Немного о DNS.

Теория.

Основная цель DNS — это преобразование доменных имен в IP адреса и наоборот.

DNS состоит из распределенной базы имен, чья структура напоминает логическое дерево, называемое пространством имен домена. Каждый узел в этом пространстве имеет свое уникальное имя. Это логическое дерево «растет» из корневого домена, который является самым верхним уровнем иерархии DNS и обозначается символом – точкой. А уже от корневого элемента ответвляются поддоменые зоны или узлы (компьютеры).

> "*.*" -> "*.com*"
>
> "*.*" -> "*.ru*" -> "*1cloud.ru*" -> "*api.1cloud.ru*"

Исторически, до появления доменной системы имен роль инструмента разрешения символьных имен в IP выполнял файл /etc/hosts, который и в настоящее время играет далеко не последнюю роль в данном деле. Но с ростом количества хостов в глобальной сети, отслеживать и обслуживать базу имен на всех хостах стало нереально затруднительно. В результате придумали DNS, представляющую собой иерархическую, распределенную систему доменных зон.

Давайте взглянем, как происходит сопоставление имен и IP-адресов. Предположим, пользователь набирает в строке браузера www.1cloud.ru и нажимает Enter. Браузер посылает запрос DNS-серверу сети, а сервер, в свою очередь, либо отвечает сам (если ответ ему известен), либо пересылает запрос

одному из высоко-уровневых доменных серверов (или корневому).

Затем запрос начинает свое путешествие – корневой сервер пересылает его серверу первого уровня (поддерживающего зону .ru). Тот – сервер у второго уровня (1cloud) и так далее, пока не найдется сервер, который точно знает запрошенное имя и адрес, либо знает, что такого имени не существует. После этого запрос начинает движение обратно.

Также стоит пару слов сказать про процедуру обратного сопоставления – получение имени по предоставленному IP-адресу. Это происходит, например, при проверках сервера электронной почты. Существует специальный домен in-addr.arpa, записи в котором используются для преобразования IP-адресов в символьные имена. Например, для получения DNS-имени для адреса 11.22.33.44 можно запросить у DNS-сервера запись 44.33.22.11.in-addr.arpa, и тот вернёт соответствующее символьное имя.

Ресурсная запись — это то, собственно ради чего в конечном счете и существует DNS.

Ресурсная запись — это единица хранения и передачи информации в DNS. Каждая такая запись несет в себе информацию соответствия какого-то имени и служебной информации в DNS, например соответствие имени домена — IP адреса.

Bind9 это пакет создающий DNS-сервер который определяет доменное имя по IP-адресу в локальной или глобальной сети. Bind9 может также работать и в режиме кеширующего DNS-сервера. BIND использует 53/TCP, UDP порт.

Однако, BIN9 это именно текстовый DNS-сервер. В том, смысле, что все настройки хранятся в виде текстовых файлов. И при

запросе данных сервер будет считывать данные именно из этих файлов.

Давайте с вами определимся на том, что DNS-сервер это всё-таки БАЗА ДАННЫХ доменных имён!

И текстовый dns-сервер сюда никак не вписывается. В том смысле, что он подойдёт скорее либо для личного использования, либо для маленькой инфраструктуры какой-либо организации. Рано или поздно любая организация разрастается до нескольких тысяч доменных имен. И управлять вручную таким количеством текста просто невозможно.

> *- Щас подожди я долистаю до твоей записи...Балин, кажись пропустил, ща пагодь пару часиков.*

Именно поэтому нормальный DNS-сервер это как минимум база данных, а как максимум DNS-серверов даже личных или корпаративных всегда должно быть минимум 2. Пусть даже это будут 2 разных VPS-ки, но их будет минимум 2. В случае сбоя одного сервера, вам даже будет не важно сохранение всей базы данных. У вас есть 2 сервер, просто настраиваете 1 заново, затем вставляете api и пароли в настройки второго и базы автоматически синхронизируются между ними. + пользователи в любом случае на время ремонта одного из серверов не потеряют доступ к своим ресурсам.

В качестве примера для собственных **DNS**-серверов вполне подойдёт **PowerDNS**.

Его можно запустить и в **docker**-е, и просто на хост машину установить.

Однако, стоит обратить ваше внимание на 1 важный фактор. При установке его в docker - база данных **mariadb -** стандартная её конфигурация не совсем подходит для этой задачи. Точнее

рано или поздно вы скорее всего получите ошибку добавления или считывания данных 400, т.е. базу необходимо немного до-настроить. Иначе после каждого обновления такие настройки придётся делать заново. Да и определить что происходит и почему вы не можете. Добавить или считать записи тоже будет проблематично. У меня на это понимание ушло аж 2 недели, потому как информации об этом толком нет.

Так происходит потому что именно **PowerDNS** требует полного **root** доступа к базе. По умолчанию в **MariaDB выключен удалённый доступ**, а также **не настроены права доступа**.

Чтобы настроить права можно поступить 2 способами. Если вы ставили базу на хост машину без docker-контейнера то просто введи ниже указанные команды. Желательно проконтролировать через **phpmyadmin**. Для докера сначала войдите в контейнер с базой. Или можете скопировать официальный **Dockerfile** и просто дополнить его парой команд к последнему **RUN** рядом с последним вызовом примерно так:

```
&& \

command
```

Примерно такими командами.

```
mysql -e "select User, host from mysql.user;"

mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'powerdns' WITH GRANT OPTION; FLUSH PRIVILEGES;"

mysql -e "use mysql; GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'powerdns' WITH GRANT OPTION; FLUSH PRIVILEGES;"

mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'172.6.0.%' IDENTIFIED BY 'powerdns' WITH GRANT OPTION; FLUSH PRIVILEGES;"
```

В разных версиях могут появляться или наоборот исчезать различные системные таблицы, которые стоит для ускорения искать через **phpmyadmin**.

Также для открытия удалённого доступа необходимо в файле **/etc/mysql/my.cnf** раскоментировать строку:

> *bind-address = 0.0.0.0*

Чтобы сервер активировал удалённый доступ для root на всех интерфейсах. Можете конечно поэскпериментировать и задать конкретный адрес, но я всё-таки рекомендую именно все сетевые интерфейсы. Да это менее безопасно, но создаёт меньше проблем для соединения с dns-сервером.

Рекомендую для каждого сервиса создать отдельных пользователей и отдельные базы со всеми привилегиями и правами через **phpmyadmin**.

И далее в **docker-compose** указывать не «image: mariadb», а «build: .», где и расположен ваш переписанный **Dockerfile**.

Зачем нужен DNS-сервер ?

Давайте представим, что у вас есть зарегистрированное доменное имя второго уровня - хорошее, короткое. Не важно для каких целей - для домашнего сервера, корпоративный сервис, корпоративный веб-сайт... Всё что угодно, цель не важна. Вам понадобилось, и вы зарегистрировали.

Более важный вопрос за какую цену? К сожалению, доменные имена только 3 уровня бесплатны. Да есть DuckDns и ему подобные, но об этом позже.

Самое интересное в том, что если вы захотите иметь более 2 доменов любого уровня, любой хостинг будет сдирать с вас деньги только за одно их содержание, т.е. иметь более 2

доменов не бесплатное удовольствие. Хотите более 2 доменов бесплатно - ресурсные записи вам в помощь, но это будет уже на уровень больше.

Да, даже **DuckDNS** всё равно домен 3 уровня. Но, все подобные сервисы ограничены количеством доменов не более 5. **Опять ограничения.**

И вот тут возникает мысль - а что если перевести управление доменом - правильнее сказать - делегировать управление на свой собственный хостинг-провайдер или свой собственный DNS-сервер? Не на **CloudFlare** и его **ddn-docker-conteiner**, а именно свой собственный сервер. Потому что со временем даже на **CloudFlare** могут ввести какие-нибудь санкции или ограничения и вы ничего с этим поделать не сможете. Просто в качестве примера.

Да, это вполне возможно.

Сколько бы у вас не было доменов и под-доменов, всеми можно управлять на своём собственном сервере. Правила уже задаёте вы сами.

Сначала поговорим про собственные хостинги.

Практически во всех панелях управления хостингами поголовно используется BIND9. Стоит только взглянуть даже не в исходный код, а на процесс установки и сразу станет ясно.

Однако, не всё так плохо. Во всех панелях BIND9 используется совместно с базой данных. Точнее говоря панель позволяет без хлопот управлять записями, хотя и хранятся они по сути в текстовом виде. Такой вот недостаток всех панелей.

Но это только про бесплатные панели. Про платные ничего не могу сказать!

Однако, они позволят вам управлять вашими доменами неограниченно. Правила задаёте уже вы сами. + можете даже предоставлять услуги хостинга уже платно и самостоятельно.

Я же обращу ваше внимание только на несколько панелей, т.к. в них управление и записи наиболее адекватные по моему мнению.

ISPanel, **BrainyCP**, **VestaCP**, **HestiaCP** (следующее поколение **VestaCP**).

Где-то больше настроек, где-то меньше. Мне больше всех **понравилась VestaCP**, но судя по **IT-новостям** некоторые старые версии могут быть уязвимыми для хакеров.

Последнюю панель можно даже расположить в Docker-контейнере.

Не забывайте про fail2ban, антивирус, фаервол и другие защиты ваших серверов, которые поддерживаются всеми указанными панелями.

А теперь поговорим про собственный DNS-сервер.

А вот здесь всё намного интереснее.

При делегирование прав на свой собственный DNS-сервер у вас появляется гораздо больше управления не только ресурсными записями, но и возможность подключения через API различных сервисов. Например, **Nginx Proxy Manager** при обращении к вручную указанному **ACME** серверу имеет возможность обратиться к **PowerDNS** серверу с указанным **API** для ускорения получения **Let's Encrypt** сертификата.

В ресурсных записях появляются кучу дополнительных полей, о которых многие вообще не подозревали. Также можно вообще

создать собственные шаблоны, разграничить права доступа и многое другое.

Т.е. например, чтобы настроить свою собственную электронную почту со своим доменом будет немного проще, т.к. у вас будут некоторые подсказки, которых в ресурсных записях ваших провайдеров никогда нет. Они всегда при делегировании прав пустые. А здесь сразу можно задать что захотите под свой шаблон.

Кстати, говоря о подсказках. Что мне сильно понравилось в панели VestaCP - там есть ресурсные записи, так сказать по умолчанию в качестве примера. Что является довольно таки удобным для настройки, так сказать, по образу и подобию.

Ну а сегодня на этом всё. Надеюсь я хоть немного вас заинтересовал.

Спасибо за внимание. Всем Удачи, до новых встреч, Пока-Пока!