

## Setup

```
docker rm $(docker ps -aq)
docker network rm my_net
```

## Demo

- Install and activate

```
yum -y install docker
systemctl start docker
systemctl enable docker
```

```
docker images
```

- Launch pre-canned container: hello-world (busybox is another good test)

```
(optional:)
docker pull hello-world
docker run -i -t hello-world
```

- Show docker-run options

```
man docker-run
```

- Launch pre-canned container: ubuntu, debian, fedora

```
docker run -it ubuntu
cat /etc/os-release
uname -a
exit
```

```
docker run -it debian
cat /etc/os-release
uname -a
exit
```

```
docker run -it fedora:25 /bin/bash
cat /etc/os-release
exit
```

- Other commands include stop/start/restart, pause/unpause

- Cleaning up

```
docker ps
docker ps -a
docker ps -aq
docker rm $(docker ps -aq)
```

- Finding images

```
docker search fedora
docker search --filter stars=200 ubuntu
docker search httpd
```

- Launch Apache

```
docker run -d -p 8081:80 --name my_server httpd
  wget -q0- http://127.0.0.1:8081/
```

```
docker logs my_server
docker exec -it my_server /bin/bash
  ps -fe
```

```
(in container) Edit /usr/local/apache2/htdocs/index.html:
  apt-get -y update
  apt-get -y install zile
  zile /usr/local/apache2/htdocs/index.html
  exit
```

```
docker ps
docker commit my_server my_new_image
docker kill my_server
docker ps
docker ps -a
docker rm my_server
docker run -d -p 8081:80 --name my_server my_new_image
  wget -q0- http://127.0.0.1:8081/
docker kill my_server
```

- Make modifications to Apache using Dockerfile

```
# For illustration, delete old image, or edit Dockerfile to invalidate cache
docker rmi psfales/apache
```

```
cd DOCKER/psfales_apache
cat index.html
cat Dockerfile
docker build -t psfales/apache .
docker rm $(docker ps -qa)
```

- Demonstrate caching by changing index.html

```
zile index.html
docker build -t psfales/apache .
docker run -d -p 8081:80 --name my_server psfales/apache
docker kill my_server
```

- Push to Docker Hub

```
(docker login needed the first time)
docker push psfales/apache
```

- Networking

```
docker network create my_net
docker network ls
docker network inspect my_net
```

```
docker run -d --net=my_net --name webserver1 httpd
docker run -d --net=my_net --name webserver2 httpd
```

```
docker run -it --net=my_net psfales/nettest
  nslookup webserver1
  ping -c3 webserver1
  wget -q0- http://webserver1/
  wget -q0- http://webserver2/
```

OR

```
docker run -it --net=my_net ubuntu
  apt-get -y update
  apt-get -y install net-tools iputils-ping dnsutils wget
  nslookup webserver1
  ping -c3 webserver1
  wget -q0- http://webserver1/
  wget -q0- http://webserver2/
```

```
docker network disconnect my_net webserver2
```

```
ping webserver2
```

- Overlay network (Skip for now - revisit to look at commands if time permits)  
<https://docs.docker.com/network/network-tutorial-overlay/#walkthrough>

Host 1

```
docker run -d -p "8500:8500" -h "consul" progrium/consul
-server -bootstrap
```

```
docker swarm init
OR (if host has more than one interface)
docker swarm init --advertise-addr=<IP-ADDRESS-OF-MANAGER>
```

```
# List nodes (can only be done on manager)
docker node ls
```

```
docker network create -d overlay --attachable my_overlay
docker run -d --net=my_overlay -p 8081:80 --name webserver1 httpd
```

```
docker run -it --net=my_overlay --name net1 psfales/nettest
```

Host 2

```
(use docker swarm join printed by host 1. Use
-advertise-addr <IP-ADDRESS-OF-WORKER-1> if more than
one interface)
```

```
docker run -d --net=my_overlay -p 8082:80 --name webserver2 httpd
```

```
docker run -it --net=my_overlay --name net2 psfales/nettest
```

```
apt-get -y update
apt-get -y install net-tools iputils-ping dnsutils wget
nslookup webserver1
ping -c3 webserver1
wget -q0- http://webserver1/
wget -q0- http://webserver2/
```

- Volumes

```
# clean out volume
rm -rf /tmp/volume
```

```
docker run -it -v /tmp/volume:/datadir ubuntu
```

- Show build.sh and Dockerfile for rpi-build

- based on another image that includes support files, toolchain, etc.

```
cd ../psfales_rpi-build
```

```
cat Dockerfile
vi build.sh
```

```
docker build -t psfales/rpi-build .
docker run -it -v /home/psfales/kernel:/kern --env KERN_BRANCH=rpi-4.11.y
psfales/rpi-build
```

- R-Studio

```
docker run -d -p 8787:8787 --name rstudio rocker/rstudio
login: rstudio, password: rstudio
```

```
(https://www.harding.edu/fmccown/r/)
# Define the cars vector with 5 values
cars <- c(1, 3, 6, 4, 9)
# Graph cars using blue points overlaid by a line
plot(cars, type="o", col="blue")
# Create a title with a red, bold/italic font
title(main="Autos", col.main="red", font.main=4)
```

```
docker kill rstudio
```