Leverage
Docker and Makefile
for
consistent testing environment

Docker Miami - 03/31/2015
@charme_g

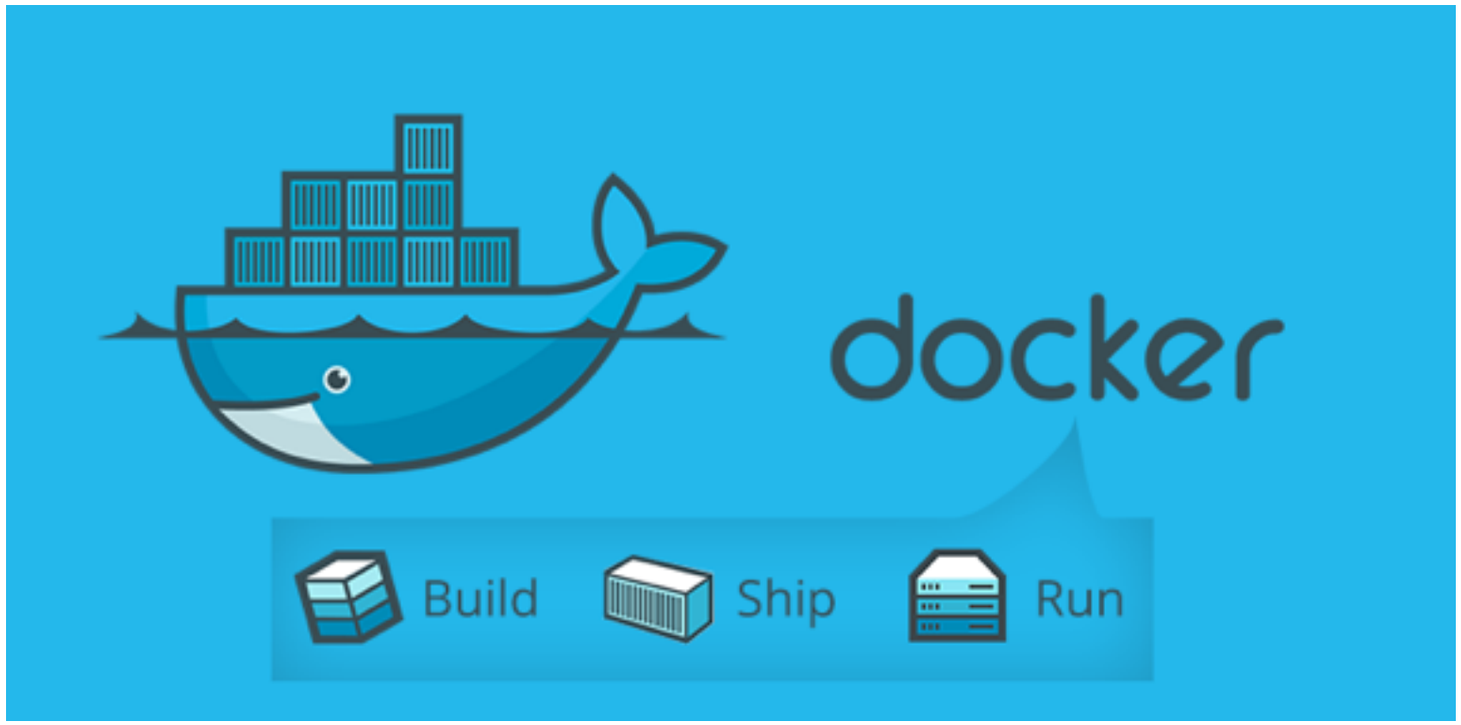
Docker 2nd Birthday Party

About me

- Guillaume J. Charmes
- @charme_g
- github.com/creack
- Lead developer on Docker from 0 to 1.0

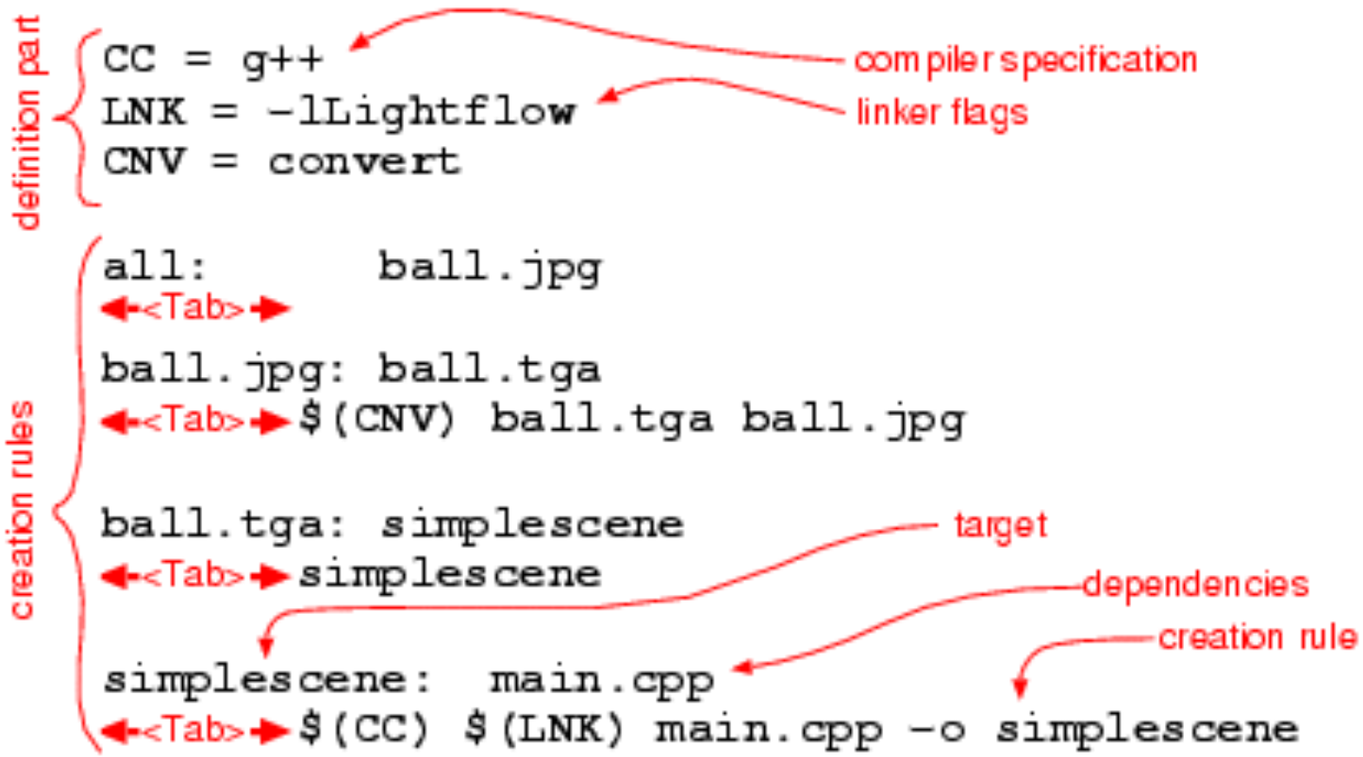
Docker

- What is Docker?
- Registry
- Use cases
- Dockerfile



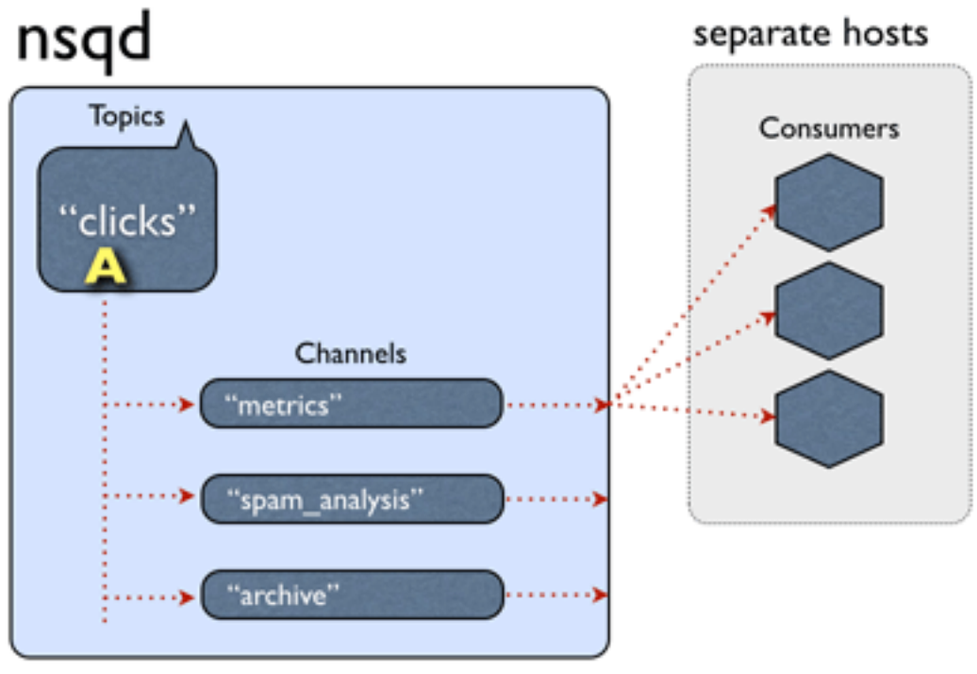
Makefile

- How it works?
- Example
- Add Docker into the mix
  - links



Multi-component Serivce

- NSQ
  - nsqd
  - nsqlookupd
  - nsqadmin
- Docker
- Makefile + Docker



```
all     :       nsq

nsq     :       nsqd

nsqlookupd:
        docker run -d --name nsqlookupd_c  nsqio/nsq /nsqlookupd

nsqd    :       nsqlookupd
        docker run -d --name nsqd_c --link nsqlookupd_c:lookupd  \
            nsqio/nsq bash -c "/nsqd
            --lookupd-tcp-address=$$LOOKUPD_PORT_4160_TCP_ADDR"


nsqadmin:       nsqd
        docker run -d --name nsqadmin_c -p 4171:4171 --link nsqlookupd_c:lookupd nsqio/nsqadmin \
            nsqui/nsq bash -c "/nsqadmin
            --lookup-tcp-address=$$LOOKUPD_PORT_4160_TCP_ADDR"
```

Use it for integration test

- Isolate integration test
  - tags
  - subdir
- Write the Makefile

```
NAME            =       sql_example
SQL_IMAGE       =       mysql
DOCKER_IMAGE    =       creack/sql_example


all             :       build

.build          :       .
                docker build -t $(NAME) .
                docker inspect -f '{{.Id}}' $(NAME) > .build

build           :       .build

mysql           :
                docker run -d --name $(SQL_IMAGE)_c $(SQL_IMAGE)
                sleep 2

test            :       build mysql
                docker run --link "$(SQL_IMAGE)_c:mysql" $(NAME) \
                        bash -c "MYSQL_ADDR=$$MYSQL_PORT_3306_TCP_ADDR go test -v"

clean           :
                docker rm -f $(SQL_IMAGE
```

Real life example

- github.com/gofinance/ib
- Java server
- Weird/complex setup
- Docker

Go further

- Use Makefiles and Docker to release code

```
NAME             = example
DOCKER_IMAGE     = 127.0.0.1:5000/$(NAME)

all              : build

.build           : .
                   docker build -t $(NAME) .
                   docker inspect -f '{{.Id}}' $(NAME) > .build

build            : .build

release/$(NAME)  : build
                   docker run --rm --entrypoint /bin/sh $(NAME) -c 'tar cf - /$(NAME) /etc/ssl' > $@ || (rm -f $@; false)
                   docker build --rm -t $(DOCKER_IMAGE) release

release          : release/$(NAME)

push             : release
                   docker push $(DOCKER_IMAGE)

clean            :
                   $(RM) .build release/$(NAME)

.PHONY           : push release build all clean
```