

Docker For Everyone

NovaLug
10-Dec-2016
dougtoppin

Agenda

- What is Docker
- Typical Use Cases
- Benefits
- Docker commands
- Docker for Everyone
- How to Dockerize Something
- Where to get containers
- Challenges
- Big Uses of Docker
- Cloud Providers for Docker
- What is a container
- What is an image
- What is a Dockerfile
- Links

What is Docker

- "Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries – anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in." (from docker.com)
- consists of a client and server (engine) that do not have to be located on the same host
- evolving rapidly, first release 2013
- run it natively on Linux, Mac and Windows
- use a container to act on local data (via volumes)
- run a remote container to provide an endpoint to do something

Typical Enterprise Use Cases

- local development environment tools (databases for example)
- with continuous integration (CI) spin up the environment that you need to build/test

Benefits

- do not have to pollute (change or add stuff to your machine) or manage your computer by installing things, particularly multiple versions of things
- do not waste space by installing things and supporting many different environments
- stop worrying about backward compatibility for your tools because they should run forever under docker

Docker commands

- run a container
 - `docker run hello-world`
 - `docker run docker/whalesay cowsay boo`
 - expose/map ports
 - `docker run -P`
 - `docker run -p a:b`
 - use host directory
 - `docker run -v $(pwd):/tmp`
- build an image from a Dockerfile
 - `docker build .`
- ps to see what containers are running
 - `docker ps [-a]`
- images to see what images are local to your machine
 - `docker images [-a]`
- logs to see what it is doing
 - `docker logs 9f9`

Docker for Everyone

- you do not have to be a developer to use Docker
 - if you can use apt-get, yum or apk you can probably make a Dockerfile
- use tools in containers to do things without installing the tool on your machine
- find ready to use docker images at <https://hub.docker.com/>
- examples of popular images <https://hub.docker.com/explore/>
- trusted/enterprise ready images at <https://store.docker.com/>
- run many distributions of linux on your machine
 - `docker run -it --rm ubuntu bash`
 - `docker run -it --rm ubuntu:14.04 bash`
 - `docker run -it --rm centos:latest bash`
 - `docker run -it --rm alpine sh`
- run popular tools
 - postgres : `docker run --name postgres -P -e POSTGRES_PASSWORD=mypassword -d postgres`

- JBoss Infinispan cache
 - : <https://github.com/dougtoppin/docker/tree/master/infinispan/rest>
- Java : https://hub.docker.com/r/_/openjdk/
- ImageMagick : docker pull starefossen/node-imagemagick
 - docker run -it -v \$(pwd):/tmp starefossen/node-imagemagick bash
- IRC
 - https://hub.docker.com/_/irssi/
 - Linux
 - docker run -it --name my-running-irssi -e TERM -u \$(id -u):\$(id -g) -log-driver=none -v \$HOME/.irssi:/home/user/.irssi:ro -v /etc/localtime:/etc/localtime:ro irssi
 - Mac
 - docker run -it --name my-running-irssi -e TERM -u \$(id -u):\$(id -g) -log-driver=none -v \$HOME/.irssi:/home/user/.irssi:ro irssi
- lenticular tool example (old perl tool that was dockerized)
 - \$ docker run -v \$(pwd):/tmp dougtoppin/lenticular infile1.jpg infile2.jpg outfile.jpg
 - this runs the tool and uses the current directory as where to find the files to act on
 - the tool source can be found at <https://github.com/dougtoppin/lenticular>
 - the ready to run docker image can be found at <https://hub.docker.com/r/dougtoppin/lenticular/>
 - note that the docker image was automatically built by dockerhub
 - use the shell 'alias' function to hide the details of invoking the docker run command like this
 - \$ alias lenticular="docker run -v \$(PWD):/tmp --rm dougtoppin/lenticular"
 - \$ lenticular file1.jpg file2.jpg output.jpg
- postgres example
 - an example of a more complicated tool
 - https://docs.docker.com/engine/examples/postgresql_service/

How to Dockerize Something

- figure out what it needs to run (FROM)
- do you need network ports to get to it (EXPOSE)
- does it need to read or write files (VOLUME)
- <https://docs.docker.com/engine/examples/>
- Create a Dockerfile or (docker-compose.yml)
 - helps to take a casual look at the Dockerfile reference <https://docs.docker.com/engine/reference/builder/>
 - contains what to start with, what to add and what to do to run it
 - docker-compose.yml
 - multiple service add-on to a Dockerfile

- allows multiple applications (services aka containers) to be run with a single command
- services can be scaled easily (but code still needs to support them)

Where to get containers (images really)

- dockerhub.com (public)
- store.docker.com (commercial)
- build your own (docker build)
- exchange as archives

Challenges

- being familiar with software development helps
- container and image cache management
- how to find tools that are ready for use
 - trusting a image, stars? pulls?
 - knowing how to use an image is not always included in the description
- lack of documentation for public images
- gui applications may be an issue for non X running machines (like macOS)
 - Xquartz for macOS
 - browser based applications will still work (note Eclipse Che)

Big Uses of Docker

- clusters of docker engines to facilitate distributing containers
 - open shift
 - <https://www.openshift.org/> (community)
 - run your own <https://blog.switchbit.io/openshift-cluster-up-with-docker-for-mac/>
 - <https://www.openshift.com/> (commercial)
 - docker datacenter
 - <https://www.docker.com/products/docker-datacenter>
 - Docker AWS Quickstart
 - docker cloud
 - <https://cloud.docker.com/>
 - clustering technologies
 - swarm
 - <https://docs.docker.com/engine/swarm/>
 - kubernetes
 - <http://kubernetes.io/>

Cloud Providers for Docker

- Docker Cloud
- AWS ECS
- RedHat OpenShift v3
- Microsoft Azure
- Google Compute Engine
- Digital Ocean

What is a container

- tarball (of a filesystem)
- `docker export -o container.tar containerId`
- lifecycle

What is an image

- tarball (of layers, which are tarballs)
- `docker save -o image.tar imageId`
- contents are a filesystem and metadata

Links

- <https://www.docker.com/>
- [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))
- <https://hub.docker.com/>
- <https://cloud.docker.com>
- <https://aws.amazon.com/ecs/>
- <https://www.openshift.com/container-platform/>
- <https://github.com/dougtoppin/lenticular>
- <https://hub.docker.com/r/dougtoppin/lenticular/>
- <http://kubernetes.io/>
- <https://www.docker.com/products/docker-datacenter>

