



Arbeta med Docker Swarm på Raspberry Pi Model 2 del 1

Förutsättningar: 3 st Raspberry Pi Model 2. Konfigurerade med värdnamn med fungerande nätverkskommunikation sinsemellan. Docker version 1.5 installerat på alla enheterna.

Mina enheter: dockerrpi1.linuxkurser.nu 192.168.0.50 Swarmkluster
 dockerrpi2.linuxkurser.nu 192.168.0.51 Nod1
 dockerrpi3.linuxkurser.nu 192.168.0.52 Nod2

Arbetsuppgift 1: Ändra konfiguration av Docker, så att Docker även lyssnar på TCP. Detta måste göras på samtliga enheter som skall ingå i Docker swarm.

Steg 1: Skriv in följande kommando: **sudo su**, klicka därefter på Enter.

Stäng ner tjänsten Docker.

Steg 2: Skriv in följande kommando: **service docker stop**, klicka därefter på Enter.

Editera `/etc/default/docker`.

Steg 3: Skriv in följande kommando: **nano /etc/default/docker**, klicka därefter på Enter.

Steg 4: Ändra raden: `DOCKER_OPTS="--storage-driver=overlay -D"` till **`DOCKER_OPTS="--storage-driver=overlay -D -H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock"`**, klicka på `ctrl+o` för att spara dina förändringar, klicka på Enter för att bekräfta filnamnet. Klicka på `ctrl+x` för att avsluta nano.

Starta tjänst för Docker.



Steg 5: Skriv in följande kommando: **service docker restart**, klicka därefter på Enter.

Verifiera att daemon körs.

Steg 6: Skriv in följande kommando: **docker info**, klicka därefter på Enter.

```
$ docker info
Containers: 2
Images: 30
Storage Driver: overlay
  Backing Filesystem: extfs
Execution Driver: native-0.2
Kernel Version: 3.18.8-hypriotos-v7+
Operating System: Raspbian GNU/Linux 7 (wheezy)
CPUs: 4
Total Memory: 925.3 MiB
Name: dockerrpi1.linuxkurser.nu
ID: ZWDA:ZS2X:SVDK:D7JN:V52E:CPC4:ZVEE:ILXN:GVEE:JNTL:UJKF:BXZQ
Debug mode (server): true
Debug mode (client): false
Fds: 27
Goroutines: 27
EventsListeners: 1
Init SHA1: 6f77311608d545807a397d65468a964df6a37519
Init Path: /usr/lib/docker/dockerinit
Docker Root Dir: /var/lib/docker
```

Steg 7: Skriv in följande kommando: **docker -H 127.0.0.1:2375**

info, klicka därefter på Enter. Skall ge ett liknande resultat.

Arbetsuppgift 2: Hämta container med Docker swarm. Detta måste göras på samtliga enheter som skall ingå i Docker swarm.

Steg 1: Skriv in följande kommando: **docker pull nimblestratus/rpi-swarm**, klicka därefter på Enter.

Arbetsuppgift 3: Starta kluster, detta görs bara på en maskin (Swarmkluster).

Steg 1: Skriv in följande kommando: **docker run --rm nimblestratus/rpi-swarm create**, klicka därefter på Enter.

```
Hypriotos: root@dockerrpi1.linuxkurser.nu in /home/pi
$ docker run --rm nimblestratus/rpi-swarm create
bea7f0bb90dbf8e71197ffba224ba60d
Hypriotos: root@dockerrpi1.linuxkurser.nu in /home/pi
$
```

Steg 2: Notera ID här: _____

Mitt exempel: bea7f0bb90dbf8e71197ffba224ba60d

Steg 3: Notera din IP-adress här: _____

Arbetsuppgift 4: Starta första nod (Nod1).

Steg 1: Skriv in följande kommando: **docker run -d nimblestratus/rpi-swarm join --addr=IP_adress_för_din_enhet:2375 token://id_för_kluster**, klicka därefter på Enter.



Exempelvis:

```
docker run -d nimblestratus/rpi-swarm join --  
addr=192.168.0.51:2375 token://  
bea7f0bb90dbf8e71197ffbba224ba60d
```

```
HyprIoTOS: root@black-pearl in /home/pi  
$ docker run -d nimblestratus/rpi-swarm join --addr=192.168.0.51:2375 token://be  
a7f0bb90dbf8e71197ffbba224ba60d  
3e05818e2644123f8fec527741fdbcb7379b5615cb6dba802d53e7623d7312a  
HyprIoTOS: root@black-pearl in /home/pi  
$
```

Verifiera att agent har startat.

Steg 2: Skriv in följande kommando: **docker ps**, klicka därefter på Enter.

```
HyprIoTOS: root@black-pearl in /home/pi  
$ docker ps  
CONTAINER ID          IMAGE                                COMMAND                  CREA  
TED                   STATUS                               PORTS                   NAMES  
3e05818e2644         nimblestratus/rpi-swarm:latest     "swarm join --addr=1   52 s  
econds ago           Up 51 seconds                       2375/tcp                determined_shockley  
HyprIoTOS: root@black-pearl in /home/pi  
$
```

Arbetsuppgift 5: Starta swarm manager (på Swarmkluster).

Steg 1: Skriv in följande kommando: **docker run -d -p 3456:2375 nimblestratus/rpi-swarm manage token:// id_för_kluster**, klicka därefter på Enter.

Exempelvis:

```
docker run -d -p 3456:2375 nimblestratus/rpi-swarm  
manage token:// bea7f0bb90dbf8e71197ffbba224ba60d
```

```
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~  
$ docker run -d -p 3456:2375 nimblestratus/rpi-swarm manage token://bea7f0bb90db  
f8e71197ffbba224ba60d  
2cf789d0c1cd400ec4a3a2ad853b1036ab74899921b99e56664a88a7b11f5bf0  
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~  
$
```



Lista noder

Steg 2: Skriv in följande kommando: **docker run --rm nimblestratus/rpi-swarm list token:// id_för_kluster**, klicka därefter på Enter.

Exempelvis:

```
docker run --rm nimblestratus/rpi-swarm list token://  
bea7f0bb90dbf8e71197ffba224ba60d
```

Vi har bara en nod igång, detta rapporteras av vårt system.

```
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~  
$ docker run --rm nimblestratus/rpi-swarm list token://bea7f0bb90dbf8e71197ffba2  
24ba60d  
192.168.0.51:2375  
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~  
$
```

Arbetsuppgift 6: Starta andra nod (Nod2).

Steg 1: Skriv in följande kommando: **docker run -d nimblestratus/rpi-swarm join -- addr=IP_adress_för_din_enhet:2375 token://id_för_kluster**, klicka därefter på Enter.

Exempelvis:

```
docker run -d nimblestratus/rpi-swarm join --  
addr=192.168.0.52:2375 token://  
bea7f0bb90dbf8e71197ffba224ba60d
```

```
HyprIoTOS: root@dockerrpi3.linuxkurser.nu in /home/pi  
$ docker run -d nimblestratus/rpi-swarm join --addr=192.168.0.52:2375 token://be  
a7f0bb90dbf8e71197ffba224ba60d  
44d917e5d75c40c90e1d5f23407e9a077a22d2431162c3328b30033c4666b4c4  
HyprIoTOS: root@dockerrpi3.linuxkurser.nu in /home/pi  
$
```

Arbetsuppgift 7: Lista noder på Swarmkluster .

I mitt fall, på dockerrpi1.linuxkurser.nu.



Steg 1: Skriv in följande kommando: **docker run --rm nimblestratus/rpi-swarm list token:// id_för_kluster**, klicka därefter på Enter.

Exempelvis:

```
docker run --rm nimblestratus/rpi-swarm list token://  
bea7f0bb90dbf8e71197ffba224ba60d
```

Nu har vi två noder igång, detta rapporteras av vårt system.

```
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~  
$ docker run --rm nimblestratus/rpi-swarm list token://bea7f0bb90dbf8e71197ffba2  
24ba60d  
192.168.0.52:2375  
192.168.0.51:2375  
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~  
$ █
```

Arbetsuppgift 8: Lista information om ingående noder i Swarmkluster.

Steg 1: Skriv in följande kommando: **docker -H tcp://localhost:3456 info**, klicka därefter på Enter.

```
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~  
$ docker -H tcp://localhost:3456 info  
Containers: 2  
Nodes: 2  
  dockerrpi2.linuxkurser.nu: 192.168.0.51:2375  
    â Containers: 1  
    â Reserved CPUs: 0 / 4  
    â Reserved Memory: 0 B / 925.3 MiB  
  dockerrpi3.linuxkurser.nu: 192.168.0.52:2375  
    â Containers: 1  
    â Reserved CPUs: 0 / 4  
    â Reserved Memory: 0 B / 925.3 MiB  
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~  
$ █
```

Information om dina två ingående noder visas.

Arbetsuppgift 9: Lista information om containers som körs på ingående noder i Swarmklustret.

Steg 1: Skriv in följande kommando: **docker -H tcp://localhost:3456 ps**, klicka därefter på Enter.



```
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~
$ docker -H tcp://localhost:3456 ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
44d917e5d75c       nimblestratus/rpi-swarm:latest   "swarm join --addr=1  17 minutes ago     Up 17 minutes      2375/tcp            dockerrpi3.linuxkurser.n
x/sick_nobel
3e05818e2644       nimblestratus/rpi-swarm:latest   "swarm join --addr=1  10 hours ago       Up 10 hours         2375/tcp            dockerrpi2.linuxkurser.n
u/determined_shockley
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~
$
```

Arbetsuppgift 10: Starta container som innehåller busybox och webbserver.

Steg 1: Skriv in följande kommando: **docker -H tcp://127.0.0.1:3456 run -d -p 8080:80 hypriot/rpi-busybox-httpd**, klicka därefter på Enter.

```
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~
$ docker -H tcp://127.0.0.1:3456 run -d -p 8080:80 hypriot/rpi-busybox-httpd
fba759164650e27d6bfff6bcd6fb63ce1f0f1b29d3030d4b62d89a6c39461c8b8f
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~
$
```

Kontrollera att container körs.

Steg 2: Skriv in följande kommando: **docker -H tcp://localhost:3456 ps**, klicka därefter på Enter.

```
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~
$ docker -H tcp://localhost:3456 ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
fba759164650       hypriot/rpi-busybox-httpd:0.1.0   "/bin/busybox httpd  3 minutes ago       Up 2 minutes       192.168.0.52:8080->80/tcp   dockerrpi3.linu
xkurser.nu/naughty_leakey
44d917e5d75c       nimblestratus/rpi-swarm:latest   "swarm join --addr=1  24 minutes ago     Up 24 minutes      2375/tcp            dockerrpi3.linu
xkurser.nu/sick_nobel
3e05818e2644       nimblestratus/rpi-swarm:latest   "swarm join --addr=1  11 hours ago       Up 11 hours         2375/tcp            dockerrpi2.linu
xkurser.nu/determined_shockley
HyprIoTOS: pi@dockerrpi1.linuxkurser.nu in ~
$
```

Testa webbserver.

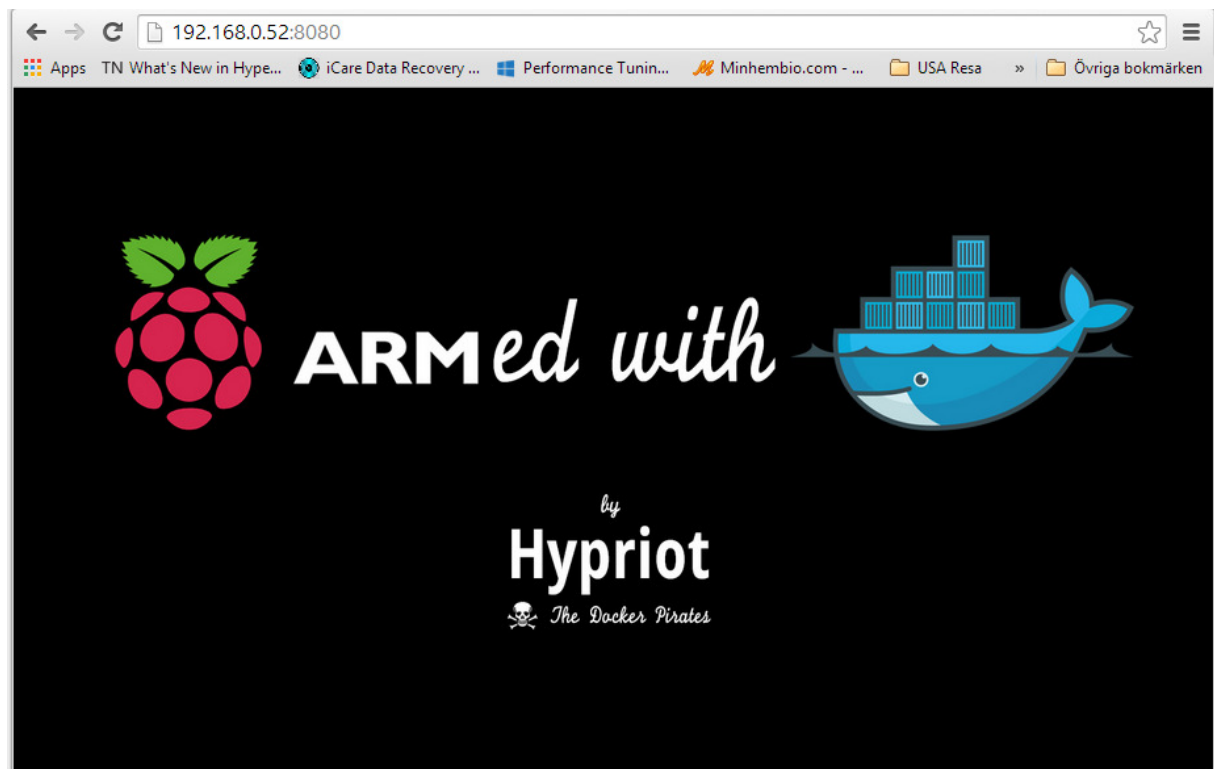
Steg 3: Kontrollera i resultatet av Steg 2, vilken IP-adress som används av container **hypriot/rpi-busybox-httpd**.

Ange adress och port här: _____.

Exempelvis: 192.168.0.52 port 8080.

Steg 4: Öppna webbläsare på klient som är ansluten på samma nätverks som dina Raspberry Pi Model 2.

Steg 5: Skriv in url:en enligt Steg 3. I mitt fall: <http://192.168.0.52:8080>, klicka därefter på Enter.



Du har slutfört del 1 av artikelserien om Docker Swarm!