Администрирование сетевых подсистем

Лабораторная работа №4

Тойчубекова Асель Нурлановна

Содержание

# 1. Цель работы

Приобретение практических навыков по установке и базовому конфигурированию HTTP-сервера Apache.

# 2. Задание

1. Установите необходимые для работы HTTP-сервера пакеты.
2. Запустите HTTP-сервер с базовой конфигурацией и проанализируйте его работу.
3. Настройте виртуальный хостинг.
4. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке HTTP-сервера во внутреннем окружении виртуальной машины server. Соответствующим образом внесите изменения в Vagrantfile.

# 3. Теоретическое введение

HTTP (HyperText Transfer Protocol) — протокол передачи данных в сети, основанный на модели «клиент–сервер». Гипертекст — это текст со ссылками на другие записи или страницы.

HTTP-сервер (например, Apache, nginx, Lighttpd) принимает запросы и отправляет ответы. HTTP-клиент (браузер или веб-приложение) отправляет запросы и получает ответы.

В Linux для работы с HTTP используется демон httpd. Основной конфигурационный файл Apache — httpd.conf, где задаются директивы для управления сервером.

# 4. Выполнение лабораторной работы

Для начала лабораторной работы я запускаю вм server. Далее перехожу к моему пользователю и и перехожу в режим суперпользователя. Затем установливаю из репозитория стандартный веб-сервер (HTTP-сервер и утилиты httpd,криптоутилиты и пр.) ([рис. 1](#fig-001) и [рис. 2](#fig-002) )

|  |
| --- |
| Рисунок 1: Установка веб-сервера |

|  |
| --- |
| Рисунок 2: Установка веб-сервера |

Перейдем в каталог /etc/httpd/conf, там мы видим конфигурационный файл httpd.conf.

|  |
| --- |
| Рисунок 3: httpd.conf |

|  |
| --- |
| Рисунок 4: httpd.conf |

В файле httpd.conf описываются основные параметры работы веб-сервера Apache.

Общие параметры сервера

В httpd.conf существуют такие группы настроек:

* ServerRoot — корневой каталог Apache.
* PidFile — файл с идентификатором процесса сервера.
* Timeout — время ожидания ответов.

Сетевые настройки

* Listen — IP-адрес и порт для приёма запросов.
* ServerName — имя сервера (домен или хост).

Документы и каталоги

* DocumentRoot — корневой каталог сайта.
* — правила доступа к папкам.
* Alias — создание псевдонимов для каталогов.

Права и безопасность

* AllowOverride — разрешение или запрет использования .htaccess.
* Require — кто может получать доступ (например, all granted или по IP).

Модули

* LoadModule — подключение дополнительных модулей (например, PHP, SSL).
* Логирование
* ErrorLog — файл с ошибками.
* CustomLog — файл с журналом запросов.

Виртуальные хосты

* — настройка нескольких сайтов на одном сервере.

Далее перейдем в каталог /etc/httpd/conf.d. Просмотрим все конфигурационные файлы.

|  |
| --- |
| Рисунок 5: autoindex.conf |

|  |
| --- |
| Рисунок 6: autoindex.conf |

Файл autoindex.conf в Apache описывает настройки модуля mod\_autoindex, который отвечает за автоматическую генерацию индексных страниц при доступе к директориям без файла index.html или index.php.

Проще говоря, если в папке нет главного файла, Apache покажет список файлов и папок. Этот файл позволяет настроить, как именно будет выглядеть этот список и что будет отображаться.

В autoindex.conf задаются:

Включение/отключение генерации списков каталогов: Options +Indexes

или отключение: Options -Indexes

Формат отображения списков:

FancyIndexing — красивый вывод с иконками и дополнительной информацией.

HeaderName — указание файла, который будет отображаться вверху списка (например, HEADER.html).

ReadmeName — файл с дополнительной информацией внизу списка.

Сортировка и отображение информации о файлах:

IndexOptions — набор опций для вывода:

NameWidth=\* — ширина столбца имени файла

SuppressDescription — скрыть описания файлов

SuppressHTMLPreamble — убрать стандартный HTML-шаблон

Иконки для файлов и папок:

Настройка, какие иконки показывать для разных типов файлов (AddIcon, AddIconByEncoding и т.д.) распиши это в виде абзацев

|  |
| --- |
| Рисунок 7: fcgid.conf |

В этом файле fcgid.conf содержится конфигурация Apache для модуля mod\_fcgid, который обеспечивает поддержку FastCGI.

В комментариях указано назначение файла — настройка FastCGI через mod\_fcgid и ссылка на документацию.

Строка

AddHandler fcgid-script fcg fcgi fpl

говорит, что файлы с расширениями .fcg, .fcgi и .fpl должны обрабатываться как FastCGI-скрипты.

Далее задаются пути для служебных файлов:

FcgidIPCDir /run/mod\_fcgid FcgidProcessTableFile /run/mod\_fcgid/fcgid\_shm

FcgidIPCDir — директория для сокетов, через которые Apache взаимодействует с FastCGI-процессами.

FcgidProcessTableFile — файл для хранения таблицы процессов и разделяемой памяти, нужной для работы FastCGI.

Иными словами, этот конфигурационный файл определяет, какие файлы запускать через FastCGI и где хранить служебные данные для обмена между Apache и FastCGI-процессами.

|  |
| --- |
| Рисунок 8: manual.conf |

В файле manual.conf настроен доступ к встроенной документации Apache.

Директива

Alias /manual /usr/share/httpd/manual

указывает, что при обращении к адресу http://localhost/manual/ сервер будет выдавать файлы из каталога /usr/share/httpd/manual.

В блоке для этой папки заданы параметры:

Options Indexes — разрешает показ списка файлов, если нет индексной страницы.

AllowOverride None — отключает переопределение настроек через .htaccess.

Require all granted — открывает доступ всем пользователям.

Директива

* RedirectMatch 301 ^/manual/(?:da|de|en|es|fr|ja|ko|pt-br|ru|tr|zh-cn)(/.\*)$ “/manual$1”

выполняет перенаправление: если пользователь заходит на документацию с указанием языка (например, /manual/ru/), его перенаправляют на общий путь /manual/….

То есть этот файл отвечает за то, чтобы встроенное руководство Apache было доступно по адресу http://localhost/manual/ и корректно открывалось независимо от выбора языка.

|  |
| --- |
| Рисунок 9: ssl.conf |

Файл ssl.conf — это конфигурация Apache для поддержки SSL/TLS (HTTPS).

В нём указываются:

Включение SSL для виртуального хоста:

<VirtualHost \*:443> SSLEngine on

Путь к сертификату и приватному ключу:

SSLCertificateFile /etc/pki/tls/certs/server.crt

SSLCertificateKeyFile /etc/pki/tls/private/server.key

(Опционально) цепочка сертификатов:

SSLCertificateChainFile /etc/pki/tls/certs/ca-bundle.crt

Протоколы и шифры, которые разрешены или запрещены:

SSLProtocol all -SSLv2 -SSLv3

SSLCipherSuite HIGH:!aNULL:!MD5

Настройки виртуальных хостов на 443 порту (аналогично httpd.conf, но с поддержкой SSL).

Параметры безопасности, например:

использование только надёжных шифров,

включение HSTS,

запрет слабых протоколов (SSLv2, SSLv3).

Иными словами, ssl.conf определяет, как Apache обслуживает HTTPS-соединения: где взять сертификаты, какие протоколы и шифры использовать и какие сайты слушать на порту 443.

|  |
| --- |
| Рисунок 10: userdir.conf |

Файле userdir.conf задаются параметры работы модуля mod\_userdir, который позволяет отображать содержимое личных веб-папок пользователей (обычно ~/public\_html) по URL типа http://site/~username/.

Обёртка <IfModule mod\_userdir.c> — настройки применяются только если модуль включён.

UserDir disabled — функциональность отключена по умолчанию (чтобы не раскрывать наличие пользователей на системе). В комментариях показано, как её включить: заменить строку на UserDir public\_html.

Блок <Directory “/home/\*/public\_html”> — правила доступа для всех public\_html в домашних каталогах:

AllowOverride FileInfo AuthConfig Limit Indexes — какие директивы из .htaccess разрешено переопределять;

Options MultiViews Indexes SymLinks IfOwnerMatch Includes NoExec — набор опций (мультипросмотры, индексирование каталогов, симлинки, включаемые файлы и запрет выполнения бинарников и т.п.);

Require method GET POST OPTIONS — какие HTTP-методы разрешены (здесь только чтение/отправка форм и preflight).

Что обычно ещё указывают в userdir.conf:

Как включить/отключить UserDir глобально или для отдельных пользователей (UserDir disabled, UserDir public\_html, или UserDir disabled username).

Путь к пользовательской папке (по умолчанию public\_html, можно изменить).

Правила доступа: Require all granted / ограничения по IP / запрет для определённых пользователей.

Какие опции и разрешения действуют в ~/public\_html (AllowOverride, Options — часто FollowSymLinks, Indexes или запрет ExecCGI).

Безопасностные настройки: запрет выполнения скриптов, запрет доступа к скрытым файлам, ограничение методов, запрет .htaccess или наоборот — лишь нужные возможности.

Комментарии о рисках (утечка информации о пользователях, безопасность исполнения кода) и примеры включения/выключения.

|  |
| --- |
| Рисунок 11: welcome.conf |

Файл welcome.conf управляет отображением стандартной стартовой страницы Apache (“Welcome page”). если пользователь заходит на корневой URL (/), а индексного файла нет, сервер отдаст страницу .noindex.html из системного каталога. разрешает доступ ко всем файлам в каталоге /usr/share/httpd/noindex, где хранится шаблон “Welcome page”. Директивы Alias создают псевдонимы, чтобы сервер мог найти файлы для отображения стандартной страницы:

/.noindex.html → /usr/share/httpd/noindex/index.html` (сама страница приветствия)

/poweredby.png → /usr/share/httpd/icons/apache\_pb3.png` (иконка Apache)

/system\_noindex\_logo.png → /usr/share/httpd/icons/system\_noindex\_logo.png` (логотип системы).

Внесем изменения в настройки межсетевого экрана узла server, разрешив работу с http. ([рис. 12](#fig-012) и [рис. 13](#fig-013))

|  |
| --- |
| Рисунок 12: Настройка межсетевого экрана |

|  |
| --- |
| Рисунок 13: Настройка межсетевого экрана |

В дополнительном терминале запустим в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы. ([рис. 14](#fig-014))

|  |
| --- |
| Рисунок 14: Расширенный лог системных сообщений |

В первом терминале активируем и запустите HTTP-сервер. ([рис. 15](#fig-015))

|  |
| --- |
| Рисунок 15: Запуск HTTP-сервера |

Просмотрев расширенный лог системных сообщений, мы видим, что веб-сервер успешно запустился. ([рис. 16](#fig-016))

|  |
| --- |
| Рисунок 16: Расширенный лог системных сообщений |

Далее запустим вм client. А на вм server просмотрим лог ошибок работы веб-сервера. ([рис. 17](#fig-017))

|  |
| --- |
| Рисунок 17: Лог ошибок работы веб-сервера |

В логе описан обычный старт Apache: включены suEXEC и SELinux, сервер успешно запущен с указанными модулями и версиями. Ошибок нет, только информационные сообщения о конфигурации.

На виртуальной машине server запустим мониторинг доступа к веб-серверу. ([рис. 18](#fig-018))

|  |
| --- |
| Рисунок 18: Мониторинг доступа к веб-серверу |

Мониторин показывает историю обращений клиента: какие файлы запрашивались, с какими результатами (код состояния HTTP), размеры ответа и браузер клиента.

На виртуальной машине client запустим браузер и в адресной строке введем 192.168.1.1. Мы видим, что на экране появилась страничка http server test page. ([рис. 19](#fig-019))

|  |
| --- |
| Рисунок 19: http server test page |

Настроим виртуальный хостинг по двум DNS-адресам: server.antoychubekova.net и www.antoychubekova.net. Остановим работу DNS-сервера для внесения изменений в файлы описания DNS-зон. ([рис. 20](#fig-020))

|  |
| --- |
| Рисунок 20: Остановка работы DNS-сервера |

Добавьте запись для HTTP-сервера в конце файла прямой DNS-зоны /var/named/master/fz/antoychubekova.net ([рис. 22](#fig-022)) и в конце файла обратной зоны /var/named/master/rz/192.168.1. ([рис. 21](#fig-021))

|  |
| --- |
| Рисунок 21: Редактирование DNS-зоны |

|  |
| --- |
| Рисунок 22: Редактирование обратной DNS-зоны |

Удалим файлы журналов DNS: antoychubekova.net.jnl и 192.168.1.jnl. ([рис. 23](#fig-023))

|  |
| --- |
| Рисунок 23: Удаление журналов |

Перезапустим DNS-сервер. ([рис. 24](#fig-024))

|  |
| --- |
| Рисунок 24: Перезапуск DNS-сервера |

В каталоге /etc/httpd/conf.d создайте файлы server.antoychubekova.net.conf и www.antoychubekova.antoychubekova.conf. ([рис. 25](#fig-025))

|  |
| --- |
| Рисунок 25: Создание конфигурационных файлов |

Откроем на редактирование файл server.antoychubekova.antoychubekova.conf и внесем некоторые изменения. ([рис. 26](#fig-026))

|  |
| --- |
| Рисунок 26: Редактирование server.antoychubekova.net.conf |

Откроем на редактирование файл www.antoychubekova.net.conf и внесем некоторые изменения. ([рис. 27](#fig-027))

|  |
| --- |
| Рисунок 27: Редактирование www.antoychubekova.net.conf |

Перейдем в каталог /var/www/html, в котором должны находиться файлы с содержимым (контентом) веб-серверов, и создадим тестовые страницы для виртуальных веб-серверов server.antoychubekova.net и www.antoychubekova.net.

Для виртуального веб-сервера server.antoychubekova.net. ([рис. 28](#fig-028))

|  |
| --- |
| Рисунок 28: Создание тестовой страницы |

Откроем на редактирование файл index.html и внесем следующее содержание: Welcome to the server.antoychubekovar.net server. ([рис. 29](#fig-029))

|  |
| --- |
| Рисунок 29: Редактирование index.html |

Для виртуального веб-сервера www.antoychubekova.net. [рис. 30](#fig-030))

|  |
| --- |
| Рисунок 30: Создание тестовой страницы |

Откроем на редактирование файл index.html и внесем следующее содержание: Welcome to the www.antoychubekova.net. ([рис. 31](#fig-031))

|  |
| --- |
| Рисунок 31: Редактирование index.html |

Скорректируем права доступа в каталог с веб-контентом. Восстановим контекст безопасности в SELinux. Перезапустим HTTP-сервер. ([рис. 32](#fig-032))

|  |
| --- |
| Рисунок 32: Запуск HTTP-сервера |

На виртуальной машине client убедимся в корректном доступе к веб-серверу по адресам server.antoychubekova.net и www.antoychubekova.net в адресной строке веб-браузера. Мы видим, что странички успешно открываются. ([рис. 33](#fig-033) и [рис. 34](#fig-034) )

|  |
| --- |
| Рисунок 33: Страница server.antoychubekova.net |

|  |
| --- |
| Рисунок 34: Страница www.antoychubekova.net |

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог http, в который поместим в соответствующие подкаталоги конфигурационные файлы HTTP-сервера. ([рис. 35](#fig-035))

|  |
| --- |
| Рисунок 35: Конфигурационные файлы HTTP-сервера |

Заменим конфигурационные файлы DNS-сервера. ([рис. 36](#fig-036))

|  |
| --- |
| Рисунок 36: Замена конфигурационных файлов DNS-сервера |

В каталоге /vagrant/provision/server создадим исполняемый файл http.sh. ([рис. 37](#fig-037))

|  |
| --- |
| Рисунок 37: Создание исполняемого файла |

Открыв его на редактирование, пропишим в нём скрипт, который по сути, повторяет произведённые нами действия по установке и настройке HTTP-сервера. ([рис. 38](#fig-038))

|  |
| --- |
| Рисунок 38: Редактирование исполняемого файла |

Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле Vagrantfile необходимо добавить в конфигурации сервера следующую запись:

server.vm.provision “server http”,

type: “shell”,

preserve\_order: true,

path: “provision/server/http.sh” ([рис. 39](#fig-039))

|  |
| --- |
| Рисунок 39: Редактирование Vagrantfile |

# 5. Вывод

В ходе выполнения лабораторной работы я приобрела практические навыки по установке и базовому конфигурированию HTTP-сервера Apache.