Containers (and Docker)

A journey beyond the traditional 'hello world' examples...

11/01/2017 - I-Talent PXL

Who am I?

Containers?

Docker?

Demo (part I)

Real life use cases

Demo (part II)

Demo (part III)

Next steps?

Q&A

Who am I?

DevOps* minded open, **Cloud** focused **Open Source** enthusiast with an increasingly growing passion for **Containers**, **Orchestrators** and **Microservices**.

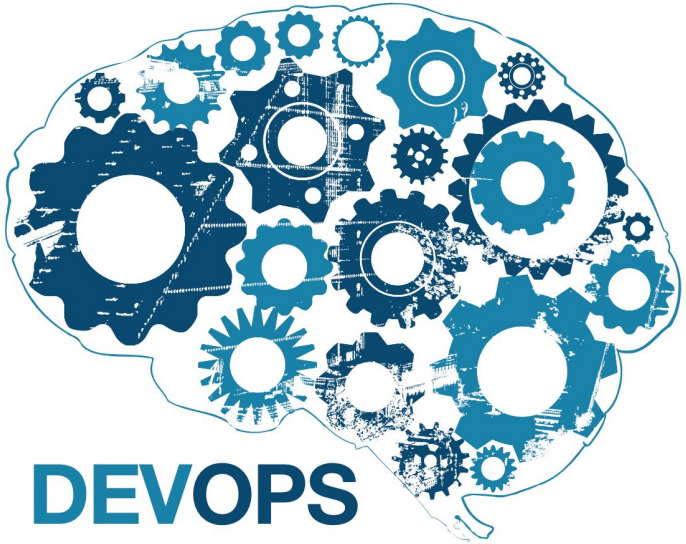
**more Ops than Dev*

Source: https://www.linkedin.com/in/trescst

gluo

*part of XPLORE GROUP POWERED BY CRONOS **

**part of CRONOS GROEP


DEVOPS

Containers?

Problem: We want to ship some hay...


MGS
GS 014
GS 014
EVERGREEN
SCANIA
114L
340

Problem: We want to ship some code...

Manual Setup

```
[root@centos-server ~]# yum install r1soft-cdp-enterprise-agent
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.hitme.net.pl
 * extras: centos.itt-consulting.com
 * updates: centos.slu.cz
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package r1soft-cdp-enterprise-agent.x86_64 0:3.18.1-16225 set to be updated
--> Processing Dependency: r1soft-cdp-agent >= 3.18.1 for package: r1soft-cdp-enterprise-agent-3.18.1-16225.x86_64
--> Processing Dependency: r1soft-setup >= 3.18.1 for package: r1soft-cdp-enterprise-agent-3.18.1-16225.x86_64
--> Running transaction check
---> Package r1soft-cdp-agent.x86_64 0:3.18.1-16225 set to be updated
--> Processing Dependency: r1soft-cdp-async-agent-2-6 >= 3.18.1 for package: r1soft-cdp-agent-3.18.1-16225.x86_64
---> Package r1soft-setup.x86_64 0:3.18.1-16225 set to be updated
--> Running transaction check
---> Package r1soft-cdp-async-agent-2-6.x86_64 0:3.18.1-16225 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                    Arch          Version              Repository    Size
================================================================================
Installing:
 r1soft-cdp-enterprise-agent x86_64       3.18.1-16225         r1soft       2.5 k
Installing for dependencies:
 r1soft-cdp-agent           x86_64        3.18.1-16225         r1soft       7.2 M
 r1soft-cdp-async-agent-2-6 x86_64        3.18.1-16225         r1soft        12 M
 r1soft-setup               x86_64        3.18.1-16225         r1soft       990 k

Transaction Summary
================================================================================
Install       4 Package(s)
Upgrade       0 Package(s)

Total download size: 21 M
Installed size: 63 M
Is this ok [y/N]:
```

difficult to replicate identical environments (documentation, human error,...)

Scripted Setup (config management)

```
- name: Update APT cache every hour
  sudo: true
  apt: update_cache=yes cache_valid_time=3600
- name: install htop
  sudo: true
  apt: pkg=htop state=latest
- name: install curl
  sudo: true
  apt: pkg=curl state=latest
- name: install make
  sudo: true
  apt: pkg=make state=latest
- name: install git
  sudo: true
  apt: pkg=git-core state=latest
```

+/- easy to replicate environments

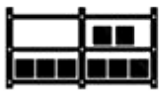
knowledge required about what is inside to create the scripts

Containers

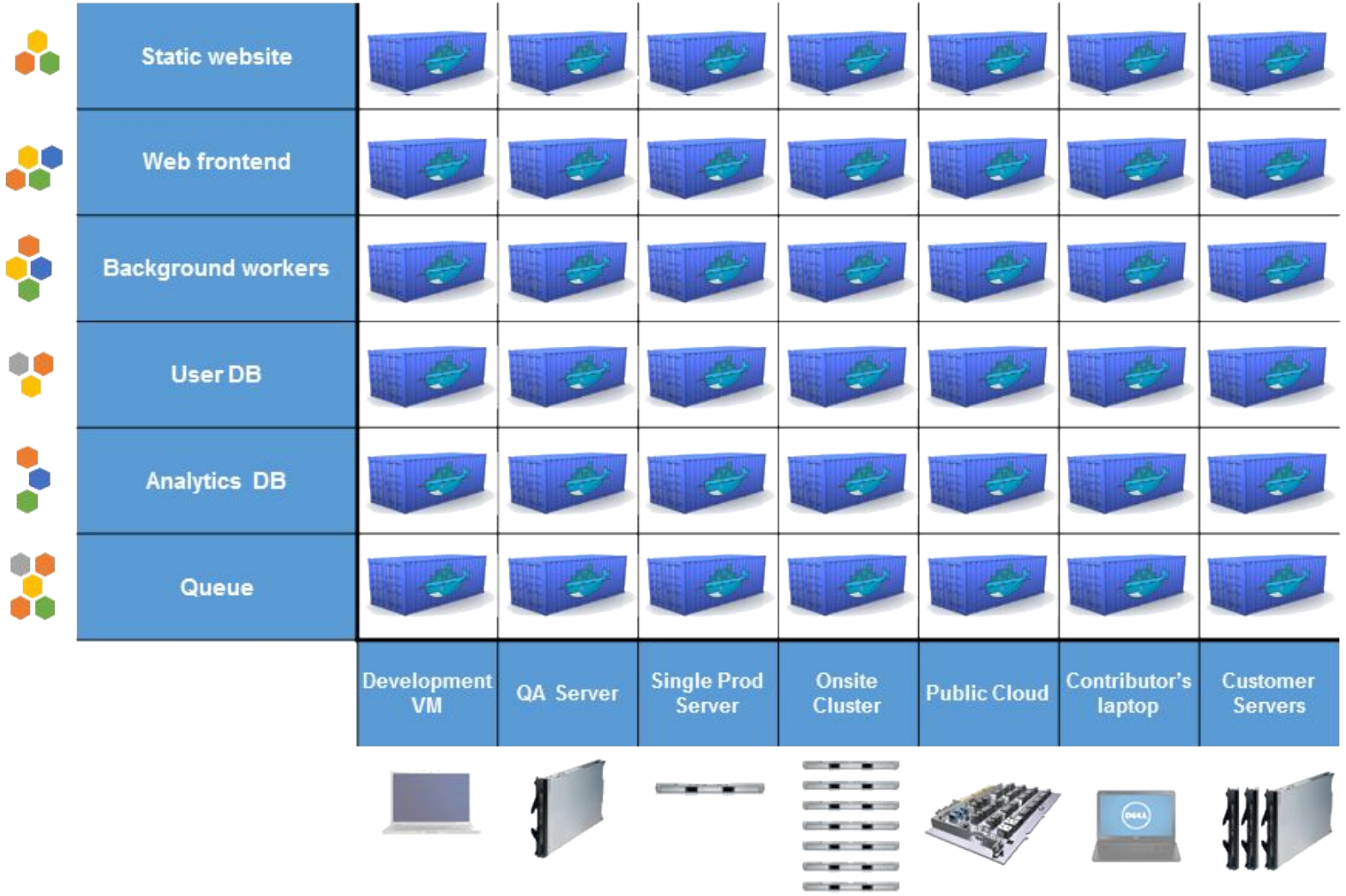
very easy to replicate (just drop it into place)

no knowledge required about what is inside

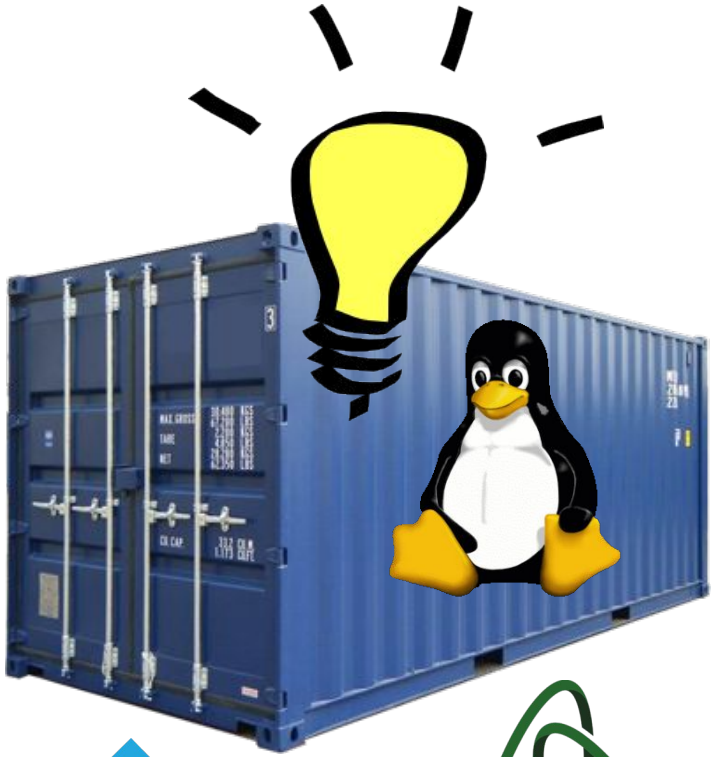
Matrix from hell...

	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?


Static website
Web frontend
Background workers
User DB
Analytics DB
Queue
Development VM
QA Server
Single Prod Server
Onsite Cluster
Public Cloud
Contributor's laptop
Customer Servers

Are containers new?



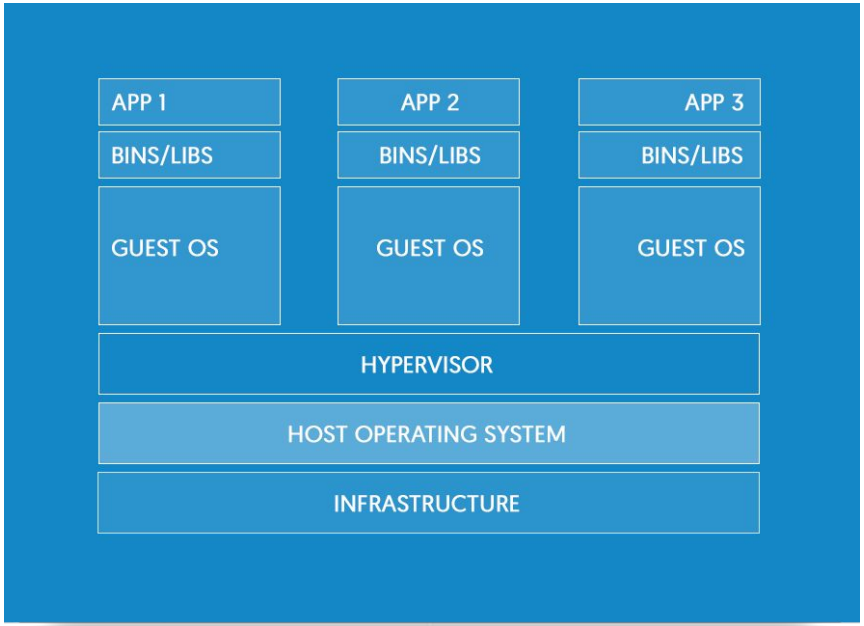
solaris

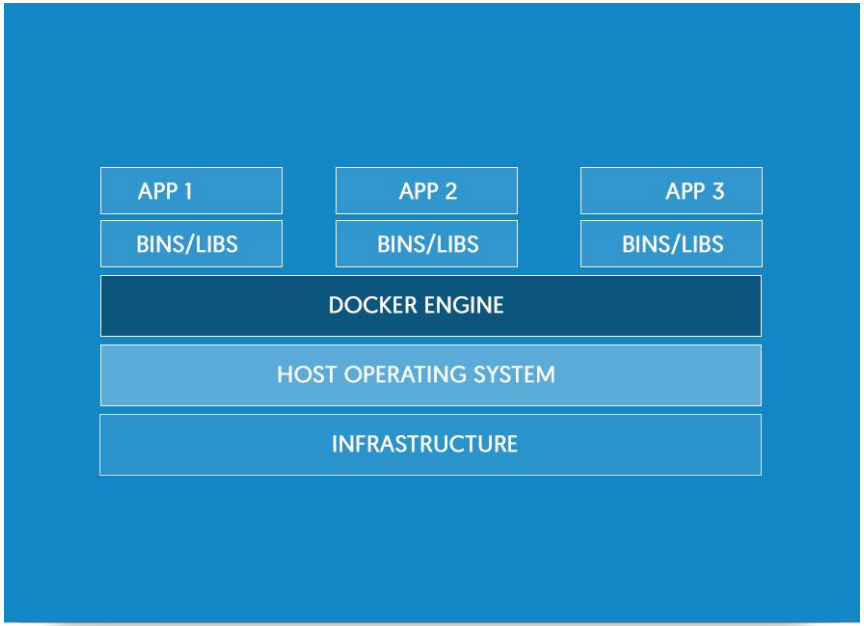
LXC

OpenVZ
Virtuozzo Containers

Containers vs. Virtual Machines

Containers vs. Virtual Machines





start-up: # minutes (boot process)
size: at least couple of GB's

start-up: less than a second (if cached)
size: couple of MB (or tens/hunderds of MB's)

What is a container?

It’s magic!

No magic at all…

Simply a combination of different kernel features that provide process isolation:

- **Namespaces**
  - “Own” network
  - “Own” process tree
  - “Own” filesystem
  - “Own” users
- **Cgroups**
  - CPU limitation
  - Memory limitation
- **“Clever” file system**

Linux/kernel/cgroup.c

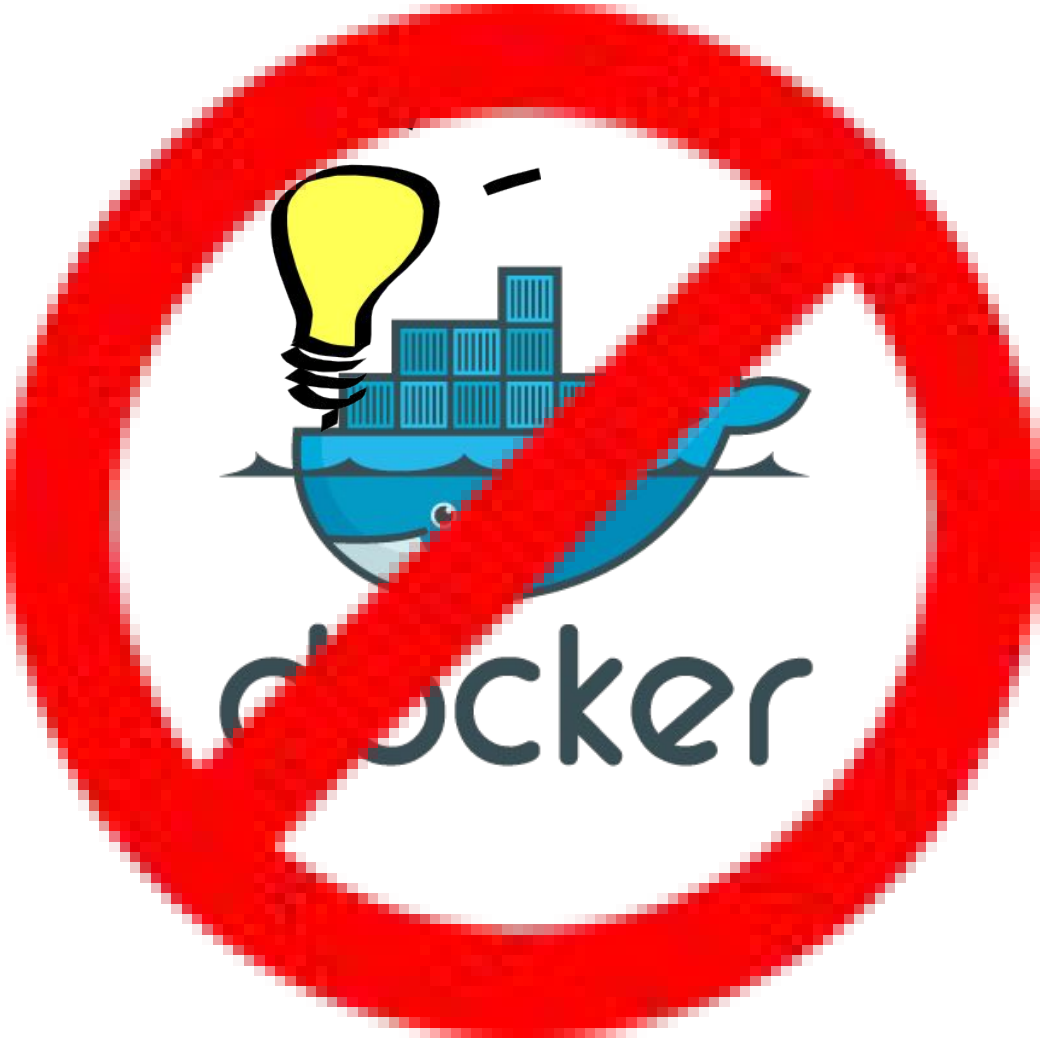
```
/*
 *  Generic process-grouping system.
 *
 *  Based originally on the cpuset system, extracted by Paul Menage
 *  Copyright (C) 2006 Google, Inc
 *
 *  Notifications support
 *  Copyright (C) 2009 Nokia Corporation
 *  Author: Kirill A. Shutemov
 *
 *  Copyright notices from the original cpuset code:
 *  --------------------------------------------------
 *  Copyright (C) 2003 BULL SA.
 *  Copyright (C) 2004-2006 Silicon Graphics, Inc.
 *
 *  Portions derived from Patrick Mochel's sysfs code.
 *  sysfs is Copyright (c) 2001-3 Patrick Mochel
 *
 *  2003-10-10 Written by Simon Derr.
 *  2003-10-22 Updates by Stephen Hemminger.
 *  2004 May-July Rework by Paul Jackson.
 *  ---------------------------------------------------
 *
 *  This file is subject to the terms and conditions of the GNU General Public
 *  License.  See the file COPYING in the main directory of the Linux
 *  distribution for more details.
 */

#define pr_fmt(fmt) KBUILD_MODNAME ": " fmt

#include <linux/cgroup.h>
#include <linux/cred.h>
#include <linux/ctype.h>
#include <linux/errno.h>
#include <linux/init_task.h>
#include <linux/kernel.h>
#include <linux/list.h>
#include <linux/magic.h>
```

Holy grail?!?

Docker?


docker

Containers before Docker

Containers with Docker

A bit of Docker history...

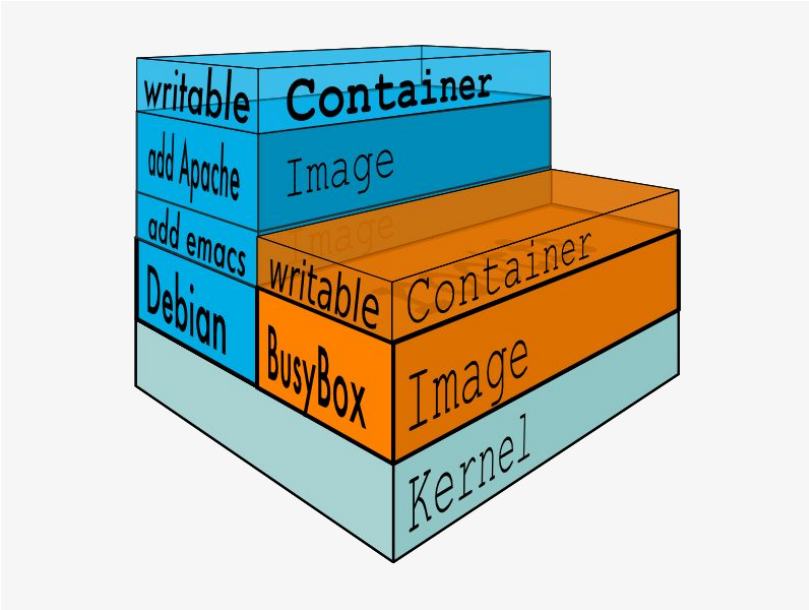
- Started as a company **dotCloud** (PaaS, cfr. Heroku)
- The dotCloud **PaaS** solution was build on container technology
- To manage/maintain their containers they created an **opensource project** called Docker
- This project became extremely popular after only short time that they completely **refocussed**
- No longer PaaS offering but completely focussed on the Docker project
- dotCloud >>> **Docker Inc**
- Docker written in **Go(lang)**
- Only **3.5 years** old
- A little under **250 employees**
- An **entire platform** developed around/for Docker

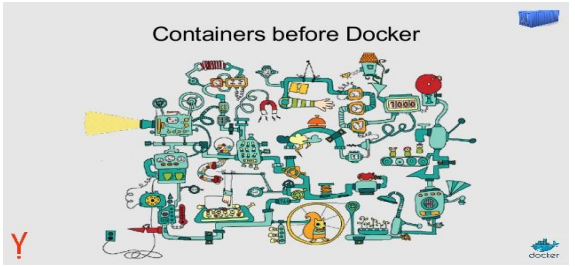
Supported by:

And many, many others...

Key components

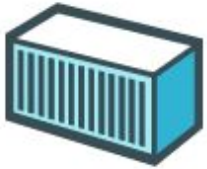
- Docker Engine:
  - heart & brains (service that needs to run on all the hosts that you want to run Docker containers on)
- Images
  - cfr. template VM
  - read-only
  - layered
- Containers
  - cfr. an active images
  - images + writeable layer
- Registry (Docker Hub)
  - cfr. repository (GitHub, Bitbucket)
  - images instead of code





Build

Develop an app using Docker containers with any language and any toolchain.



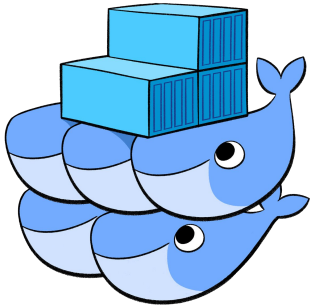
Ship

Ship the “Dockerized” app and dependencies anywhere - to QA, teammates, or the cloud - without breaking anything.

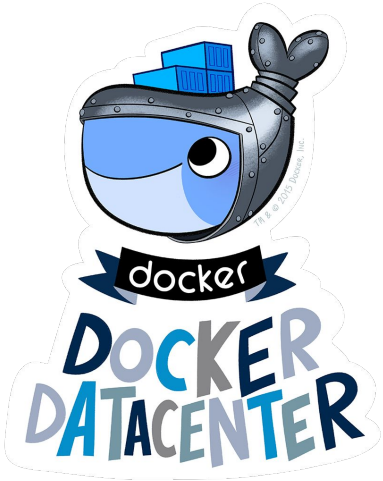
Run

Scale to 1000s of nodes, move between data centers and clouds, update with zero downtime and more.

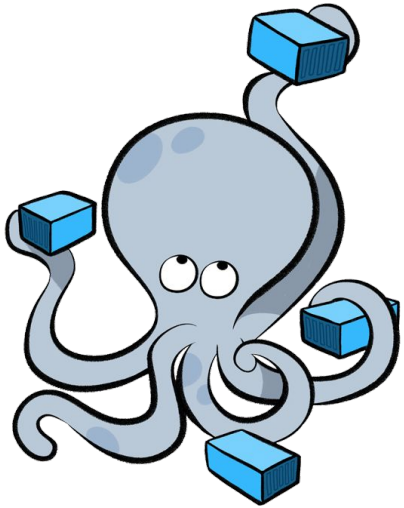
Build: **# docker build -t trescst/ticonference .**
Ship: **# docker push trescst/ticonference && docker pull trescst/ticonference**
Run: **# docker run -d trescst/ticonference**



Docker Swarm



Docker Datacenter



Docker Compose

Docker Registry

Many more to come….


CoreOS


RANCHEROS


snappy


PROJECT
ATOMIC

Demo (part I)


DEMO GODS
PLEASE LET THESE DEMOS WORK
memegenerator.net

Demo 1

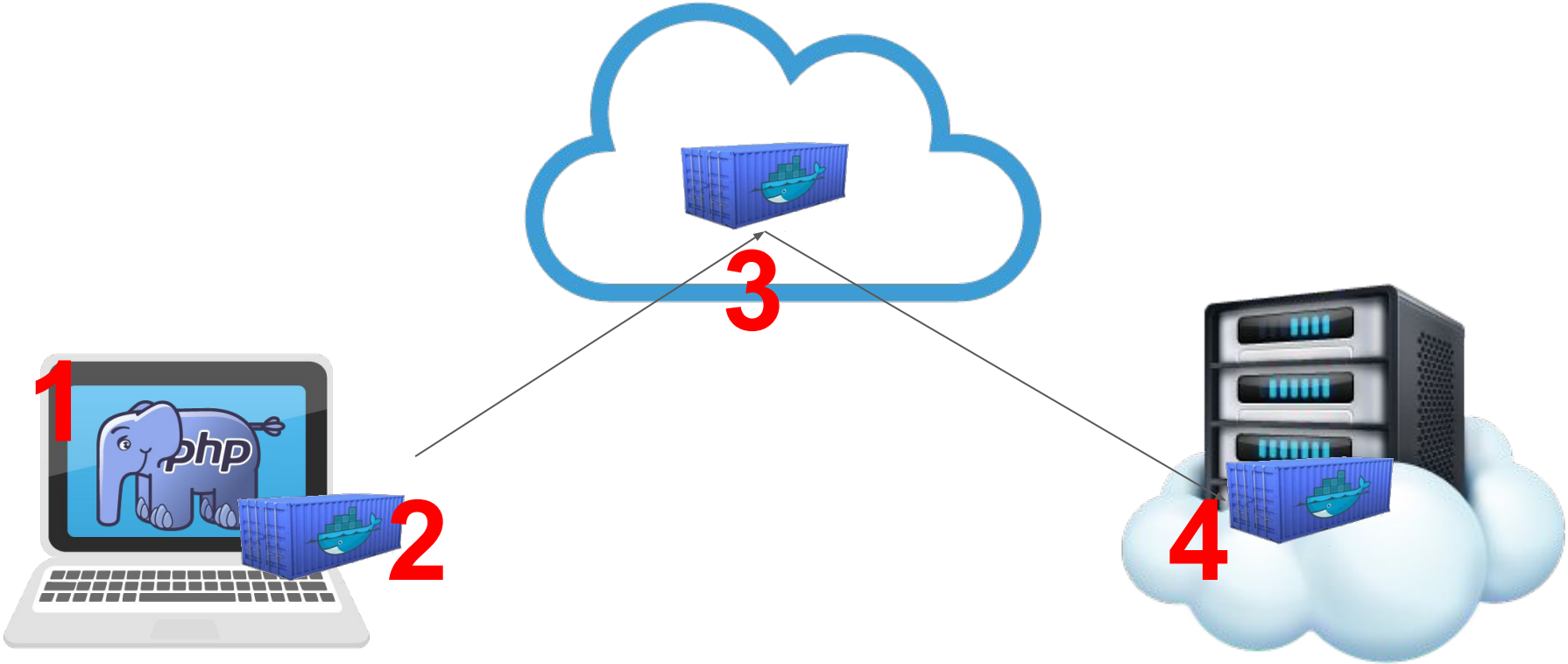
- Mac:
  - https://docs.docker.com/docker-for-mac/
    - Download **Docker.dmg**
    - Install **Docker.dmg**
- Windows:
  - https://docs.docker.com/docker-for-windows/
    - Download **InstallDocker.msi**
    - Install **InstallDocker.msi**
- Linux:
  - `curl -sSL https://get.docker.com/ | sh` (or check https://docs.docker.com/engine/installation/)

Demo 2

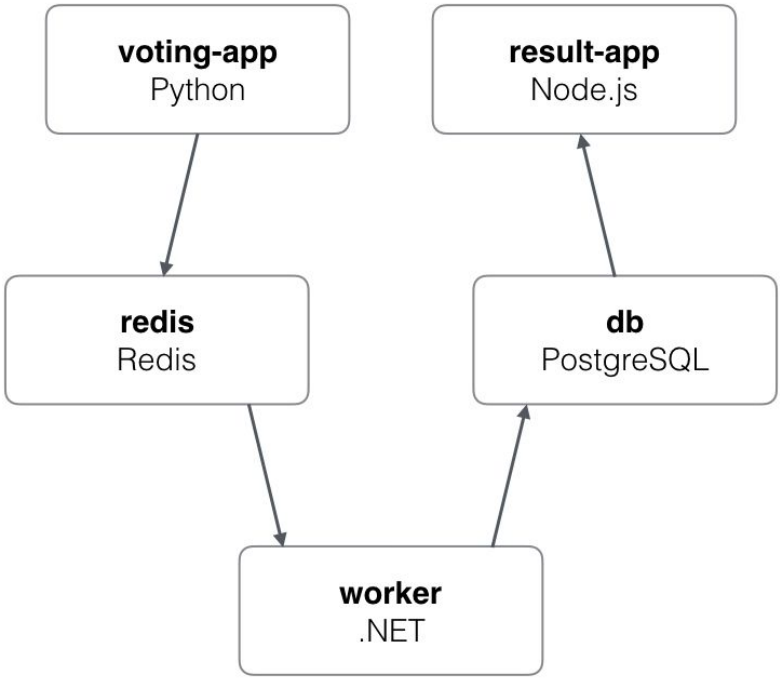
- Run different distros quickly:
  - docker run -ti ubuntu bash
    - cat /etc/debian_version
  - docker run -ti centos bash
    - cat /etc/redhat-release
- Dockerfile
  - docker run -ti ubuntu bash
    - elinks (not installed)
    - apt-get update && apt-get install elinks
    - elinks (installed)
    - exit
  - docker run -ti ubuntu bash
    - elinks (again not installed)
  - docker build -t ubuntu-with-elinks .
- rm -rf / ?
  -

Real life use cases

Develop > Build > Ship > Run



Complex environments



Demo (part II)


LET'S PRAY
TO THE DEMO GODS
memegenerator.net

Demo 3

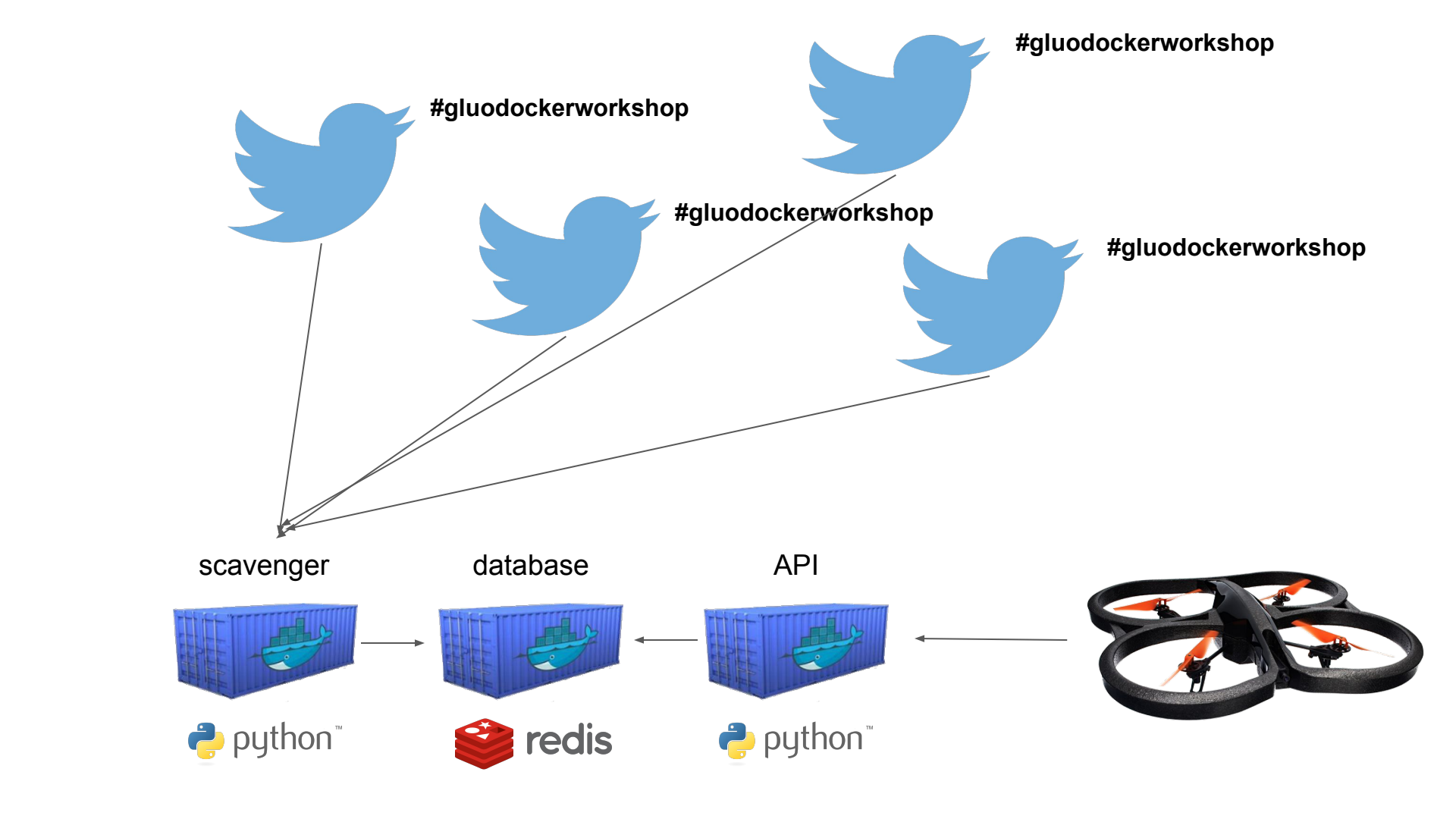
- Laptop:
  - Develop code
  - Show different PHP version (easy to switch)
  - Build
  - Ship
- Server:
  - Run!

Demo 4

- git clone https://github.com/docker/example-voting-app
- docker-compose up

Demo (part III)


TOO MUCH TIME ON MY HANDS?
OR THE APPROPRIATE SKILLS TO NOT NEED MUCH TIME?
memegenerator.net


#gluodockerworkshop
#gluodockerworkshop
#gluodockerworkshop
#gluodockerworkshop
scavenger
database
API
python
redis
python


DEMO GODS
PLEASE LET THESE DEMOS WORK
memegenerator.net


LET'S PRAY
TO THE DEMO GODS
memegenerator.net

Next steps?

Suggestions

- build your own images (using Dockerfile)
- create an application stack with Docker Compose of your Docker-ized application
- setup a Kubernetes cluster
- setup a Docker Swarm cluster

Q&A

steven@gluo.be