Docker

Docker is Software Containerization.
- Similar in concept to the old VMs but different in reality.

Docker – builds containers.
Docker-Machine – creates container run environments on hypervisors
Docker-Compose – combines several docker containers into one “grouping”

Basic Configuration

Install

On host (MAC)
- Install Docker Toolbox
  - https://www.docker.com/community-edition
- Install VirtualBox
  - https://www.virtualbox.org/wiki/Downloads

Busy Box

How to use this image

Run BusyBox shell

```
$ docker run -it --rm busybox
```

This will drop you into an sh shell to allow you to do what you want inside a BusyBox system.

Create dockerfile.df for a binary

```
FROM busybox

COPY ./my-static-binary /my-static-binary

CMD ["/my-static-binary"]
```

***B**uild the image*

“docker build -t sbeeker/busybox -f dockerfile.df .”

Python Base

- Create a directory

  “mkdir python_docker”

- Create Dockerfile

  ### Create Dockerfile (vi python_base.df)

  ```
  # Ubuntu docker image with python
  FROM ubuntu:latest
  RUN apt-get update && apt-get -y upgrade && apt-get install -y python
  RUN groupadd -r -g 3000 worker && useradd -rM -g worker -u 3000 worker
  ENV DATADIR="/data" VERSION="1.0"
  #LABEL base.name="Python Script Run" base.version=$VERSION
  WORKDIR $DATADIR
  ENTRYPOINT ["python"]
  #ENTRYPOINT ["/data/python_script.py"]
  ```

Build the image

“docker build -t sbeeker/python_base -f python_base.df .”

Run container

“`docker run --rm -it sbeeker/python_base`”

Python Application (python_app.py)

```
#!/usr/bin/python
  import csv
  import sys
  f = open(sys.argv[1],'rt')
  table = sys.argv[2];
try:reader = csv.reader(f) i = 0;
    for row in reader:
        if i == 0:
            header = row
        else:
            print "INSERT INTO " + table + " (" +
  ",".join(header) + ") VALUES ('" + row[0] + "','" + row[1]+
  "','" + row[2] + "','" + row[3] + "');"
    i += 1
finally:
  f.close()
```

Python Application

Dockerfile (python_app.df)

```
# Ubuntu docker image with python
FROM sbeeker/python_base
RUN apt-get update && apt-get -y upgrade && apt-get install -y python
COPY ["./script_app`.py", "${DATADIR}"]
COPY ["./books.csv", "${DATADIR}"]
RUN chmod +x ${DATADIR}/script.py && chown worker:worker $
{DATADIR}
USER worker:worker
CMD ["books.csv", "books"]
```

B*uild the image*

“docker build -t sbeeker/python_app -f python_app.df .
”

Run container

“docker run --rm -it sbeeker/python_app”

Configure the SWARM

SWARM Master

Terminal 1:

docker-machine create swarm-manager1
docker-machine env swarm-manager1
<run the suggested command>

docker-machine ip swarm-manager1
docker swarm init --advertise-addr <IP Address>

Will get back something like this:

```
  docker swarm join \
    --token SWMTKN-1-0pj8q3ch3baws26wcjptw48a6e1rd0lvkypxugvlnih89a8ooj-
2oysbi0g0r5fhtlmgxe1in7zp \
    192.168.99.101:2377
```

docker node ls

docker-machine active
swarm-manager1

docker node ls # repeat after each swarm join.

SWARM Workers

Terminals [2-N+1]

docker-machine create swarm-worker[N]
docker-machine env swarm-worker[N]
<run the suggested command>

docker-machine ls
docker swarm –help
docker swarm join --help

docker-machine active

// Run suggested command from SWARM init.
docker swarm join --token SWMTKN-1-0pj8q3ch3baws26wcjptw48a6e1rd0lvkypxugvlnih89a8ooj-2oysbi0g0r5fhtlmgxe1in7zp 192.168.99.101:2377

Visualizer

Terminal 1

docker-machine active
swarm-manager1
C:\labs>**docker run -it -d -p 8090:8080 -v /var/run/docker.sock:/var/run/docker.sock dockersamples/visualizer**

docker-machine ip swarm-manager1
192.168.99.101
From the browser, go to <IP-address-of-swarm-manager1>:8090,
http:// 192.168.99.101:8090 in the example below.

Actual Work

Terminal 1

docker-machine active
docker service create --replicas 2 --name croc-hunter -p 8080:8080 --update-delay 5s rgardler/croc-hunter
docker service ls

/repeat this as service starts
docker service ps croc-hunter

docker service inspect croc-hunter

Browser port 8080

docker service inspect croc-hunter --pretty

docker service ps croc-hunter

docker service scale croc-hunter=5

docker service create --replicas 3 --name python_app