|  |  |
| --- | --- |
|  |  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования«Московский технологический университет» МИРЭА |

Институт Информационных Технологий

Кафедра Корпоративных Информационных Систем

**ОТЧЕТ**

по Лабораторной Работе №8

на тему

«Сети доставки контента (Content Delivery Networks)»

по дисциплине

«Разработка клиент-серверных приложений»

Выполнил студент группы ИКБО-08-18 Валяев Д.A.

Принял старший преподаватель Мирзоян Д.И.

Выполнено «04» января 2021 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_ 2020 г.

Москва, 2020

**Задание**

Установить, настроить и проверить работу веб-сервера nginx в кэширующем режиме

**Теоретическая часть**

Официальный выпуск NGINX состоялся в октябре 2004 года. Создатель программного обеспечения, Игорь Сысоев, начал свой проект в 2002 году, пытаясь ответить на проблему C10k. C10k — это задача одновременного управления десятью тысячами соединений. Сегодня существует ещё больше подключений, которые должны обрабатывать веб-серверы. По этой причине NGINX предлагает управляемую событиями и асинхронную архитектуру. Эта функция делает NGINX одним из самых надёжных ПО для серверов по скорости и масштабируемости.

Благодаря превосходной способности обрабатывать большое количество соединений и скорости, многие сайты с высоким трафиком используют сервис NGINX. Некоторые из этих онлайн-гигантов — Google, Netflix, Adobe, Cloudflare, WordPress.com и многие другие.

Среди популярных веб-серверов Apache является одним из главных конкурентов NGINX. Он существует с 90-х годов и имеет большое сообщество пользователей. Если вам интересно, какой веб-сервер лучше всего подходит для ваших нужд, взгляните на это краткое и информативное сравнение между NGINX и Apache.

Совместимость — это одна из маленьких деталей, которую следует учитывать при выборе программного обеспечения. И NGINX, и Apache могут работать во многих операционных системах, поддерживающих систему Unix. К сожалению, производительность NGINX в Windows не так велика, как на других платформах.

NGINX может одновременно выполнять 1000 подключений статического контента в два раза быстрее, чем Apache, и использует немного меньше памяти. Однако при сравнении их производительности при работе с динамическим контентом обе имеют одинаковую скорость. NGINX это лучший выбор для тех, у кого более статичный сайт.

В отличие от обычного веб-сервера, Nginx не создаёт один поток под каждый запрос, а разделяет его на меньшие однотипные структуры, называемые рабочими соединениями. Каждое такое соединение обрабатывается отдельным рабочим процессом, а после выполнения они сливаются в единый блок, возвращающий результат в основной процесс обработки данных. Одно рабочее соединение может обрабатывать до 1024 запросов одного вида одновременно.

Практическое применение

1. Отдельный порт/IP. При наличии большого количества статичного контента или файлов для загрузки, можно настроить на отдельном порту или IP, чтобы осуществлять раздачу. При большом количестве запросов рекомендуется ставить отдельный сервер и подключать к нему Nginx.
2. Акселерированное проксирование. В таком случае все пользовательские запросы на статичный контент (картинки, простой HTML, JavaScript, CSS-файлы) поступают сначала на Nginx, а он их обрабатывает самостоятельно. При этом никаких изменений исходного кода не требуется.
3. Nginx и FastCGI. Если поддерживается технология FastCGI, Apache вообще можно не использовать. Но в таком случае может потребоваться модификация кодов скриптов.

Поскольку широкий функционал Nginx требует и значительно больших ресурсов системы, постоянно применять полноценную связку «Nginx + Apache» нецелесообразно. Чаще оба веб-сервера используются в симбиозе — Nginx отдает статику и перенаправляет обработку скриптов Apache.

Apache – это открытое программное обеспечение для создания веб-сервера (HTTP-сервера). Его главная функция – быстрая и надежная доставка контента в сети Интернет. Веб-сервер принимает запросы от клиентов через веб-браузер по протоколу HTTP/HTTPS. В ответ Apache отправляет браузеру искомый контент (документы, изображения, видео) в виде статических HTML-страниц.

Apache HTTP Server был выпущен в 1995 году разработчиком Робертом Маккулом из Университета штата Иллинойс (UIUC). Продукт возник как доработанная версия другого HTTP-клиента – NCSA HTTPd 1.3, созданного Робертом ранее.

Основой для модификации стали многочисленные «патчи» или программные «заплатки» для NCSA. Именно отсюда (а не от индейского племени апачей) изначально и происходит название Apache. Оно расшифровывается как «a patchy server» или «сервер с патчами».

Разработкой и поддержкой продукта с 1999 года занимается организация Apache Software Foundation (ASF) – сообщество экспертов-энтузиастов со всего мира. Этим же некоммерческим фондом была создана официальная лицензия ПО – Apache License.

В 2000 году ASF представило новую версию Apache 2.0 с полностью переработанной архитектурой, свободной от кода NCSA. С этого момента веб-сервер развивается по двум основным веткам – 1.х и 2.х.

Apache состоит из ядра и динамической модульной системы. Параметры системы изменяются с помощью конфигурационных файлов. Для запуска на сервере нескольких веб-проектов одновременно используется механизм виртуальных хостов.

Ядро Apache разработано Apache Software Foundation на языке C. Основные функции — обработка конфигурационных файлов, протокол HTTP/HTTPS и загрузка модулей. Ядро может работать без модулей, но будет иметь ограниченный функционал.

Модуль – отдельный файл, подключение которого расширяет изначальный функционал ядра. Они могут включаться в состав ПО при первоначальной установке или подгружаться позже через изменение конфигурационного файла.

Большинство из них отвечает за определенный аспект обработки клиентского запроса – поддержку различных языков программирования, безопасность, кэширование, аутентификацию и т.д. Таким образом, большая задача разбивается на несколько фаз, каждую из которых решает отдельный, узкоспециализированный модуль.

Для Apache существует больше 500 модулей. Многие популярные веб-приложения сразу выпускаются в виде модуля к Apache. Например, ISPmanager и VDSmanager.

**Выполнение работы**

Для работы по внутренней сети нам необходимо включить и настроить 3й сетевой адаптер в настройках виртуальной машины. Предполагается, что 1й сетевой адаптер используется для доступа виртуальной машины в интернет, а 2й для подключения к виртуальной машине с помощью SSH для облегчения процесса настройки.



Рис1. – Настройка сетевых адаптеров

В данной внутренней сети ВМ с Apache будет иметь IP 192.168.99.10, а ВМ с nginx — 192.168.99.11

В сети хоста, соответственно, IP адреса будем считать следующими: ВМ с Apache 192.168.56.100, ВМ с nginx 192.168.56.120.

Теперь нам необходимо установить веб-сервер nginx. Выполнив команду

yum install nginx

Можно убедиться, что этого не произошло.



Рис2. – Установка nginx

В современных дистрибутивах Linux присутствует большое количество пакетов, некоторые из которых отличаются по лицензиям, поддержке и пр. Такого рода пакеты часто выносятся в отдельные репозитории. Нужно подключить нужный репозиторий:

yum install epel-release

после чего установить nginx

yum install nginx



Рис 3. – Установка nginx

Теперь остается только настроить веб-сервер

vi /etc/nginx/nginx.conf

# For more information on configuration, see:

# \* Official English Documentation: http://nginx.org/en/docs/

# \* Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;

worker\_processes auto;

error\_log /var/log/nginx/error.log;

pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.

# include /usr/share/nginx/modules/\*.conf;

events {

worker\_connections 1024;

multi\_accept on;

}

http {

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

types\_hash\_max\_size 2048;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

access\_log off;

error\_log /var/log/nginx/error.log;

gzip on;

gzip\_disable "msie6";

gzip\_comp\_level 6;

gzip\_proxied any;

gzip\_vary on;

gzip\_types text/plain application/javascript text/javascript text/css application/json application/xml text/xml application/rss+xml;

gunzip on;

proxy\_temp\_path

/var/cache/tmp;

proxy\_cache\_path

/var/cache/cdn

levels=1:2

keys\_zone=cdn:64m

max\_size=20g

inactive=7d;

proxy\_cache\_bypass $http\_x\_update;

server {

listen 80;

server\_name cdn.localhost;

location / {

proxy\_cache cdn;

proxy\_cache\_key $uri$is\_args$args;

proxy\_cache\_valid 90d;

proxy\_pass http://192.168.99.10;

}

}

}

Запускаем веб-сервер командой

systemctl start nginx

и убеждаемся, что он работает, открыв адрес 192.168.56.120 в браузере.

**Заключение**

Данная лабораторная работа позволила узнать подробную информацию про Сети доставки контента (Content Delivery Networks). А также удалось узнать много другой полезной информации и поднять свой уровень в сфере информационных технологий.