

12.11.2021



Docker

Курс виртуализация



Илья Веретенников

Docker

Курс виртуализация

1. Описание подготовленной инфраструктуры.

1.1. Заготовлены 3 сервера Ubuntu 20.04, со следующими характеристиками:

RAM – 2Gb
CPU – 2
HDD – 25Gb
Сетевой адаптер – 1

Имена хостов: docker01, docker02, docker03

1.2. Сетевые настройки проведены в netplan:

Для docker01:

```
# This is the network config written by 'subiquity'  
network:  
  ethernets:  
    ens33:  
      dhcp4: yes  
      addresses: [192.168.197.150/24]  
      gateway4: 192.168.197.2  
      nameservers:  
        addresses: [8.8.8.8]  
      version: 2
```

Для docker02:

```
# This is the network config written by 'subiquity'  
network:  
  ethernets:  
    ens33:  
      dhcp4: yes  
      addresses: [192.168.197.151/24]  
      gateway4: 192.168.197.2  
      nameservers:  
        addresses: [8.8.8.8]  
      version: 2
```

Для docker03:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens33:
      dhcp4: yes
      addresses: [192.168.197.152/24]
      gateway4: 192.168.197.2
      nameservers:
        addresses: [8.8.8.8]
      version: 2
```

1.3. Внесены данные в файл /etc/hosts

Для docker01:

```
root@docker01:~# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 docker01

192.168.197.150 docker01
192.168.197.151 docker02
192.168.197.152 docker03

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Для docker02:

```
root@docker02:~# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 docker02

192.168.197.150 docker01
192.168.197.151 docker02
192.168.197.152 docker03

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Для docker03:

```
root@docker03:~# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 docker03

192.168.197.150 docker01
192.168.197.151 docker02
192.168.197.152 docker03

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

2. Установка необходимого ПО

Последовательно на всех виртуальных узлах:

1. Создать каталог /root/docker
2. Установить пакет apt-transport-https – пакет который позволяет получить доступ к apt репозиториям через https протокол.
3. Скачать и добавить ключ gpg со страницы репозитория в apt, что позволит установить пакет соответствующей версии, это действие можно провести следующей командой:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add  
OK
```

4. Добавить репозиторий с требуемой версией в список apt:

```
root@docker02:~# add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu focal stable"  
Сущ:1 http://ru.archive.ubuntu.com/ubuntu focal InRelease  
Пол:2 http://ru.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Пол:3 https://download.docker.com/linux/ubuntu focal InRelease [57,7 kB]  
Пол:4 http://ru.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]  
Пол:5 http://ru.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]  
Пол:6 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages  
[12,3 kB]  
Пол:7 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages  
[1 344 kB]  
Пол:8 http://ru.archive.ubuntu.com/ubuntu focal-updates/main Translation-en  
[276 kB]  
Пол:9 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f  
Metadata [14,5 kB]  
Пол:10 http://ru.archive.ubuntu.com/ubuntu focal-updates/universe amd64  
Packages [875 kB]  
Пол:11 http://ru.archive.ubuntu.com/ubuntu focal-backports/universe amd64  
Packages [6 592 B]  
Пол:12 http://ru.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f  
Metadata [580 B]  
Пол:13 http://ru.archive.ubuntu.com/ubuntu focal-security/main amd64 Packages  
[987 kB]  
Пол:14 http://ru.archive.ubuntu.com/ubuntu focal-security/main amd64 c-n-f  
Metadata [8 880 B]  
Пол:15 http://ru.archive.ubuntu.com/ubuntu focal-security/universe amd64  
Packages [651 kB]  
Получено 4 561 kB за 3с (1 341 kB/s)  
Чтение списков пакетов... Готово
```

5. Убедиться в том, что нет недостающих пакетов, даже после добавления репозитория:

```
apt update
```

6. Провести установку docker-ce

```
apt install docker-ce
```

7. Установка Docker завершена.

3. Инициализация узлов Docker и предоставление ролей в Swarm

1. Следующей командой определяется узел который в последствии будет мастер нодой Docker Swarm, выполняем команду на узле docker01, указываем ip адрес docker01:

```
docker swarm init --advertise-addr 192.168.197.150
```

Swarm initialized: current node (5kg3chhwwzugcrakx5v58dwvk) is now a manager.

To add a worker to this swarm, run the following command:

```
docker swarm join --token SWMTKN-1-  
2wc1o49k9lf9inxb03mef25dgf4wvmf2fbpd8nl9awqgw1wwnw-  
a5r73ymhsjl491ly3kb70z4rl 192.168.197.150:2377
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

Следует обратить внимание на то, что в полученном при исполнении выводе указана команда для подключения второго мастера.

2. Получаем команду подключения узла к Docker Swarm в роли `worker`.
Для этого выполняем:

На `docker01`:

```
root@docker01:~# docker swarm join-token worker
To add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-
2hbfrq1nbffs88wrgre23aitw1lw9tuof58amfexx4l8l5cc72-cxxercfofoine26zidjifkp57
192.168.197.150:2377
```

Затем исполняем полученную команду на узле `docker02`:

```
root@docker02:~# docker swarm join --token SWMTKN-1-
2hbfrq1nbffs88wrgre23aitw1lw9tuof58amfexx4l8l5cc72-cxxercfofoine26zidjifkp57
192.168.197.150:2377
This node joined a swarm as a worker.
```

На `docker03`:

```
root@docker03:~# docker swarm join --token SWMTKN-1-
2hbfrq1nbffs88wrgre23aitw1lw9tuof58amfexx4l8l5cc72-cxxercfofoine26zidjifkp57
192.168.197.150:2377
This node joined a swarm as a worker.
```

На любом из узлов можно просмотреть список и статусы нод Docker Swarm

```
root@docker01:~# docker node ls
ID                HOSTNAME  STATUS  AVAILABILITY  MANAGER STATUS  ENGINE VERSION
q6u0fzptol4icf6fe53p72sk * docker01  Ready   Active        Leader          20.10.10
hxz79pyjo4qkbowt21wcxvnc0 docker02  Ready   Active                20.10.10
7pnfao8dvmqkuedn5y8l1jgu20 docker03  Ready   Active                20.10.10
```

4. Установка и настройка Traefik

Все дальнейшие операции, кроме случаев, когда это оговорено отдельно проводятся на docker01.

Создать каталог /srv/traefik

```
root@docker01:~# cd /srv/traefik/
```

Проинициализировать сеть

```
root@docker01:/srv/traefik# docker network create --driver=overlay traefik-net  
0g00klga11evs0d0jcl8se1e5
```

Разместить рассмотренный при выполнении практической работы конфигурационный файл в каталоге /srv/traefik/traefik.yml.

Объявить текущий узел узлом развёртывания traefik.

```
export NODE_ID=$(docker info -f '{{.Swarm.NodeID}}')  
docker node update --label-add traefik-net.traefik-net-certificates=true $NODE_ID
```

Объявить переменные, используемые в конфигурационном файле, задать имя пользователя, пароль и закрыть пароль в целях повышения безопасности.

```
export USERNAME=<ваш пользователь>  
export PASSWORD=<ваш пароль>  
export HASHED_PASSWORD=$(openssl passwd -apr1 $PASSWORD)
```

На текущем этапе все готово для разворачивания traefik сервиса. Выполним

```
docker stack deploy -c traefik.yml traefik
```

5. Установка и настройка Swarmpit

В каталоге `/srv/traefik/` разместить конфигурационный файл `swarmpit.yml` рассматриваемый на практическом задании.

Подготовить переменную, содержащую Docker ID узла.

```
export NODE_ID=$(docker info -f '{{.Swarm.NodeID}}')
```

Объявить узел узлом размещения `swarmpit-db` и `swarmpit.influx`.

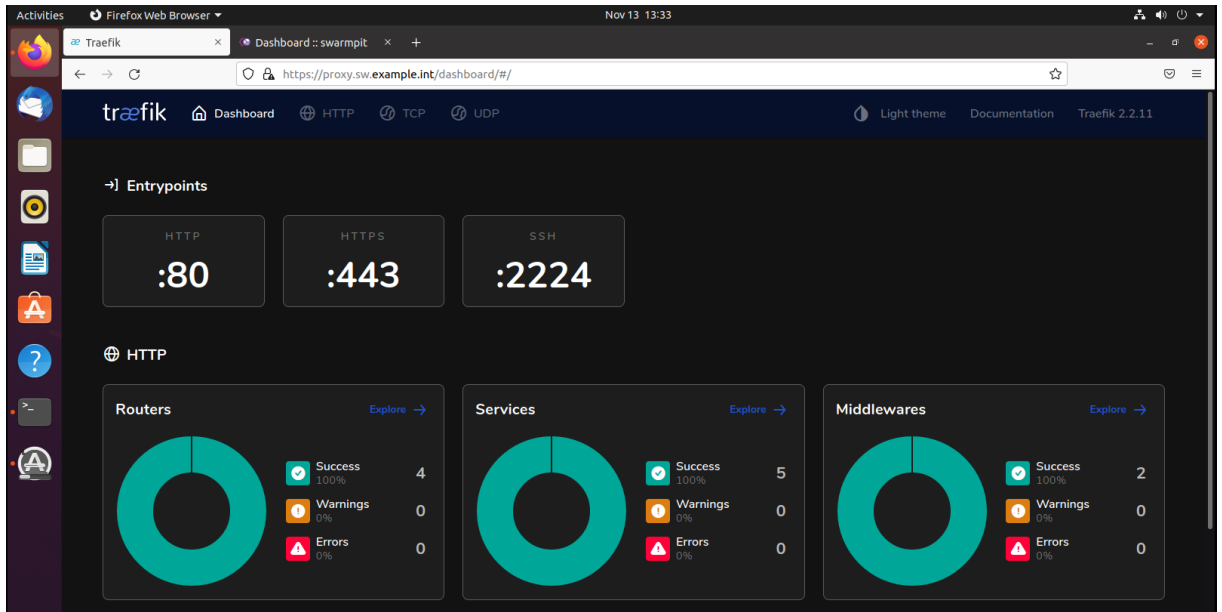
```
root@docker01:/srv/traefik# docker node update --label-add swarmpit.db-data=true
$NODE_ID
q6u0fzpzntol4icf6fc53p72sk
root@docker01:/srv/traefik# docker node update --label-add swarmpit.influx-data=true
$NODE_ID
```

Произвести разворачивание `swarmpit`

```
docker stack deploy -c swarmpit.yml swarmpit
```

После этого можно проверять доступность веб интерфейсов развернутых сервисов.

Traefik:



Swarmpit:

