

Docker

containers + filesystem overlays

Container layer

- Docker relies on libcontainer (previously LXC) for process level containerization
- Probably spiritually most similar Solaris zones but also to a lesser degree BSD jails or chroot in Linux
- Process and fs based containment in contrast to full hardware virtualization and hence much less overhead than full virtual machines

Overlay file system

- A union filesystem allows for a merged view of one or more filesystem layers. Allows for composing layers derived separate build steps (or from a base image) to contribute to a single unified filesystem view.
- In terms of Docker a single Docker command may correspond to a single filesystem layer which enables to Docker to fairly intelligently cache Docker build steps.

Ecosystem

- Technology layer (process and fs containerization, union fs / AUFS)
- Images
- Registries (Public, Private)

What does it provide?

- Predictable shared nothing builds
- High parity from development to production environments
- Trivial access to alternative environments (OS)
- Application delivery options (Google Cloud Engine, AWS ECS etc, Self)
- A description format for grouping related applications into a larger service definition
- A controlled approach to managing the application dependency lifecycle

Hello world

```
$ docker run ubuntu:14.04 /bin/echo 'Hello world'  
Unable to find image 'ubuntu:14.04' locally  
14.04: Pulling from library/ubuntu  
0bf056161913: Pull complete  
1796d1c62d0c: Pull complete  
e24428725dd6: Pull complete  
89d5d8e8bafb: Pull complete  
Digest: sha256:d3b59c1d15c3cfb58d9f2eaab8a232f21fc670c67c11f582bc48fb32df17f3b3  
Status: Downloaded newer image for ubuntu:14.04  
Hello world
```

Notice the first run requires fetching layer dependencies

```
$ docker run ubuntu:14.04 /bin/echo 'Hello world'  
Hello world
```

and subsequent runs take advantaged of cached layers

Commands

\$ docker build

Produces an Docker image from a Dockerfile image definition

```
FROM ubuntu:14.04

RUN apt-get install ...
ADD my_app_files /my_app

CMD ["/my_app/start.sh"]
```

```
$ docker images
REPOSITORY          TAG                IMAGE ID           CREATED            VIRTUAL SIZE
label-service       latest            3ed388e94d39      22 hours ago      1.22 GB
<none>              <none>            304473645cda      22 hours ago      969.7 MB
<none>              <none>            12b6f4ae3cda      23 hours ago      1.099 GB
<none>              <none>            9953fd902caa      4 days ago        1.099 GB
<none>              <none>            bb5bd067a2eb      4 days ago        1.099 GB
ubuntu              14.04             89d5d8e8bafb      12 days ago       187.9 MB
golang              latest            08ff0c215f8f      2 weeks ago       703.8 MB
```


\$ docker ps

List the running docker containers

```
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
eee159342182  label-service  "/bin/sh -c 'sudo ser"  27 minutes ago  Up 27 mi
```

\$ docker run

run starts a new container by container tag or id and executes the provided command. The container is shut down once the command exits.

```
# ex.  
$ docker run -i -t IMAGE-NAME /bin/bash  
  
$ docker run ubuntu:14.04 /bin/uname -a  
Linux 889b5021f65f 4.1.13-boot2docker #1 SMP Fri Nov 20 19:05:50 UTC 2015 x86_64 x86_64 x86_64
```

\$ docker exec

exec runs a new command in an existing container. Useful for inspecting or debugging the state of running containers. In contrast with run, exec spawns a new process in the container (I think) and as long as the containers entrypoint command is running the container will continue to run.

```
$ docker exec -i -t CONTAINER-ID /bin/bash
```

\$ docker logs

Fetch any output generated to stdout/err by the container process.

```
$ docker logs modest_liskov | tail
[2015-12-20T07:41:00Z: INFO] GET 304 /api/labels/v1/swagger-ui/lib/marked.js 192.168.99.1 163
[2015-12-20T07:41:00Z: INFO] GET 304 /api/labels/v1/swagger-ui/lib/highlight.7.3.pack.js 192.1
[2015-12-20T07:41:00Z: INFO] GET 304 /api/labels/v1/swagger-ui/lib/swagger-oauth.js 192.168.9
[2015-12-20T07:41:00Z: INFO] GET 304 /api/labels/v1/swagger-ui/css/print.css 192.168.99.1 126
[2015-12-20T07:41:00Z: INFO] GET 304 /api/labels/v1/swagger-ui/images/logomark.png 192.168.99
[2015-12-20T07:41:00Z: INFO] GET 304 /api/labels/v1/swagger-ui/fonts/DroidSans-Bold.ttf 192.1
[2015-12-20T07:41:00Z: INFO] GET 304 /api/labels/v1/swagger-ui/fonts/DroidSans.ttf 192.168.99
[2015-12-20T07:41:00Z: INFO] GET 200 /api/labels/v1/docs 192.168.99.1 1.690528ms 5697b
[2015-12-20T07:41:00Z: INFO] GET 200 /api/labels/v1/swagger-ui/images/favicon-16x16.png 192.1
[2015-12-20T07:41:00Z: INFO] GET 200 /api/labels/v1/swagger-ui/images/favicon-32x32.png 192.1
```

\$ docker attach

attach attaches your terminal to a running instance (i.e. the process indicated by CMD) and shows the output of the currently running entrypoint command and provides a method for interacting with the running process.

```
docker attach CONTAINER-ID
```

Think `screen -r` OR `tmux attach`

Dockerfile

A simple description format for creating a new container image. Most often used commands are

- ADD
- CMD
- EXPOSE
- FROM
- RUN

```
FROM ubuntu:14.04

RUN apt-get install ...
ADD my_app_files /my_app

CMD ["/my_app/start.sh"]
```

Getting started

- Install docker (<https://www.docker.com/docker-toolbox>)
- Start a docker terminal (or add the Docker env vars to your shell configs)
- Generate a Dockerfile
- `build` and `run`
- Label service example

.dockerignore

```
$ cat .dockerignore  
godeps  
.git
```


Dockerfile

```
FROM golang
MAINTAINER Paul Howe "paul@socialcode.com"
ENV DEBIAN_FRONTEND noninteractive

RUN apt-get update -qq
RUN apt-get install -y build-essential
RUN apt-get install -y sudo
RUN apt-get install -y telnet
RUN apt-get install -y mysql-server
RUN apt-get install -y golang
RUN apt-get install -y python
RUN apt-get install -y python-setuptools
RUN apt-get install -y build-essential
RUN apt-get install -y git-core
RUN apt-get install -y libncurses5-dev

# Setup the app dir structure
ENV APP_HOME /app
RUN mkdir $APP_HOME
WORKDIR $APP_HOME
ADD . $APP_HOME

EXPOSE 1234
CMD sudo service mysql start && \
  echo "create database label_service" | mysql -uroot && \
  cd /app && \
  make apply_migrations && \
  make devreqs && \
  make server
```

make docker

...

docker:

```
echo "building a new $(PROJECT) container"  
docker build -t $(PROJECT) .  
echo "starting container"  
docker run -d -p 1234:1234 $(PROJECT)  
open "http://`docker-machine ip default`:1234/api/docs"
```

...

make docker

```
$ make docker
echo label-service
label-service
echo "building a new label-service container"
building a new label-service container
docker build -t label-service .
Sending build context to Docker daemon 14.24 MB
Step 1 : FROM golang
----> 08ff0c215f8f
Step 2 : MAINTAINER Paul Howe "paul@socialcode.com"
----> Using cache
----> 8b5571315361
Step 3 : ENV DEBIAN_FRONTEND noninteractive
----> Using cache
----> 631d8df8ce0a
Step 4 : RUN apt-get update -qq
----> Using cache
----> ab75d889cd16
Step 5 : RUN apt-get install -y build-essential
----> Using cache
----> 0a3550841469
Step 6 : RUN apt-get install -y sudo
----> Using cache
----> 0d247dc315ab
Step 7 : RUN apt-get install -y telnet
----> Using cache
----> 185d35e6101a
Step 8 : RUN apt-get install -y mysql-server
----> Using cache
----> 4a7e64043b1f
Step 9 : RUN apt-get install -y golang
```

make docker

```
----> Using cache
----> d8953de6171b
Step 10 : RUN apt-get install -y python
----> Using cache
----> 6cdf06ce8d56
Step 11 : RUN apt-get install -y python-setuptools
----> Using cache
----> 6c93a3ebf5bc
Step 12 : RUN apt-get install -y build-essential
----> Using cache
----> 19285b1af54a
Step 13 : RUN apt-get install -y git-core
----> Using cache
----> 18a58e2a020b
Step 14 : RUN apt-get install -y libncurses5-dev
----> Using cache
----> a079a1ecdd40
Step 15 : ENV APP_HOME /app
----> Using cache
----> d63a1842890e
Step 16 : RUN mkdir $APP_HOME
----> Using cache
----> e4639b1e984e
Step 17 : WORKDIR $APP_HOME
----> Using cache
----> 058dc3704460
Step 18 : ADD . $APP_HOME
----> 5b4b852a3e8b
Removing intermediate container d488e566c7a0
```

make docker

```
Step 19 : EXPOSE 1234
---> Running in 793b5412b9d3
---> 455e29cb3c80
Removing intermediate container 793b5412b9d3
Step 20 : CMD sudo service mysql start &&      echo "create database label_service" | mysql -u
---> Running in 24ed58ed5acf
---> 12b6f4ae3cda
Removing intermediate container 24ed58ed5acf
Successfully built 12b6f4ae3cda
echo "starting container"
starting container
docker run -d -p 1234:1234 label-service
4685e2f67d38016d2f9b6d000e76016db3255341caa26b300edd89cf1ec30f1a
open "http://`docker-machine ip default`:1234/api/docs"
```

Container output

If we attach or check the logs we should see

```
Starting MySQL database server: mysqld ..  
Checking for tables which need an upgrade, are corrupt or were  
not closed cleanly..  
mkdir -p /go/src/github.com/SocialCodeInc  
ln -snf /app /go/src/github.com/SocialCodeInc/label-service  
mkdir -p env/bin  
go build -o env/bin/migrate_lock ./migrateLock  
env/bin/migrate_lock  
mkdir -p /go/src/github.com/SocialCodeInc  
ln -snf /app /go/src/github.com/SocialCodeInc/label-service  
go get github.com/tools/godep  
go get github.com/davecheney/gcvis  
go get github.com/jstemmer/go-junit-report  
go get github.com/t-yuki/gocover-cobertura  
go run webhead/main.go --settings=config/local_settings.json  
Running server at http://127.0.0.1:1234
```

Container output (kitematic)

The screenshot displays the Kitematic interface for a container named 'loving_khorana' (label-service) which is in a 'RUNNING' state. The interface includes a sidebar with a '+ NEW' button and a 'CONTAINERS' list. The main area is divided into two sections: 'CONTAINER LOGS' and 'IP & PORTS'.

CONTAINER LOGS

```
Starting MySQL database server: mysqld ..
Checking for tables which need an upgrade, are corrupt or were
not closed cleanly..
mkdir -p /go/src/github.com/SocialCodeInc
ln -snf /app /go/src/github.com/SocialCodeInc/label-service
mkdir -p env/bin
go build -o env/bin/migrate_lock ./migrateLock
env/bin/migrate_lock
mkdir -p /go/src/github.com/SocialCodeInc
ln -snf /app /go/src/github.com/SocialCodeInc/label-service
go get github.com/tools/godep
go get github.com/davecheney/gcvis
go get github.com/jstemmer/go-junit-report
go get github.com/t-yuki/gocover-cobertura
go run webhead/main.go --settings=config/local_settings.json
Running server at http://127.0.0.1:1234
[2015-12-20T06:19:57Z] [INFO] GET 404 /api/docs 192.168.99.1 604.471µs 10b
[2015-12-20T06:19:58Z] [INFO] GET 404 /favicon.ico 192.168.99.1 945.448µs 10b
[2015-12-20T06:20:33Z] [INFO] GET 200 /ping 192.168.99.1 5.444µs 4b
```

IP & PORTS

You can access this container using the following IP address and port:

DOCKER PORT	ACCESS URL
1234/tcp	192.168.99.100:1234

The interface also features a 'Home' and 'Settings' navigation bar, a 'LOGIN' button, and a 'DOCKER CLI' button at the bottom left.

Ping

```
$ curl "http://192.168.99.100:1234/ping"  
Pong
```


Search

```
curl --silent "http://192.168.99.100:1234/api/labels/v1/search" | jq '.'  
{  
  "meta": {  
    "limit": 20,  
    "offset": 0  
  },  
  "objects": []  
}
```

API Explorer

The screenshot shows a web browser window with the URL `192.168.99.100:1234/api/labels/v1/swagger-ui/`. The page title is "SocialCode Docs". The main content area displays the API documentation for the "Label Service".

Label Service

label Show/Hide | List Operations | Expand Operations

- GET** /label
- POST** /label
- PUT** /label
- POST** /label/batch
- DELETE** /label/{ID}
- GET** /label/{ID}
- PUT** /label/{ID}

search Show/Hide | List Operations | Expand Operations

[BASE URL: /api/labels/v1 , API VERSION: v1]

Dependencies

What if we wanted to change one of the core app dependencies? Perhaps the database?

Maria

```
diff --git a/Dockerfile b/Dockerfile
index 0ad9d40..02abf7b 100644
--- a/Dockerfile
+++ b/Dockerfile
@@ -6,7 +6,7 @@ RUN apt-get update -qq
  RUN apt-get install -y build-essential
  RUN apt-get install -y sudo
  RUN apt-get install -y telnet
-RUN apt-get install -y mysql-server
+RUN apt-get install -y mariadb-server
  RUN apt-get install -y golang
  RUN apt-get install -y python
  RUN apt-get install -y python-setuptools
```

Maria: make docker

Update our Dockerfile and rebuild

```
$ make docker
```

The build should proceed as normal, re-using a lot of the previously stored layers (sudo, build essential, telnet etc).

When the new dependency is encountered Docker will run that command and cache the results in a new layer with a unique hashed id.

Maria: make docker

```
...snip...
Step 6 : RUN apt-get install -y sudo
----> Using cache
----> 0d247dc315ab
Step 7 : RUN apt-get install -y telnet
----> Using cache
----> 185d35e6101a
Step 8 : RUN apt-get install -y mariadb-server
----> Running in 3f1633364ddd
Reading package lists...
Building dependency tree...
Reading state information...
The following extra packages will be installed:
  libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
  libreadline5 libterm-readkey-perl mariadb-client-10.0
  mariadb-client-core-10.0 mariadb-common mariadb-server-10.0
  mariadb-server-core-10.0 mysql-common psmisc
Suggested packages:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl
  libipc-sharedcache-perl mailx mariadb-test tinyca
The following NEW packages will be installed:
  libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
  libreadline5 libterm-readkey-perl mariadb-client-10.0
  mariadb-client-core-10.0 mariadb-common mariadb-server mariadb-server-10.0
  mariadb-server-core-10.0 mysql-common psmisc
0 upgraded, 15 newly installed, 0 to remove and 2 not upgraded.
Need to get 13.8 MB of archives.
After this operation, 135 MB of additional disk space will be used.
Get:1 http://security.debian.org/ jessie/updates/main mysql-common all 5.5.46-0+deb8u1 [85.5
...
```

Maria

The build completed, now we have to check the app

```
$ curl --silent "http://192.168.99.100:1234/ping"  
Pong
```

/ping still works, thats encouraging. But it is just /ping so lets try a few more requests.

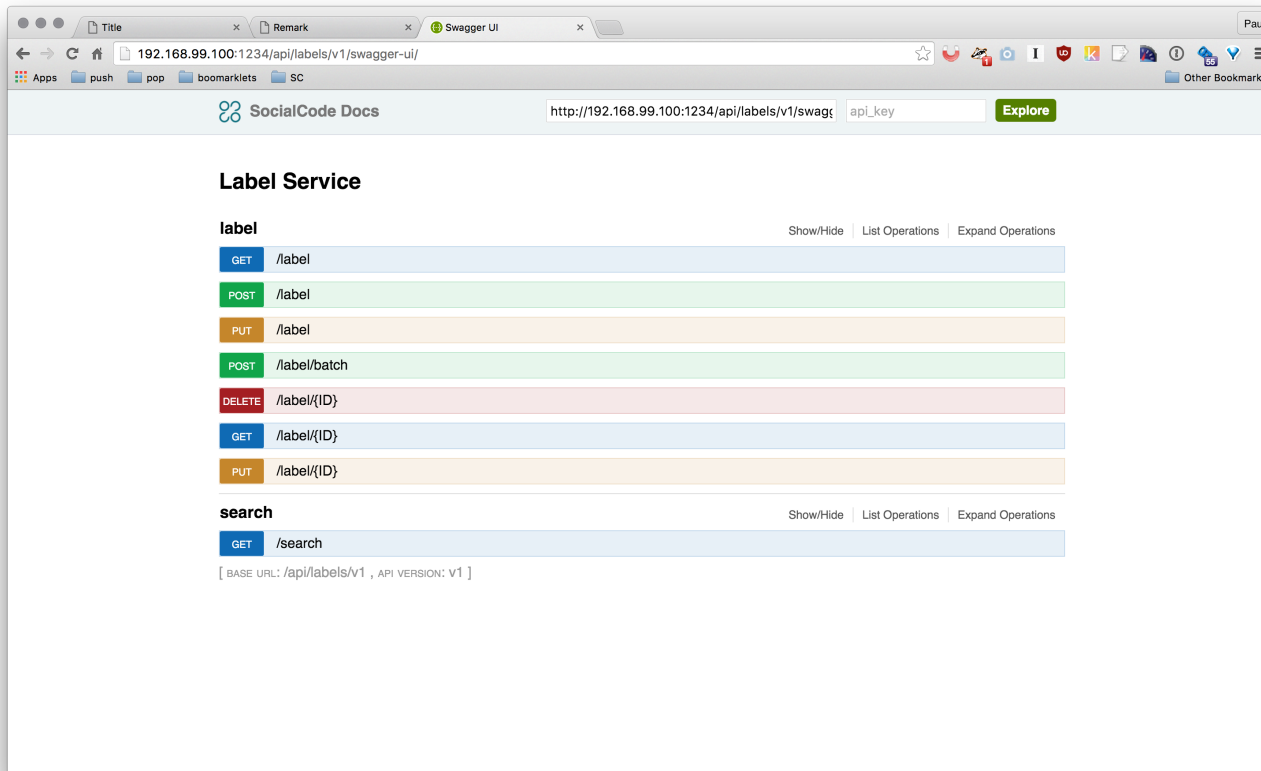
Maria /search

```
curl --silent "http://192.168.99.100:1234/api/labels/v1/search" | jq '.'  
{  
  "meta": {  
    "limit": 20,  
    "offset": 0  
  },  
  "objects": []  
}
```

Looks good

Maria API explorer

The API explorer also seems to look good



Maria

Lets check one more thing.

```
$ docker exec -i -t eee159342182 /bin/bash
root@eee159342182:/app# TERM=vt100 mysql -uroot label_service

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 50
Server version: 10.0.22-MariaDB-0+deb8u1 (Debian)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [label_service]> show tables;
+-----+
| Tables_in_label_service |
+-----+
| actions                  |
| crawl_source             |
| crawl_source_field       |
| crawl_state              |
| keywords                 |
| label_keywords           |
| label_relationships       |
| labels                   |
| oauth                    |
| schema_migrations        |
+-----+
10 rows in set (0.00 sec)

MariaDB [label_service]>
```

Maria

So we changed one line and rebuilt the container which retained no leftover materials from the previous container which referred to MySQL

We could push this new image to a registry and then deploy to a container runner like ecs

Container runners

- <https://www.tutum.co/>
- <https://aws.amazon.com/ecs/>
- Google compute engine

Misc

- Compose
 - Defines all of the containers that compose your application service (web, db, cache, etc)
- Swarm

Compose

A minimal `docker-compose.yml`

```
web:
  build: .
  ports:
    - "5000:5000"
  volumes:
    - ./code
  links:
    - redis
    - db
    - cache

redis:
  image: redis

db:
  image: mysql-ubuntu

cache:
  image: memcache
```

Future

- Jenkins integration
- Development builds
- A more flexible runtime context / choice of providers
- Simpler handoff between ops and dev
 - Imagine an ops managed base image with developer managed Dockerfiles in repos

```
# OPS managed
FROM socialcode-base:0.0.1

# Dev managed
RUN apt-get install golang
RUN apt-get install zlib
RUN apt-get install tesseract

ADD label-service /var/label-service
CMD ["/var/label-service/webhead"]
```

[DEAFENING APPLAUSE]

References

- Label service docker branch <https://github.com/pauladam/label-service/tree/docker>
- Label service Dockerfile <https://github.com/pauladam/label-service/blob/docker/Dockerfile>

