

Оглавление

Предисловие от издательства	10
Предисловие.....	11
Глава 1. Основы	12
1.0. Введение.....	12
1.1. Установка на компьютер с Debian/Ubuntu	12
1.2. Установка на компьютер с RedHat/CentOS	13
1.3. Установка NGINX Plus.....	14
1.4. Проверка установки.....	15
1.5. Ключевые файлы, команды и каталоги.....	16
1.6. Обслуживание статического контента	18
1.7. Аккуратная перезагрузка	19
Глава 2. Высокопроизводительная балансировка нагрузки.....	20
2.0. Введение.....	20
2.1. Балансировка нагрузки для HTTP.....	21
2.2. Балансировка нагрузки для TCP	23
2.3. Балансировка нагрузки UDP.....	24
2.4. Методы балансировки нагрузки.....	26
2.5. Директива sticky cookie	28
2.6. Директива sticky learn.....	29
2.7. Директива sticky route	30
2.8. Осушение соединения	32
2.9. Пассивные проверки работоспособности	33
2.10. Активные проверки работоспособности	34
2.11. Медленный запуск.....	36
2.12. Проверки работоспособности TCP	37
Глава 3. Управление трафиком	39
3.0. Введение.....	39
3.1. A/B-тестирование	39
3.2. Использование модуля GeoIP и базы данных.....	40
3.3. Ограничение доступа в зависимости от страны	43

3.4. Поиск исходного клиента.....	44
3.5. Ограничение подключений.....	45
3.6. Ограничение скорости.....	46
3.7. Ограничение пропускной способности.....	48
Глава 4. Массивно масштабируемое кеширование контента.....	50
4.0. Введение.....	50
4.1. Кеширование зон.....	50
4.2. Хеш-ключи кеширования.....	52
4.3. Обход кеширования.....	53
4.4. Производительность кеширования.....	54
4.5. Продувка.....	55
4.6. Директива slice.....	56
Глава 5. Программируемость и автоматизация.....	58
5.0. Введение.....	58
5.1. API NGINX Plus.....	59
5.2. Хранилище типа ключ/значение.....	63
5.3. Установка с использованием приложения Puppet.....	65
5.4. Установка с использованием системы Chef.....	67
5.5. Установка с использованием системы Ansible.....	69
5.6. Установка с использованием SaltStack.....	70
5.7. Автоматизация конфигураций с помощью Consul.....	72
Глава 6. Аутентификация.....	75
6.0. Введение.....	75
6.1. Базовая HTTP-аутентификация.....	75
6.2. Подзапросы аутентификации.....	77
6.3. Валидация токенов в формате JWT.....	78
6.4. Создание веб-ключей в формате JSON.....	79
6.5. Аутентификация пользователей с помощью существующего протокола единого входа OpenID Connect.....	81
6.6. Получение ключа в формате JSON от Google.....	82
Глава 7. Контроль безопасности.....	84
7.0. Введение.....	84
7.1. Доступ на основе IP-адреса.....	84
7.2. Разрешение совместного использования ресурсов между разными источниками.....	85

7.3. Шифрование на стороне клиента.....	87
7.4. Восходящее шифрование	89
7.5. Безопасность местоположения.....	90
7.6. Генерация безопасного соединения при помощи ключа безопасности.....	91
7.7. Безопасность местоположения при помощи ограниченной даты.....	92
7.8. Генерация ссылки с ограниченным сроком.....	94
7.10. Перенаправление на HTTPS, когда SSL/TLS прекращается до NGINX.....	97
7.11. Строгая безопасность доставки HTTP	98
7.12. Удовлетворение любого числа методов безопасности.....	98
7.13. Динамичное ослабление DDoS.....	100
Глава 8. HTTP/2	102
8.0. Введение.....	102
8.1. Базовая настройка.....	102
8.2. gRPC.....	103
8.3. Сервер активной доставки HTTP/2	106
Глава 9. Управление сложными потоками медиа	107
9.0. Введение.....	107
9.1. Обслуживание MP4 и FLV	107
9.2. Организация потоков с помощью HLS.....	108
9.3. Организация потоков с помощью HDS	110
9.4. Пределы полосы пропускания.....	110
Глава 10. Развертывание в облачных решениях.....	112
10.0. Введение	112
10.1. Автоматическая настройка в AWS.....	112
10.2. Маршрутизация в узлы NGINX без ELB.....	115
10.3. NLB-сэндвич	116
10.4. Развертывание из AWS Marketplace	117
10.5. Создание образа виртуальной машины NGINX в Azure	119
10.6. Балансировка нагрузки поверх наборов масштабирования NGINX в Azure.....	122
10.7. Развертывание через Azure Marketplace	123
10.8. Развертывание в Google Compute Engine	124
10.9. Создание образа Google Compute.....	125
10.10. Создание прокси-сервера для Google App Engine	126

Глава 11. Контейнеры/Микросервисы	128
11.0. Введение	128
11.1. Записи DNS SRV	128
11.2. Использование официального образа NGINX.....	130
11.3. Создание Dockerfile NGINX.....	131
11.4. Сборка образа NGINX Plus	133
11.5. Использование переменных среды в NGINX.....	135
11.6. Контроллер Ingress в Kubernetes.....	137
11.7. Маршрутизатор OpenShift.....	140
Глава 12. Режимы развертывания высокой доступности	142
12.0. Введение	142
12.1. Режим высокой доступности NGINX.....	142
12.2. Балансировка нагрузки балансировщиками с помощью DNS	143
12.3. Балансировка нагрузки в EC2	144
12.4. Синхронизация конфигурации	145
12.5. Совместное использование состояния с помощью Zone Sync.....	148
Глава 13. Расширенный мониторинг активности	150
13.0. Введение	150
13.1. Активация модуля Stub Status с открытым исходным кодом.....	150
13.2. Активация инструментальной панели мониторинга NGINX Plus.....	151
13.3. Сбор метрик с помощью API NGINX Plus.....	153
Глава 14. Отладка и устранение неполадок с помощью журналов доступа, журналов ошибок и отслеживания запросов	157
14.0. Введение	157
14.1. Настройка журналов доступа	157
14.3. Отправка журналов в Syslog.....	159
14.4. Трассировка запросов	161
Глава 15. Настройка производительности.....	163
15.0. Введение	163
15.1. Автоматизация тестов с помощью драйверов нагрузки	163
15.2. Сохраняем подключения открытыми для клиентов	164
15.3. Сохраняем подключения открытыми для вышестоящих серверов.....	165
15.4. Буферизация ответов.....	166
15.5. Буферизация журналов доступа	167
15.6. Настройка ОС.....	168

Глава 16. Советы по практической эксплуатации и заключение	170
16.0. Введение	170
16.1. Использование директивы include для чистых настроек.....	170
16.2. Отладка конфигураций	171
16.3. Заключение.....	173
Сведения об авторе	174
Предметный указатель	175

Предисловие от издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге, – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и O'Reilly очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую предоставлять вам качественные материалы.

Предисловие

Добро пожаловать в обновленное издание «NGINX. Книга рецептов». Прошло почти два года, с тех пор как издательство O'Reilly опубликовало первую версию книги. С того времени многое изменилось, кроме одного: каждый день все больше и больше сайтов по всему миру отдают предпочтение NGINX. Сегодня их 300 млн, почти вдвое больше, когда вышло первое издание.

Есть много причин, по которым использование NGINX набирает популярность спустя 14 лет после его появления. Это швейцарский армейский нож: NGINX может быть веб-сервером, балансировщиком нагрузки, кешировать контент и быть API-шлюзом. Но, возможно, еще более важным фактом является то, что он надежен.

Эта книга покажет вам, как получить максимальную отдачу от NGINX с открытым исходным кодом и NGINX Plus. Вы найдете свыше 150 страниц, содержащих простые в использовании рецепты, охватывающие все от того, как правильно установить NGINX, настроить все основные функции до отладки и устранения неполадок.

Эта обновленная версия также включает в себя описание новых функций с открытым исходным кодом, таких как поддержка gRPC, сервер активной доставки HTTP/2 и алгоритм балансировки нагрузки Random with Two Choices для кластерных сред, а также новые функции NGINX Plus, такие как поддержка совместного использования состояния, новый API NGINX Plus и хранилище типа ключ/значение. На этих страницах рассказывается почти все, что вам нужно знать о NGINX.

Мы надеемся, что вам понравится эта книга и она будет способствовать вашему успеху в создании и развертывании приложений, которые все мы используем.

*Файсаль Меном,
менеджер по маркетингу продукции, NGINX Inc.*

Глава 1

ОСНОВЫ

1.0. Введение

Чтобы начать работу с NGINX с открытым исходным кодом или NGINX Plus, для начала необходимо установить их в системе и познакомиться с основами. В этой главе вы узнаете, как установить NGINX, где находятся основные файлы конфигурации и команды администрирования. Вы также узнаете, как проверить свою установку и выполнить запросы к серверу, установленному по умолчанию.

1.1. Установка на компьютер с Debian/Ubuntu

Задача

Вам необходимо установить NGINX с открытым исходным кодом на машине с Debian или Ubuntu.

Решение

Создайте файл с именем `/etc/apt/sources.list.d/nginx.list` следующего содержания:

```
deb http://nginx.org/packages/mainline/OS/ CODENAME nginx
deb-src http://nginx.org/packages/mainline/OS/ CODENAME nginx
```

Измените файл, заменив буквосочетание `os` в конце URL-адреса на `ubuntu` ИЛИ `debian` в зависимости от дистрибутива. Замените слово `CODENAME` на кодовое имя вашего дистрибутива: `jessie` или `stretch`, если это Debian, либо `trusty`, `xenial`, `artful` или `bionic`, если это Ubuntu. Затем выполните приведенные ниже команды:

```
wget http://nginx.org/keys/nginx_signing.key
apt-key add nginx_signing.key
```



```
apt-get update
apt-get install -y nginx
/etc/init.d/nginx start
```

Обсуждение

Файл, который вы только что создали, инструктирует систему управления пакетами `apt` использовать Официальный репозиторий пакетов NGINX. С помощью следующих далее команд вы скачиваете ключ подписи пакета NGINX GPG и импортируете его в `apt`. Предоставление `apt` ключа подписи активирует систему, чтобы выполнить проверку пакетов из репозитория. Команда `apt-get update` велит системе `apt` обновить свои листинги пакетов из известных репозиториях. После обновления списка пакетов вы можете установить NGINX Open Source из официального репозитория NGINX. После его установки последняя команда запускает NGINX.

1.2. Установка на компьютер с RedHat/CentOS

Задача

Вам необходимо установить NGINX с открытым исходным кодом на компьютер с RedHat или CentOS.

Решение

Создайте файл с именем `/etc/yum.repos.d/nginx.repo` следующего содержания:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/mainline/OS/OSRELEASE/$basearch/
gpgcheck=0
enabled=1
```

Измените файл, заменив буквосочетание `os` в конце URL-адреса на `rhel` или `centos` в зависимости от дистрибутива. Замените слово `OSRELEASE` на цифру 6 или 7 для версии 6.x или 7.x соответственно. Затем выполните приведенные ниже команды:

```
yum -y install nginx
systemctl enable nginx
```

```
systemctl start nginx  
firewall-cmd --permanent --zone=public --add-port=80/tcp  
firewall-cmd --reload
```

Обсуждение

Файл, который вы только что создали, инструктирует систему управления пакетами yum использовать Официальный репозиторий пакетов NGINX. С помощью следующих далее команд вы устанавливаете NGINX Open Source из Официального репозитория, сообщаете systemd активировать NGINX во время загрузки и велите ей запустить его сейчас. Команды брандмауэра открывают порт 80 для протокола TCP, который является значением порта по умолчанию для HTTP. С помощью последней команды вы выполняете перезагрузку брандмауэра для фиксации изменений.

1.3. Установка NGINX Plus

Задача

Вам необходимо установить NGINX Plus.

Решение

Посетите страницу по адресу http://cs.nginx.com/repo_setup. В раскрывающемся меню выберите устанавливаемую вами ОС и следуйте инструкциям. Эти инструкции аналогичны инструкциям по установке решений с открытым исходным кодом; однако вам необходимо установить сертификат для аутентификации в репозитории NGINX Plus.

Обсуждение

NGINX постоянно обновляет данное руководство по установке репозитория с инструкциями по установке NGINX Plus. В зависимости от вашей операционной системы и ее версии эти инструкции несколько отличаются, но есть кое-что общее. Вы должны войти на портал NGINX, чтобы скачать сертификат и ключ к вашей системе, которые используются для прохождения аутентификации в репозитории NGINX Plus.

1.4. Проверка установки

Задача

Вам нужно оценить правильность установки NGINX и проверить версию.

Решение

Вы можете убедиться, что NGINX установлен правильно и проверить его версию, используя приведенную ниже команду:

```
$ nginx -v
nginx version: nginx/1.15.3
```

Как показывает этот пример, в ответе отображается версия.

Можно подтвердить, что NGINX работает с помощью приведенной ниже команды:

```
$ ps -ef | grep nginx
root      1738    1 0 19:54 ? 00:00:00 nginx: master process
nginx     1739 1738 0 19:54 ? 00:00:00 nginx: worker process
```

Команда `ps` выводит список запущенных процессов. Отправив его в `grep`, можно искать конкретные слова в выводе. В этом примере мы используем `grep` для поиска `nginx`. Результат показывает два запущенных процесса, `master` и `worker`. Если NGINX запущен, вы всегда будете видеть главный процесс. Чтобы увидеть инструкции по запуску NGINX, см. следующий раздел. Чтобы понять, как запустить NGINX в качестве демона, примените методологии `init.d` или `systemd`.

Чтобы убедиться, что NGINX возвращает запросы правильно, используйте свой браузер и отправьте запрос на ваш компьютер или воспользуйтесь командой `curl`:

```
$ curl localhost
```

Вы увидите страницу приветствия NGINX.

Обсуждение

Команда `nginx` позволяет взаимодействовать с двоичным файлом NGINX, чтобы проверить версию, вывести список установленных модулей, протестировать конфигурации и отправить сигналы в главный процесс. Чтобы обслуживать запросы, NGINX должен быть запущен. Команда

ps – верный способ определить, работает ли NGINX в качестве демона или в приоритетном режиме. В исходной конфигурации, предоставляемой по умолчанию в NGINX, запускается HTTP-сервер статического сайта на порту 80. Этот сайт можно протестировать, отправив HTTP-запрос на компьютер на локальном хосте, а также на IP-адрес хоста и имя хоста.

1.5. Ключевые файлы, команды и каталоги

Задача

Вам нужно разобраться в важных каталогах и командах NGINX.

Решение

Файлы и каталоги NGINX

/etc/nginx/

Каталог */etc/nginx/* является корневым каталогом конфигурации по умолчанию сервера NGINX. В этом каталоге вы найдете файлы конфигурации, которые инструктируют NGINX, как вести себя.

/etc/nginx/nginx.conf

Файл */etc/nginx/nginx.conf* является точкой входа конфигурации по умолчанию, используемой службой NGINX. Этот файл конфигурации устанавливает глобальные параметры для таких вещей, как рабочий процесс, настройка, ведение журнала, загрузка динамических модулей и ссылки на другие файлы конфигурации NGINX. В исходной конфигурации файл */etc/nginx/nginx.conf* включает в себя блок `http` верхнего уровня, который содержит все файлы конфигурации в каталоге, описанном далее.

/etc/nginx/conf.d/

Каталог */etc/nginx/conf.d/* содержит файл конфигурации HTTP-сервера по умолчанию. Файлы в этом каталоге, оканчивающиеся на *.conf*, включены в блок `http` верхнего уровня из файла */etc/nginx/nginx.conf*. Рекомендуется использовать операторы `include` и упорядочивать свою конфигурацию таким образом, чтобы ваши файлы конфигурации были краткими. В некоторых репозиториях пакетов эта папка называется *site-enabled*, а файлы конфигурации связываются из папки с именем *site-available*; данная конвенция является устаревшей.

`/var/log/nginx/`

Каталог `/var/log/nginx/` является местоположением журнала по умолчанию для NGINX. В этом каталоге вы найдете файлы `access.log` и `error.log`. Файл `access.log` содержит запись каждого запроса, который обслуживает NGINX. В файле `error.log` содержатся события с ошибками и отладочная информация, если модуль отладки включен.

Команды NGINX

`nginx -h`

Показывает справочное меню NGINX.

`nginx -v`

Показывает версию NGINX.

`nginx -V`

Показывает версию NGINX, информацию о сборке и аргументы конфигурации, где даны модули, встроенные в двоичный файл NGINX.

`nginx -t`

Проверяет конфигурацию NGINX.

`nginx -T`

Проверяет конфигурацию NGINX и выводит ее проверенной на экран. Эта команда полезна при поиске поддержки.

`nginx -s signal`

Флаг `-s` отправляет сигнал главному процессу NGINX. Вы можете отправлять такие сигналы, как `stop`, `quit`, `reload` и `reopen`. `stop` немедленно прекращает процесс NGINX, `quit` останавливает процесс NGINX после завершения обработки запросов в полете, `reload` перезагружает конфигурацию, `reopen` дает NGINX команду повторно открывать файлы журнала.

Обсуждение

Освоив эти ключевые файлы, каталоги и команды, вы можете приступить к работе с NGINX. Обладая данными знаниями, вы можете изменять файлы конфигурации по умолчанию и тестировать свои изменения с помощью команды `nginx -t`. Если ваш тест пройдет успешно, вы также будете знать, как проинструктировать NGINX для перезагрузки конфигурации с помощью команды `nginx -s reload`.

1.6. Обслуживание статического контента

Задача

Вам необходимо обслуживать статический контент с помощью NGINX.

Решение

Перепишите конфигурацию HTTP-сервера по умолчанию, расположенную в `/etc/nginx/conf.d/default.conf`, используя приведенный ниже пример конфигурации NGINX:

```
server {
    listen 80 default_server;
    server_name www.example.com;

    location / {
        root /usr/share/nginx/html;
        # alias /usr/share/nginx/html;
        index index.html index.htm;
    }
}
```

Обсуждение

Данная конфигурация обслуживает статические файлы по протоколу HTTP через порт 80 из каталога `/usr/share/nginx/html/`. Первая строка в этой конфигурации определяет новый блок `server`. Это определяет новый контекст, который NGINX будет слушать. Вторая строка дает NGINX команду слушать порт 80, а параметр `default_server` указывает NGINX использовать этот сервер в качестве контекста по умолчанию для порта 80. Директива `server_name` определяет имя хоста или имена запросов, которые должны быть направлены на этот сервер. Если конфигурация не определила этот контекст как `default_server`, NGINX будет направлять запросы на этот сервер, только если заголовок HTTP-узла соответствует значению, указанному в директиве `server_name`.

Блок `location` определяет конфигурацию на основе пути в URL-адресе. Путь или часть URL-адреса после домена называется URI. NGINX лучше всего будет соответствовать запрошенному URI для блока `location`. В этом примере используется символ `/` для сопоставления всех запросов. Директива `root` показывает NGINX, где искать статические файлы при

обслуживании контента для данного контекста. URI запроса добавляется к значению директивы `root` при поиске запрошенного файла. Если бы мы указали префикс URI для директивы `location`, он был бы включен в добавленный путь, если только мы не использовали каталог `alias` вместо `root`. И наконец, директива `index` предоставляет NGINX файл по умолчанию или список файлов для проверки в случае, если в URI не указан дальнейший путь.

1.7. Аккуратная перезагрузка

Задача

Вам необходимо перезагрузить конфигурацию, не отбрасывая пакеты.

Решение

Используйте метод `reload` для аккуратной перезагрузки конфигурации без остановки сервера:

```
$ nginx -s reload
```

В этом примере выполняется перезагрузка системы NGINX с использованием двоичного файла NGINX для отправки сигнала главному процессу.

Обсуждение

Перезагрузка конфигурации NGINX без остановки сервера предоставляет возможность изменять конфигурации на лету, не отбрасывая никаких пакетов. В динамичной среде с высокой продолжительностью работы в какой-то момент вам потребуется изменить конфигурацию балансировки нагрузки. NGINX позволяет делать это, сохраняя балансировщик нагрузки в сети. Эта функция предоставляет бесчисленные возможности, такие как повторный запуск управления конфигурацией в реальной среде или создание модуля с поддержкой приложений и кластеров для динамической настройки и перезагрузки NGINX, чтобы удовлетворять потребностям среды.