Оглавление

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 репозиториев. 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.

Глава 2

Высокопроизводительная балансировка нагрузки

2.0. Введение

Сегодня пользователям интернета требуется производительность и безотказная работа. Для этого запускается несколько копий одной и той же системы, и нагрузка распределяется между ними. По мере увеличения нагрузки в рабочее состояние может вводиться еще одна копия такой системы. Этот метод называется *горизонтальным масштабированием*. Программная инфраструктура приобретает все большую популярность благодаря своей гибкости, открывая огромный мир возможностей. Независимо от того, является ли вариант использования небольшим (как набор из двух для обеспечения высокой доступности) или крупнее (из тысяч по всему миру), необходимо создать решение для балансировки нагрузки, которое будет таким же динамичным, как и инфраструктура. NGINX удовлетворяет эту потребность несколькими способами, между серверами HTTP, TCP и UDP, о чем мы расскажем в этой главе.

При балансировке нагрузки важно, чтобы влияние на клиента было только положительным. Многие современные веб-архитектуры используют уровни приложений без фиксации состояния, храня состояние в системе с общей памятью или базах данных. Однако такое происходит не у всех. Состояние сеанса чрезвычайно ценно и обширно в интерактивных приложениях. Оно может храниться локально на сервере приложений по ряду причин; например, в приложениях, для которых обрабатываемые данные настолько велики, что нагрузка на сеть слишком высока по производительности. Когда состояние хранится

локально на сервере приложений, пользователю крайне важно, чтобы последующие запросы продолжали доставляться на тот же сервер. Еще один аспект этой ситуации заключается в том, что серверы не должны быть освобождены до завершения сеанса. Для работы с масштабными приложениями, которые фиксируют состояние, требуется интеллектуальный балансировщик нагрузки. NGINX Plus предлагает несколько способов решения этой проблемы путем отслеживания куки-файлов или маршрутизации. В этой главе рассматривается постоянство сеансов, поскольку оно касается балансировки нагрузки с помощью NGINX и NGINX Plus.

Также важно обеспечить работоспособность приложения, которое обслуживает NGINX. По ряду причин приложения могут не работать. Это может быть вызвано подключением к сети, сбоем в работе сервера или приложения среди прочего. Прокси-серверы и балансировщики нагрузки должны быть достаточно умными, чтобы обнаруживать сбои в работе вышестоящих серверов и прекращать передачу трафика к ним; в противном случае клиент будет пребывать в состоянии ожидания, получая лишь сообщения о тайм-ауте. Способ смягчить ухудшение качества обслуживания в случае сбоя сервера состоит в том, чтобы прокси-сервер проверял работоспособность вышестоящих серверов. NGINX предлагает два типа проверки работоспособности: пассивный, доступный в версии с открытым исходным кодом, и активный, доступный только в NGINX Plus. Активные проверки работоспособности через регулярные промежутки времени установят соединение или выполнят запрос к вышестоящему серверу и смогут убедиться, что ответ верный. Пассивные проверки работоспособности контролируют соединение или ответы вышестоящего сервера, когда клиенты выполняют запрос или устанавливают соединение. Возможно, вы захотите использовать пассивные проверки, чтобы уменьшить нагрузку на свои вышестоящие серверы, и активные – чтобы определить отказ вышестоящего сервера, прежде чем клиент столкнется со сбоем. В конце этой главы рассматривается мониторинг работоспособности вышестоящих серверов приложений, для которых вы выполняете балансировку нагрузки.

2.1. Балансировка нагрузки для HTTP

Задача

Вам необходимо распределить нагрузку между двумя или более серверами HTTP.

Решение

Используйте HTTP-модуль NGINX для выполнения балансировки по HTTP-серверам с использованием блока `upstream`:

```
upstream backend {
    server 10.10.12.45:80 weight=1;
    server app.example.com:80 weight=2;
}
server {
    location / {
        proxy_pass http://backend;
    }
}
```

Эта конфигурация выполняет балансировку нагрузки по двум HTTP-серверам на порту 80. Параметр `weight` указывает NGINX передавать вдвое больше подключений на второй сервер, а значение этого параметра по умолчанию устанавливается на 1.

Обсуждение

Модуль `upstream` управляет балансировкой нагрузки для HTTP. Этот модуль определяет пул адресатов – любую комбинацию сокетов Unix, IP-адресов и записей DNS или их комбинацию. Модуль `upstream` также определяет, каким образом любой отдельный запрос назначается любому из вышестоящих серверов.

Каждый получатель в восходящем потоке определяется в восходящем пуле директивой `server`. В этой директиве указывается сокет Unix, IP-адрес или полное доменное имя, а также ряд необязательных параметров. Необязательные параметры дают больший контроль над маршрутизацией запросов. Эти параметры включают в себя вес сервера в алгоритме балансировки; находится ли сервер в режиме ожидания, доступен он или недоступен; и как определить, что сервер недоступен. NGINX Plus предоставляет ряд других удобных параметров, таких как ограничения подключения к серверу, расширенный контроль над DNS-преобразованием и возможность медленного увеличения количества подключений к серверу после его запуска.

2.2. Балансировка нагрузки для TCP

Задача

Вам необходимо распределить нагрузку между двумя или более серверами TCP.

Решение

Используйте модуль `stream` для балансировки нагрузки на TCP-серверы с использованием блока `upstream`:

```
stream {
    upstream mysql_read {
        server read1.example.com:3306 weight=5;
        server read2.example.com:3306;
        server 10.10.12.34:3306 backup;
    }

    server {
        listen 3306;
        proxy_pass mysql_read;
    }
}
```

Блок `server` в этом примере дает NGINX команду слушать TCP-порт 3306, выполняет балансировку нагрузки между двумя репликами на чтение базы данных MySQL и перечисляет еще одну в качестве резервной копии, в которой будет передаваться трафик, если указанные первичные будут остановлены. Эту конфигурацию не нужно добавлять в папку `conf.d`, поскольку она включена в блок `http`; вместо этого следует создать другую папку с именем `stream.conf.d`, открыть блок `stream` в файле `nginx.conf` и включить в состав новую папку для потоковых конфигураций.

Обсуждение

Балансировка нагрузки TCP определяется модулем `stream`. Модуль `stream`, как и модуль HTTP, позволяет определять пулы вышестоящих серверов и настраивать слушающий сервер. При настройке сервера для прослушивания данного порта вы должны определить порт для прослушивания или – по желанию – адрес и порт. Оттуда нужно настроить