Расширенное администрирование Linux. Блок 8.

v 1.03

Оглавление

WEB сервер Apache....1
Конфигурационный файл httpd.conf....2
Параметры WEB сервера Apache....2
Контейнеры....3
Опции....3
AllowOverride....4
Контроль доступа к ресурсам....4
Модуль mod_userdir.c....4
Сообщения об ошибках....5
Виртуальный хостинг....6
Лабораторная работа....6

WEB сервер Apache

WEB сервер Apache является самым популярным сервером в мире UNIX. Эта популярность достигнута по следующим причинам:

- Сервер выполнен по модульной структуре. То есть, все основные особенности работы сервера реализованы в виде подключаемых модулей. Благодаря этому при появлении новых возможностей они могут быть легко добавлены при помощи новых модулей. При обнаружении какой либо ошибки, легче исправить один модуль, а не весь сервер целиком. Так же, поскольку Apache распространяется в исходных кодах, с открытым API, Вы сами можете добавлять новые функции к WEB серверу.
- Apache можно использовать в различных операционных системах, в том числе и Windows. Запуск в Windows был возможен и в версии 1.3.x, но она не учитывала всех особенностей Windows и работала не так хорошо как в UNIX.
- С Apache великолепно интегрированы различные языки программирования, при помощи которых можно создавать динамические WEB сайты. Это в первую очередь Perl, PHP и Java.

WEB сервер Apache версии 2 имеет несколько возможных режимов управления процессами. Для каждого режима существует свой модуль (multiprocessing modules — MPM):

- MPM с ветвлением — имитирует архитектуру Apache предыдущей версии и создает столько дочерних процессов, сколько будет обслуживаться запросов.
- Потоковый MPM — во второй версии Apache реализована поддержка потоков (POSIX threads). Каждый процесс может иметь несколько потоков. Каждый поток обслуживает запросы клиентов. Если в Apache запущено 15 процессов и каждому процессу разрешено иметь 10 потоков, тогда сервер может обслужить 15x10=150 запросов одновременно.
- MPM с фиксированным числом дочерних процессов — основное различие

между этим модулем и потоковым состоит в том, что количество дочерних процессов неизменно и каждый процесс можно запускать различные идентификаторы пользователей и групп.

- MPM для других операционных систем — существует несколько MPM, предназначенных для работы сервера в других операционных системах: Windows, OS/2, Novell.

Чтобы узнать, какой модуль был использован при сборке WEB сервера, следует запустить программу httpd с опцией -l чтобы был показан список модулей.
В этом разделе будут рассмотрены общие вопросы конфигурации WEB сервера Apache, а также вопрос организации виртуального хостинга.

Конфигурационный файл httpd.conf

После запуска WEB сервер Apache читает свой конфигурационный файл httpd.conf. В Ubuntu Linux этот файл находиться в директории /etc/apache2.

Параметры WEB сервера Apache.

Ниже будут рассмотрены основные параметры, которые могут быть определены в конфигурационном файле WEB сервера Apache.

Параметр	Описание
`ServerRoot`	Параметр определяет вершину дерева каталогов WEB сервера. Этот каталог не предназначен для хранения html файлов.
`PidFile`	Определяет путь к файлу, в котором WEB сервер после запуска помещает свой PID.
`KeepAlive`	Если параметр имеет значение On — WEB сервер позволяет за одно подключение передавать сразу несколько ресурсов.
`MaxKeepAliveRequests`	Параметр определяет максимальное количество запросов в одном соединении.
`KeepAliveTimeout`	Параметр определяет время в секундах. Если в течении этого времени клиент не запросит следующий ресурс, сервер разорвет соединение.
`IfModule`	Директива проверяет — загружен ли соответствующий модуль и, если он загружен, позволяет использовать специфичные для этого модуля параметры.
`StartServers`	Количество дочерних серверов, создаваемых при запуске WEB сервера.
`MinSpareServers`	Минимальное количество экземпляров WEB сервера.
`MaxSpareServers`	Максимальное количество экземпляров WEB сервера.
`MaxClients`	Определяет количество запросов, обрабатываемых сервером.
`MaxRequestsPerChild`	Определяет максимальное количество запросов, которое может обработать один процесс.
`Listen`	Определяет IP адрес и порт, на которых WEB сервер слушает запросы. Если IP адрес не указан — сервер слушает запросы на всех сетевых интерфейсах.
`LoadModule`	Параметр связывает имя модуля и файл, в котором он находится.

Параметр	Описание
`Include`	Директива подключает внешние конфигурационные файлы.
`Port`	Определяет порт на котором будет слушать запросы WEB сервер.
`User`	Определяет пользователя, с правами которого будет работать WEB сервер.
`Group`	Определяет группу, с правами которой будет работать WEB сервер.
`ServerAdmin`	Определяет Email администратора сервера.
`DocumentRoot`	Определяет директорию, в которой находятся html файлы.

Контейнеры

В Apache можно использовать различные контейнеры, которые позволяют определять параметры для объектов:

- Directory — директорий
- DirectoryMatch — то же, что и предыдущий контейнер, но в ка- честве параметра используется регулярное выражение
- Files — файлов
- FilesMatch — то же, что и предыдущий контейнер, но в качестве параметра используется регулярное выражение
- Limit — методов протокола HTTP

Опции

Внутри контейнеров Directory можно использовать параметр Options, при помощи которого определяются различные опции. В таблице приведены опции, которые можно использовать в параметре:

Опция	Описание
None	Нет опций.
All	Все опции.
ExecCGI	Разрешает выполнение сценариев CGI.
FollowSymLinks	Сервер следует по символьным ссылкам.
Includes	Разрешается выполнение команд SSI (Server Side Includes).
IncludesNoExec	При использовании SSI запрещается ис- пользовать команды #exec и #include.
Indexes	Если в запросе не указывается конкрет- ный файл, а только директория, и для дан- ной директориия не определен параметр DirectoryIndex или не найден файл по умолчанию, сервер выдает список файлов и директорий, находящихся в запрашивае- мой директории.

SymLinksIfOwnerMatch	Сервер следует по символьной ссылке только в том случае, если объект, на кото- рый указывает ссылка, принадлежит тому же пользователю, что и сама символьная ссылка.
MultiViews	Позволяет выводить документ согласно языка клиента.

AllowOverride

Параметр AllowOverride используют внутри контейнера Directory. Параметр определяет, какие параметры, объявленные в файле .htaccess (или любом другом, заданном AccessFileName), могут быть переопределены пользователями.

В таблице показаны значения, которые можно указывать в качестве параметра AllowOverride:

Опция	Описание
AuthConfig	Разрешает использование параметров аутентификации и управления доступом.
FileInfo	Разрешает использование параметров, управляющих типами документов.
Indexes	Разрешает использование параметров, управляющих индексами директорий.
Limit	Разрешает использовать параметры, управляющие доступом.
Options	Разрешает использовать параметры, управляющие свойствами директорий.
None	Запрещает использовать любые парамет- ры.
All	Разрешает использовать все параметры.

Контроль доступа к ресурсам

Контроль доступа к ресурсам осуществляется при помощи параметров:

- Allow from — разрешает доступ
- Deny from — запрещает доступ
- Order — определяет порядок рассмотрения параметров Allow и Deny

Эти параметры можно использовать только в контейнерах или файлах .htaccess.

Модуль mod_userdir.c

При включении модуля можно использовать параметр UserDir, определяющий директорию в домашней директории пользователя, в которой WEB сервер будет искать html файлы при запросе к WEB серверу пользователя.

Например, в системе существует пользователь any с домашней директорией /home/any. В его домашней директории создана директория (обычно public_html), в которой находятся html файлы. В конфигурационном файле WEB сервера

определен параметр:

```
UserDir public_html
```

Тогда для доступа к этим html файлам необходимо использовать следующий URL:

```
http://host/~any
```

Параметры (продолжение)

Параметр	Описание
DirectoryIndex	Определяет имя (имена) файла, который будет выводиться при обращении к серверу без указания конкретного файла. Можно определить несколько файлов, разделив их пробелами.
AccessFileName	Определяет имя файла, в котором пользователи могут переопределять параметры WEB сервера.
ErrorLog	Определяет файл, в который будет помещаться отладочная информация.
LogLevel	Определяет количество отладочной информа-ции, помещаемой в файл, определенный при по- мощи параметра ErrorLog.
LogFormat	Определяет формат строки и имя формата.
CustomLog	Определяет имя файла, куда будет помещаться информация о доступе к WEB серверу. А так же имя формата строки.
Alias	Определяет псевдоним.
ScriptAlias	Определяет псевдоним директории. Но в от-личии от Alias сервер не выдает содержимое файла. Он выполняет программу и выдает пользователю то, что программа выводит на стандартный вывод.

Сообщения об ошибках

При возникновении ошибок WEB сервер Apache выдает страницы с указанием ошибок. Вы можете заменить стандартные страницы на свои собственные. Для этого необходимо использовать параметр ErrorDocument. В качестве опций следует указывать номер ошибки, а также:

- Текст сообщения
- Html файл
- GCI сценарий
- URL

Например:

```
ErrorDocument 500 "The server made a boo boo." ErrorDocument 404 /missing.html
ErrorDocument 404 "/cgi-bin/missing_handler.pl" ErrorDocument 402 http://www.example.com/
subs_info.html
```

Виртуальный хостинг

В Apache можно использовать два наиболее распространенных метода виртуального хостинга WEB сайтов:

- Базирующийся на IP адресах — когда каждый WEB сайт ассоциируется с отдельным IP адресом. Этот метод не является лучшим вариантом, так как потребляет большое количество IP адресов.
- Базирующийся на именах. В протоколе HTTP 1.1 при запросе клиента к WEB серверу предусмотрено указание имени сервера. По этому имени Apache может однозначно определить WEB сервер. Метод является наиболее предпочтительным, т.к. требуется только один IP адрес.

При использовании виртуального хостинга, базирующегося на именах, обязательно должен быть определен параметр NameVirtualHost. На каждый виртуальных хост должен быть определен собственный контейнер VirtualHost.

Например, необходимо обеспечить виртуальный хостинг сайта www.any.com на нашем WEB сервере.

Первое, что необходимо сделать — это в DNS сервере, отвечаю- щем за зону any.com, определить запись A, связывающую имя машины www с IP адресом Вашего WEB сервера. На машине, где находиться WEB сервер, добавить пользователя, например, any. В его домашней директории (/home/any) создать директорию public_html. В конфигурационном файле httpd.conf добавить следующие строки:

```
NameVirtualHost *:80
<VirtualHost *:80>
ServerAdmin webmaster@any.com
DocumentRoot /home/any/public_html
ServerName www.any.com
ErrorLog /home/any/err/error_log
CustomLog /home/any/err/access_log common
</VirtualHost>
```

Параметр NameVirtualHost определяется один раз. Количество блоков VirtualHost должно соответствовать количеству виртуальных серверов. Администратор сервера www.any.com получает доступ к своему сайту по ftp. Для каждого виртуального хоста рекомендуется делать отдельные журнальные файлы. В примере это файлы error_log и access_log.

Внутри блока VirtualHost можно определять любые параметры сервера Apache.

При включении механизма виртуального хостинга обязательно опишите основной сайт, который до этого обслуживался сервером Apache, в контейнере VirtualHost. Если этого не сделать, сайт не
будет доступен.

Лабораторная работа

Настройка виртуального хостинга.

Цель работы.
Научиться настраивать виртуальный хостинг WEB сайтов в сервере Apache.

Задача.

Настройте виртуальный хостинг сайта www.any.com.

Задачи	***Описание***
1. Добавление пользователя.	1. Добавьте пользователя any с паролем any. **`useradd -m -g nogroup -s /bin/false any`** **`passwd any`** 2. Создайте директорию public_html: **`mkdir /home/any/public_html`** **`chmod 755 /home/any/public_html`** 3. Создайте директорию, в которой сервер будет размещать логи: **`mkdir /home/any/err`** **`chmod 777 /home/any/err`** 4. В директории public_html создайте файл index.html со следующим содержанием: **`<HTML>`** **`<HEAD>`** **`<TITLE>TEST FILE</TITLE>`** **`</HEAD>`** **`<BODY>test</BODY>`** **`</HTML>`**
2. Настройка вир- туального хостинга.	1. Откройте на редактирование конфигурационный файл WEB сервера Apache: **`mcedit /etc/apache2/apache2.conf`** 3. Добавьте описание своего сервера и виртуального хоста: **`<VirtualHost *:80>`** **`ServerAdmin root@localhost`** **`DocumentRoot /var/www`** **`ServerName c1.u1.unix.specilaist.ru`** **`ErrorLog /var/log/apache/error_log`** **`CustomLog /var/log/apache/access_log common`** **`</VirtualHost>`** **`<VirtualHost *:80>`** **`ServerAdmin any@any.com`** **`DocumentRoot /home/any/public_html`** **`ServerName www.any.com`** **`ErrorLog /home/any/err/error_log`** **`CustomLog /home/any/err/access_log common`** **`</VirtualHost>`** 5. Сохраните файл. 6. Проверьте конфигурационный файл на наличие синтаксических ошибок: **`apachectl configtest`**
3. Запуск сервера.	1. Сделайте исполняемым стартовый скрипт сервера: **`chmod a+x /etc/init.d/apache2`** 2. Запустите сервер: **`/etc/init.d/apache2 start`** 3. В файле /etc/hosts в строке с IP адресом Вашего компьютера добавьте имя www.any.com ***При настройке реального хостинга этот шаг выполнять не надо!***
4. Проверка работоспособности.	1. Откройте браузер и введите FQDN имя вашей машины. Например: ***http://c1.u1.unix.specialist.ru*** В результате вы получите содержимое старого сайта (документация по Apache). 2. В браузере введите URL: www.any.com. В результате Вы получите содержимое тестовой странички WEB сервера www.any.com.

Вопросы.

1. Какие версии WEB сервера Apache Вам известны?

2. Как называется и где располагается конфигурационный файл WEB сервера Apache?

3. Какой параметр конфигурационного файла WEB сервера Apache определяет, где будут располагаться html файлы, доступные серверу?

4. Какой параметр следует обязательно определить при организации виртуального хостинга, базирующегося на именах WEB сайтов?

5. Какой контейнер используется в конфигурационном файле WEB сервера Apache для описания виртуальных WEB серверов?