

Docker from A to Z

Including **Oracle Container Cloud Service**

OUGN Conference 2017

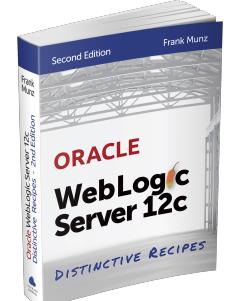
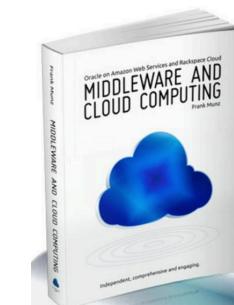
Dr. Frank Munz





Frank Munz

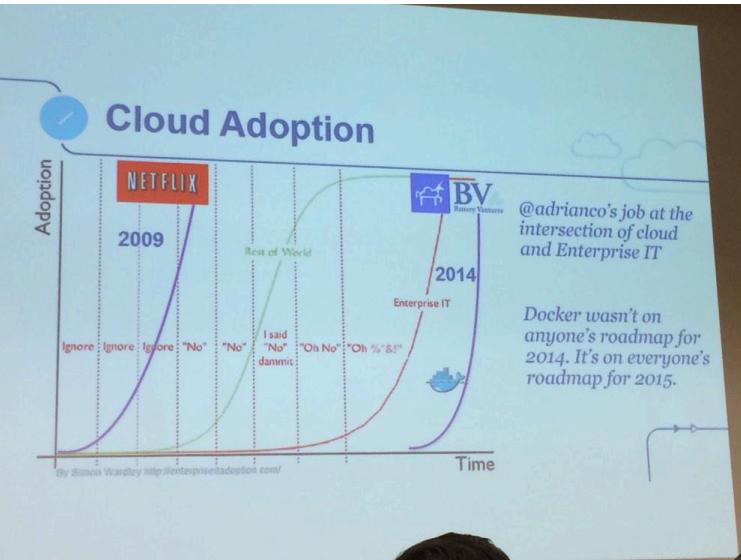
- Founded munz & more in 2007
- 17 years Oracle Middleware, Cloud, and Distributed Computing
- Consulting and High-End Training
- Wrote two Oracle WLS and one Cloud book





... some basics

Adrian Cockcroft
Netflix



“Docker wasn’t on anyone’s agenda for 2014.
It’s on every ones roadmap for 2015.”

Docker

- Open Source (evolving), written in Go
- Container technology
- Portable standard
- Runs on Linux (Microsoft, MacOS, Solaris)

The Register logo: **The Register®**
Biting the hand that feeds IT

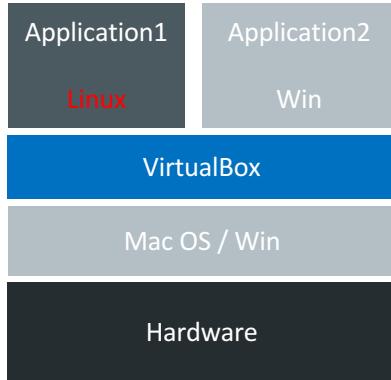
DATA CENTRE SOFTWARE NETWORKS SECURITY INFRASTRUCTURE BUSINESS HARDWARE SCIENCE

Google: 'EVERYTHING at Google runs in a container'

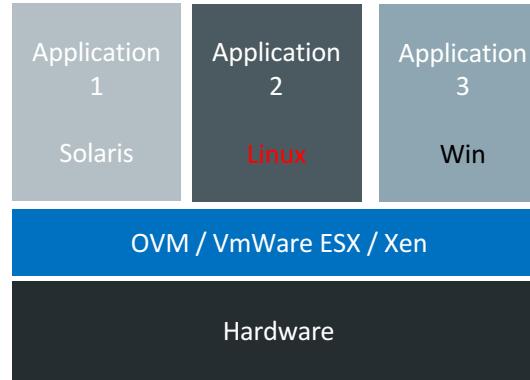
Ad giant lifts curtain on how it uses virtualization's successor
TWO BILLION TIMES A WEEK

Google starts
2.000.000.000
containers
per week!

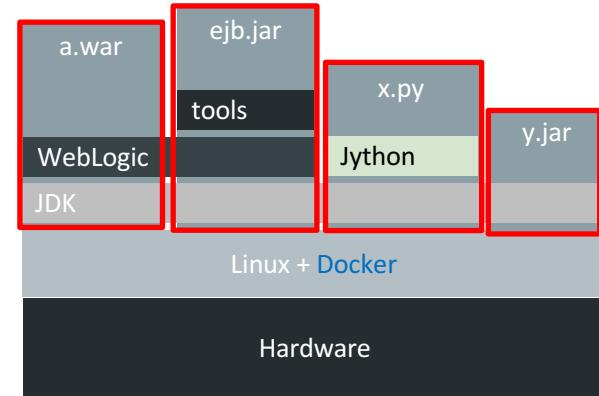
Virtualization vs. Isolation



Desktop Virtualization:
Type 2 hypervisor
= with host OS

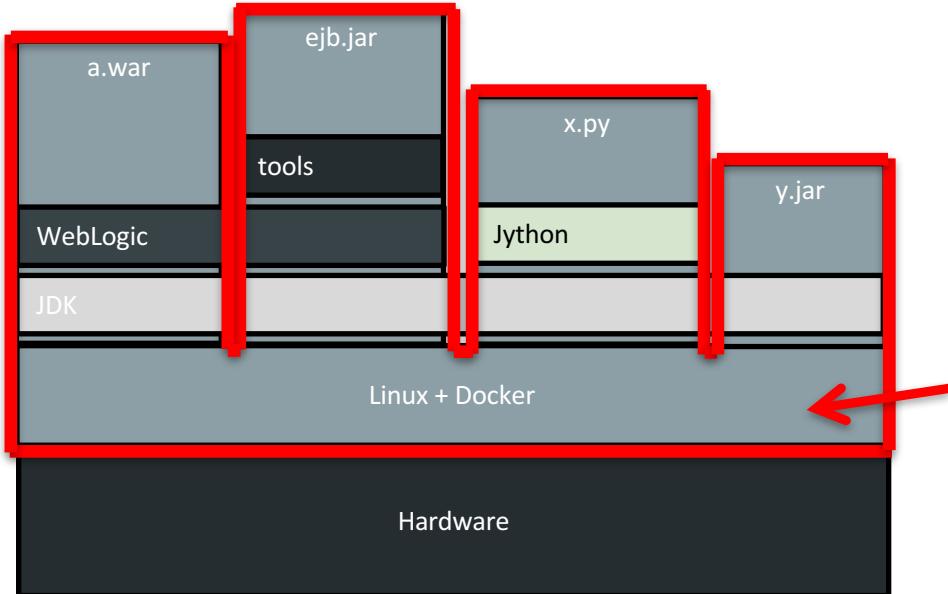


Server Virtualization
Type 1 hypervisor
= on bare metal



Immutable Docker **container** in Linux
with own FS, network stack /
IP address, process space and
resource limits.
-> **Isolation**

Docker



Docker is not a lightweight VirtualBox
- it's about **isolation**.

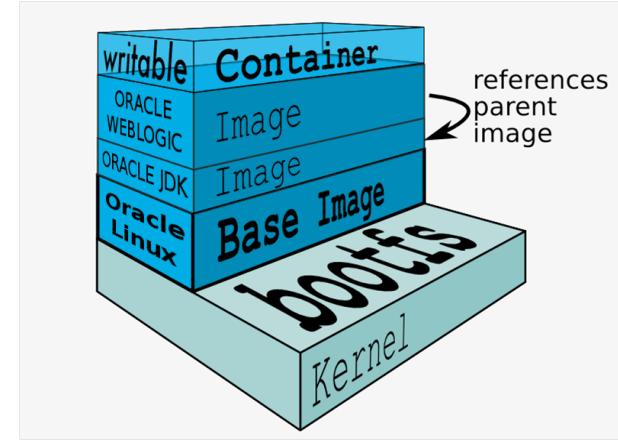
Containers run on Linux kernel of host

-> Containers are visible on host

Docker Images

- Package format
- Layered incremental,
copy on write file system
- “Application with all dependencies”
- Create image yourself
or get it from Docker Hub

docker image ls



Example Layers:

- WLS Domain
- WebLogic
- Java
- Linux Image

Docker Container

- Isolated runtime of Docker image
- Starts up in milliseconds
- Sandboxing uses Linux namespaces and cgroups (RAM, CPU, filesystem)
-> isolated part of your Linux
- Open Container Standard / Linux Foundation

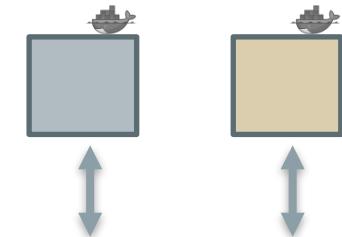
```
docker run -d -p 8080:9999 fmunz/micro
```

Docker Limitations

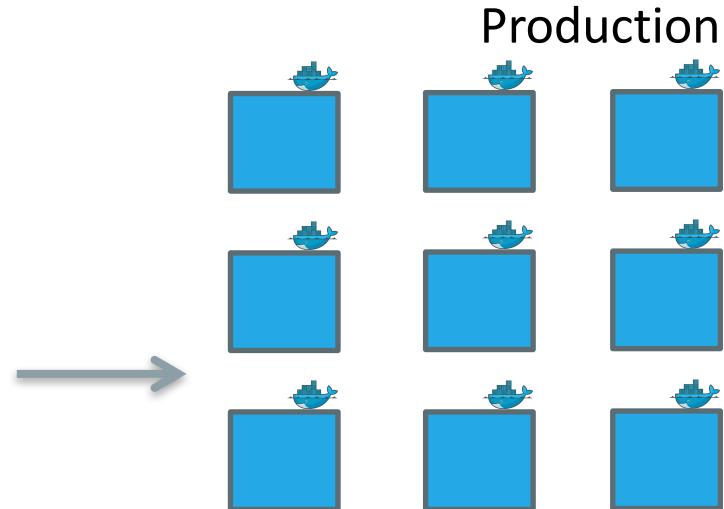
- Cannot load kernel modules
- Applications that manipulate namespaces
- Kernel config per container
- Some SW not (yet) supported when running in Docker container: ~~Oracle DB~~ etc.

Solves the “Worked For Me!” issue

OS utils, JDK, patches, database driver, libs, appserver, domain, **deployment**, tools, scripts



Testing



You can pass environment variables for specific settings e.g. in prod



Docker Registry

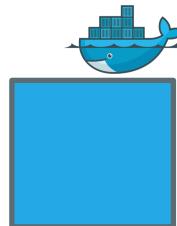
Dockerfile

Dockerfile

Manually create container with
docker build

GitHub  + 

Automatic build:
`gitub push trigger image build`



Docker Image

Automated Builds

- Automatically build your images:
GitHub account with Dockerfile
 - Registry uses GitHub directory structure
as build context
 - Image is uploaded automatically
to Docker hub
- > Trust, up to date, and transparent



Dockerfile

```
33 # Pull base image
34 # -----
35 FROM oraclelinux:7
36
37 # Maintainer
38 # -----
39 MAINTAINER Bruno Borges <bruno.borges@oracle.com>
40
41 # Environment variables required for this build (do NOT change)
42 # -----
43 ENV JAVA_RPM jdk-7u79-linux-x64.rpm
44 ENV WLS_PKG fmw_12.1.3.0.0_wls.jar
45 ENV JAVA_HOME /usr/java/default
46 ENV CONFIG_JVM_ARGS -Djava.security.egd=file:/dev/.urandom
47
48 # Setup required packages (unzip), filesystem, and oracle user
49 # -----
50 RUN mkdir /u01 && \
51     chmod a+xr /u01 && \
52     useradd -b /u01 -m -s /bin/bash oracle
53
54 # Copy packages
55 COPY $WLS_PKG /u01/
56 COPY $JAVA_RPM /u01/
57 COPY install.file /u01/
58 COPY oraInst.loc /u01/
59
60 # Install and configure Oracle JDK 8u25
61 # -----
62 RUN rpm -i /u01/$JAVA_RPM && \
63     rm /u01/$JAVA_RPM
64
65 # Change the open file limits in /etc/security/limits.conf
66 RUN sed -i '/.*EOF/d' /etc/security/limits.conf && \
67     echo "* soft nofile 16384" >> /etc/security/limits.conf && \
68     echo "* hard nofile 16384" >> /etc/security/limits.conf && \
69     echo "*/EOF" >> /etc/security/limits.conf
70 munichmon
```

Manually create container:

`docker build -t name .`

And Then Automate

- Build Docker images for testing from continuous delivery pipeline
- Use Jenkins / Hudson hooks or a maven plugin to create / start / stop / delete Docker containers

Docker Maven Plugins



fabric8: docker-maven

- Alternative approach to Dockerfile
- Various plugins
- Few only active

Goal	Description
<code>docker:start</code>	Create and start containers
<code>docker:stop</code>	Stop and destroy containers
<code>docker:build</code>	Build images
<code>docker:watch</code>	Watch for doing rebuilds and restarts
<code>docker:push</code>	Push images to a registry
<code>docker:remove</code>	Remove images from local docker host
<code>docker:logs</code>	Show container logs
<code>docker:source</code>	Attach docker build archive to Maven project
<code>docker:volume-create</code>	Create a volume to share data between containers
<code>docker:volume-remove</code>	Remove a created volume

the registry

Registry

- Docker image is not found?
-> pulled from registry
- Push your image to registry
`docker push yourname/newimage`
- Free account includes 1 private registry

Also private, containerized registry for download with fs and optional in-memory, S3, or Azure data store

Docker Hub Registry - Rep x Settings demo

https://registry.hub.docker.com

 docker What is Docker? Use Cases Try It! Browse Install & Docs Log In Sign Up

Search the Registry

Official Repositories

 redis

 The Official Ubuntu base image

 MySQL™ Popular open-source relational database management system

 NGINX

 mongoDB Document-oriented NoSQL database

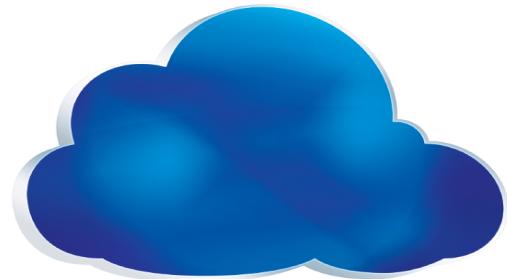
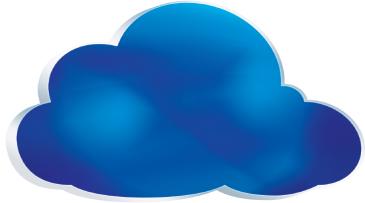
 CentOS Official CentOS base image



Docker Registry

what should be your
biggest nightmare:

unknown and
unofficial images
(>14000)



clouds

Docker in the Cloud?

Supported by every major cloud provider:



Oracle Cloud and Docker

Oracle Container Cloud Service OCCS (covered in part II)

- Available now!

Oracle Application Container Cloud Service

- Uses Docker containers to run your Java or JavaScript application

Compute Cloud Service

- Manually run your containers

Docker Data Center, AWS ECS, Google Container Service

- Docker Clouds

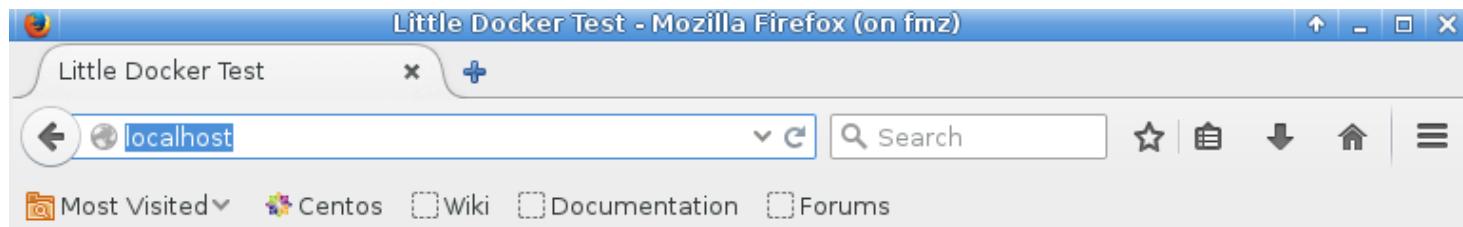
fancy a demo?

Small Images / Microservices

You can have a real service in ...

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
oracle/dom1	latest	bc2020c6bd0f	3 hours ago	2.027 GB
fmdom1	latest	bc2020c6bd0f	3 hours ago	2.027 GB
oracle/weblogic	12.1.3-dev	2aa8eadf6c86	3 hours ago	2.025 GB
micro	latest	dfdcbb33a11fe	6 hours ago	8.488 MB
ubuntu	latest	8251da35e7a7	4 days ago	188.4 MB
centos	latest	7322fbe74aa5	7 weeks ago	172.2 MB
oraclelinux	7	8a2b759d9dd8	8 weeks ago	189.6 MB



Possible Options:
busybox and
static binary

Simple Life Inside Container

```
Mem: 5023576K used, 2950096K free, 0K shrd, 0K buff, 140346646331393K cached  
CPU: 0% usr 0% sys 0% nic 100% idle 0% io 0% irq 0% sirq  
Load average: 0.12 0.19 0.13 2/1612 13
```

PID	PPID	USER	STAT	VSZ	%VSZ	%CPU	COMMAND
1	0	root	S	7192	0%	0%	python /webserver.py
9	0	root	S	3176	0%	0%	/bin/sh
13	9	root	R	3168	0%	0%	top

processes

```
/ #
```

```
/ # ls  
bin etc lib linuxrc mnt proc run sys usr webserver.py  
dev home lib64 media opt root sbin tmp var  
/ #
```

FS

```
/ # df  
Filesystem 1K-blocks Used Available Use% Mounted on  
none 19049892 2568556 15490612 14% /  
tmpfs 3986836 0 3986836 0% /dev  
shm 65536 0 65536 0% /dev/shm  
/dev/sda1 19049892 2568556 15490612 14% /etc/resolv.conf  
/dev/sda1 19049892 2568556 15490612 14% /etc/hostname  
/dev/sda1 19049892 2568556 15490612 14% /etc/hosts  
tmpfs 3986836 0 3986836 0% /proc/kcore  
tmpfs 3986836 0 3986836 0% /proc/timer_stats
```

mounts

#25

#3

Security

```
$ docker run -d -p  
8080:9999 fmunz/micro
```

VS.

A stranger gives you a box at night and asks you to connect it to your company network:



Would you do it?

Suggestions

- Use trusted images / with known Dockerfile
- Kernel features are well established
 - cgroups (2006, merged into 2.6.24 kernel)
 - namespaces (initial kernel patch 2.4.19)
- Docker can use TLS (client to daemon)
- Docker images can be signed
- Think twice about pulling images from public repos / Docker hub



FUD

"Docker is like chroot() on steroids."

- Yes: It's easy to escape chroot() environment
- No: Docker does **not** use chroot()
-> it uses namespaces

Do namespaces solve it?

6 different namespace, but
not everything is namespaced, eg:

- /proc/sys|irq|bus
- /sys, /sys/fs
- /dev/mem
- /dev/sd*
- kernel modules
- User namespaces since 1.9

Docker uses **read-only mounts** where possible

Linux Capabilities

- Privileged container: like being root on host
- Capabilities -> Break down power of root
- Examine PID 1 capabilities with getpcaps:

```
$ getpcaps 1
Capabilities for `1': = cap_chown, cap_dac_override, cap_dac_read_search, cap_fowner, cap_fsetid, cap_kill,
cap_setgid, cap_setuid, cap_setpcap, cap_linux_immutable, cap_net_bind_service, cap_net_broadcast, cap_net_admin,
cap_net_raw, cap_ipc_lock, cap_ipc_owner, cap_sys_module, cap_sys_rawio, cap_sys_chroot, cap_sys_ptrace,
cap_sys_pacct, cap_sys_admin, cap_sys_boot, cap_sys_nice, cap_sys_resource, cap_sys_time, cap_sys_tty_config,
cap_mknod, cap_lease, cap_audit_write, cap_audit_control, cap_setfcap, cap_mac_override, cap_mac_admin, cap_syslog, 35, 36+ep
ccloud:/home/frank
$
ccloud:/home/frank
$ docker run -it centos getpcaps 1
Capabilities for `1': = cap_chown, cap_dac_override, cap_fowner, cap_fsetid, cap_kill, cap_setgid, cap_setuid,
cap_setpcap, cap_net_bind_service, cap_net_raw, cap_sys_chroot, cap_mknod, cap_audit_write, cap_setfcap+ep
```

User Namespaces

- Before Docker 1.10 root process in container was root process on host
 - with limited capabilities
- With Docker 1.10 user namespaces a root process in container can be mapped to user process on host
 - Disabled per default
 - To enable start Docker daemon with `--userns-remap`

... more Suggestions

- Drop privileges as quickly as possible
- Treat root in container as root outside (although it isn't)
- No secrets in images
- Combine Docker with SELinux, AppArmor and / or virtualization
- Host can always access container

Note: Public PaaS do not simply spin up Docker containers!

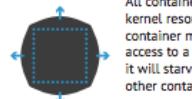
Cheat Sheet

TYPES OF SECURITY THREATS AND HOW TO AVOID THEM



KERNEL EXPLOITS

If a container can cause a kernel panic or similar, it will bring down the whole host.



DENIAL OF SERVICE (DoS) ATTACKS

All containers share kernel resources. If one container monopolizes access to a resource, it will starve out the other containers.



CONTAINER BREAKOUTS

If an attacker can breakout of a container, they can gain access to the host and other containers.



POISONED IMAGES

Images may be injected with trojan or virus infected software. Or they may simply be running outdated, known-vulnerable versions of software.



COMPROMISED SECRETS

API keys and database passwords must be kept secure to prevent attackers gaining access.

SEGREGATE CONTAINER GROUPS WITH VMs



DEFANG SETUID/SETGID BINARIES



BE AWARE OF CPU SHARES



VERIFY IMAGES



SET CONTAINER FILE SYSTEM TO READ-ONLY



SET A USER



DO NOT USE ENVIRONMENT VARIABLES TO SHARE SECRETS



DO NOT RUN CONTAINERS WITH THE --privileged FLAG



TURN OFF INTER-CONTAINER COMMUNICATION



SET VOLUMES TO READ-ONLY



SET MEMORY LIMITS



DO NOT INSTALL UNNECESSARY PACKAGES IN THE CONTAINER



Source: Container-Solutions.com

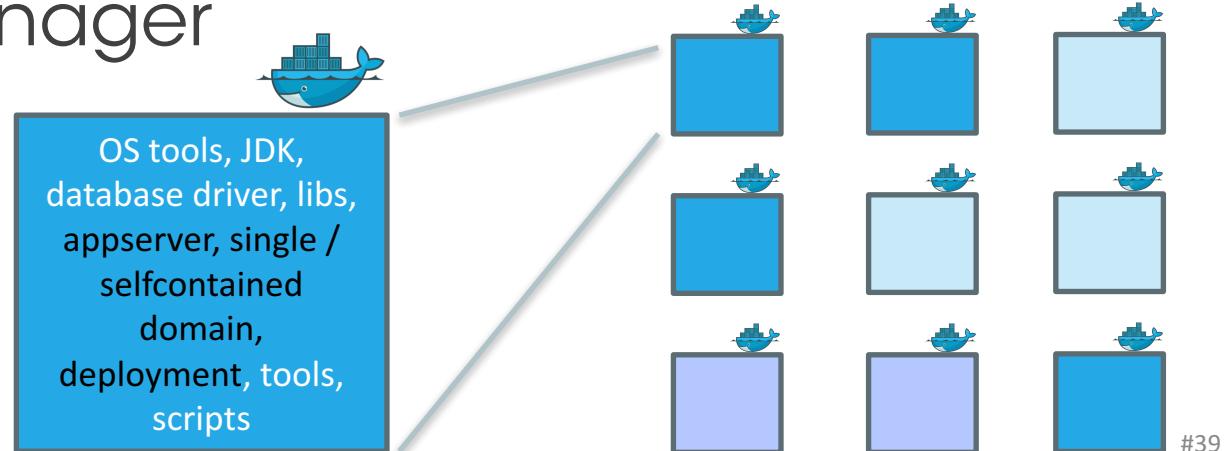
Conclusion

- You have to deal with Docker security depending on your use case
- Note: Public PaaS are not just spinning up Docker containers they use SELinux, VMs,...
- Docker is not a risk per se
but new technology with different challenges.

Oracle Options

Docker Style

- Independent appserver in container
- Microservices / 12-factor app architecture
- Just add your favorite Docker cluster manager



WebLogic Example

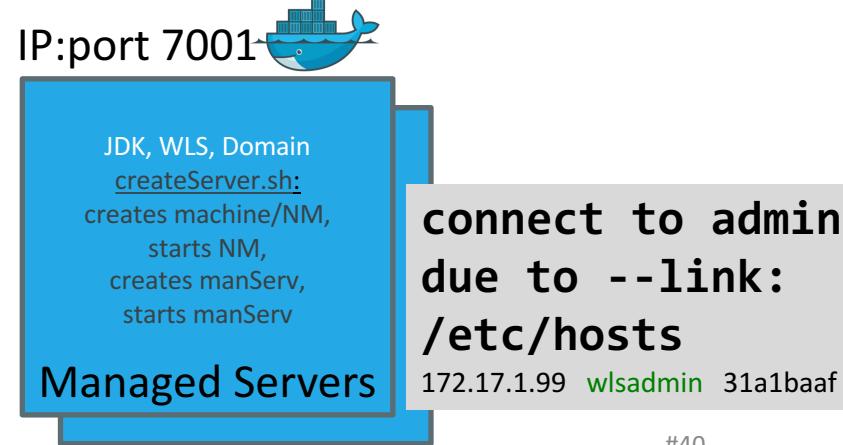
```
$docker run -d -p 8001:8001  
--name=wlsadmin  
fmdom1  
startWebLogic.sh
```



OLD STYLE!
Use **Networks** now...

--link

```
$docker run -d  
--link wlsadmin:wlsadmin  
fmdom1  
createServer.sh
```



Official Support

Oracle Product	Official Support
GlassFish	
MySQL	yes
NoSQL	
OpenJDK	
Oracle Linux	yes
Oracle Coherence	yes
Oracle Database	yes
Oracle HTTP Server	yes
Oracle JDK	yes
Oracle Tuxedo	yes
Oracle WebLogic 	yes

Oracle support does not require you to use the provided Docker files

Oracle and Docker

Different approaches:

1. DIY: Dockerfile from Oracle github
2. Oracle Docker registry
3. Oracle Container Cloud Service

DIY: Github

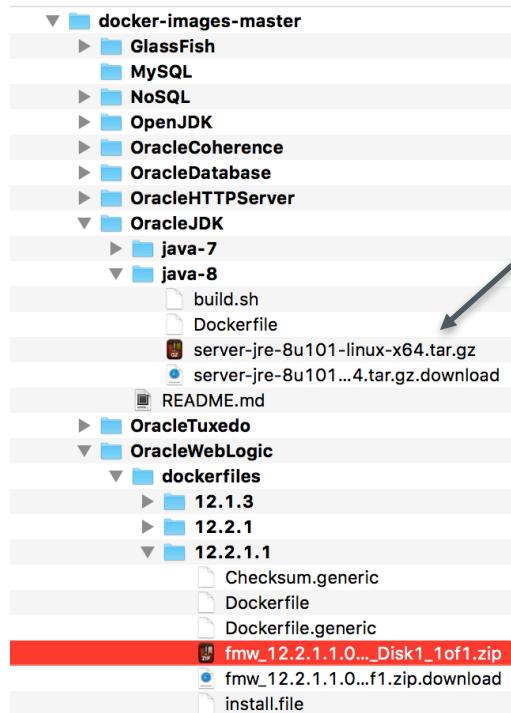
- Github repo with set up scripts based on Oracle Linux
- Docker is a supported environment for WebLogic 12.1.3+

The screenshot shows a GitHub repository page for the 'oracle/docker' repository. The repository name is at the top left, followed by a search bar and navigation links for Explore, Features, Enterprise, and Blog. Below the header is a pull request summary for 'Update to Weblogic 12.1.3 update2 dev and OpenJDK 7u79' by 'asziranyi' on May 11. The main content area displays the 'master' branch of the 'docker/OracleWebLogic/dockerfiles/12.1.3' directory. The files listed are .gitignore, Checksum.developer, Checksum.generic, Dockerfile.developer, Dockerfile.generic, and fmw_12.1.3.0.0_wls.jar.download, each with a brief description of its purpose.

File	Description
.gitignore	Update to Weblogic 12.1.3 update2 dev and OpenJDK 7u79
Checksum.developer	Update to Weblogic 12.1.3 update2 dev and OpenJDK 7u79
Checksum.generic	Update to Weblogic 12.1.3 update2 dev and OpenJDK 7u79
Dockerfile.developer	Update to Weblogic 12.1.3 update2 dev and OpenJDK 7u79
Dockerfile.generic	Update to Weblogic 12.1.3 update2 dev and OpenJDK 7u79
fmw_12.1.3.0.0_wls.jar.download	project restructured

<https://github.com/oracle/docker-images>
munz & more

Just Drop Server JRE and WLS Installer



```
$ cd java-8  
$ docker build -t oracle/jdk:8 .  
Sending build context to Docker daemon 4.096 kB  
Step 1 : FROM oraclelinux:latest  
latest: Pulling from library/oraclelinux  
10ec637c060c: Downloading 4.865 MB/97.84 MB  
...
```

```
$ sh buildDockerImage.sh -g -v 12.2.1.1  
...
```

Extend the WLS image

Sample script provided:

- Dockerfile to extend WLS image
- Run WLST script to create domain
- Create boot.properties
- Expose NM, Server ports

WLS Domain Image

WebLogic Image

JDK Image

Linux Base Image

Docker Registry (New)

Docker images from Oracle container registry
(restricted to US / OZ / GB!)

Get image:

```
$ docker login container-registry.oracle.com
$ docker pull container-registry.oracle.com/java/serverjre:8
```



Thank you for accessing www.oracle.com

In compliance with U.S. and applicable Export laws we are unable to process your request. Please contact RPLS-Ops_ww@oracle.com if you believe you are receiving this notice in error.

Container Cloud Service (New)

Run your Docker in Oracle cloud with OCCS

- Dashboards
- Services
- Stacks of services
- Ressource pools
- Registries



Thank you for accessing www.oracle.com

In compliance with U.S. and applicable Export laws we are unable to process your request. Please contact RPLS-Ops_ww@oracle.com if you believe you are receiving this notice in error.

OCCS

OCCS

- Easy way to run Docker container in Oracle cloud
- No need to install Docker, buy machines etc.
- Integrates with Docker hub and Oracle Container registry
- Does not use Kubernetes or Docker Swarm



Dashboard

ORACLE® Container Cloud Service

Dashboard

Quick Start Wizard

Deployments 0 OK There are no running deployments

Hosts 1 OK All hosts are active and reachable.

Resource Pools 3 Healthy

Name	Hosts	Memory	CPU
default	1	8%	0%
Development	0	0%	0%
Production	0	0%	0%

Services 32
Stacks 4
Deployments 0
Resource Pools 3

Containers 0
Images 3
Hosts 1

Services

Services

Quick Start Wizard

Actions				ID	Name ▲	Description
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		alpine-linux	Alpine Linux	Runs an example alpine linux container that executes for 10 minutes and prints out a hello world message to the terminal. This example is provided as-is for educational purposes and should not be used in production.
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		apache	Apache	A Docker image with Apache and PHP. After the container deploys successfully, visit port 9001 on the host to see the phpinfo page. This example is provided as-is for educational purposes and should not be used in production.
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		busybox	BusyBox	Runs a BusyBox image that executes for 10 minutes and prints out a hello world to the terminal. This example is provided as-is for educational purposes and should not be used in production.
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		cadvisor	cAdvisor	A cAdvisor container that runs on port 8080 on the host. By default, this image will be deployed to all hosts. This example is provided as-is for educational purposes and should not be used in production.
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		cassandra	Cassandra	The officially supported Apache Cassandra image. This example is provided as-is for educational purposes and should not be used in production.
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		datadog	Datadog	The officially supported Datadog image. You will need to add your API key to publish data to your datadog account. This example is provided as-is for educational purposes and should not be used in production.
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		golang	Go Example Application	An example golang container that runs for 10 minutes and prints out a hello world to the terminal. This example is provided as-is for educational purposes and should not be used in production.
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		haproxy	HAProxy	An example HAProxy service. This serves as a building block to a larger deployments as shown in the stacks tab for the loadbalancers. Access the stats page at http://YOUR_HOST:1936/haproxy?stats . This example is provided as-is for educational purposes and should not be used in production.
<button>Deploy</button>	<button>Edit</button>	<button>Remove</button>		helloworld	Hello World Application	A simple hello world web application that runs on port 9000 on the host. This example is provided as-is for educational purposes and should not be used in production.

Ressource Pools

ORACLE® Container Cloud Service

Get Help ▾ Settings ▾ Logout

Resource Pools

Host Pools

New Host Pool

Host Pool	Active	Inactive
default	1	0
Development	0	0
LoadTest	0	0
Production	0	0

v16.4.5-1014 ©2016 Oracle and/or its affiliates. All rights reserved.

Service Editor

Service Editor

Service Name	Service ID	Health Check
grafana	grafana	No health checks defined. Add
Service Description <small>Optional</small>		
<input type="text"/>		
Scheduler	Availability	
Random - Starts N containers	per-pool (across hosts in thi	
Builder	Docker Run	YAML

Docker Run Command
This is a `docker run` command that can be executed to launch this service manually through Docker. Paste a `docker run` command here to turn it into a service.

The current version of Docker supported is 1.10.3. To learn more about supported Docker Run commands, click [here](#).

```
docker run \
-e="occs:availability=per-pool" \
-e="occs:scheduler=random" \
-e="GF_INSTALL_PLUGINS=grafana-clock-panel,grafana-simple-json-
datasource,raintank-worldping-app" \
-grafana/grafana"
```

Docker Run Validation:
✓ No validation errors



Deployment Pulls Image

Deployments ▶ grafana

Service **grafana** SERVICE

Status **Starting...**

Deployment ID **grafana-20161217-181247**

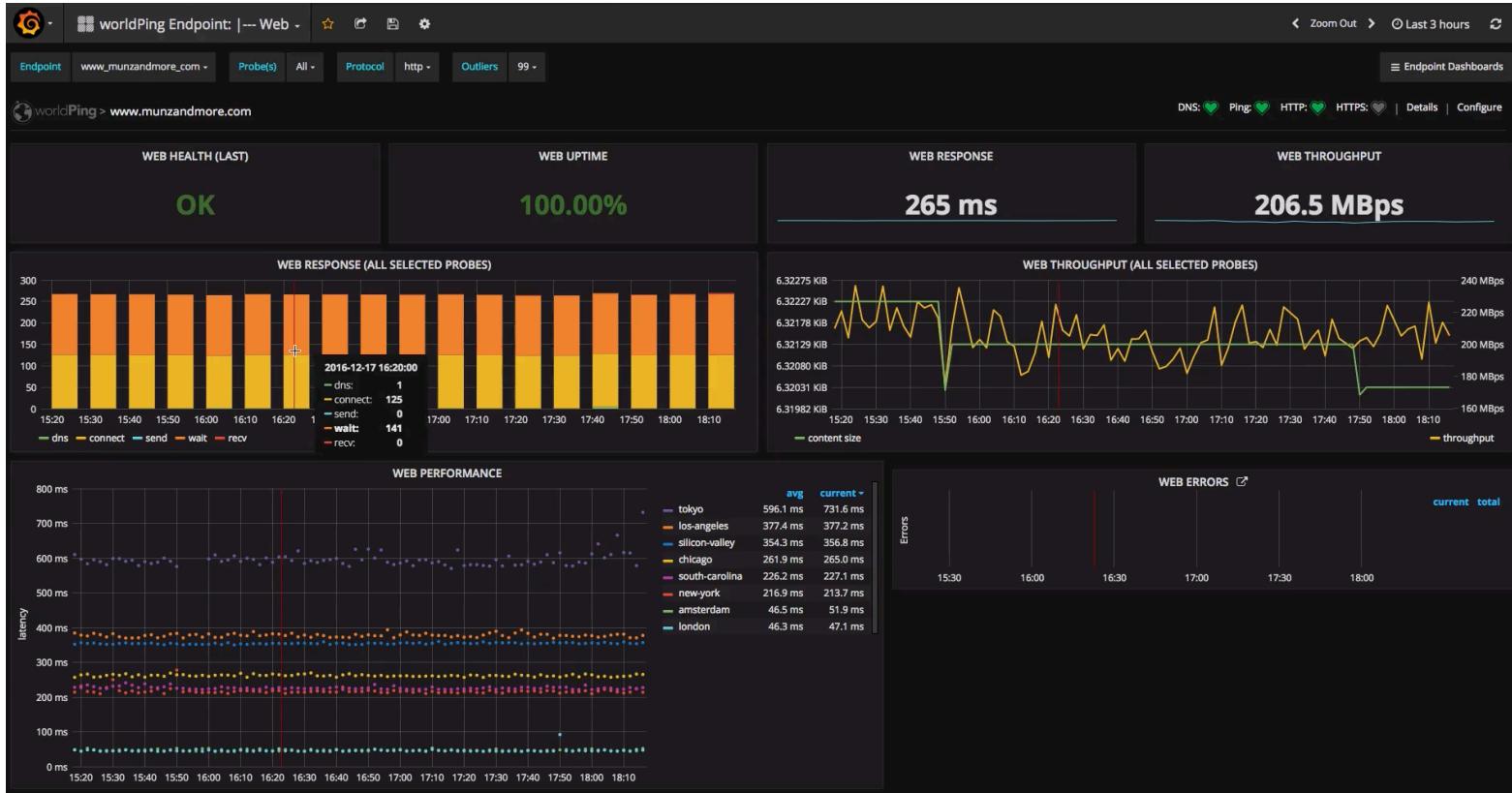
Resource Pool **default**

Started Dec 17, 2016 6:12 PM +01:00

Change Scaling **Stop** **Start**

Containers	Stack YAML	Webhook		
grafana				
Status	Container Name	Hostname	Uptime	Health Check
PENDING	<pre>pull image grafana/grafana layer latest: Pulling from grafana/grafana</pre>			Details

Running Grafana in OCCS



Webcasts on Youtube

The screenshot shows the OCCS dashboard with the following key metrics:

- Deployments:** 0 (OK)
- Hosts:** 1 (OK, 1 Active)
- Resource Pools:** 3 (default, Development, Production, all healthy)
- Services:** 32
- Images:** 4
- Deployments:** 0
- Resource Pools:** 3

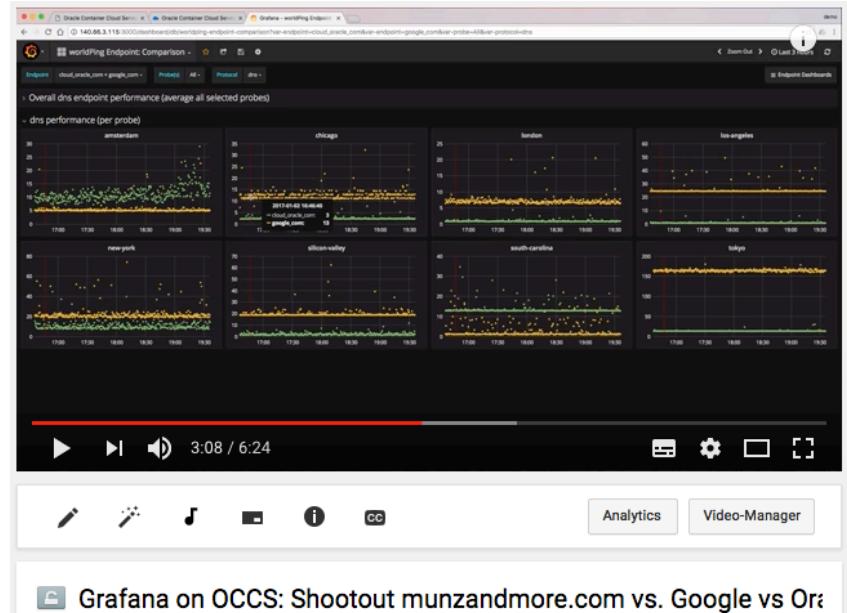
Current Status: There are no problems to report.

Video controls at the bottom indicate a video length of 6:41.

Analytics and Video-Manager buttons are visible at the bottom right.

Header: Oracle Container Cloud Service (OCCS)

<https://www.youtube.com/watch?v=YFWAUEjtTpk>

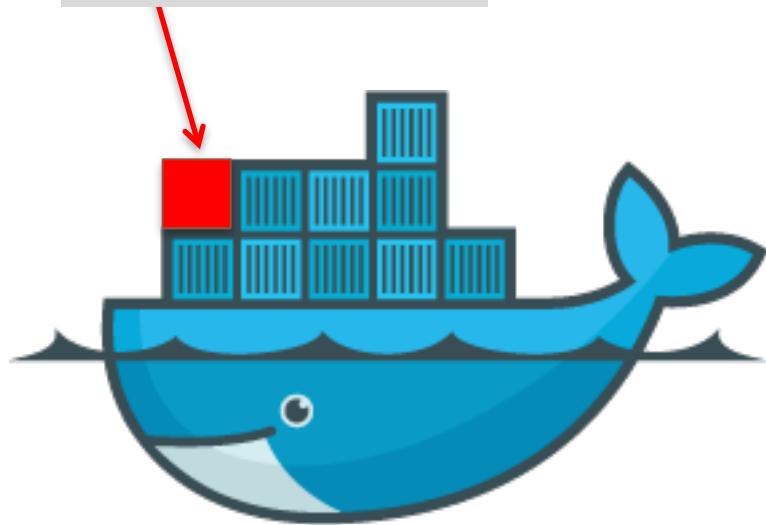


<https://www.youtube.com/watch?v=aRj0WK6uids>

Is Docker the new PaaS?

Docker in Production?

WebLogic
in a Docker
Container!

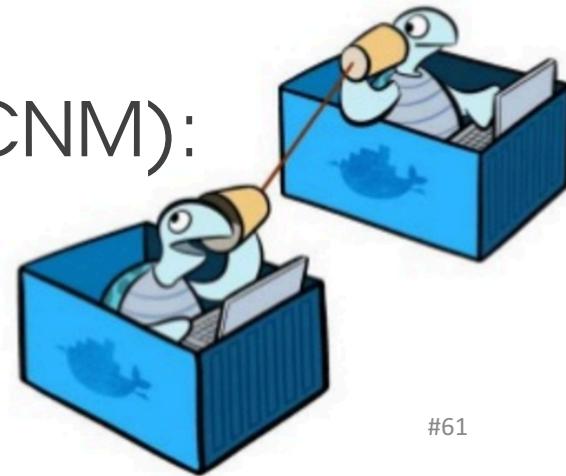


docker

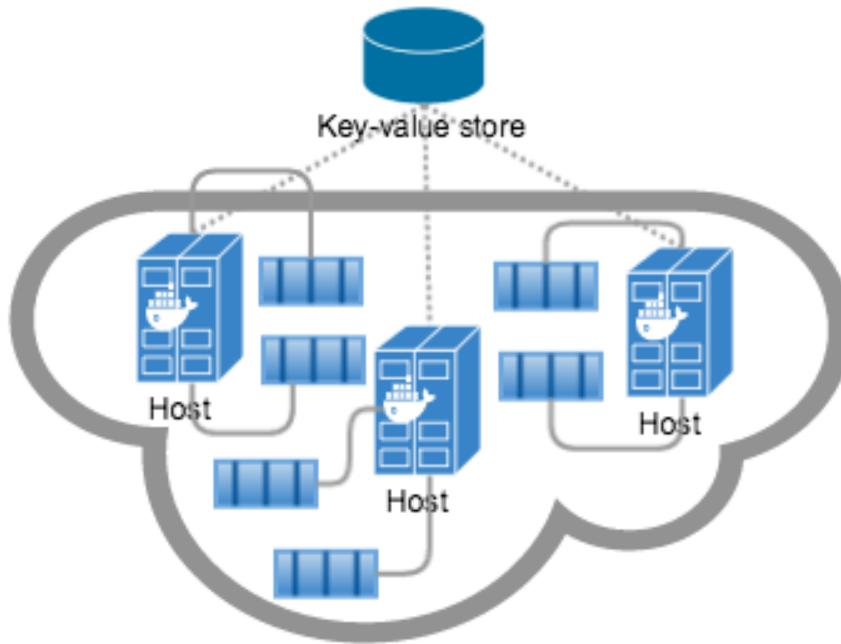
Docker Networking

Networking: Facts to Know

- Docker --link only works on single host
-> regarded as deprecated now
- Networking supported since Docker 1.9
- SDN network that spans hosts:
Libnetwork implements
Container Networking Model (CNM):
Endpoint / Network / Sandbox



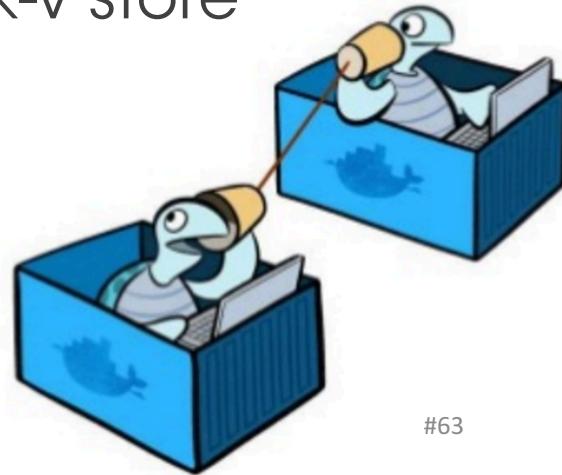
Overlay Network



```
docker network create -d overlay
```

Networking: Facts to Know

- Integrated with Compose / Swarm
- Two types:
 - Bridge: Single host
 - Overlay: Multiple hosts, based on k-v store



Orchestration / Cluster Manager

Cluster Manager Options

	On Premise	Cloud
Docker Swarm	Docker 1.12	Docker DC (EE)
Kubernetes	DIY	Google
Oracle	Oracle Registry / Git	Oracle CCS

Kubernetes (K8s)



- Kicked off by Google's borg
- Most active github project
- Orchestration for containers, e.g. Docker
- Declarative configuration
- Service discovery
- Rolling upgrades

K8s

- YAML configuration
- Pod: Container(s) sharing IP, network, filesystem
 - IP is ephemeral
 - Uses Labels
- Replication Controller -> Replica Set
 - Manages PODs (restarts replicas based on labels)
- Services
 - Proxy for pod
 - Permanent IP

Kubernetes

- Not easy to install / manage
- Recommended for local: minicube
- Cloud: Google container engine
- Oracle: somehow announced (?)
- OCCS is not K8s

Docker Swarm

- Native Docker cluster with same API as a single engine
- Fast provisioning, about 500 msec
- Scheduling: spread, binpack, rand
- No insecure mode ☺



Docker Swarm

Since Docker 1.12

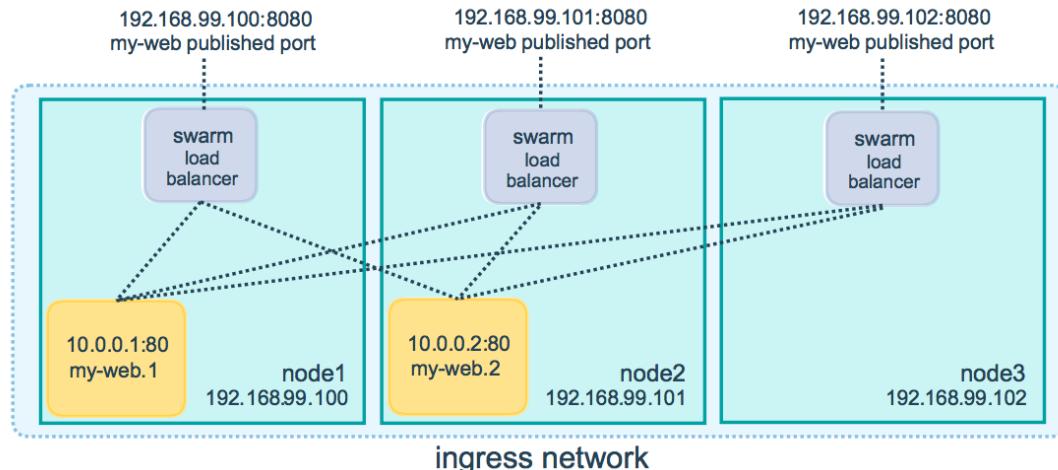
- Swarm is merged with Docker engine:
 - Load balancer included
 - Service discovery
 - Cluster scheduler
- > Swarm has now more features similar to Kubernetes
-> easier to get started

Step to Create a Swarm

1. Create Swarm master node
2. Create worker nodes
3. Create / scale / update service
4. Use routing mesh

Swarm Routing Mesh

- All nodes participate in routing mesh
- Load balancing for services
- Service discovery

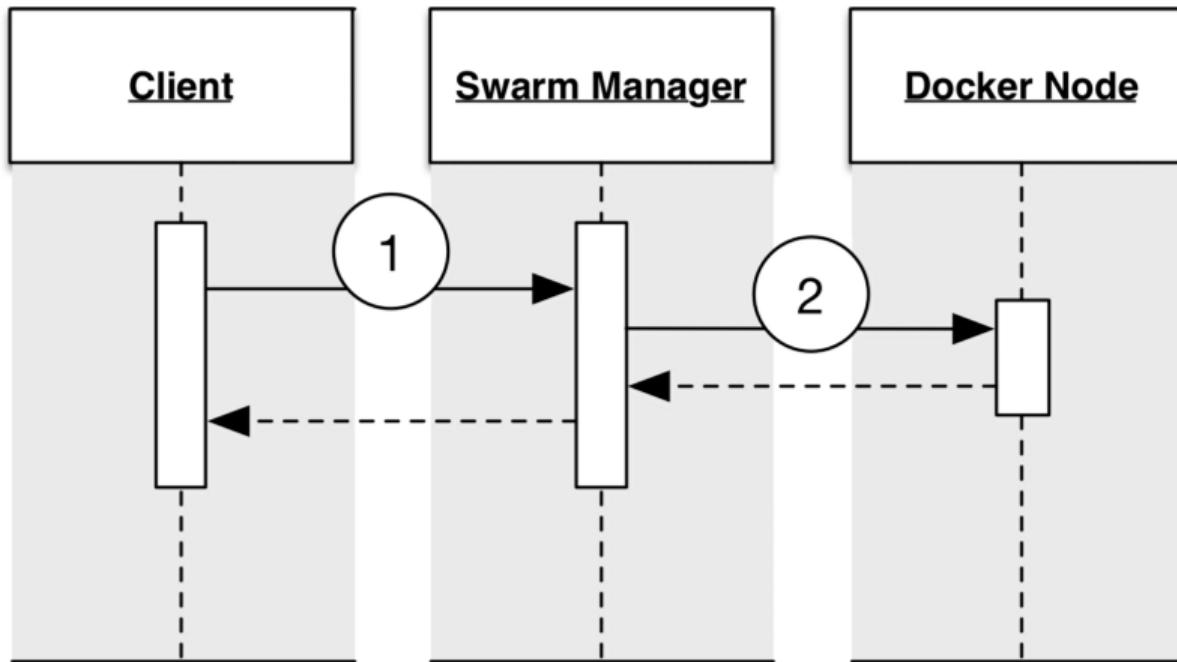


[https://docs.docker.com/engine/swarm/ingress/
#publish-a-port-for-tcp-only-or-udp-only](https://docs.docker.com/engine/swarm/ingress/#publish-a-port-for-tcp-only-or-udp-only)

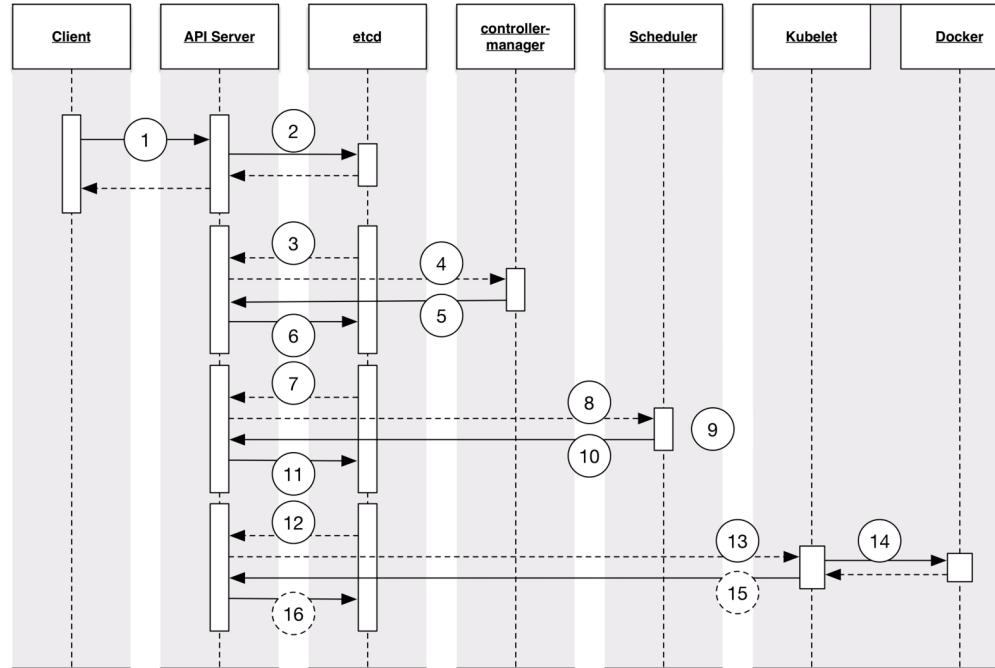
Kubernetes or Swarm?

- Swarm wasn't impressive when released, but this has changed now
- Swarm is much easier to understand and to operate
- Swarm covers a lot of what K8s does
- Swarm is tightly linked to Docker API
-> harder to replace Docker with e.g. Rocket

Docker Swarm



Google Kubernetes



Docker Containers in Kubernetes

```
frank — /home/pirate — ssh pirate@mini0 — 183x51 — #2
home/~

$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
 NAMES
85e3f09618b4        gcr.io/google_containers/kubedns-arm:1.9   "/kube-dns --domain..."   About an hour ago   Up About an hour
k8s_kube-dns_8872a1e7_kube-dns-4211557627-hvt10_kube-system_93f10303-f061-11e6-8748-b827eb0d0586_f9af066e
f78f04374e51        gcr.io/google_containers/exehealthz-arm:1.2    "/exehealthz '--c..."   About an hour ago   Up About an hour
k8s_healthz_85e03f2c_kube-dns-4211557627-hvt10_kube-system_93f10303-f061-11e6-8748-b827eb0d0586_b1de63d2
a64079451af8        gcr.io/google_containers/dnsmasq-metrics-arm:1.0   "/dnsmasq-metrics ..."  About an hour ago   Up About an hour
k8s_dnsmasq-metrics_6b5e5ecf_kube-dns-4211557627-hvt10_kube-system_93f10303-f061-11e6-8748-b827eb0d0586_fdc8128
ea6b6761c6fa        gcr.io/google_containers/kube-dnsmasq-arm:1.4   "/usr/sbin/dnsmasq..."  About an hour ago   Up About an hour
k8s_dnsmasq_e7bb89d9_kube-dns-4211557627-hvt10_kube-system_93f10303-f061-11e6-8748-b827eb0d0586_f2b2483
b9691378d692        gcr.io/google_containers/pause-arm:3.0          "/pause"                About an hour ago   Up About an hour
k8s_POD_daf6fe110_kube-dns-4211557627-hvt10_kube-system_93f10303-f061-11e6-8748-b827eb0d0586_58fbddc4
8f1b80796c72        quay.io/coreos/flannel:v0.7.0-arm           "/opt/bin/flanneld..."  About an hour ago   Up About an hour
k8s_kube-flannel_7bc68b9f_kube-flannel-ds-3q4f9_default_12cf3c58-f062-11e6-8748-b827eb0d0586_af646301
a1c1350103ec        alexelliz/arm-pingcurl@sha256:934aa8636be7ad7bde3888519c387cbaa432f6713dd3253ffec39cb9496f65f0  "ping google.com"
ping1.2.zgr2by0v024mk16c8tdezigr4
7dc272d8fed8        quay.io/coreos/flannel:v0.7.0-arm           "/bin/sh -c 'set -..."  About an hour ago   Up About an hour
k8s_installer_cni_6fe287d4_kube-flannel-ds-3q4f9_default_12cf3c58-f062-11e6-8748-b827eb0d0586_70f6557d
2bf83cd5d757        gcr.io/google_containers/kube-discovery-arm:1.0  "/usr/local/bin/ku..."  About an hour ago   Up About an hour
k8s_kube-discovery_fd59cadf_kube-discovery-1659614412-pz8ch_kube-system_519ff412-f061-11e6-8748-b827eb0d0586_2fb60162
f772670e4370        gcr.io/google_containers/kube-proxy-arm:v1.5.2   "/kube-proxy --kube..."  About an hour ago   Up About an hour
k8s_kube-proxy_d733b4b_kube-proxy-6218g_kube-system_93d408a0-f061-11e6-8748-b827eb0d0586_cb76db7e
21d731a590da        gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up About an hour
k8s_dummy_8a021bad_dummy_2501624643-2sv16_kube-system_4eb69889-f061-11e6-8748-b827eb0d0586_b43b88df
f096faeaed7b4       gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up About an hour
k8s_POD_daf6fe110_kube-proxy-6218g_kube-system_93d408a0-f061-11e6-8748-b827eb0d0586_dab5fb6d
fb8c62a0b522        gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up About an hour
k8s_POD_daf6fe110_kube-discovery-1659614412-pz8ch_kube-system_519ff412-f061-11e6-8748-b827eb0d0586_b5830b6e
fb743775f867        gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up 14 hours
k8s_POD_daf6fe110_kube-flannel-ds-3q4f9_default_12cf3c58-f062-11e6-8748-b827eb0d0586_ae98df5a
744beeed5339        gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up 14 hours
k8s_POD_daf6fe110_dummy_2501624643-2sv16_kube-system_4eb69889-f061-11e6-8748-b827eb0d0586_828f3f1b
4f4ae57840d8        gcr.io/google_containers/etcd:3.0.14_kubeadmin   "etcd --listen-cl..."  14 hours ago      Up 14 hours
k8s_etcd_ff4b9813_etcd-minio_kube-system_2bbfa1c47be87a58102b5ec053341827_3bb2f1ae
c1d25683bfff1       gcr.io/google_containers/kube-scheduler-arm:v1.5.2   "kube-scheduler --..."  14 hours ago      Up 14 hours
k8s_kube-scheduler_690d6f5_kube-scheduler-mini0_kube-system_72bf0117ced4dcc9ed6a7e9c8c54bba1_5288c935
218396698f2f        gcr.io/google_containers/kube-apiserver-arm:v1.5.2   "kube-apiserver --..."  14 hours ago      Up 14 hours
k8s_kube-apiserver_1315f351_kube-apiserver-mini0_kube-system_2d18fe099ae9677dc2c122348b1e3ade_c1cfbf44
2880227a6ab6        gcr.io/google_containers/kube-controller-manager-arm:v1.5.2   "kube-controller-m..."  14 hours ago      Up 14 hours
k8s_kube-controller-manager_a2a1b28a_kube-controller-manager-mini0_kube-system_bf782036e5d3e2de6654547f692d7fb2_e7b5cbc
d97792506f07        gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up 14 hours
k8s_POD_daf6fe110_kube-apiserver-mini0_kube-system_2d18fe099ae9677dc2c122348b1e3ade_35c3ba97
582c2bb57037        gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up 14 hours
k8s_POD_daf6fe110_etcd-mini0_kube-system_2bbfa1c47be87a58102b5ec053341827_6cf84ec0
05e157004175        gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up 14 hours
k8s_POD_daf6fe110_kube-controller-manager-mini0_kube-system_bf782036e5d3e2de6654547f692d7fb2_acbe46d8
68d44af734e1        gcr.io/google_containers/pause-arm:3.0          "/pause"                14 hours ago      Up 14 hours
k8s_POD_daf6fe110_kube-scheduler-mini0_kube-system_72bf117ced4dcc9ed6a7e9c8c54bba1_2df3b368
HypriotOS/armv7: pirate@mini0 ~
```

Welcome Mini ☺



Design Goals for Mini

- Hands-on Raspi experience
- Kubernetes setup -> Docker Swarm setup
- HA / failover / rolling updates
- Build something cool

Design Decisions for Mini

- Hypriot Linux
- 4 node cluster
- ARM based -> runs ARM Docker images
- Preferred Gbit switch, but WIFI only
 - Better I/O throughput with separate WIFI chip
 - Simplicity
 - DHCP reservation bug in router -> no MAC reservation for switch
- Wifi router in hotspot mode
 - Mac OS issue with NAT
- DIY: github <https://github.com/fmunz/raspicluster>

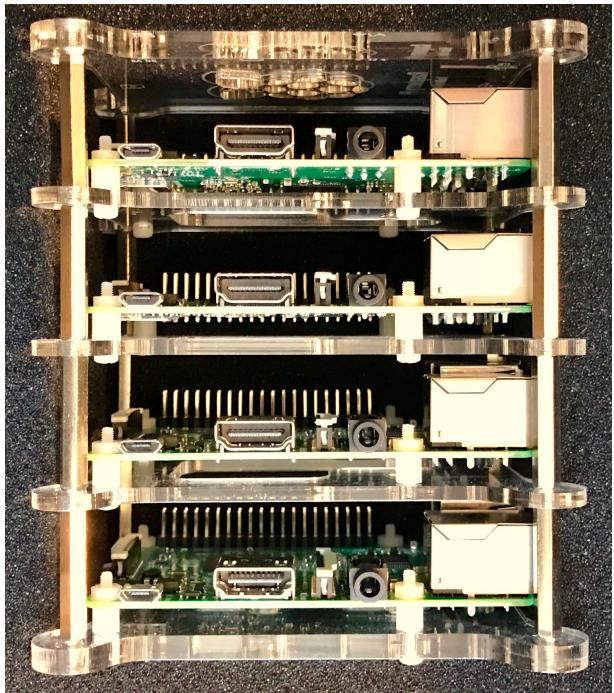
Paper bag Computer Raspberry Pi

- 1.2 GHz Quad Core ARM cortex-a53
- RAM: 1 GB LPDDR2 Speicher
- Dual Core VideoCore, 1920x1080
- Bluetooth
- 802.11 B/G WIFI
- Boots from micro SD card
- 38 €

DIY Raspi Docker Cluster



Component	Price
Raspi	4x 38€
Micro SD	4x 11€
Power	28€
Wifi	22€
Case	30€
Wires	10€



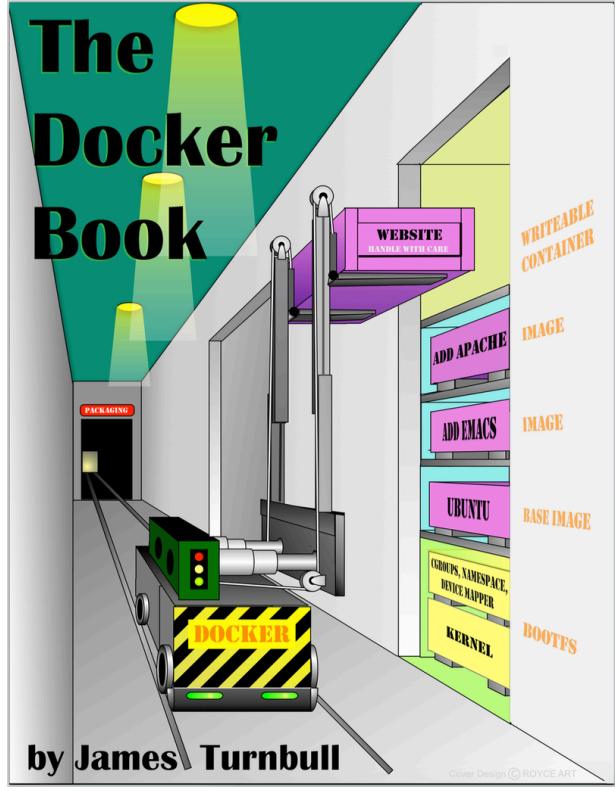
Oracle WebLogic Server on Docker Containers

ORACLE WHITE PAPER | OCTOBER 2015



Oracle Whitepaper WebLogic on Docker

<http://www.oracle.com/us/products/middleware/cloud-app-foundation/weblogic/weblogic-server-on-docker-wp-2742665.pdf>



THE DOCKER BOOK

CONTAINERIZATION IS THE NEW VIRTUALIZATION



SIMPLE

A hands-on book that teaches you Docker™.



SCALABLE

Start small with a single container and then build on what you learn to deploy multi-container applications.



UP-TO-DATE

Current, accurate and up-to-date.



DIFFERENT

Written for both developers and sysadmins with real-world examples and use cases.

Good Docker book by
J. Turnbull / \$ 9.99
(covering Docker 1.13)

Predictions

- Swarm will take its share from Kubernetes.
- You will **not** dockerize 90% of your enterprise IT in the next 24 months.
- **Docker is the new Linux.**
Be ready to experience that feeling we had with Linux 13 years ago ☺
- Sometimes Docker is the new PaaS.

Conclusion

- Docker is used in production
- Docker, but more so cluster managers are still evolving
- Docker is not a security risk, but tick off the referenced security checklist
- Oracle caught the trend early - good!
- Many Oracle products supported already, more to come?

TL;DR @docker / #cloud

works for me: solved / cross cloud
/ not everything must be a
container / just Docker is not
enough / #swarm is the easier #k8s
/ not a security risk / many
products offered as images /
official Oracle support / consider
cloud services.

Don't be
shy



e www.munzandmore.com/blog

f facebook.com/cloudcomputingbook

f facebook.com/weblogicbook

@frankmunz

YouTube youtube.com/weblogicbook
-> more than 50 web casts

