Контейнер на роутере с radius и portainer (web для управления докерами) · Wiki · ecotelecom / ecotree

Последовательность действий для быстрого развёртывания docker-контейнера с freeradius и web-интерфейсом управления в среде EcoRouterOS

1. Включаем поддержку контейнеров

включить docker-сервер на ecorouter

```
ecorouter(config)#enable container
ecorouter(config)#
```

UPDATE 05.09.18 ВАЖНО!!!
появилась новая команда (теперь в CLI доступны обе)

```
ecorouter(config)#enable docker
ecorouter(config)#
```

enable container - запускает докер в неймспейсе роутера и использует для маршрутизации iptables + RIB линукса. Плюсы: ничего не нужно настраивать самому, минусы: обнаружен баг с натом в схеме, когда пакет адресован в сеть, которая не является directly connected.

enable docker - запускает докер в неймспейсе mgmt и предполагается, что функции маршрутизации и NAT'а возьмет на себя роутер. Плюсы: можно избежать бага выше, после ребута роутера все будет работать, минусы: потребуется дополнительная настройка роутера + проблемы с неправильной udp/tcp checksum.

Пример настройки через enable docker:

```
в Linux:
docker network create --subnet=192.168.149.0/29 MyNet

в CLI роутера:
port virt.0
 virtual-network container MyNet
 service-instance 1
  encapsulation untagged

interface Docker
 ip mtu 1500
 connect port virt.0 service-instance 1
```

```
 ip nat inside
 ip address 192.168.149.6/24
!
+ Необходимо настроить маршрутизацию или NAT, чтобы вывести контейнеры в сеть
```

После настроек роутера важно не забыть поменять gw в контейнере, я это делал скриптом:

```
#!/bin/sh
set -x
ip route del default
ip route add default via 192.168.149.6
ethtool --offload eth0 tx off sg off tso off
radiusd -X
```

ethtool --offload eth0 tx off sg off tso off - команда, для пересчета checksum.
а дальше запускал контейнер командой:

```
docker run -it -v radius_volume:/etc/raddb/ --network MyNet --rm --cap-add
NET_ADMIN hub.rdp.ru/freeradius
```

2. Загружаем image с freeradius и portainer на роутер

Подключится под root на роутер и если есть ip-связность сделать docker pull , если же ip-связности нет, то:
1) На локальной машине сделать docker save -o portainer.tar portainer/portainer:develop и docker save -o radius.tar hub.rdp.ru/freeradius:latest 2) scp'шнуть эти архивы на роутер
3) Выполнить на роутере

```
cat portainer.tar | docker image load
```

4) Выполнить

```
cat radius.tar | docker image load
```

5) Проверить, что images установлены командой docker images (там должны появится два наших имейджа)

3. Отключаем hbm

На роутере есть нехороший функционал(для тех кто работает с докерами) hbm, который запрещает работу некоторых docker-команд, например, docker create volume
На момент написания статьи у нас не получилось добавить правила для конкретных программ, поэтому мы полностью отключали данный функционал:

```
/nix/store/h67369xpz36wydpwm0ay4fpvgs5lyj6f-hbm-0.11.0-bin/bin/hbm config set
```

```
authorization false
```

4. Скачиваем сертификаты

Для подключения к docker-серверу с удалённой машины необходимы следующие сертификаты

- корневой сертификат, которым подписаны сертификаты пользователей и docker-сервера
- сертификат пользователя **admin**
- секретный ключ пользователя **admin**

Выводим текущий корневой сертификат на консоль:

```
ecorouter#show crypto ca export | nopager
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            e3:28:e3:36:8d:0e:d1:1d
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=ecorouter
        Validity
            Not Before: Oct 27 11:31:10 2017 GMT
            Not After : Oct 28 11:31:10 2027 GMT
        Subject: CN=ecorouter
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:ad:e5:88:1e:45:d1:26:9f:b8:c7:72:89:73:18:
                    . . .
                    f0:b7:65
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                B1:72:ED:2D:E4:A4:F6:77:83:98:3A:2E:1D:3B:19:A7:3B:53:FC:65
            X509v3 Authority Key Identifier:

keyid:B1:72:ED:2D:E4:A4:F6:77:83:98:3A:2E:1D:3B:19:A7:3B:53:FC:65

            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
         24:78:83:24:3c:f6:63:3e:b9:79:e7:f2:69:46:22:66:4b:d8:
         . . .
         11:19:0b:a8:26:7a:56:b9
-----BEGIN CERTIFICATE-----
MIIE+zCCAuOgAwIBAgIJAOMo4zaNDtEdMA0GCSqGSIb3DQEBCwUAMBQxEjAQBgNV
```

```
. . .
gFm7ELWEJfR2vTG/O3pnkGsPKhGpJlkRGQuoJnpWuQ==
-----END CERTIFICATE-----

ecorouter#
```

Сохраним в файл `ca.pem` все строки вывода этой команды (начиная с `Certificate:`, кончая `-----END CERTIFICATE-----`) и дадим права chmod 444

````
Теперь выводим на консоль сертификаты пользователя admin:

```text
ecorouter#show crypto certificate export admin |nopager
User: admin
Certificate: Valid
-----BEGIN CERTIFICATE-----
MIIESTCCAjGgAwIBAgIBATANBgkqhkiG9w0BAQsFADAMMQowCAYDVQQDDAEqMB4X
DTE4MDcyMzEyNTkyN1oXDTE4MDgyMzEyNTkyN1owgZ4xCzAJBgNVBAYTAkVSMQ8w
. . .
VEW199D1UffzatoZF4SFMNTN0+OqHwx2MRZOJz8deYWELppG3miCD/vDZ0wXweZw
vaBUTpbYH2XXyooDJugfdZVGTF430/bjEEnA9XH9ba9lSl+P/CvSPXx74kH7
-----END CERTIFICATE-----

ecorouter#
````

Скопируем в файл `cert.pem` содержимое сертификата для пользователя **admin** и дадим права chmod 440:

Теперь выводим на консоль секретный ключ пользователя **admin**:

```
ecorouter#show crypto key export |nopager
User: admin
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEArIHVaWr4pAKdNLg5XjL9hbRoydrG1v7rZZhVonDcWvI5bUWn
. . .
28/ao4HDTgiNNZCvKvwfb0gEaVVGQWy9p/LMVix3fphQH0nqroNb
-----END RSA PRIVATE KEY-----

ecorouter#
```

Скопируем в файл `key.pem` содержимое секретного ключа для пользователя **admin** и дадим права chmod 440:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEArIHVaWr4pAKdNLg5XjL9hbRoydrG1v7rZZhVonDcWvI5bUWn
```
```

```
. . .
28/ao4HDTgiNNZCvKvwfb0gEaVVGQWy9p/LMVix3fphQH0nqroNb
-----END RSA PRIVATE KEY-----
```

5. Подключаемся извне к docker-серверу

Для подключения к docker-серверу с другого компьютера необходимо внести изменения в файл `/etc/hosts`, добавив в него следующую строку:

```
10.210.10.1 ecorouter
```

Без этой строки в нашем случае подключиться к docker-серверу будет невозможно.

Пробуем подключиться к docker-серверу при помощи скачанных сертификатов:

```
$ docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem -
H=ecorouter:2376 version
Client:
 Version:         1.12.6
 API version:     1.24
 Package version: docker-1.12.6-61.git85d7426.el7.centos.x86_64
 Go version:      go1.8.3
 Git commit:      85d7426/1.12.6
 Built:           Tue Oct 24 15:40:21 2017
 OS/Arch:         linux/amd64

Server:
 Version:         17.03.1-ce
 API version:     1.27
 Package version:
 Go version:      go1.8.3
 Git commit:      c6d412e
 Built:           Thu Aug 17 04:51:01 2017
 OS/Arch:         linux/amd64
$
```

6. Запускаем контейнер freeradius

1) Создаем volume, для хранения файлов конфигурации радиуса:

```
docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem -
H=ecorouter:2376 volume create radius_volume
```

где radius_volume - имя volume(может быть любым)

2) Проверяем, что volume создался успешно, для этого вводим команду docker volume ls:

```
[root@dc77-er1004:~]# docker volume ls
DRIVER              VOLUME NAME
local               portainer_data
local               radius_volume
```

3) Запускаем контейнер с радиусом:

```
docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem -
H=ecorouter:2376 run -d -v radius_volume:/etc/raddb/ --network MyNet -p
1812:1812/udp -p 1813:1813/udp --cap-add NET_ADMIN --restart=always
hub.rdp.ru/freeradius /etc/raddb/run.sh
```

где -d - запустить в фоновом режиме
-v radius_volume:/etc/raddb/ - сохранять все изменения в контейнере в /etc/raddb/ в созданный ранее volume
-p 1812:1812/udp - проброс портов в контейнер, в данном примере udp 1812
--restart=always - контейнер будет автоматически запускаться при старте системы, если администратор не выключал контейнер принудительно
--cap-add NET_ADMIN - права, позволяющие изменять маршруты в контейнере /etc/raddb/run.sh - запустить скрипт run.sh при старте контейнера

4)Проверить, что контейнер запустился можно командой ps:

```
docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem -
H=ecorouter:2376 ps
CONTAINER ID        IMAGE                    COMMAND             CREATED
STATUS              PORTS                                NAMES
f2fe2dfbe408        hub.rdp.ru/freeradius   "radiusd -X"        2 hours ago
Up 2 hours          0.0.0.0:1812-1813->1812-1813/udp   angry_kowalevski
```

P.S. Контейнер с радиусом может быть в состоянии restarted, одна из причин почему так может быть: в конфиге радиуса включена поддержка ipv6, ее нужно отключить: зайти на контейнер с помощью команды

```
docker run --rm -it -v radius_volume:/etc/raddb/ hub.rdp.ru/freeradius
```

и отредактировать файл /etc/raddb/sites-enabled/default, (закомментить все секции listen для ipv6)

7. Запускаем контейнер portainer

1) Создаем volume, для хранения файлов portainer'a:

```
docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem -
H=ecorouter:2376 volume create portainer_data
```

2)Запускаем докер с portainer

```
docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem -
H=ecorouter:2376 run -d -p 9000:9000 -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data --
restart=always --add-host=ecorouter:10.210.10.1 portainer/portainer
```

аргументы все те же, кроме одного:
--add-host=ecorouter:10.210.10.1 - добавляет запись "ecorouter 10.210.10.1" в /etc/hosts внутри контейнера, без данной записи не получится подключится к контейнеру через web

8. Подключаемся к web-интерфейсу

1)С локальной машины вбиваем в браузере x.x.x.x:9000, для того, чтобы попасть на web(где x.x.x.x - ip адрес интерфейса роутера, в моей примере это 10.210.10.1)
2)Придумываем логин/пароль администратору
3)Прописываем url: ecorouter:2376, где ecorouter - имя, которое прописано в etc/hosts внутри контейнера, а 2376 порт, на котором работает docker server
4)Добавляем 3 ранее сохраненных сертификата
Если все сделали правильно, то в web должны увидеть 2 контейнера, 2 имейджа, 2 вольюма и теперь управлять контейнерами можно из вебы:
посмотреть логи радиуса: containers -> выбираем freeradius -> в столбце quick actions выбираем logs
зайти на контейнер и, например, изменить конфиг: containers -> выбираем freeradius -> в столбце quick actions выбираем console
перезапустить контейнер - выбираем контейнер и нажимаем restart