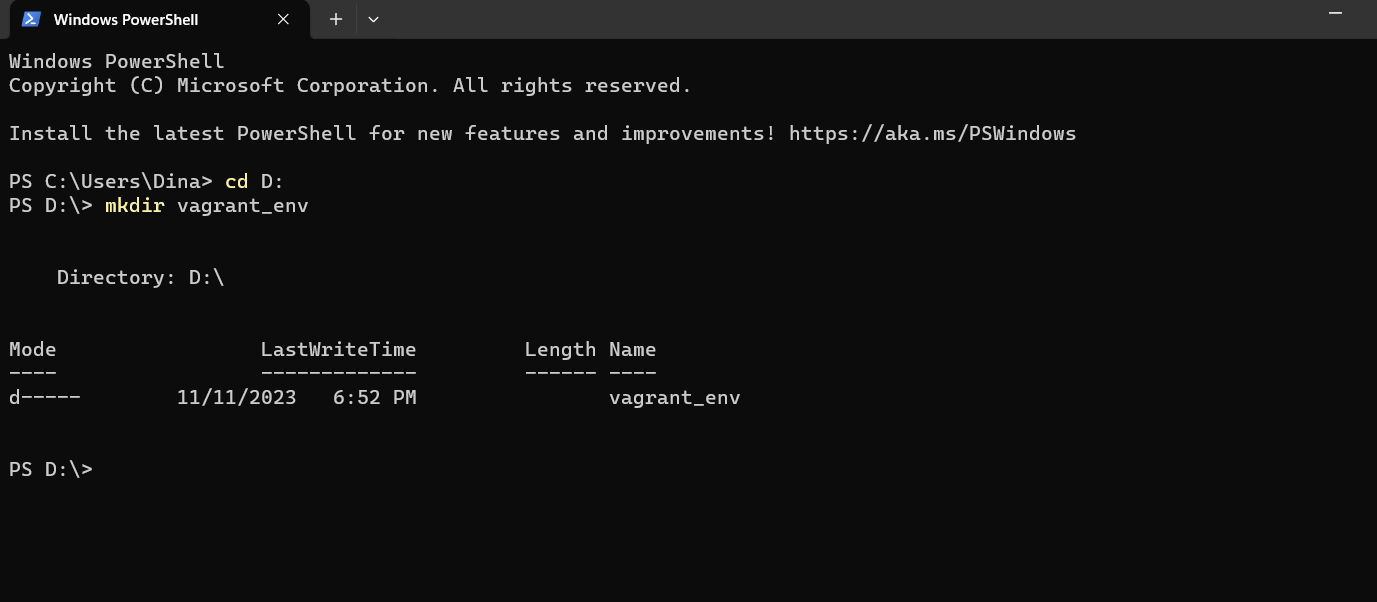
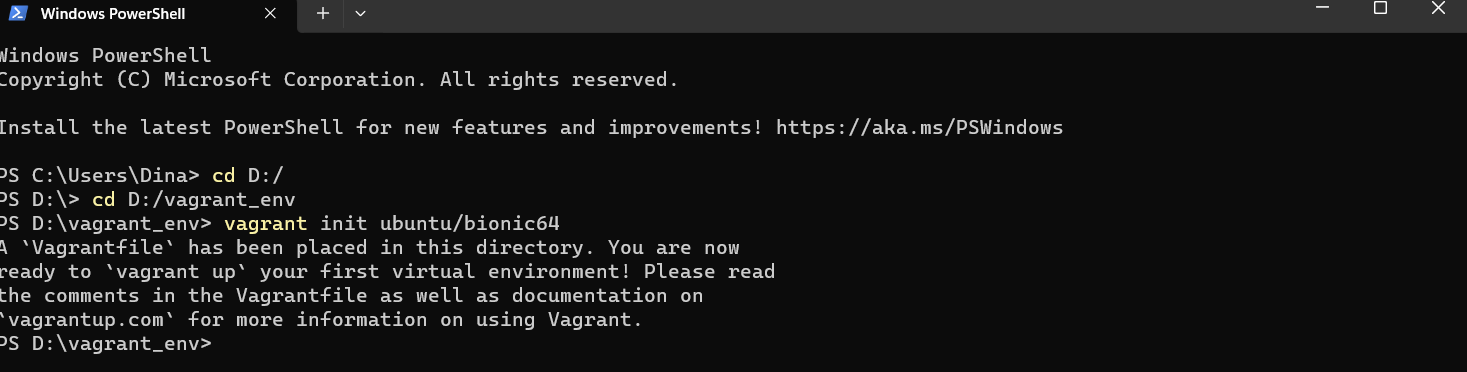
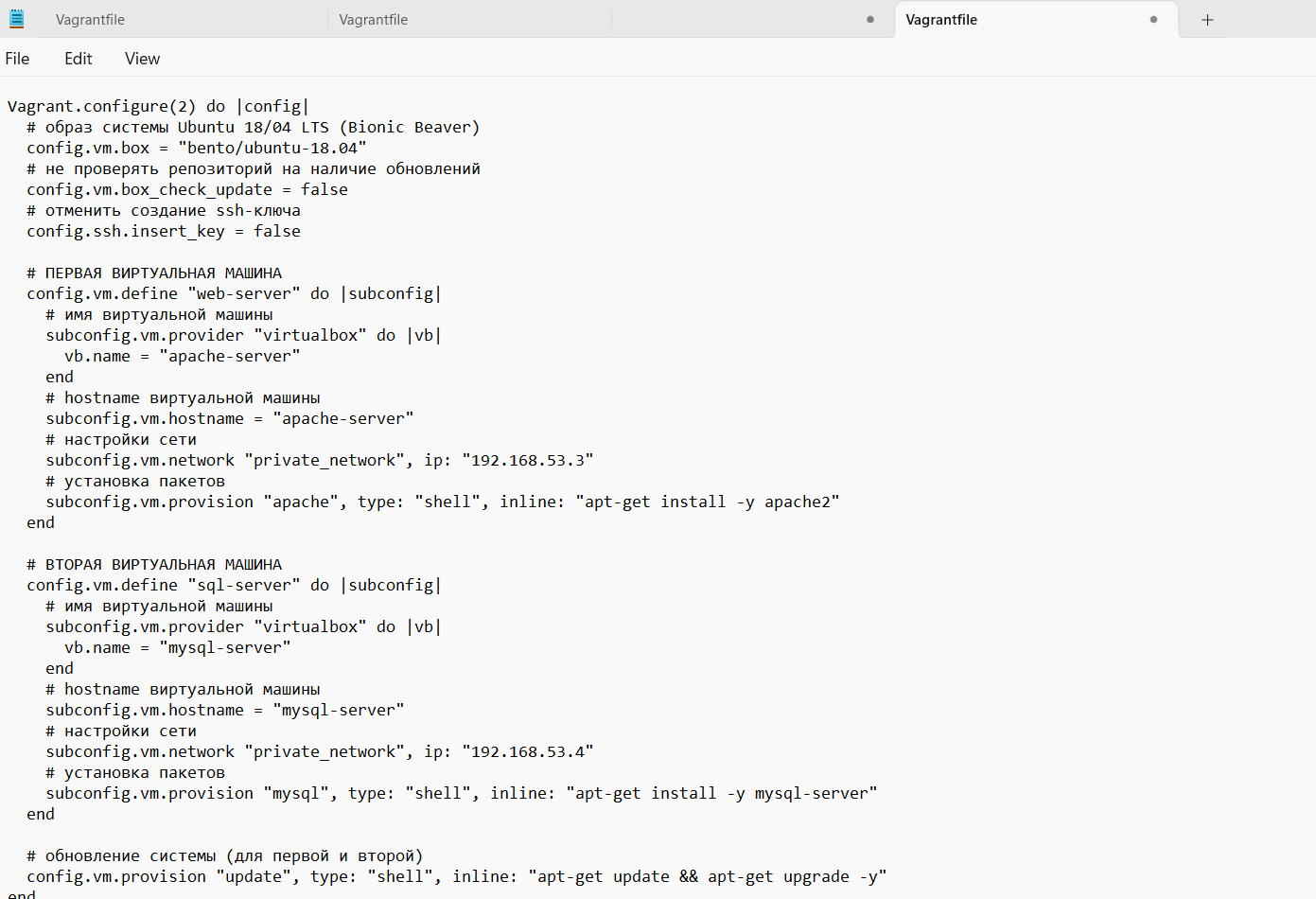
Задача 2.2: Создание многокомпонентного окружения с использованием Vagrant и VirtualBox После успешного овладения основами Vagrant и VirtualBox, вашей задачей будет создать многокомпонентное виртуальное окружение для тестирования и разработки. Это окружение будет включать в себя несколько виртуальных машин с разными конфигурациями.

**Инициализация проекта**: Создайте новую директорию для вашего многокомпонентного окружения. В этой директории выполните команду vagrant init, чтобы создать файл Vagrantfile.

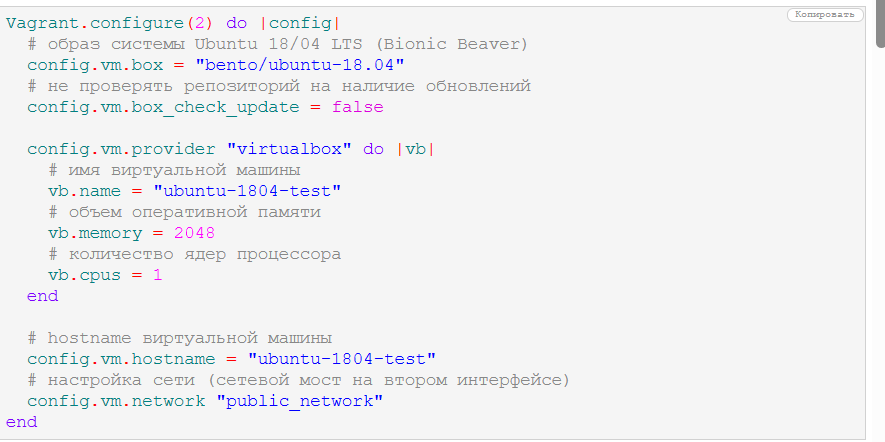
**Создание нескольких виртуальных машин**: В файле Vagrantfile определите несколько виртуальных машин с разными характеристиками (разное количество CPU, объем оперативной памяти и др.). Каждая из них должна использовать базовый образ вашей операционной системы по выбору.







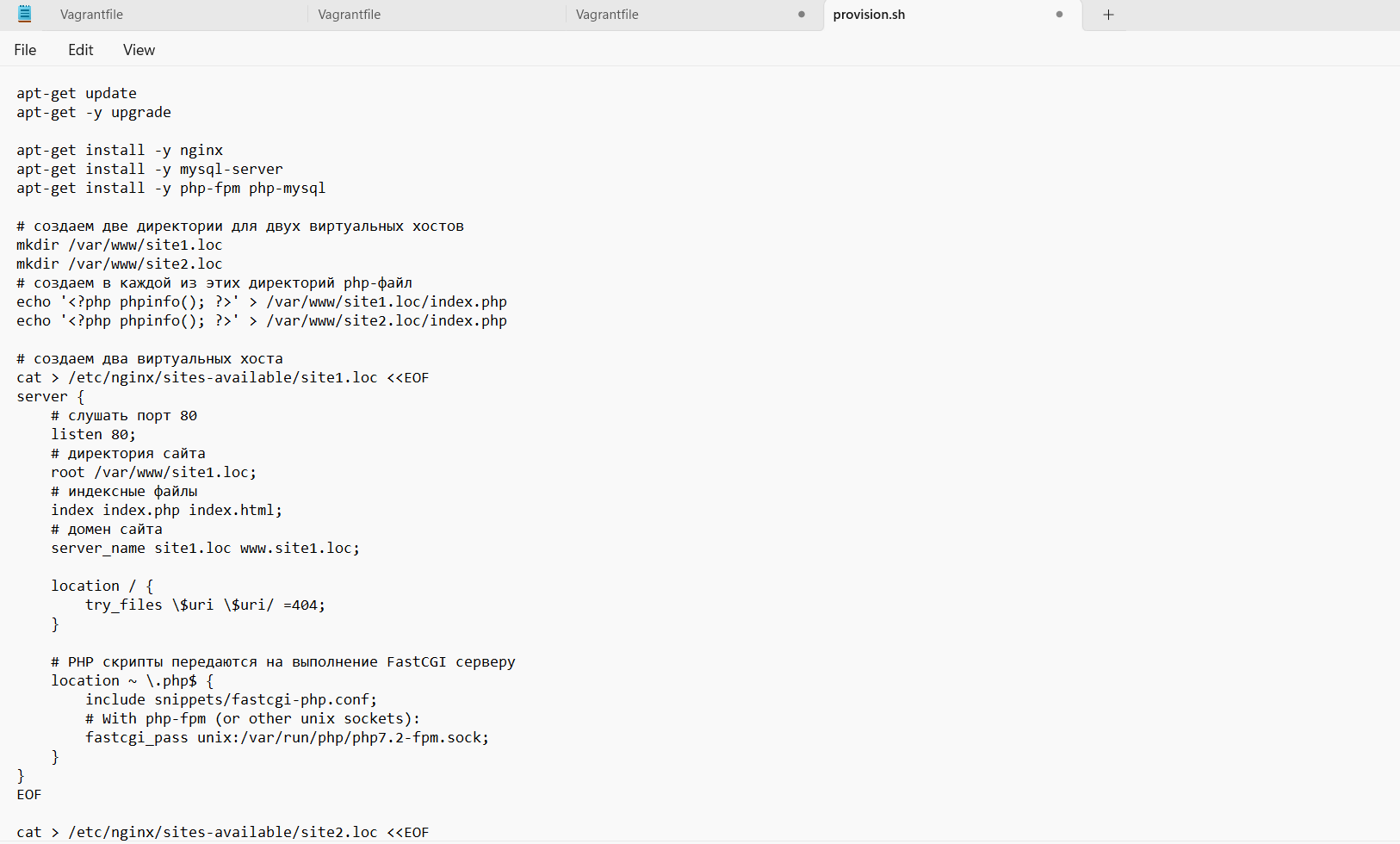
**Конфигурация сети**: Настройте сеть так, чтобы ваши виртуальные машины могли взаимодействовать друг с другом внутри виртуальной сети, а также имели доступ к интернету через NAT.



**Скрипты установки:** Создайте скрипты установки (Bash-скрипты) для каждой виртуальной машины, которые будут выполняться при их создании. Скрипты должны устанавливать и настраивать необходимое программное обеспечение или компоненты.

Установка Nginx, MySQL и PHP

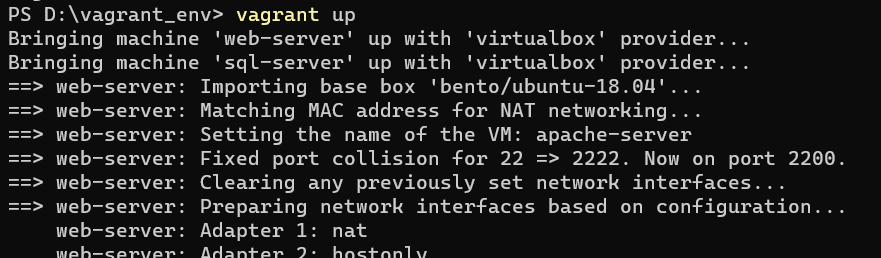
Во время создания виртуальной машины Vagrant запустит скрипт provision.sh, который установит Nginx, MySQL, PHP + создаст два виртуальных хоста.



Изображение выглядит как текст, снимок экрана, Шрифт, документ

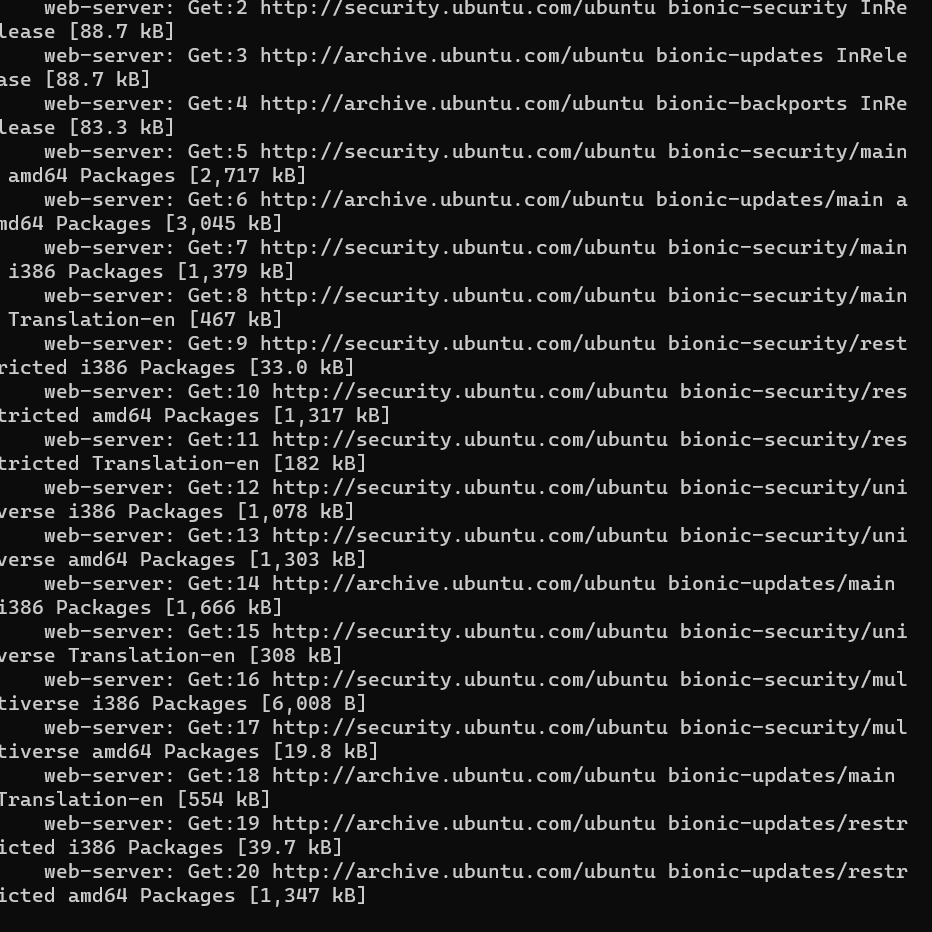
Автоматически созданное описание

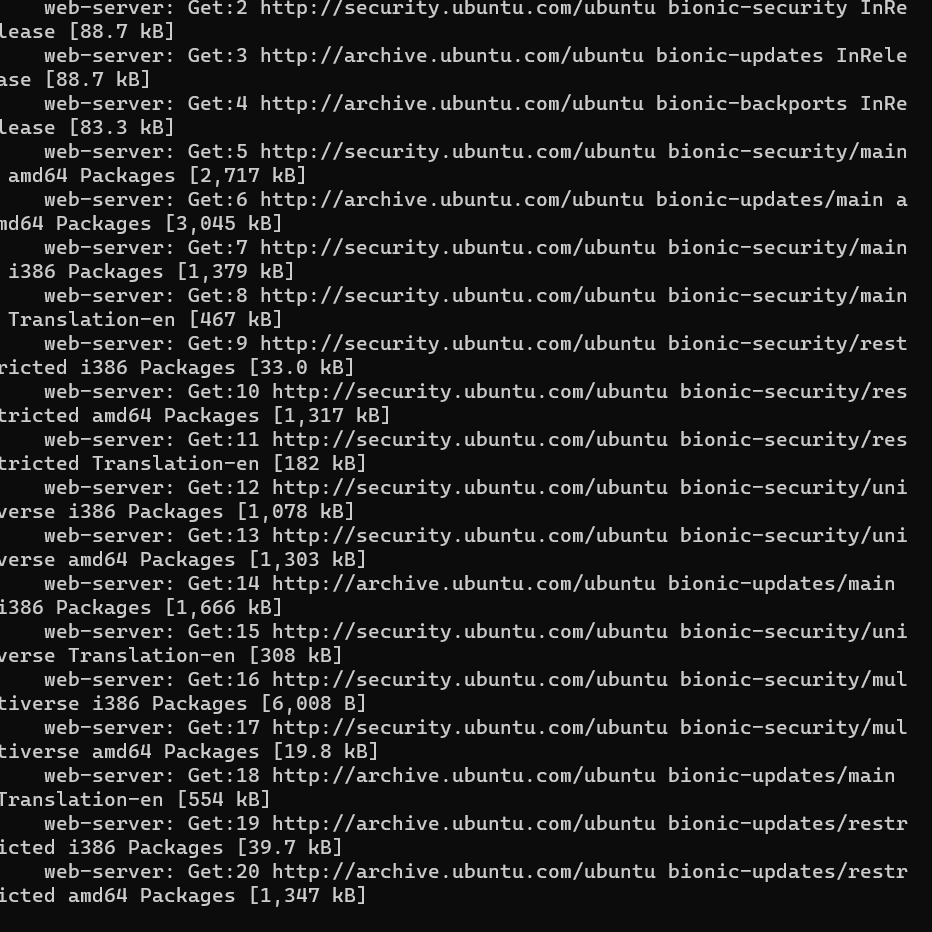
Установка

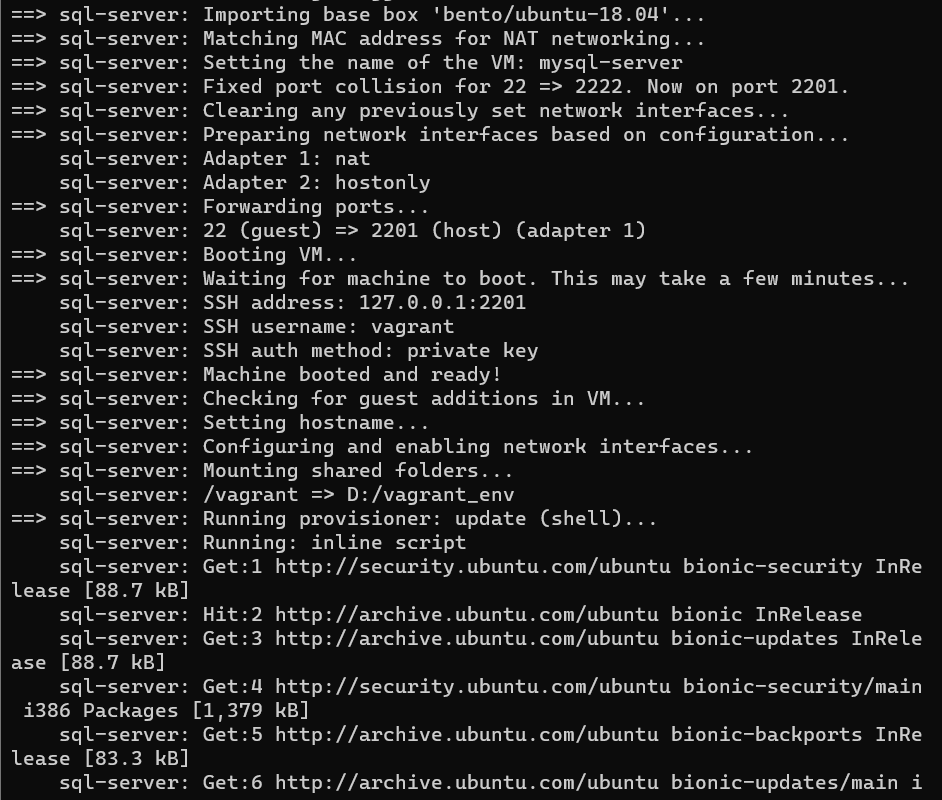


**Запуск и настройка**: Запустите все виртуальные машины с помощью команды vagrant up. Убедитесь, что они корректно настроены и могут взаимодействовать друг с другом.

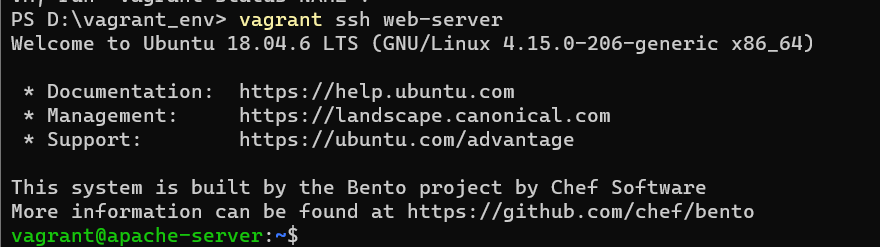
Ход установки первой машины:

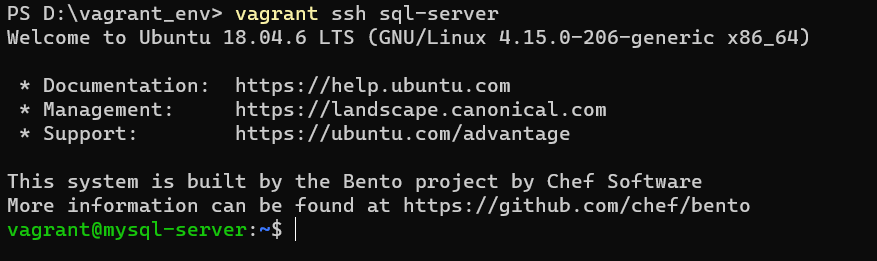


Ход установки второй машины:



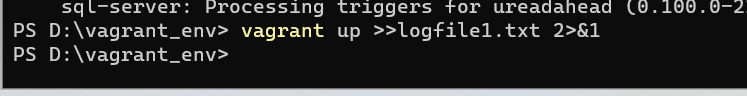
Взаимодействие и тестирование: Войдите в каждую виртуальную машину через SSH с помощью vagrant ssh и проверьте их работоспособность. Попробуйте взаимодействовать между машинами, например, пинговать друг друга.





Логирование и документация: Ведите лог своих действий и подготовьте документацию, описывающую создание и настройку каждой виртуальной машины. Укажите, какие проблемы возникли, и как вы их решали.

Для логирования наших действий в файле в командной строке Windows мы использовали оператор >>, который направляет вывод команды в файл. Вот пример логирования в файл logfile.txt:

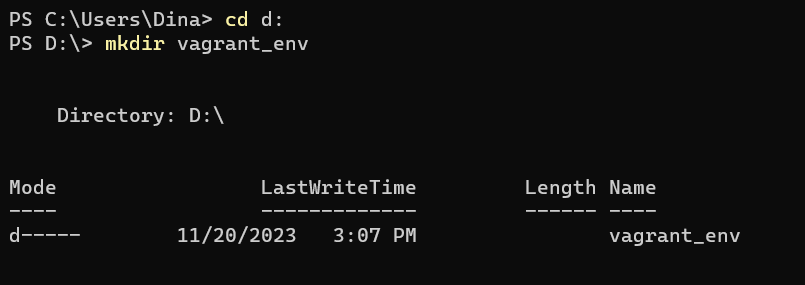


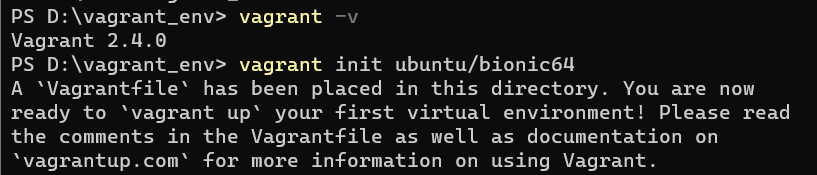
Результатом нашего логирования является текстовой файл с расширением logfile.txt

Задача 2.3: Автоматизация конфигурации и администрирование виртуальной машины с помощью Ansible

В этой задаче вам предстоит создать виртуальную машину с использованием Vagrant и VirtualBox, а затем автоматизировать её конфигурацию и выполнить несколько задач по администрированию Linux с помощью Ansible.

Инициализация проекта: Создайте новую директорию для вашего проекта. В этой директории выполните команду vagrant init, чтобы создать файл Vagrantfile.

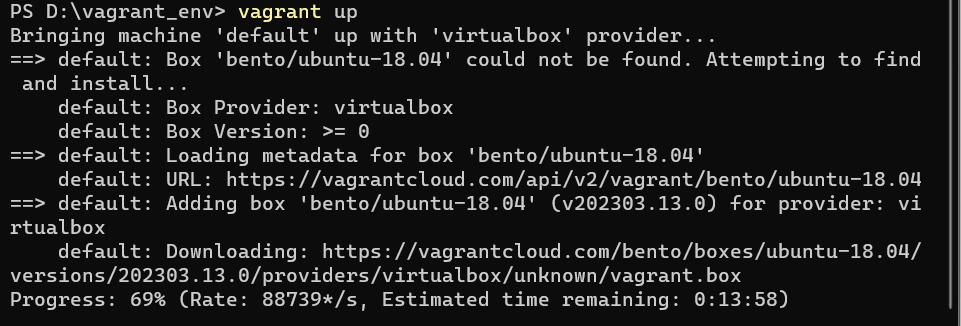




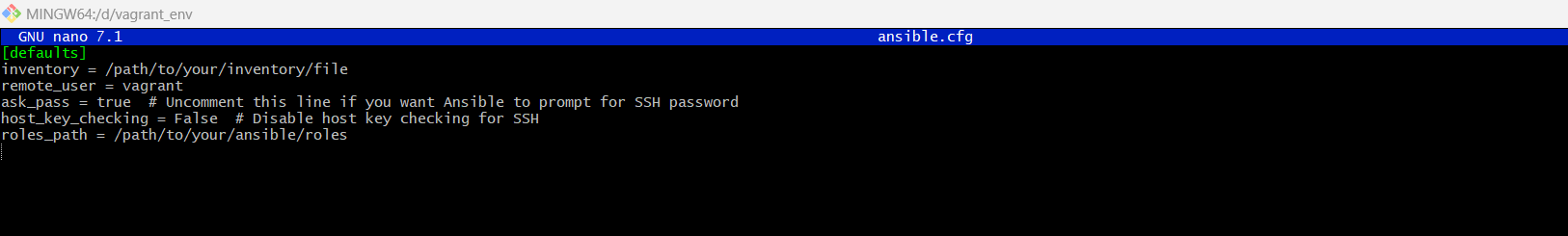
Создание виртуальной машины: В файле Vagrantfile определите виртуальную машину с базовым образом Ubuntu 20.04 (или другой по вашему выбору). Укажите параметры CPU, RAM и другие настройки виртуальной машины. Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Запуск виртуальной машины: Запустите виртуальную машину с помощью команды vagrant up.



Настройка Ansible: Установите Ansible на вашем локальном компьютере, если он ещё не установлен. Создайте файл ansible.cfg и настройте его для использования виртуальной машины.



Написание Ansible-плейбука:

Создайте Ansible-плейбук для конфигурации вашей виртуальной машины.

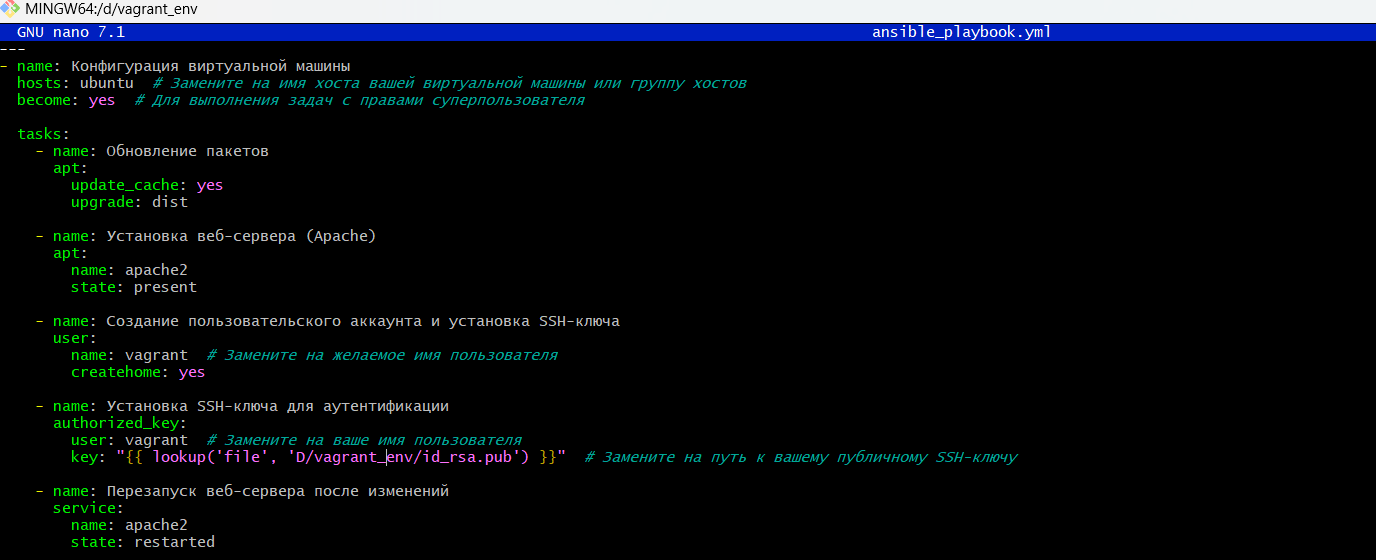
Плейбук должен включать в себя следующие задачи:

● Обновление пакетов на виртуальной машине.

● Установка и настройка веб-сервера (например, Apache или Nginx).

● Создание пользовательского аккаунта и установка SSH-ключа для аутентификации.

● Применение плейбука: Запустите Ansible-плейбук для конфигурации виртуальной машины. Убедитесь, что все задачи выполнились успешно.

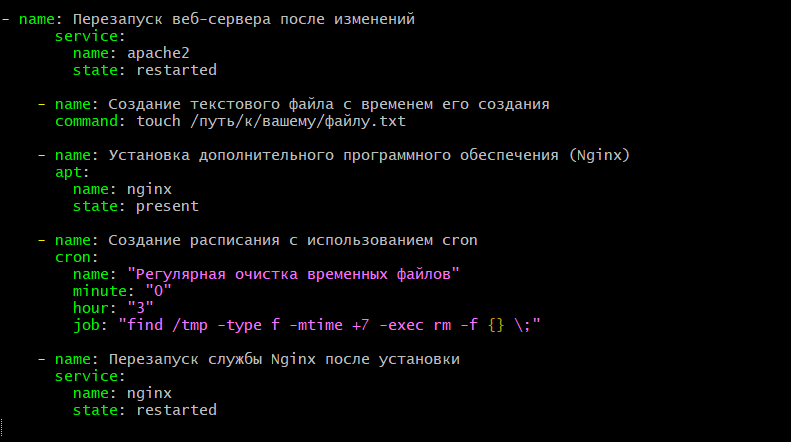


Задачи по администрированию Linux: Добавьте следующие задачи в ваш Ansible-плейбук:

● Создание текстового файла с временем его создания.

● Установка дополнительного программного обеспечения (например nginx, apache2, etc).

● Создание расписания с использованием cron для выполнения задачи (например, регулярной очистки временных файлов).



Логирование: Ведите лог своих действий и операций, выполненных с использованием Ansible.

Отчет: Подготовьте отчет, описывающий все действия, выполненные вами при создании и настройке виртуальной машины с использованием Ansible. Укажите, какие команды и плейбуки использовались, и какие результаты были получены. Также укажите, если возникли какие-либо проблемы и как вы их решали