Container Tutorials

Docker Daemon Configuration

Docker daemon options can also be controlled using files such as /etc/sysconfig/docker or /etc/default/docker on Ubuntu. Also, note that Docker in daemon mode can be identified as having '-d' as the argument to docker service.

Restrict network traffic between containers

By default, unrestricted network traffic is enabled between all containers on the same host. Thus, each container has the potential of reading all packets across the container network on the same host. This might lead to unintended and unwanted disclosure of information to other containers. Hence, restrict the inter container communication.

Create a nginx container;

```
$ docker run -d -p 4915:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
1565e86129b8: Pull complete
a604b236bcde: Pull complete
84cef7523477: Pull complete
fd07be91d48e: Pull complete
fb36adbd81bf: Pull complete
097ca86c0784: Pull complete
8e0912ce0c1b: Pull complete
306fa48b674e: Pull complete
e51a2e0cfd66: Pull complete
8ccac0b28146: Pull complete
20ea32821467: Pull complete
198a73cfd686: Pull complete
Digest: sha256:a3e3180ab6ac4a3095cf6d6225223e10fa5f5ac1b20939a4ba1d777918f63f9f
Status: Downloaded newer image for nginx:latest
39fc070bf79d1668580829a7b0bae125ec7e1812fae229246466358e29002579


$ curl http://localhost:4915
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
```

```
<p><em>Thank you for using nginx.</em></p>
</body>
</html>

$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATU
39fc070bf79d        nginx               "nginx -g 'daemon off"   About a minute ago   Up Ab
```

Get the IP adress of nginx container.

```
$ docker inspect --format '{{  .NetworkSettings.IPAddress }}' 39f
172.17.0.2
```

Create a new centos container and try to ping the nginx container from it;

```
$ docker run -it centos bash
Unable to find image 'centos:latest' locally
latest: Pulling from library/centos
fa5be2806d4c: Pull complete
0cd86ce0a197: Pull complete
e9407f1d4b65: Pull complete
c9853740aa05: Pull complete
e9fa5d3a0d0e: Pull complete
Digest: sha256:c96eeb93f2590858b9e1396e808d817fa0ba4076c68b59395445cb957b524408
Status: Downloaded newer image for centos:latest
[root@df006460c9d1 /]# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.147 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.084 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.109 ms
^C
--- 172.17.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.084/0.113/0.147/0.027 ms
[root@df006460c9d1 /]# exit
exit
```

Now we will disbale the inter-communication between the containers by setting the –icc flag as false.

```
$ service docker stop

$ docker -d -icc=false &
[1] 12482
Warning: '-d' is deprecated, it will be removed soon. See usage.
Warning: '-icc' is deprecated, it will be replaced by '--icc' soon. See usage.
WARN[0000] please use 'docker daemon' instead.
INFO[0000] API listen on /var/run/docker.sock
INFO[0001] [graphdriver] using prior storage driver "aufs"
INFO[0003] Firewalld running: false
INFO[0003] Default bridge (docker0) is assigned with an IP address 172.17.0.1/16. Daemon op
WARN[0003] Your kernel does not support swap memory limit.
INFO[0004] Loading containers: start.
......
INFO[0004] Loading containers: done.
INFO[0004] Daemon has completed initialization
INFO[0004] Docker daemon                                 commit=a34a1d5 execdriver=native-0

$ docker ps
INFO[0017] GET /v1.21/containers/json
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
```

We will create both the containers again and we can check that they are not pingable.

```
$ docker run -d -p 4915:80 nginx
INFO[0088] POST /v1.21/containers/create
34e34a5354977206b358e305c08a152f5146b6126a94175639e35579f6a68571
INFO[0088] POST /v1.21/containers/34e34a5354977206b358e305c08a152f5146b6126a94175639e35579f
INFO[0090] No non-localhost DNS nameservers are left in resolv.conf. Using default external
INFO[0090] IPv6 enabled; Adding default IPv6 external servers : [nameserver 2001:4860:4860:

$ docker ps
INFO[0102] GET /v1.21/containers/json
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
34e34a535497        nginx               "nginx -g 'daemon off"   14 seconds ago      Up 10

$ docker inspect --format '{{  .NetworkSettings.IPAddress }}' 34e
INFO[0134] GET /v1.21/containers/34e/json
172.17.0.2

$ docker run -it centos bashINFO[0166] POST /v1.21/containers/create
INFO[0167] POST /v1.21/containers/08c82f15168a846037f1aa4ef2d0f16cb8757fa2e1754f07e2155a250
INFO[0167] POST /v1.21/containers/08c82f15168a846037f1aa4ef2d0f16cb8757fa2e1754f07e2155a250
INFO[0167] No non-localhost DNS nameservers are left in resolv.conf. Using default external
INFO[0167] IPv6 enabled; Adding default IPv6 external servers : [nameserver 2001:4860:4860:
INFO[0169] POST /v1.21/containers/08c82f15168a846037f1aa4ef2d0f16cb8757fa2e1754f07e2155a250
[root@08c82f15168a /]# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
^C
--- 172.17.0.2 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3024ms
```

Do not bind Docker to another IP/Port or a Unix socket

By default, Docker daemon binds to a non-networked Unix socket and runs with ‘root’ privileges. If you change the default docker daemon binding to a TCP port or any other Unix socket, anyone with access to that port or socket can have full access to Docker daemon and in turn to the host system. Hence, you should not bind the Docker daemon to another IP/Port or a Unix socket.

Stop the docker service and then try to run it on different port which can be vulnerable.

```
$ docker -H tcp://0.0.0.0:2375 -H unix:///var/run/example.sock -d
Warning: '-d' is deprecated, it will be removed soon. See usage.
WARN[0000] please use 'docker daemon' instead.
WARN[0000] /!\ DON'T BIND ON ANY IP ADDRESS WITHOUT setting -tlsverify IF YOU DON'T KNOW WH
INFO[0000] API listen on [::]:2375
INFO[0000] [graphdriver] using prior storage driver "aufs"
INFO[0000] API listen on /var/run/example.sock
INFO[0000] Firewalld running: false
INFO[0000] Default bridge (docker0) is assigned with an IP address 172.17.0.1/16. Daemon op
WARN[0000] Your kernel does not support swap memory limit.
INFO[0001] Loading containers: start.
.....
INFO[0001] Loading containers: done.
INFO[0001] Daemon has completed initialization
INFO[0001] Docker daemon                                 commit=a34a1d5 execdriver=native-0
```

In new terminal check that -H option is present which should not exist during the default run of the docker service. After this tutorial, restart the default docker service.

```
$ ps -ef | grep docker
```