

Введение

В этой инструкции расскажу как сделать непробиваемую защиту для ваших проектов и быстроразворачиваемую и восстанавливаемую экосистему для ваших проектов.

Идея родилась когда я занимался дорами. Изначально проблема была в том, что балансировка нагрузки стала для меня очень важна.

Понятнее будет на примере. Например, у нас есть 1000 доменов и 10 серверов (впс). Но суть дора в том, что он может выстрелить, а может и не выстрелить (что чаще и бывает, к сожалению).

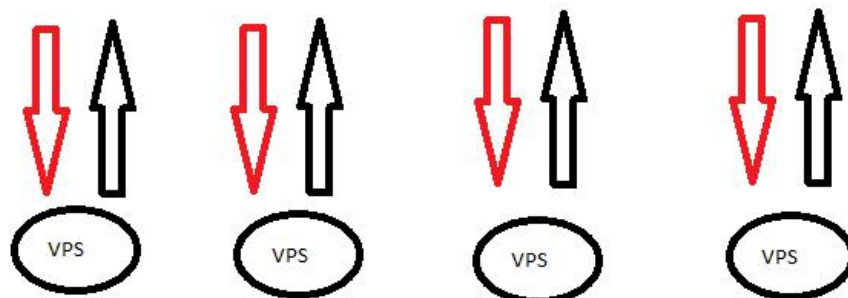
Пусть вышеназванные 1000 доменов у нас распределены по 100 штук на сервер. Типичной картиной станет то, что на одном сервере выстрелили 10, а на втором ни одного. Получается, что 1 сервер у нас трудится на пределе, тупит, а второй простаивает. В итоге мы теряем трафик из-за тупящего сервера и платим за то, чем не пользуемся. Казалось бы, все просто: перенеси выстрелившие на другие сервера и будет тебе счастье. Но! Вы хотите ежедневно мониторить и таскать домены?

Используя балансировку нагрузки мы сможем существенно снизить расходы на хостинг и получить более ровную и плавную работу наших проектов практически независимо от нагрузки.

Часть 1: теория.

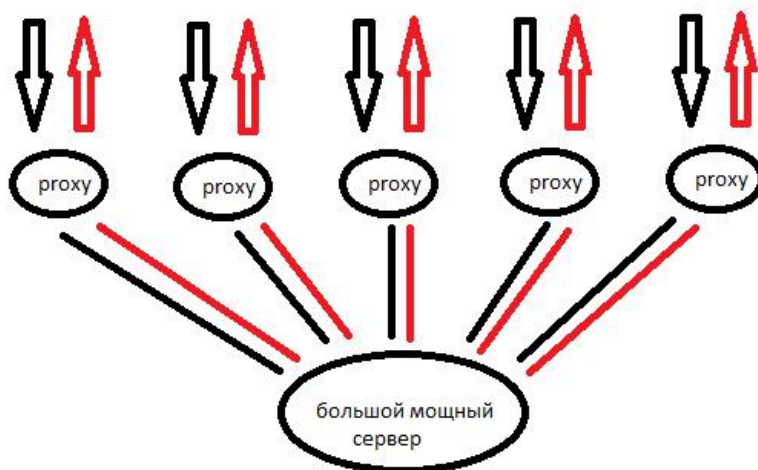
Несколько схем:

Классический подход



Тут все просто: приходит траф, обрабатывается на сервере и клиенту возвращается результат. Клиент видит ip сервера, на котором происходит непосредственная обработка запросов - взаимодействие прямое. В итоге мы имеем для обработки мощность конкретно взятого сервера.

А теперь схема с проксированием запросов:



На схеме прокси - это прокси впс, не http, socks или другой прокси. Это совершенно другое, не путайте. Из официальной документации nginx:

Одним из частых применений nginx является использование его в качестве прокси-сервера, то есть сервера, который принимает запросы, перенаправляет их на проксируемые сервера, получает ответы от них и отправляет их клиенту.

Итак. На наш прокси-впс (далее прокси) приходит какой-то запрос, но он не обрабатывается непосредственно на прокси, а передается выше, на наш большой сервер, где обрабатывается и результат возвращается на прокси, который возвращает результат клиенту. В этой схеме клиент видит и обращается к ип прокси, а большой сервер (далее БС) для него не видим. И как результат, мы имеем для обработки данных ресурсы уже не слабенькой впс, а всего БС. И даже если на одном прокси выстреливает десяток дорвов, вся нагрузка ложится на БС, которому это по сути семечки. В то же время, домены без трафа лежат и не едят ресурсы.

Ну и финансовая сторона вопроса.

Рассмотрим мою ситуацию. У меня было 6 небольших впс под мои дорвеи. Примерный конфиг каждой был такой: 1 гб оперативки, 1-2 неизвестных ядер частотой 2,7 ГГц и 20 гб ссд диска. Тут отдельно стоит упомянуть наших добросовестных хостеров с их оверселлом. Сменил много хостеров, и, что удивительно, у всех на примерно одинаковых конфигах наблюдалась поразительно большая разница в реальной производительности: один без проблем вывозил по 200 доменов со средним трафом, у другого 100 условно таких же доменов наглухо клали впс. Вернемся к арифметике. 6 таких серверов стоили около 6 долларов в месяц, итого 36 долларов в месяц. В данный момент мой БС это Intel core i5, 16 GB RAM, 2 TB hdd. Аренда БС мне обходится в 24 доллара в месяц, плюс каждый прокси по 1 доллару, итого 30 долларов. Разница не так велика, НО по факту, за 30 долларов в месяц я имею 6 серверов производительностью сравнимой с мощностью БС.

Часть 2: подготовка

Для аренды БС рекомендую услуги ДЦ <https://www.kimsufi.com> (сам у них арендую). Если подходящего конфига не нашлось, можно воспользоваться услугами <https://bill.ovh/> (русские ребята, имеют сетку проектов по популяризации услуг OVH), но сам я у них брал только впс на тест, по поводу серверов думаю проблем не будет, очень приятные в общении парни (у них еще сайт интересный есть <https://ruovh.ru/>, там они часто выкладывают освободившиеся сервера + инструкции, рекомендую почитать). Мой БС находится в дата-центре Kimsufi во Франции, но можете выбрать любое местоположение. Однако учитывайте, что чем ближе БС и прокси - тем лучше: банально данные шустрее проходить будут.










Для прокси подойдут абсолютно все впс: нам не важен тип виртуализации, железо и т.д. Лишь бы было 64 мб оперативки и хоть какой то диск.

Регистрация аккаунтов

Тут я приведу цитату с <https://ruovh.ru/blog/kimsufi/48.html>, за что им огромное спасибо.

1. Идем на сайт www.kimsufi.com/ru/#servers

2. Выбираем доступный сервер

Цена /мес.			Количество	
29.99 € без НДС	-	-	-	-
24.99 € без НДС	-	-	-	-
24.99 € без НДС	-		1 <input type="text"/>	
19.99 € без НДС			1 <input type="text"/>	
14.99 € без НДС		-	1 <input type="text"/>	
9.99 € без НДС	-	-	-	-

3. Выбираем дата-центр, Франция или Канада.



Choose your datacentre

Datacentres*

 The fastest delivery time in Europe
I would like the fastest possible delivery, so the first a

4. Подтверждаем заказ. Тут можно выбрать период, 1 3 6 или 12 мес. На kimsufi никаких скидок нет, поэтому оплачивать на 1 год, ну если хочется, других причин нет.

Preview of your order

Dedicated Server(s)

SUBSCRIPTION	Domain	Quantity	Unit price	Price +vat
Kimsufi - 16G i7 1x2TB Server - 1 month	*001	1	24.99 €	24.99 €
SUB-TOTAL				24.99 €

INSTALLATION	Domain	Quantity	Unit price	Price +vat
Setup fee	*001	1	9.99 €	9.99 €
Datacentre France	*001	1	0.00 €	0.00 €
SUB-TOTAL				9.99 €

Change the duration of the lease Chosen duration: 1 month
 Enter a promo code

SUBSCRIPTION	24.99 €
INSTALLATION	9.99 €

PRICE EXCL. VAT	34.98 €
VAT (20 %)	7.00 €
TOTAL INCL. VAT	41.98 €

5. Регистрируем новый аккаунт

Identification

You are a new customer
 You are already a customer

6. Заполняем данные

Фамилия, имя.

Потом страна Россия обязательно.

Выбираем физ лицо, потом Улица и дом. Ну прописка, которую вы отправите вместе с паспортом.

Индекс и город.

И язык — английский например.

email и дополнительный/резервный email

А в самом конце можете написать, кто вас привел. Мне ничего не дают за это. Но просто так, я бы попросил вас указывать какой-ниб мой новостной проект.

General information:

Name* : Koko

Surname* : Hekmatyar

Country* : Russian Federation

Choose your profile:

You are* : an individual a company an association Other

Address* : ul. Jormungand 40
(can be checked by letter)

Post code* : 630119

Town* : Novosibirsk

Contacts:

Email* : koko@hekmatyar.ru

Confirm email* : koko@hekmatyar.ru

Telephone (+33-122456789)* : +79139126666

Fax:

Spare email (optional): alice2k@default.im
Can be used instead of your main email contact, in the event it is no longer accessible

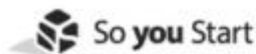
Language:

Language* : English

7. На последнем шаге будет ваш номер.

На всех сайтах OVH они маркируются по линейке. Первые буквы это имя и фамилия, потом идет номер и суфикс от линейки.

- kimsufi — kr34350-ks
- soyostart — kr98212-sys
- ovh — kr34297-ovh



Bon de commande :


Número : 28287079

Date : 01 Juin 2014 13:55:07

Expiration : 02 Juin 2014 13:55:07

Russia, Severo-Kyrl

[Version imprimable](#)



Votre compte d'utilisateur n'a pas encore été validé.

Pour passer commande, merci de nous faire parvenir les deux documents suivants à validation@ovh.net en renseignant votre Nichandle : [ka99037-sys](#)

Une preuve d'identité:

- Carte d'identité
- Permis
- Passeport

Un justificatif d'adresse:

- Facture (Gaz, électricité, téléphone)
- Relevé bancaire
- Correspondance officielle

[Retour sur le site](#)

8. Пишем на email pagalba@ovh.lt

Прилагаете этот номер kr34350-ks и скан паспорта или другой любой документ, подтверждающий ваши данные, что вы там указали. И пишете, что хотите снять налог VAT и активировать аккаунт. На каждом сайте нужен новый аккаунт. Поэтому рекомендую сразу зарегистрироваться везде, на kimsufi, на soyoustart, на ovh и потом разом подтвердить на паспорт.

Кстати, первая покупка, т.е. первый метод оплаты, если карты оплаты меняются, то значит так же долго будут, а если уже хоть раз платилось, что моментально. Иногда могут еще отказать, если ФИО на карте не совпадает с аккаунтом, довольно проблематично иногда получается, особенно если ты даешь друзьям ссылки для оплаты. Так что 100% моментальной оплаты у них нет, нужно заранее знать о этом и иметь резерв времени. Это не горячие сервера для переноса.

Свой сервер я заказывал именно по этой инструкции. Первый раз прошла оплата с виртуалки киви, сейчас плачу виртуалкой другой платежки, все проходит без проблем.

При заказе напрямую вы заплатите setup fee (плата за установку) в любом случае, учитывайте это. После активации заказа вам пришлют данные для доступа в панель управления. Логинимся, выбираем шаблон установки (я рекомендую Centos 6 x64, дальше ее будем рассматривать), устанавливаем.

Далее заказываем впс для прокси у любого хостера. Берем минимальный тариф, нам на железо абсолютно параллельно. И переходим к настройке.

Часть 3: настройка БС

В свое время я долго любилсяс вестой, потом плюнул, накатил голые пакеты. На весте было 80 доменов - тупили, загрузка доходила до 6.x . На голых пакетах экспериментально наваливал те 80 доменов + 320 доменов - нагрузка была в районе 2.x (для двухядерного впс норм). После этого эксперимента я понял, что к весте я вернусь только по принуждению.

Рассмотрим привычную мне работу на Centos 6 и связке NGINX+PHP-FPM. Для полноты картины установим дополнительно MySQL, monit.

Логинимся по SSH под рутом на наш БС

```
//ставим репозитории
```

```
rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
```

```
rpm -Uvh https://mirror.webtatic.com/yum/el6/latest.rpm
```

```
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

```
wget http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
rpm -Uvh remi-release-6*.rpm epel-release-6*.rpm
yum -y install epel-release
cd /etc/yum.repos.d
wget http://rpms.remirepo.net/enterprise/remi.repo
```

```
//ставим нужные полезные утилиты
yum install htop nano mc screen monit -y
yum --enablerepo=remi,remi-php56 install nginx18 php-fpm php-common -y
yum --enablerepo=remi,remi-php56 install php-opcache php-pecl-apcu php-cli php-pear php-pdo
php-mysqlnd php-pgsql php-pecl-mongo php-pecl-sqlite php-pecl-memcache php-pecl-memcached
php-gd php-mbstring php-mcrypt php-xml php-ioncube-loader -y
yum install --enablerepo=remi mysql-server
```

```
//открываем 80 порт
iptables -I INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
service iptables save
/etc/init.d/iptables restart
```

```
// добавлем сервисы в автозагрузку
chkconfig monit on
chkconfig php-fpm on
chkconfig nginx on
chkconfig mysqld on
service mysqld start
service monit start
service php-fpm start
service nginx start
```

```
//настраиваем monit (мониторинг процессов)
nano /etc/monit.d/nginx
И вставляем туда
```

```
check process nginx with pidfile /var/run/nginx.pid
    start program = "/sbin/service nginx start"
    stop program = "/sbin/service nginx stop"
```

Жмем Ctrl+X и Y (сохраняем)

```
nano /etc/monit.d/mysqld
```

```
Вставляем
check process mysqld with pidfile /var/run/mysqld/mysqld.pid
    start program = "/sbin/service mysqld start"
    stop program = "/sbin/service mysqld stop"
```

И сохраняем

```
nano /etc/monit.d/php
```

Вставляем

```
check process php-fpm with pidfile /var/run/php-fpm/php-fpm.pid
    start program = "/sbin/service php-fpm start"
    stop program = "/sbin/service php-fpm stop"
```

Сохраняем

```
monit -t
monit reload
monit start all
```

Настроим NGINX
Nano /etc/nginx/nginx.conf

worker_processes 1; заменяем 1 на количество ядер нашего БС*2. В моем случае это 8

Теперь у нас есть почти полностью настроенный веб-сервер на NGINX+PHP-FPM с MySQL и всеми необходимыми модулями для PHP. Этого хватит для работы 90% CMS и скриптов.
Теперь добавим наш первый домен.

```
mkdir /home/www/наш_домен
```

Создадим тестовый индексный файл
nano /home/www/наш_домен/index.php

В него вставляем

```
<?
phpinfo();
?>
```

Теперь нам нужно создать конфиг NGINX для нашего домена

```
nano /etc/nginx/conf.d/наш_домен.conf
```

В него вставляем

```
server {
listen          80;
server_name     наш_домен www.наш_домен;
index index.php;
root /home/www/наш_домен;
location / {
    try_files $uri $uri/ =404;
}

error_page 404 /404.html;
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root /usr/share/nginx/html;
}
location ~ \.php$ {
    try_files $uri =404;
    fastcgi_split_path_info ^(.+\.(php|php5))(/.+)$;
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}
}
```


И перезагрузим NGINX для применения нового конфига
Service nginx restart

На этом собственно настройка закончена. Если привязать домен к айпи БС и открыть его в браузере, то откроется страница с информацией о PHP. Но нам ведь нужно спрятать БС от посторонних глаз. Переходим к настройке прокси-впс. Именно этот сервер клиенты будут видеть при обращении к нашему домену.

Настройка прокси-впс

Заходим на прокси по ssh

```
//ставим репозитории
```

```
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

```
wget http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
```

```
rpm -Uvh remi-release-6*.rpm epel-release-6*.rpm
```

```
yum -y install epel-release
```

```
cd /etc/yum.repos.d
```

```
wget http://rpms.remirepo.net/enterprise/remi.repo
```

```
//ставим нужные полезные утилиты
```

```
yum install htop nano mc screen monit -y
```

```
yum --enablerepo=remi,remi-php56 install nginx18 -y
```

```
//открываем 80 порт
```

```
iptables -I INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
```

```
service iptables save
```

```
/etc/init.d/iptables restart
```

```
// добавлем сервисы в автозагрузку
```

```
chkconfig nginx on
```

```
service nginx start
```

Настроим NGINX для работы с Cloudflare

```
Nano /etc/nginx/nginx.conf
```

В секцию http { добавляем следующие строки после index index.html index.htm;

```
set_real_ip_from 103.21.244.0/22;  
set_real_ip_from 103.22.200.0/22;  
set_real_ip_from 103.31.4.0/22;  
set_real_ip_from 104.16.0.0/12;  
set_real_ip_from 108.162.192.0/18;  
set_real_ip_from 131.0.72.0/22;  
set_real_ip_from 141.101.64.0/18;  
set_real_ip_from 162.158.0.0/15;  
set_real_ip_from 172.64.0.0/13;  
set_real_ip_from 173.245.48.0/20;  
set_real_ip_from 188.114.96.0/20;  
set_real_ip_from 190.93.240.0/20;  
set_real_ip_from 197.234.240.0/22;  
set_real_ip_from 198.41.128.0/17;  
set_real_ip_from 199.27.128.0/21;  
set_real_ip_from 2400:cb00::/32;  
set_real_ip_from 2606:4700::/32;  
set_real_ip_from 2803:f800::/32;  
set_real_ip_from 2405:b500::/32;  
set_real_ip_from 2405:8100::/32;  
set_real_ip_from 2c0f:f248::/32;
```

```
set_real_ip_from 2a06:98c0::/29;
real_ip_header CF-Connecting-IP;
```

Если Cloudflare не используете, добавлять ничего не надо.

На этом настройка прокси закончена. Переходим к созданию домена

```
nano /etc/nginx/conf.d/наш_домен.conf
```

И вставляем туда

```
server {
listen          80;
server_name     наш_домен www.наш_домен;
location / {
    proxy_pass      http://ip_нашего_БС:80;
    proxy_set_header Host      $host;
    proxy_set_header X-Real-IP $remote_addr;
    }
}
```

Сохраняем и перезапускаем NGINX
Service nginx restart

Теперь, чтобы ip-адреса клиентов определялись на БС корректно, нам нужно его немного до настроить. Возвращаемся в консоль БС.

```
Nano /etc/nginx/nginx.conf
```

И в секцию http { добавляем следующую строку после index index.html index.htm;

```
set_real_ip_from ip_прокси_впс;
real_ip_header X-Real-IP;
real_ip_recursive on;
```

Сохраняем и перезапускаем NGINX.

На этом настройка закончена.

Заключение.

Таким образом у меня работала большая сетка дорвеев, я рассказал то, что делал сам. На самом деле, все что уместилось на 10 листах стоило мне очень много времени и сил. И областей применения подобных связей крайне велико: от сетей дорвеев до поддержания отказоустойчивой инфраструктуры, которая может быть переконфигурирована в любой момент. Вы не потеряете никакие данные из-за абузы конкурентов или правообладателей. Даже если ваш прокси остановит хостер, вам будет достаточно запустить новый, настроить его буквально за 5 минут и вы снова в строю и злите нехороших дядек своей живучестью. Ваши дорвеи будут летать как ошпаренные, а расходы на хостинг будут минимальны.

По любым вопросам можете обращаться в telegram <https://t.me/behemostte>. Дополнительно могу предоставить баш-скрипты для настройки серверов и прокси, добавлению доменов и т.д.