Если вы видите что-то необычное, просто сообщите мне.

Сети

- Настройка статического сайта с помощью Nginx
- DNS, DHCP, маршрутизация.
- Nginx, как частный пример сервера.
- Пример настройки iptables

Настройка статического сайта с помощью Nginx

Не забудьте установить nginx, для этого используйте команду `sudo apt install nginx`.

Для начала создадим файл, который мы будем раздавать с помощью nginx.

Поместим в файл `index.html` по адресу `/srv/www/test.ru` текст:

```
<html>
    <head>
        <title>Приветственная страничка</title>
    </head>
    <body>

        Привет, Мир!

    </body>
</html>
```

Теперь добавим конфигурационный файл `test.ru.conf` по адресу `/etc/nginx/sites-enable/` для nginx:

```
server {
       listen 80;

       server_name  test.ru;

       access_log    /var/log/nginx/shkolapobedy.access.log;
       error_log     /var/log/nginx/shkolapobedy.error.log;

       root /srv/www/test.ru;
```

```
        index index.html;

        location / {
        □□   autoindex on;
                try_files $uri $uri/ =404;
        }
}
```

Для примера приведем конфигурационные файл, которым можно заменить конфиг выше:

```
server {
        listen 80;

        server_name  test.ru;

        access_log    /var/log/nginx/shkolapobedy.access.log;
        error_log     /var/log/nginx/shkolapobedy.error.log;

        location / {
        □□   root /srv/www/test.ru;
        □□   autoindex on;
                try_files $uri $uri/ =404;
        }
}
```

Проверяем, что мы не допустили ошибок и конфигурация рабочая:

```
nginx -t
```

Перезапускаем:

```
systemctl restart nginx
```

Проверяем доступность сайта

Так как у нас нет своего домена и привзяанного к нему ip, то сайт не будет доступен по адресу `test.ru`. Для того, чтобы исправить ситуацию необходимо внести `test.ru` в файл `/etc/hosts` Откройте файл `hosts` для редактирования и внесите следующую строку:

```
127.0.0.1 test.ru
```

Теперь можно открыть браузер и зайти по адресу `test.ru`

Вы должны увидеть страничку, которую мы положили по адресу `/srv/www/test.ru`

DNS, DHCP, маршрутизация.

DNS

Что такое DNS-сервер?

Принцип работы DNS похож на поиск и вызов контактов из телефонной книги смартфона. Ищем имя, нажимаем «позвонить», и телефон соединяет нас с нужным абонентом. Понятно, что смартфон в ходе звонка не использует само имя человека, вызов возможен только по номеру телефона. Если вы внесете имя без номера телефона, позвонить человеку не сможете.

Так и с сайтом. Каждому имени сайта соответствует набор цифр формата 000.000.000.000. Этот набор называется IP-адресом, примером реального IP-адреса является 192.168.0.154 или 203.113.89.134. Когда пользователь вводит в адресной строке браузера имя сайта, например google.com, компьютер запрашивает IP-адрес этого сайта на специальном DNS-сервере и после получения корректного ответа открывает сам сайт.

Зачем нужны DNS-серверы и какие они бывают?

Основное предназначение DNS-серверов — хранение информации о доменах и ее предоставление по запросу пользователей, а также кэширование DNS-записей других серверов. Это как раз «книга контактов», о которой мы писали выше.

В случае кэширования все несколько сложнее. Дело в том, что отдельно взятый DNS-сервер не может хранить вообще всю информацию об адресах сайтов и связанных с ними IP-адресами. Есть исключения — корневые DNS-серверы, но о них позже. При обращении к сайту компьютера пользователя браузер первым делом проверяет локальный файл настроек DNS, файл hosts. Если там нет нужного адреса, запрос направляется дальше — на локальный DNS-сервер интернет-провайдера пользователя.

Локальный DNS-сервер в большинстве случаев взаимодействует с другими DNS-серверами из региона, в котором находится запрошенный сайт. После нескольких обращений к таким серверам локальный DNS-сервер получает искомое и отправляет эти данные в браузер — запрошенный сайт открывается. Полученные данные сохраняются на локальном сервере, что значительно ускоряет его работу. Поскольку, единожды «узнав» IP-адрес сайта, запрошенного пользователем, локальный DNS сохраняет эту информацию. Процесс сохранения полученных ранее данных и называется кэшированием.

Если пользователь обратится к ранее запрошенному сайту еще раз, то сайт откроется быстрее, поскольку используется сохраненная информация. Правда, хранится кэш не вечно, время хранения зависит от настроек самого сервера.

IP-адрес сайта может измениться — например, при переезде на другой хостинг или сервер в рамках прежнего хостинга. Что происходит в этом случае? В этом случае обращения пользователей к сайту, чей IP-адрес поменялся, некоторое время обрабатываются по-старому, то есть перенаправление идет на прежний «айпишник». И лишь через определенное время (например, сутки) кэш локальных серверов обновляется, после чего обращение к сайту идет уже по новому IP-адресу.

DNS-серверы верхнего уровня, которые содержат информацию о корневой DNS-зоне, называются корневыми. Этими серверами управляют разные операторы. Изначально корневые серверы находились в Северной Америке, но затем они появились и в других странах. Основных серверов — 13. Но, чтобы повысить устойчивость интернета в случае сбоев, были созданы запасные копии, реплики корневых серверов. Так, количество корневых серверов увеличилось с 13 до 123.

Что такое DNS-зоны?

- A — адрес веб-ресурса, который привязан к конкретному имени домена.
- MX — адрес почтового сервера.
- CNAME — чаще всего этот тип записи используется для подключения поддомена.
- NS — адрес DNS-сервера, который отвечает за содержимое других ресурсных записей.
- TXT — любая текстовая информация о доменном имени.
- SPF — данные с указанием списка серверов, которые входят в список доверенных для отправки писем от имени указанного домена.
- SOA — исходная запись зоны, в которой указаны сведения о сервере и которая содержит шаблонную информацию о доменном имени.

DHCP

DHCP — протокол прикладного уровня модели TCP/IP, служит для назначения IP-адреса клиенту. Это следует из его названия — Dynamic Host Configuration Protocol. IP-адрес можно назначать вручную каждому клиенту, то есть компьютеру в локальной сети. Но в больших сетях это очень трудозатратно, к тому же, чем больше локальная сеть, тем выше возрастает вероятность ошибки при настройке. Поэтому для автоматизации назначения IP был создан протокол DHCP.

Принцип работы DHCP Из вступления ясно, какие функции предоставляет DHCP, но по какому принципу он работает? Получение адреса проходит в четыре шага. Этот процесс называют DORA по первым буквам каждого шага: Discovery, Offer, Request, Acknowledgement.

Сетевая маршрутизация

Понятие «маршрутизация» включает в себя несколько значений, одно из них — это передача информации от отправителя к получателю. В IT-среде маршрутизацией называется аппаратное вычисление маршрута движения пакетов данных между сетями с использованием специального сетевого устройства – маршрутизатора.

Маршруты могут быть статическими и задаваться администратором сети, или динамическими и рассчитываться сетевыми устройствами по определенным алгоритмам (протоколам) маршрутизации, которые основаны на данных о топологии сети.

Виды маршрутизации:

- Различают два вида маршрутизации: программная и аппаратная.
- Программная маршрутизация — это специализированное программное обеспечение, установленное на компьютере с несколькими сетевыми интерфейсами, которые входят в состав различных сетей.
- Аппаратная маршрутизация осуществляется специальным оборудованием, способным анализировать и перенаправлять входящие потоки данных.

Что такое NAT

При проектировании сетей обычно применяются частные IP-адреса 10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16. Их используют внутри сети площадки или организации для поддержания локального взаимодействия между устройствами, а не для маршрутизации во всемирной сети. Чтобы устройство с адресом IPv4 могло обратиться к другим устройствам или ресурсам через интернет, его частный адрес должен быть преобразован в публичный и общедоступный. Такое преобразование — это главное, что делает NAT, специальный механизм преобразования приватных адресов в общедоступные.

Терминология NAT:

- внутренний локальный — видимый во внутренней сети адрес источника, собственный локальный адрес устройства.
- внутренний глобальный — видимый из внешней сети адрес источника. При передаче трафика, например, с локального компьютера на веб-сервер, его Inside local address преобразуется маршрутизатором во внутренний глобальный адрес.
- внешний локальный — видимый из внешней сети адрес получателя. Присвоенный хосту глобально маршрутизируемый адрес IPv4.
- внешний глобальный — видимый из внутренней сети адрес получателя. Часто совпадает с локальным внешним адресом.

Типы NAT:

- Static NAT — статическая адресная трансляция. Предусматривает сопоставление между глобальными и локальными адресами «один к одному».
- Dynamic NAT — динамическая адресная трансляция. Сопоставление адресов осуществляется по принципу «многие ко многим».
- Port Address Translation (NAT Overload) — трансляция с использованием портов. Предусматривается многоадресное сопоставление.

Проброс портов.

«Пробросить» порты — это дать команду роутеру зарезервировать один порт и все приходящие на него данные для передачи на определенный компьютер. Другими словами, сделать исключение из правила отклонения неинициированных внешних запросов и принимать их при заданных условиях.

Для этого задается правило перенаправления любого свободного порта интерфейса WAN на роутере на определенный порт указанного устройства.

После того, как правило перенаправления будет создано, входящие на указанный внешний порт запросы будут адресованы указанному порту определенного устройства.

Пример такого подключения я вляется перенапралвение, когда на роутере мы указываем, что входящий запрос на 80 порт должен перевестись на определенный компьютер в локальной сети. В этом случае любой запрос из интернета на ip адрес, если он не за NAT, откроет приложение запущенно на компьютере в локальной сети.

Nginx, как частный пример сервера.

NGINX — это веб-сервер и почтовый прокси, который работает под управлением операционных систем семейства Linux/Unix и Microsoft.

Область применения

Веб-сервер применяется в следующих ситуациях:

- ыделенный порт или IP-адрес. Если на сервере присутствует большое количество статичного материала (картинки, тексты и т. д.) либо файлов для загрузки пользователями, то Nginx используют, чтобы выделить под данные операции отдельный IP-адрес либо порт. Таким образом нагрузка на сервер распределяется.
- Прокси-сервер. Когда пользователь загружает страницу сайта, на которой расположен статичный контент, Nginx сначала кэширует данные у себя, а потом возвращает результат. При следующих запросах данной страницы ответ происходит в разы быстрее.
- Распределение нагрузки. При запросе страницы сайта, пользователю выдается ответ в синхронной последовательности. Nginx использует асинхронный режим. Все запросы обрабатываются на разных этапах. Такой подход повышает скорость обработки.
- Почтовый сервер. Поскольку в веб-сервер встроены механизмы аутентификации, то его часто используют для перенаправления на почтовые сервисы после прохождении авторизации клиентом.

Установка сервера

```
sudo apt install nginx
```

Команды для взаимодействия с демоном

Запустить/остановить демон nginx

```
systemctl start/stop nginx
```

Включить/выключить автоматический запуск, `--now` - говорит не ждать следующий загрузки системы, а выполнить условие прямо сейчас.

```
systemctl disable/enable nginx --now
```

Проверить конфигурацию на правильность.

```
nginx -t
```

```
При изменении конфигурации nginx требуется перезагрузить.
```

Пример конфигурации:

Внимание! Конфиг ниже приведет в упрощенном виде.

```
worker_processes  auto;

pid  /var/run/nginx.pid;

events {
    worker_connections  1024;
    multi_accept on;
    use epoll;
}
```

```
http {
    default_type application/octet-stream;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    types_hash_max_size 2048;
    client_max_body_size 1G;

    include /etc/nginx/mime.types;

    server {
        listen 80;
        server_name test.ru;

        # gzip begin
        gzip on;
        gzip_disable "msie6";
        # gzip end

        location /static/ {
            root /code/public;
            expires max;
            try_files $uri$args $uri =404;
        }

        location / {
         auth_basic "Restricted Content";
         auth_basic_user_file /etc/nginx/htpasswd;
            proxy_pass http://web:8080;
            proxy_set_header Host $http_host;
            proxy_set_header Connection "upgrade";
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header Upgrade $http_upgrade;
            proxy_connect_timeout       600;
            proxy_send_timeout          600;
            proxy_read_timeout          600;
```

```
            send_timeout                600;
        }
    }
}
```

Ключевые слова, важные для нас:

- http - общие настройки сервера применяемые
- server - настройки применяемые исключительно для конкретного сервера.
- server_name - название сервера которое мы будем использовать
- listen - порт на котоом будет запущен nginx
- location - настройка различных префиксов, которые в свою очередь могут перенаправлять запросы в различные сервисы.
- include - включает конфиг, расположенный по указанному адресу

Пример настройки iptables

```
#./iptables_rules.sh
#! /bin/bash

IPT="iptables"
## Внешний интерфейс
WAN=ens224
WAN_IP=ВАШ_внешний_IP
## Внутренний интерфейс
LAN1=ens192
LAN1_IP=ВАШ_внутренний_IP
LAN1_IP_RANGE=192.168.0.0/24
LAN2=IP_Jitsi

## Очищаем все цепочки перед применением новых правил
$IPT -F
$IPT -F -t nat
$IPT -F -t mangle
$IPT -X
$IPT -t nat -X
$IPT -t mangle -X

## Блокируем весь трафик, который не соответствует ни одному из правил
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP

## Разрешаем весь трафик в локалхост и локальной сети
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A INPUT -i $LAN1 -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT
$IPT -A OUTPUT -o $LAN1 -j ACCEPT

## Этот блок разрешает icmp-запросы (пинговать сервер) (опционально)
$IPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
```

```
$IPT -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

## Открываем доступ в интернет самому серверу
$IPT -A OUTPUT -o $WAN -j ACCEPT
#$IPT -A INPUT -i $WAN -j ACCEPT

## Разрешаем все установленные соединения и дочерние от них
$IPT -A INPUT -p all -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A OUTPUT -p all -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -p all -m state --state ESTABLISHED,RELATED -j ACCEPT

## Защита от наиболее распространенных сетевых атак. (опционально)
## Отбрасываем все пакеты без статуса (опционально)
$IPT -A INPUT -m state --state INVALID -j DROP
$IPT -A FORWARD -m state --state INVALID -j DROP

## Блокируем нулевые пакеты (опционально)
$IPT -A INPUT -p tcp --tcp-flags ALL NONE -j DROP

## Закрываемся от syn-flood атак (опционально)
$IPT -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
$IPT -A OUTPUT -p tcp ! --syn -m state --state NEW -j DROP

## Запрет доступа с определенных IP (опционально)
#$IPT -A INPUT -s ipaddres -j REJECT

## Разрешаем доступ в интернет из локальной сети
$IPT -A FORWARD -i $LAN1 -o $WAN -j ACCEPT

## Запрещаем доступ из интернета в локальную сеть (опционально)
#$IPT -A FORWARD -i $WAN -o $LAN1 -j REJECT

## Чтобы в локальной сети был интернет включаем nat
$IPT -t nat -A POSTROUTING -o $WAN -s $LAN1_IP_RANGE -j MASQUERADE
##Должен быть включен ip forwarding

## Разрешаем ssh (порт 22 - это порт по умолчанию для ssh соединения, если вы его меняли, то
обязательно значение 22 надо заменить своим, иначе будет потерян доступ к серверу)
$IPT -A INPUT -i $WAN -p tcp --dport 22 -j ACCEPT
```

```
## Перенаправление трафика на сервер JITSI
$IPT -A FORWARD -i $WAN -d $LAN2 -p tcp -m tcp --dport 80 -j ACCEPT
$IPT -A FORWARD -i $WAN -d $LAN2 -p tcp -m tcp --dport 443 -j ACCEPT
$IPT -A FORWARD -i $WAN -d $LAN2 -p udp -m udp --dport 10000 -j ACCEPT
$IPT -t nat -A PREROUTING -i $WAN -p tcp --dport 80 -j DNAT --to $LAN2
$IPT -t nat -A PREROUTING -i $WAN -p tcp --dport 443 -j DNAT --to $LAN2
$IPT -t nat -A PREROUTING -i $WAN -p udp --dport 10000 -j DNAT --to $LAN2
$IPT -t nat -A POSTROUTING -j MASQUERADE
```

Применить конфигу

```
chmod +x iptables_rules.sh
./iptables_rules.sh
iptables-save > /etc/iptables-conf/iptables_rules.ipv4
```