и оптимизировать работу Apache.