

20150831- docker_basics

Michael Pope

michael@dtcorp.com.au



docker

What is Docker

- Easy to use containers
- Containers are like LXC, FreeBSD jails.
- Uses kernel feature 'cgroups' & 'namespaces' to isolate
 - cgroups - CPU/memory
 - namespaces - file system, hostname, users, networking

Advantages of using Docker

- Faster than VMs
- Light on resources
- Portable
- Quick to build
- Share base images
- Easy to build complex systems

Uses

- Development & Staging servers for devs
- Testing (CI software)
- Production

Installation

- Ref: [Docker Installation](#)
- Requires Linux kernel 3.16+
- debian 8

```
sudo apt-get install docker.io
```

- Add yourself to the 'docker' group

```
sudo groupadd docker  
sudo gpasswd -a ${USER} docker
```

Windows & Mac

- Runs in small VM
- Docker is still running within Linux
- Containers are using Linux

Debian 7 install

1. Upgrade kernel from backports

```
sudo apt-get install -t wheezy-backports linux-image  
-amd64
```

2. Reboot into the new kernel

3. Install Docker from script

```
curl -sSL https://get.docker.com/ | sh
```


Run command

- Runs a container
- Downloads if it doesn't exist
- Most complex command (due to amount of options)

Test

- Using debian container (85MB)

```
docker run debian echo "Hello World"
```

- Run in interactive mode

```
docker run -i -t debian /bin/bash
```

-i	interactive
----	-------------

-t	tty attached
----	--------------

- Using the hello world container

```
docker run --rm hello-world
```

--rm	delete the container when exiting
------	-----------------------------------

Images/containers

- Images are made up from many different layers
- Uses UFS the file system (devicemapper, AUFS, btrfs or vfs)
- images turn into containers when using run/create commands
- Running an image adds R/W, IP address, name, etc.

Create an image

- To keep containers create an image from them.
- Create a container

```
docker run -it --name darkstat --hostname darkstat debian bash
```

<code>-name <NAME></code>	Name your container
<code>-hostname <HOSTNAME></code>	Give container a hostname

- Install darkstat as per normal
- Create an image from a container

```
docker commit darkstat test/darkstat
```

Export your image

- Useful for;
 - backups
 - transfer from one server to another
- Save (save format is a tar file)

```
docker save -o test-darkstat.docker test/darksta  
t
```

Load image

- Load

```
docker load -i test-darkstat.do  
cker
```

Dockerfiles

- Build script
- Run commands in order
- Define the base image (debian is a recommend)
- Copy files
- Define an 'entrypoint' script

Basic Example

- Dockerfile

```
FROM debian
```

```
RUN apt-get update && apt-get install -y darkstat && apt-get clean
```

- Build from the Dockerfile (run everytime you change Dockerfile)

```
docker build -t test/darkstat-dockerfile .
```

- Run

```
docker run test/darkstat /usr/bin/darkstat
```


COPY

- Copy a file from the current dir to the container
- Usage

```
COPY myscript.sh /
```

ENTRYPOINT command

- Gives a starting point binary/script
- Usage

```
ENTRYPOINT ["/usr/bin/darkstat"]
```

OR

```
ENTRYPOINT ["/myscript.sh"]
```

Demo

```
cd ~/docker/using_doker/chap3
docker build -t test/cowsay-dockerfile .
docker run test/cowsay-dockerfile "Moo"
```

Volumes

- Mount a host directory inside a container
- Data should not be part of the container
- Bind a directory (stored on the server)

```
-v <container dir>
```

- Link a directory

```
-v <host dir>:<container dir>
```

Demo

1. Bind a directory called '/data' in your container

```
docker run -it --name test -v /data debian /bin/bash  
exit
```

2. Look at where that physically is

```
docker inspect -f {{.Volumes}} test
```

3. Remove container

```
docker ps  
docker rm <container ID>
```

Demo2

- Bind local directory

```
docker run -it -h test -v /home/map7/code:/data debian /bin/bash
touch /data/x
```

Demo3

- Share volumes with other containers

```
docker run -it -h NEW --volumes-from test debian /bin/bash
ls /data
```

Linking containers

- Containers are supposed to do one thing only
- Split your containers for data, database, web app, web server, etc.
- link syntax

```
--link CONTAINER:ALIAS
```

alias is used within the main container.

- link communications happen over Docker internal network

Demo

- Linking redis & redis-cli Run the redis db

```
docker run --name tut-redis -d redis
```

Run another container to talk to it

```
docker run --rm -it --link tut-redis:redis redis /bin/bash
> ping
> set "abc" 123
> get "abc"
> exit
```

(Example from Using Docker book.)

Tag

- tag an image
- latest tag is always called 'latest'
- tag command;

```
docker tag <image ID> <repository>/<name>:<tag>
```

EG

```
docker tag 5902eb16f45c test/darkstat:latest
```

- TIP: Use a git hash as the tag name

What is Docker Hub

- A container repository
- Create & share repositories
- Offer private repositories (at a cost)
- Docker Hub is not open source.

docker-compose

- Store configuration (in YAML)
- Automate all the command line options

Tips

Images

- Try and use official images where possible (official images don't define a user, eg: instead of john/postgresql the official would be postgresql or library/postgresql)

RUN commands

- When installing do installation & cleanup in one line eg;

```
RUN apt-get update && apt-get install darkstat && apt-get clean
```

- Or when compiling

```
RUN curl -SL "https://unix4lyfe.org/darkstat/darkstat-3.0.719.tar.bz2" -o darkstat
.tar.bz2 \
  && tar xvf darkstat.tar.bz2 -C /tmp \
  && rm darkstat.tar.bz2 \
  && cd /tmp/darkstat \
  && ./configure \
  && make install \
  && rm -fr /tmp/darkstat
```

Get IP address

- Find running containers

```
docker ps
```

- Find the IP address of a running container

```
docker inspect --format {{.NetworkSettings.IPAddress}} <container name>
```


Jump into a running container

- Using exec you can jump into a running container (docker 1.5+)

```
docker exec -it <container ID> /bin/bash
```

Cleaning up

- Remove old exited containers

```
docker rm $(docker ps -aq -f status=ex  
ited)
```

Manage multiple containers

- **Percheron** by Ash McKenzie
- Ash gave a talk at Melbourne Ruby last Wednesday
- Programmed in Ruby 2.x
- Graphing, build/create dependencies, version control, etc.
- Uses YAML for config

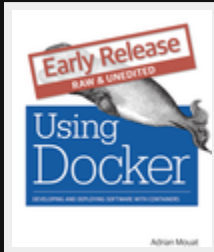
Other uses

Ref: [Docker the Desktop #linuxcon #containercon - InternetNews.](#)

Jessie Frazelle uses it for;

- [Willy Wonka of Containers - Jessie Frazelle - YouTube](#)
- [Spotify container](#)
- Run browsers in separate containers to limit memory
- Run tor
- Everything in a container keeping it nice and tidy.

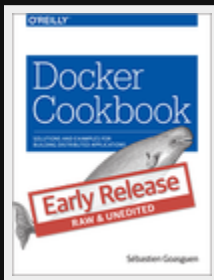
References



- Using Docker by Adrian Mouat



- Get Started with Docker for Linux



- Docker Cookbook by Sébastien Goasguen
- Linux Format Issue 190 & 191 docker tutorials

Questions

Email	map7777@gmail.com
-------	--

Twitter	@map7
---------	--

Github	github: map7
--------	--