SINE NOMINE
ASSOCIATES

Introduction to Docker

Neale Ferguson

2016-06-24

Preface

- Examples built and run using ClefOS 7.2
  - CentOS Clone with name change
  - Available for z Systems
- However, as we will see this is irrelevant

What is Docker

- An open source project to pack, ship and run any application as a lightweight container
- Container: self-contained receptacle
  - Filesystem
  - Apps
  - Static data
  - Network

SINE NOMINE
ASSOCIATES

Docker

UNDERLYING TECHNOLOGIES

cgroups…

- A kernel feature that limits, accounts for, and isolates the resource of a collection of processes
- Similar to processes:
  - They are hierarchical
  - Child cgroups inherit certain attributes from their parent cgroup
- Difference: multiple cgroup hierarchies

…cgroups…

- Can span multiple “subsystems”
  - blkio —sets limits on input/output access to and from block
  - cpu —uses the scheduler to provide cgroup tasks access to the CPU
  - cpuacct —generates automatic reports on CPU resources used by tasks in a cgroup.

…cgroups…

- Can span multiple “subsystems”
  - cpuset —assigns individual CPUs (on a multicore system) and memory
  - devices —allows or denies access to devices by tasks in a cgroup
  - freezer —suspends or resumes tasks in a cgroup

…cgroups…

- Can span multiple “subsystems”
  - memory —sets limits on memory use by tasks in a cgroup, & generates automatic reports on memory
  - net_cls —tags network packets with a class identifier (classid) that allows the Linux traffic controller (tc) to identify packets originating from a particular cgroup task.

…cgroups…

- Can span multiple “subsystems”
  - net_prio —provides a way to dynamically set the priority of network traffic per network interface
  - ns — the namespace subsystem

Namespaces…

- CLONE_NEWIPC: IPC Namespaces: SystemV IPC and POSIX Message Queues can be isolated.
- CLONE_NEWPID: PID Namespaces: PIDs are isolated, meaning that a virtual PID inside of the namespace can conflict with a PID outside of the namespace. PIDs inside the namespace will be mapped to other PIDs outside of the namespace. The first PID inside the namespace will be '1' which outside of the namespace is assigned to init

…Namespaces…

- CLONE_NEWNET: Network Namespaces: Networking (/proc/net, IPs, interfaces and routes) are isolated. Services can be run on the same ports within namespaces, and "duplicate" virtual interfaces can be created.
- CLONE_NEWNS: Mount Namespaces. We have the ability to isolate mount points as they appear to processes. Using mount namespaces, we can achieve similar functionality to chroot() however with improved security.

…Namespaces

- CLONE_NEWUTS: UTS Namespaces. This namespaces primary purpose is to isolate the hostname and NIS name.
- CLONE_NEWUSER: User Namespaces. Here, user and group IDs are different inside and outside of namespaces and can be duplicated.

Copy-on-Write

- Allows Docker to instantiate containers very quickly
- Instead of having to make full copies of the which files comprise a container, it can use “pointers” back to existing files
- Containers are easily “linked” (or “stacked” or “layered”) to other containers

Docker Registry (optional)

- A stateless, highly scalable server-side application that stores and distributes Docker images
- Enables:
  - Tight control where images are stored
  - Full ownership of distribution pipeline
  - Integration of image storage & distribution into an in-house development workflow

Docker Daemon

- Manages containers
  - Creates volumes
  - Starts/stops containers

```
docker daemon -H tcp://0.0.0.0:4243 -H
unix:///var/run/docker.sock
```

SINE NOMINE
ASSOCIATES

Docker

BUILDING CONTAINERS

Creating a Starter System

SINE NOMINE ASSOCIATES

- Base image: containers built from it or its descendants
- Create a chroot-like environment
  - File system including /dev
  - yum install packages
  - Trim unwanted stuff
  - Create tar ball
  - Import to Docker

The Dockerfile

- A recipe for building a container
- Build from an existing container
- Install requirements
- Define network and volume requirements
- Specify command to run on startup

```
FROM    clefos71-base-s390x:latest

MAINTAINER The ClefOS Project <neale@sinenomine.net>
LABEL   Vendor="ClefOS" License="GPLv2" Version="8.0-10.1"

COPY    ibm-java-sdk-8.0-1.10-s390x-archive.bin java.rsp dummy-
java-1.8-0.el7.noarch.rpm /

RUN     yum install -y tar zip && \
        mkdir -p /opt/ibm && \
        echo "Installing IBM JDK" && \
        /ibm-java-sdk-8.0-1.10-s390x-archive.bin -f /java.rsp -i silent && \
        yum install -y dummy-java-1.8-0.el7.noarch.rpm && \
        yum erase -y tar zip vim-minimal && \
        yum clean all && \
        rm -f /*.rpm /java.rsp /*.bin

ENV     JAVA_HOME=/opt/ibm/java PATH=$JAVA_HOME/bin:$PATH
```

```
FROM    brunswickheads/clefos71-nodejs-s390x
MAINTAINER      The ClefOS project <neale@sinenomine.net>
ADD     epel.repo /etc/yum.repos.d/epel.repo
RUN     yum install -y git tar gcc gcc-c++ make mongodb mongodb-server \
                mongo-tools krb5-devel perl-Digest-SHA && \
                npm install -g express && \
                npm install -g mongodb && \
                npm install -g tar mkdirp
WORKDIR /mean
EXPOSE  27017 28017
VOLUME  /mongodb/data
RUN     echo "mongod --fork --logpath /mongodb/data/log/mongod.log \
        --dbpath /mongodb/data --smallfiles --noprealloc --httpinterface --rest \
        > /start.sh && \
        echo "node \$1" >> /start.sh && \
        yum erase -y git tar gcc gcc-c++ make perl-Digest-SHA && \
        rm -f /etc/yum.repos.d/epel.repo && \
        rm -rf /tmp/* /var/cache/yum/* /root/* /root/.[a-zA-Z0-9]* /src
ENV     NODE_PATH=/opt/ibm/nodejs/lib/node_modules:/mean/node_modules
ENTRYPOINT ["sh", "/start.sh"]
```

Building Images

- Each step corresponds to a layer
- Stop build at one point
- Rebuild starts from last change

Managing Images

```
[root@docker docker]# docker images
REPOSITORY                          TAG     IMAGE ID     CREATED       VIRTUAL SIZE
brunswickheads/fluentd-s390x        latest  b3e3d646f313 4 days ago    515.2 MB
brunswickheads/amhub-s390x          latest  76a2c4a387f0 7 days ago    795 MB
brunswickheads/ade-s390x            latest  5dc6c7c6191c 5 weeks ago   645.8 MB
brunswickheads/compose-ui-s390x     latest  ff5b9eda68ec 8 weeks ago   315.9 MB
brunswickheads/nginx-1.8-s390x      latest  4f87e1292531 8 weeks ago   211 MB
brunswickheads/clefos71-base-s390x  latest  60ef3a8ba174 3 months ago  110.5 MB
clefos71-base-s390x                 latest  60ef3a8ba174 3 months ago  110.5 MB
brunswickheads/clefos71-nodejs-s390x latest d76f12128dde 5 months ago  548.7 MB
brunswickheads/mariadb-5.5-s390x    latest  91233ea5a5c1 5 months ago  311.3 MB
brunswickheads/clefos71-java-s390x  latest  3cb8ef8fd562 5 months ago  480.2 MB
```

Making Images Available

```
[root@docker ~]# docker push brunswickheads/fluentd-s390x:latest
The push refers to a repository [docker.io/brunswickheads/fluentd-s390x] (len: 1)
b3e3d646f313: Pushed
1b11901fbead: Pushed
5f6ab7c78e8b: Pushed
288d092713a6: Pushed
f86e5eb99f4b: Pushed
745463ff1ff7: Pushed
c1eb69a620cb: Pushed
67cc290487fe: Pushed
f295869eaedd: Pushed
530a81592b4f: Pushed
d69fc3fad8fa: Pushed
732e18ef67b6: Pushed
7196f6de1451: Pushed
7118afa06d84: Pushed
ec3ec425b681: Pushed
60ef3a8ba174: Pushed
latest: digest:
sha256:120519d3d8f0cf00a0caddb3fd8c0c6148b8145dbf6fed2897b36e965d35424d size: 29665
```

Dockerhub

SINE NOMINE ASSOCIATES

Dashboard Explore Organizations Search Create brunswickheads

Repos Stars

brunswickheads

Neale Ferguson

Ashburn, Virginia, USA
Joined November 2015

brunswickheads/clefos71-base-s390x public	2 STARS	117 PULLS	DETAILS
brunswickheads/openchain-peer public	0 STARS	69 PULLS	DETAILS
brunswickheads/openchain-build-s390x public	0 STARS	58 PULLS	DETAILS
brunswickheads/golang1.5-s390x public	0 STARS	54 PULLS	DETAILS
brunswickheads/spark-1.5.2-s390x public	0 STARS	45 PULLS	DETAILS

Running Containers

- Persistent data goes to [a] volume[s]
- Run a standalone container
  - All functionality within the container
- Run a “swarm” of containers
  - Typically database server
  - Web server
  - Application server

Running Containers

- `docker run --name=mariadb -v /var/local/mariadb:/var/lib/mysql -d -p 3306:3306  -e MYSQL_ROOT_PASSWORD=passw0rd brunswickheads/mariadb-5.5-s390x:latest mysqld_safe --connect-timeout=30`

- `docker run --rm -i -t --name=ade -p 8022:22 --link=mariadb -v /var/local/ade:/var/local/ade -e MARIADB_ROOT_PASSWORD=passw0rd -e MARIADB_ADE_PASSWORD=passw0rd brunswickheads/ade-s390x`

Running Containers

- Containers run as daemons or interactively
- Multiple containers wanting to use same port?
  - Docker can remap:

    -p <host port>:<container port>

Running Containers

- What about environment variables?
  - -e option
  - Dockerfile
- What is my container doing:
  - docker top <image id>
  - Top
- What is my container config?
- docker inspect <image>

```
"Networks": {
        "bridge": {
            "EndpointID":
"0448a9a68ed5a9c5f89435b3d62d78bbc42d4f601a25e65b2e149ed8f6949
93c",
            "Gateway": "172.18.0.1",
            "IPAddress": "172.18.0.3",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:12:00:03"
        }
```

Running Containers

- Command line
  - docker run
  - kubernetes
- GUIs
  - Compose-UI
  - AMHub
- Images are automatically downloaded

Running Containers

- Build on ClefOS / Run on Ubuntu
- Build on ClefOS / Build upon image on Ubuntu
- Builders meet all pre-requisites
- Self-contain requirements
  - No conflicts with other containers
  - Unlike multi-tenancy apps

SINE NOMINE
ASSOCIATES

Docker on z

DEMO TIME

Docker Compose UI

Projects

hello-node

nginx

node-redis

volumes-demo

filter by project name:

Reload projects

Create new project

Settings

nginx

view yml

Actions

build | pull | logs | kill | stop | start | up | down

Services

no containers found

Docker Compose UI v.0.13 Licensed under the MIT license.

nginx

close yml

```yml
nginx:
  image: brunswickheads/nginx-1.8-s390x
  container_name: nginx
  hostname: nginx
  ports:
  - 80:80
  - 443:443
```

nginx

close yml

```yml
nginx:
  image: brunswickheads/nginx-1.8-s390x
  container_name: nginx
  hostname: nginx
  ports:
  - 80:80
  - 443:443
```

Services

nginx scale

nginx logs details

Network

0.0.0.0:443 : 443/tcp

0.0.0.0:80 : 80/tcp

nginx / nginx

Key	Value
name	nginx
ip	172.18.0.4
number	1
short id	5a9fb7a01a
log config	json-file
command	nginx -g daemon off;
environment	• PATH = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
labels	• com.docker.compose.config-hash = fe4fe73c044ff3dd65431737d3340ed20a0ad8ab1c7288b30e69fbb416da19fe • com.docker.compose.container-number = 1 • com.docker.compose.oneoff = False • com.docker.compose.project = nginx • com.docker.compose.service = nginx • com.docker.compose.version = 1.6.0

SINE NOMINE ASSOCIATES

Welcome to **nginx** on ClefOS!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on ClefOS. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

NGINX Powered by ClefOS

nginx

close yml

```yaml
nginx:
  image: brunswickheads/nginx-1.8-s390x
  container_name: nginx
  hostname: nginx
  ports:
  - 80:80
  - 443:443
```

SINE NOMINE ASSOCIATES

nginx

close yml

```
nginx:
  image: brunswickheads/nginx-1.8-s390x
  container_name: nginx
  hostname: nginx
  ports:
  - 80:80
  - 443:443
```

Actions

build | pull | logs | kill | stop | start | up | down

Services

no containers found

```
db:
  image: 'postgres:9.4'
  restart: always
redis:
  image: 'redis:latest'
  restart: always
result:
  autoredeploy: true
  image: 'docker/example-voting-app-result:latest'
  ports:
    - '80:80'
  restart: always
```

```
lb:
  autoredeploy: true
  image: 'dockercloud/haproxy:latest'
  links:
    - vote
  ports:
    - "80:80"
  roles:
    - global
  restart: always
```

```
vote:
  autoredeploy: true
  image: 'docker/example-voting-app-vote:latest'
  restart: always
  target_num_containers: 5

worker:
  autoredeploy: true
  image: 'docker/example-voting-app-worker:latest'
  restart: always
  target_num_containers: 3
```