Mitigate complexity in cloud micro services analysis

Swiss army knife tool for live container analysis

Cisco CSIRT

Fabio Nigi

FIRST TC April 20, 2015

Last login: Thu Apr 20 16:30:00 on ttyfirst2016

$

$whoami

Fabio Nigi / @fnigi / fnigi@cisco.com

$sudo su –

Cisco csirt

$finger

Member of the winner pool of global Docker hack 2015

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

"Never upgrade a server again. Never update your code. Instead, create new servers, and throw away”Jerome Petazzo,
Docker Tinkerer Extraordinaire

```
FROM ubuntu:14.04

RUN apt-get update && \
apt-get install -y curl openjdk-7-jre-headless
```

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Introduction to the technology Docker from the OS stack to minimal infrastructure design

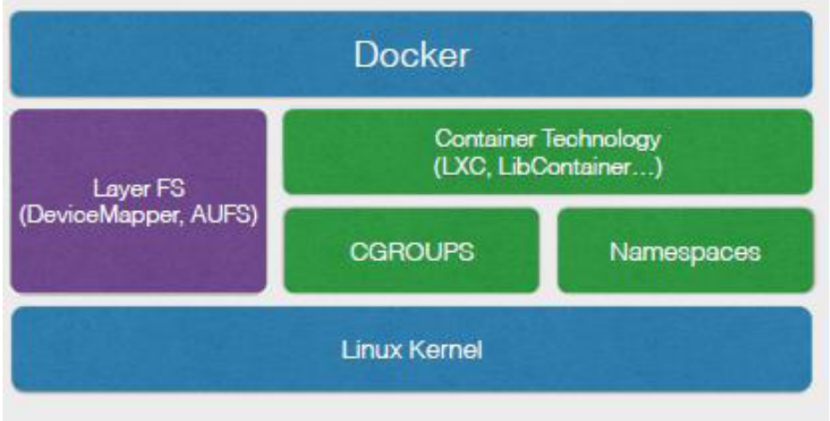


Image source:
http://www.ociweb.com/resources/publications /sett/ march -20 15-docker-vs-vagrant/

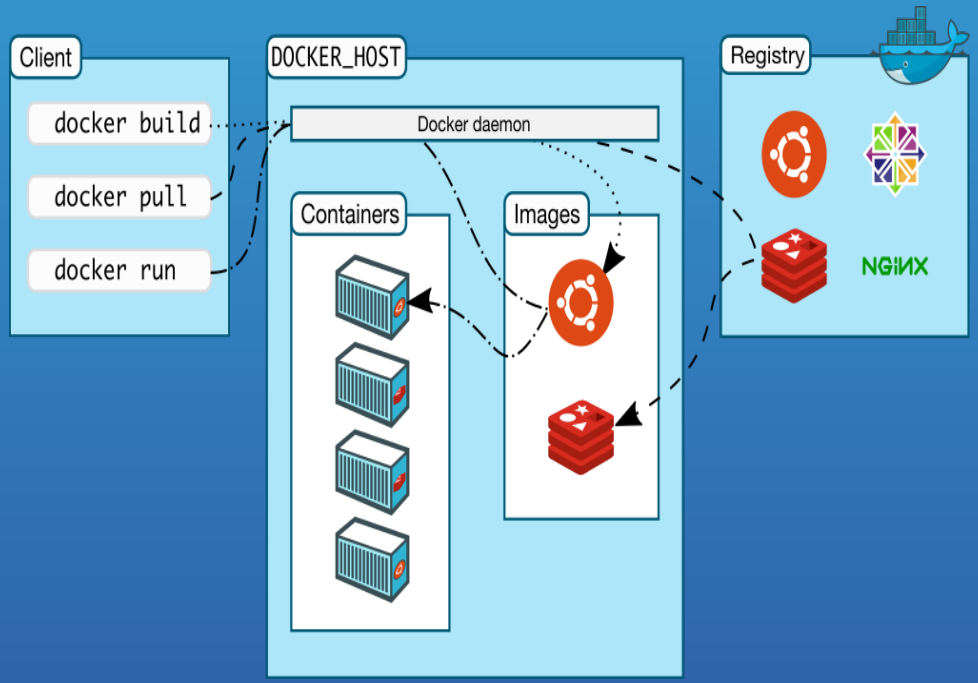
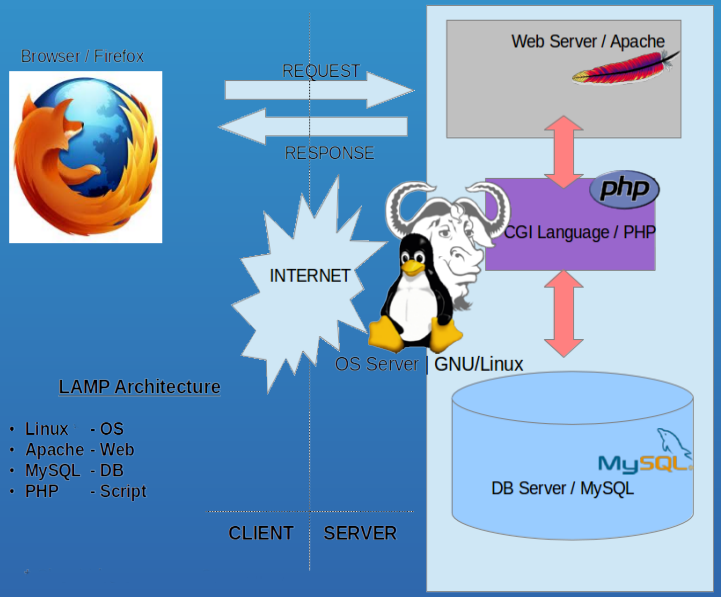


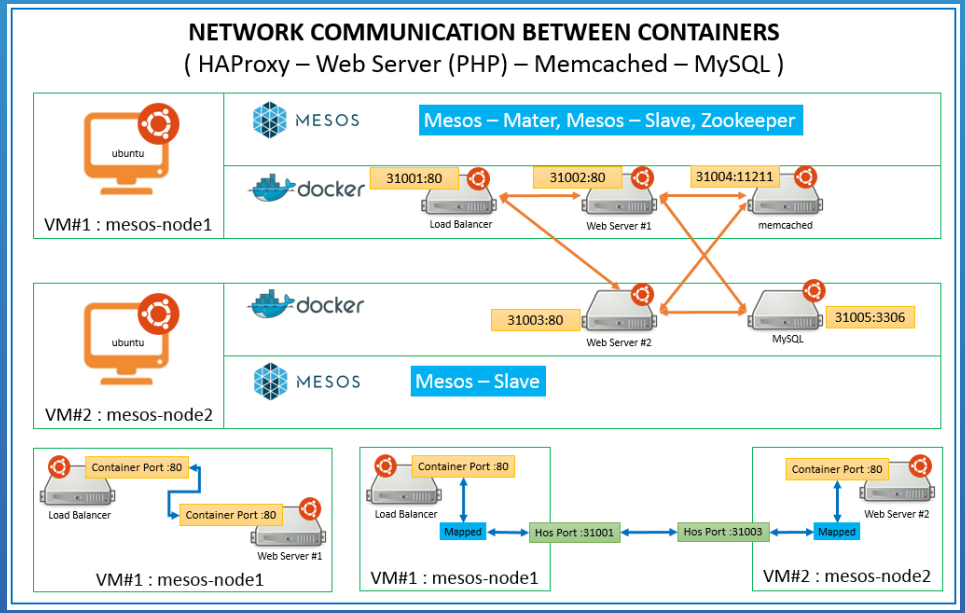
Image source: https://docs.docker.com/engine/un derst anding -dock er/

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Serve a website: architectural approach: with or without micro-services



https://en.wikipedia.org/wiki/LAMP_(software_b undle)



https://selfieblue.files.wordpress.com/2 015/1 0/ima ge11 5.png

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Orchestration at scale w/Docker

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

-bind

```
docker run -v /Users/<path>:/<container path> ...
```

-privileged

```
docker run --privileged
```

Bypass security without warning

Capability Key	Capability Description
SETPCAP	Modify process capabilities.
SYS_MODULE	Load and unload kernel modules.
SYS_RAWIO	Perform I/O port operations (iopl(2) and ioperm(2)).
SYS_PACCT	Use acct(2), switch process accounting on or off.
SYS_ADMIN	Perform a range of system administration operations.
SYS_NICE	Raise process nice value (nice(2), setpriority(2)) and change the nice value for arbitrary processes.
SYS_RESOURCE	Override resource Limits.
SYS_TIME	Set system clock (settimeofday(2), stime(2), adjtimex(2)); set real-time (hardware) clock.
SYS_TTY_CONFIG	Use vhangup(2); employ various privileged ioctl(2) operations on virtual terminals.
MKNOD	Create special files using mknod(2).
AUDIT_WRITE	Write records to kernel auditing log.
AUDIT_CONTROL	Enable and disable kernel auditing; change auditing filter rules; retrieve auditing status and filtering rules.
MAC_OVERRIDE	Allow MAC configuration or state changes. Implemented for the Smack LSM.
MAC_ADMIN	Override Mandatory Access Control (MAC). Implemented for the Smack Linux Security Module (LSM).
NET_ADMIN	Perform various network-related operations.
SYSLOG	Perform privileged syslog(2) operations.
CHOWN	Make arbitrary changes to file UIDs and GIDs (see chown(2)).
NET_RAW	Use RAW and PACKET sockets.
DAC_OVERRIDE	Bypass file read, write, and execute permission checks.
FOWNER	Bypass permission checks on operations that normally require the file system UID of the process to match the UID of the file.
DAC_READ_SEARCH	Bypass file read permission checks and directory read and execute permission checks.
FSETID	Don't clear set-user-ID and set-group-ID permission bits when a file is modified.
KILL	Bypass permission checks for sending signals.
SETGID	Make arbitrary manipulations of process GIDs and supplementary GID list.
SETUID	Make arbitrary manipulations of process UIDs.
LINUX_IMMUTABLE	Set the FS_APPEND_FL and FS_IMMUTABLE_FL i-node flags.
NET_BIND_SERVICE	Bind a socket to internet domain privileged ports (port numbers less than 1024).
NET_BROADCAST	Make socket broadcasts, and listen to multicasts.
IPC_LOCK	Lock memory (mlock(2), mlockall(2), mmap(2), shmctl(2)).
IPC_OWNER	Bypass permission checks for operations on System V IPC objects.
SYS_CHROOT	Use chroot(2), change root directory.
SYS_PTRACE	Trace arbitrary processes using ptrace(2).
SYS_BOOT	Use reboot(2) and kexec_load(2), reboot and load a new kernel for later execution.
LEASE	Establish leases on arbitrary files (see fcntl(2)).
SETFCAP	Set file capabilities.
WAKE_ALARM	Trigger something that will wake up the system.
BLOCK_SUSPEND	Employ features that can block system suspend.

https://en.wikipedia.org/wiki/MV_Rena

Overriding Dockerfile image defaults

When a developer builds an image from a *Dockerfile* or when she commits it, the developer can set a number of default parameters that take effect when the image starts up as a container.

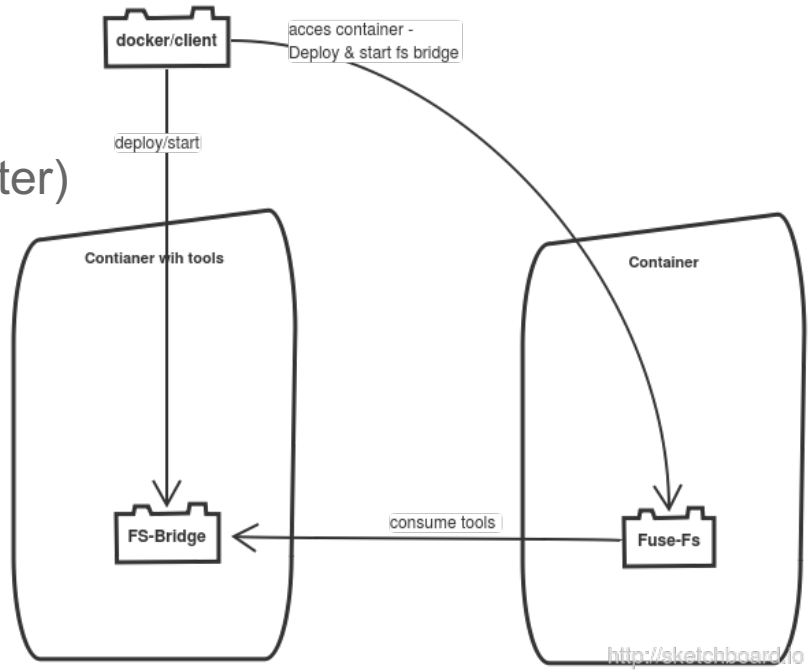
Four of the Dockerfile commands cannot be overridden at runtime: `FROM`, `MAINTAINER`, `RUN`, and `ADD`. Everything else has a corresponding override in `docker run`. We'll go through what the developer might have set in each Dockerfile instruction and how the operator can override that setting.

- CMD (Default Command or Options)
- ENTRYPOINT (Default Command to Execute at Runtime)
- EXPOSE (Incoming Ports)
- ENV (Environment Variables)
- VOLUME (Shared Filesystems)
- USER
- WORKDIR

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Design and feature

- Injecting tools into running container
- Network transparent or 9pfs client server
- Browsing container FS
- Scriptable - easy to use in bash scripts
- Live analysis without trace (removing all after)



How to run and execute the cloud micro services analysis:

1) setup a tools container

$docker build -t csk-tools Dockerfiles/debian-tools

2) setup a csk container

docker build -t csk .

3) enhance desired container with your tool and follow the instruction to use it!

docker run -ti -v /var/run/docker.sock:/var/run/docker.sock csk <container to enhance>

4) hack it!

5) remove tools from a container

docker run csk --remove <container to enhance>

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Usage & cases

System debug
File comparison (hashing)
Compromised applications testing

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

fnigi@cisco.com

@fnigi

https://github.com/nigifabio

The tool: https://github.com/docker/global-hack-day-3/tree/master/container-swiss-knife