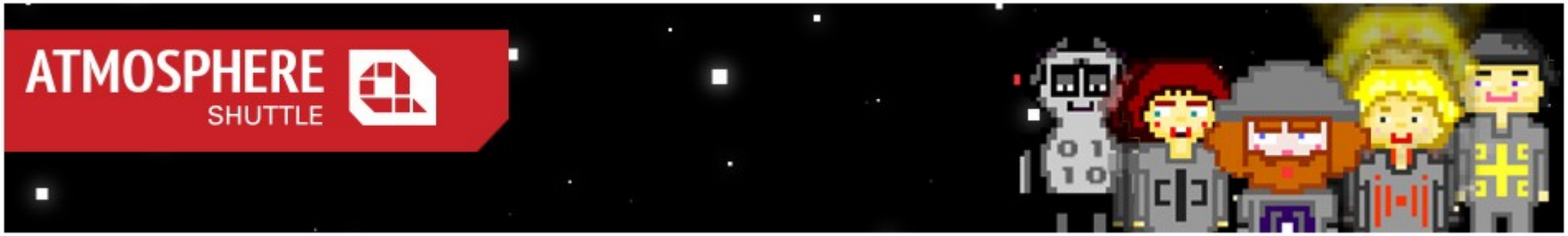
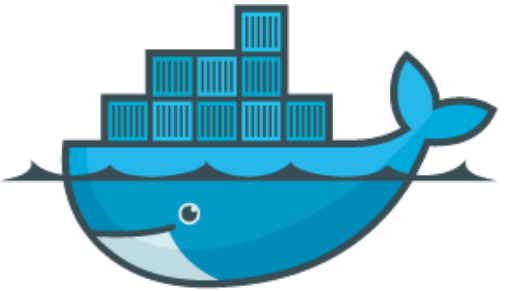


Linux containers & Devops

Maciej Lasyk

Atmosphere Shuttle #02 – Wrocław

2015-04-17



Join Fedora Infrastructure!

- learn Ansible
- learn Docker with Fedora Dockerfiles

http://fedoraproject.org/en/join-fedora

Quick survey

How many of you...

Quick survey

How many of you...

→ Knows what Docker is?

Quick survey

How many of you...

- → Knows what Docker is?
- → Played with Docker?

Quick survey

How many of you...

- → Knows what Docker is?
- → Played with Docker?
- → Runs it on production?

Why use Docker?

With Docker we can solve many problems

Why use Docker?

With Docker we can solve many problems

→ “it works on my machine”

Why use Docker?

With Docker we can solve many problems

→ “it works on my machine”

→ reducing build & deploy time

Why use Docker?

With Docker we can solve many problems

→ “it works on my machine”

→ reducing build & deploy time

→ Infrastructure configuration spaghetti – automation!

Why use Docker?

With Docker we can solve many problems

→ “it works on my machine”

→ reducing build & deploy time

→ Infrastructure configuration spaghetti – automation!

→ Libs dependency hell

Why use Docker?

With Docker we can solve many problems

→ “it works on my machine”

→ reducing build & deploy time

→ Infrastructure configuration spaghetti – automation!

→ Libs dependency hell

→ Cost control and granularity

Docker – what is it?

Docker – what is it?

“automates the deployment of any application as a lightweight, portable, self-sufficient container that will run virtually anywhere”

Docker – what is it?

Java’s promise: Write Once. Run Anywhere.

Docker – what is it?

Java’s promise: Write Once. Run Anywhere.

Even on Windows now!

https://blog.docker.com/2014/10/docker-microsoft-partner-distributed-applications/

Docker – what is it?

Is Docker is lightweight?

Docker – what is it?

Is Docker is lightweight?

```
================================================================================
 Package              Arch              Version                Repository          Size
================================================================================
Installing:
 docker-io            x86_64            1.3.0-1.fc20           updates            4.3 M
```

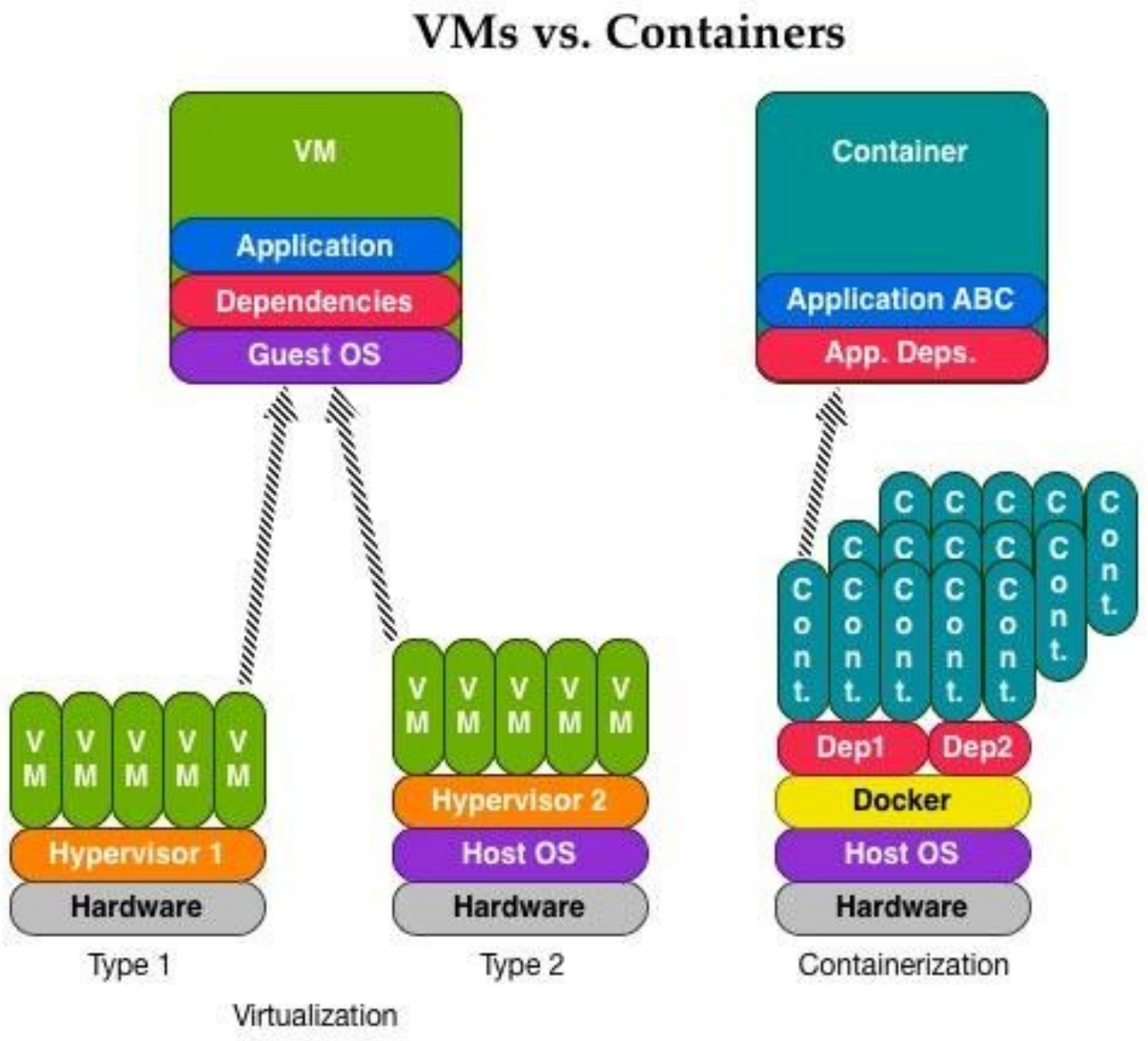
Docker – what is it?

Is Docker is lightweight?

```
================================================================================
 Package              Arch              Version                  Repository          Size
================================================================================
Installing:
 docker-io            x86_64            1.3.0-1.fc20             updates            4.3 M
```

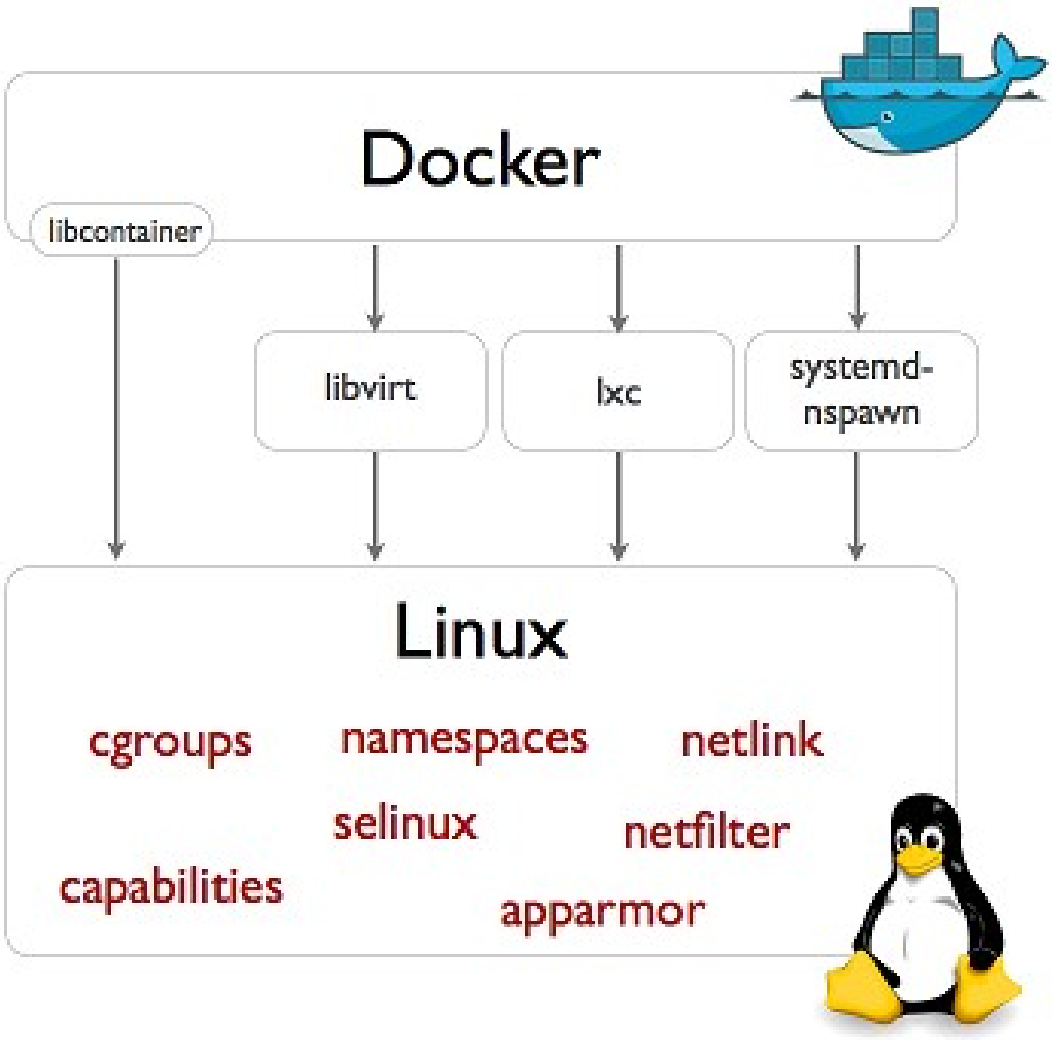
```
================================================================================
 Package              Arch              Version                  Repository          Size
================================================================================
Installing:
 docker-io            x86_64            1.5.0-2.fc21             updates            26 M
```

Docker – what is it?



http://sattia.blogspot.com/2014/05/docker-lightweight-linux-containers-for.html

Docker – how it works?



http://blog.docker.com/2014/03/docker-0-9-introducing-execution-drivers-and-libcontainer/

Docker – how it works?

→ LXC & libcontainer

Docker – how it works?

→ LXC & libcontainer

→ control groups

Docker – how it works?

→ LXC & libcontainer

→ control groups

→ kernel namespaces

Docker – how it works?

- → LXC & libcontainer
- → control groups
- → kernel namespaces
- → layered filesystem
  - → devmapper thin provisioning & loopback mounts
  - → no more AUFS (perf sucks)
  - → OverlayFS!

control groups (cgroups)

Control Groups provide a mechanism for aggregating/partitioning sets of tasks, and all their future children, into hierarchical groups with specialized behavior

control groups (cgroups)

Control Groups provide a mechanism for aggregating/partitioning sets of tasks, and all their future children, into hierarchical groups with specialized behavior

- → grouping processes
- → allocating resources to particular groups
  - → memory
  - → network
  - → CPU
  - → storage bandwidth (I/O throttling)
  - → device whitelisting

control groups (cgroups)

little demo #1

Kernel Namespaces

Providing a unique views of the system for processes.

Kernel Namespaces

Providing a unique views of the system for processes.

→ PID – PIDs isolation

Kernel Namespaces

Providing a unique views of the system for processes.

→ PID – PIDs isolation

→ NET – network isolation (via virt-ifaces; demo)

Kernel Namespaces

Providing a unique views of the system for processes.

→ PID – PIDs isolation

→ NET – network isolation (via virt-ifaces; demo)

→ IPC – won't user this

Kernel Namespaces

Providing a unique views of the system for processes.

→ PID – PIDs isolation

→ NET – network isolation (via virt-ifaces; demo)

→ IPC – won't user this

→ MNT – chroot like; deals w/mountpoints

Kernel Namespaces

Providing a unique views of the system for processes.

→ PID – PIDs isolation

→ NET – network isolation (via virt-ifaces; demo)

→ IPC – won't user this

→ MNT – chroot like; deals w/mountpoints

→ UTS – deals w/hostname

Kernel Namespaces

little demo #2

OverlayFS

+ hell fast (you'll see)

OverlayFS

+ hell fast (you'll see)
+ page cache sharing

OverlayFS

+ hell fast (you'll see)
+ page cache sharing
+ finally in upstream kernel (in rhel from 7.2, 3.18)

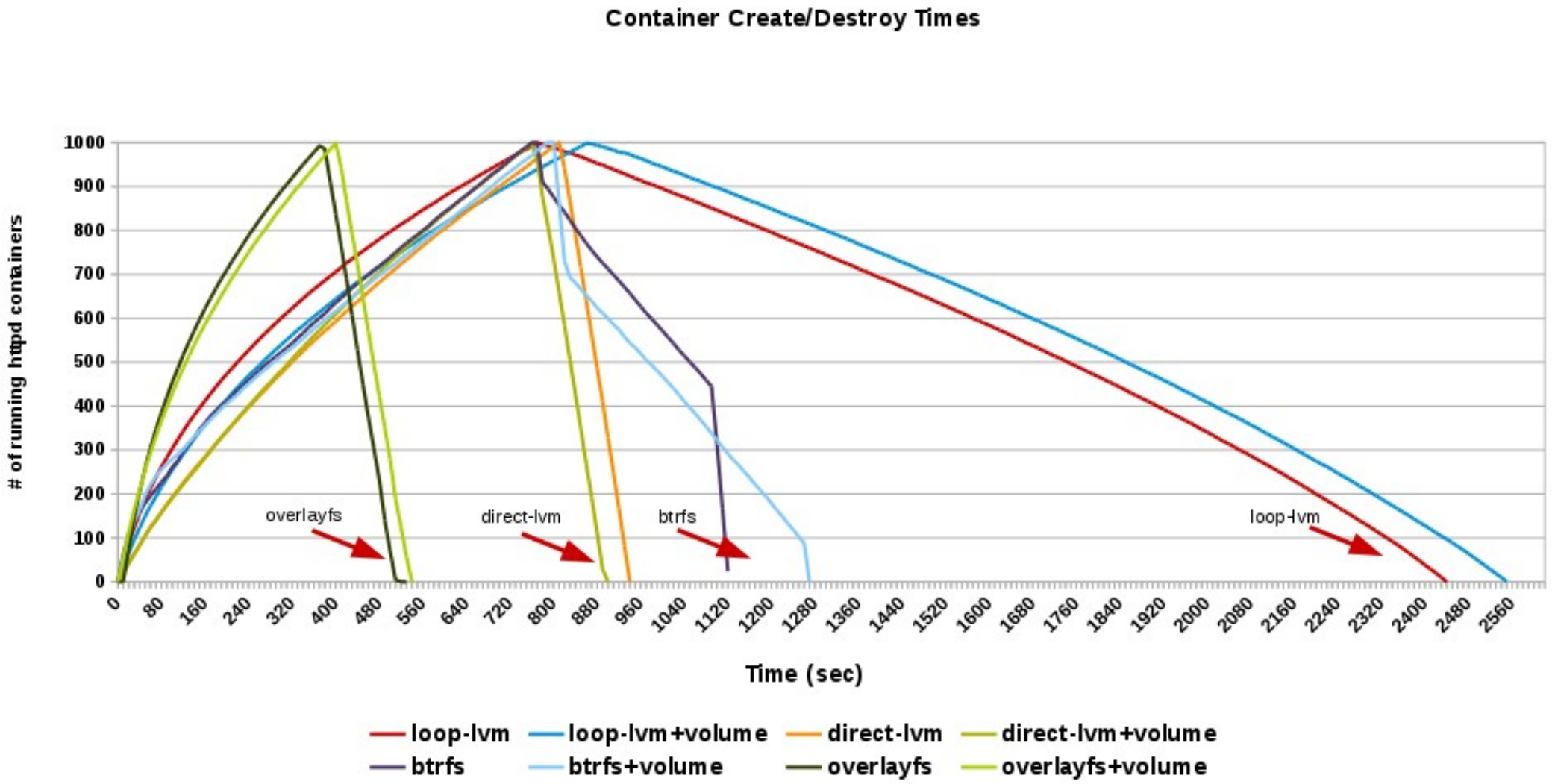
OverlayFS

+ hell fast (you'll see)
+ page cache sharing
+ finally in upstream kernel (in rhel from 7.2, 3.18)
+ finally supported by docker (-s overlay)

OverlayFS

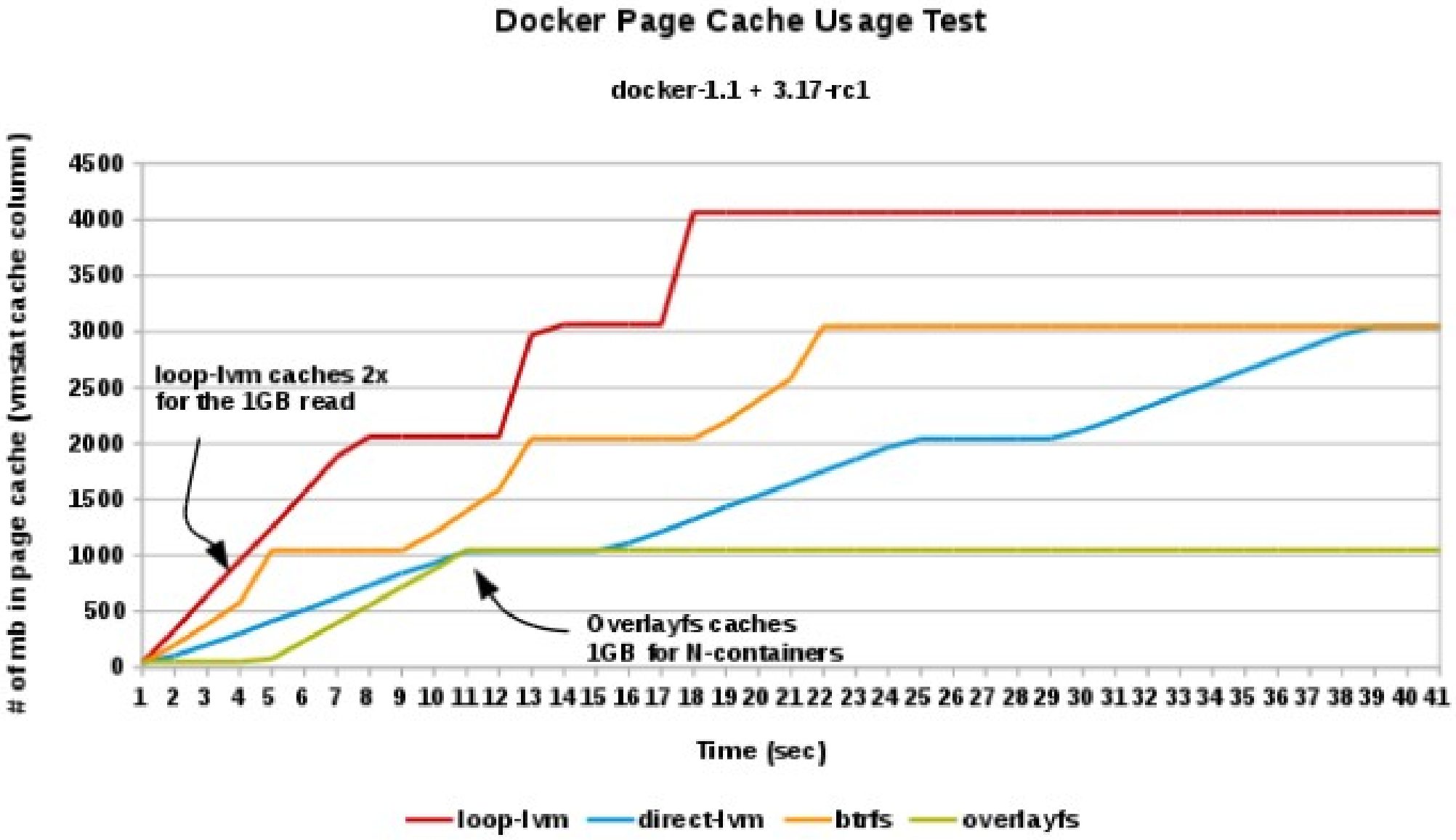
+ hell fast (you'll see)
+ page cache sharing
+ finally in upstream kernel (in rhel from 7.2, 3.18)
+ finally supported by docker (-s overlay)
- SELinux not there yet (but will be)

OverlayFS



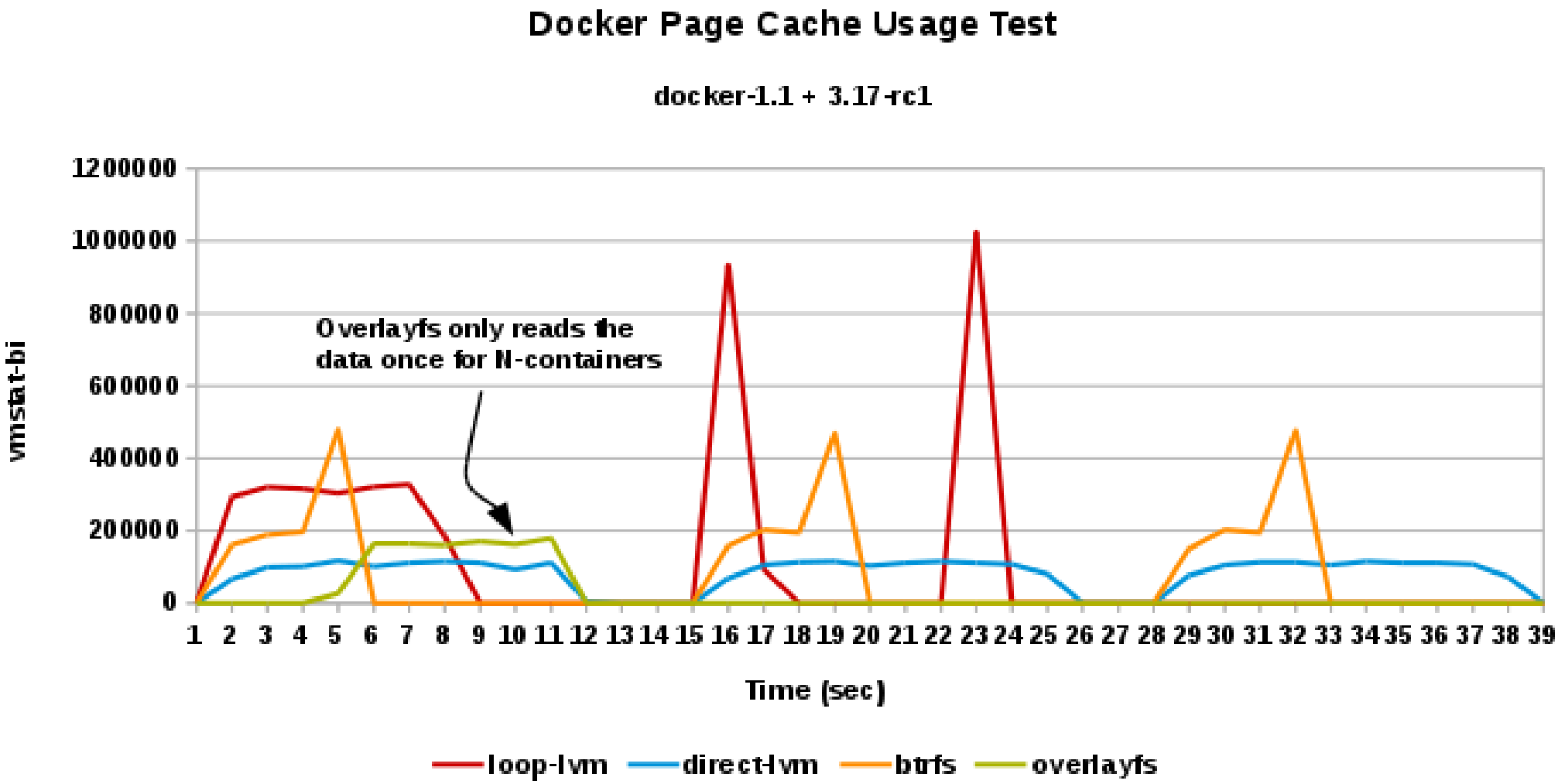
http://developerblog.redhat.com/2014/09/30/overview-storage-scalability-docker/

OverlayFS



http://developerblog.redhat.com/2014/09/30/overview-storage-scalability-docker/

OverlayFS



http://developerblog.redhat.com/2014/09/30/overview-storage-scalability-docker/

OverlayFS

let's demo again

Linux containers equation

Linux Containers = namespaces + cgroups + storage

Docker – concepts

- images
  - read only
  - act as templates

Docker – concepts

- → images
  - → read only
  - → act as templates
- → Dockerfile
  - → like a makefile
  - → commands order & cache'ing
  - → extends the base image
  - → results in a new image

Docker – concepts

- images
  - read only
  - act as templates
- Dockerfile
  - like a makefile
  - commands order & cache'ing
  - extends the base image
  - results in a new image
- Containers: instances running apps

Docker – concepts

- → images
  - → read only
  - → act as templates
- → Dockerfile
  - → like a makefile
  - → commands order & cache'ing
  - → extends the base image
  - → results in a new image
- → Containers: instances running apps

dockerfile + base image = docker container

Dockerfile

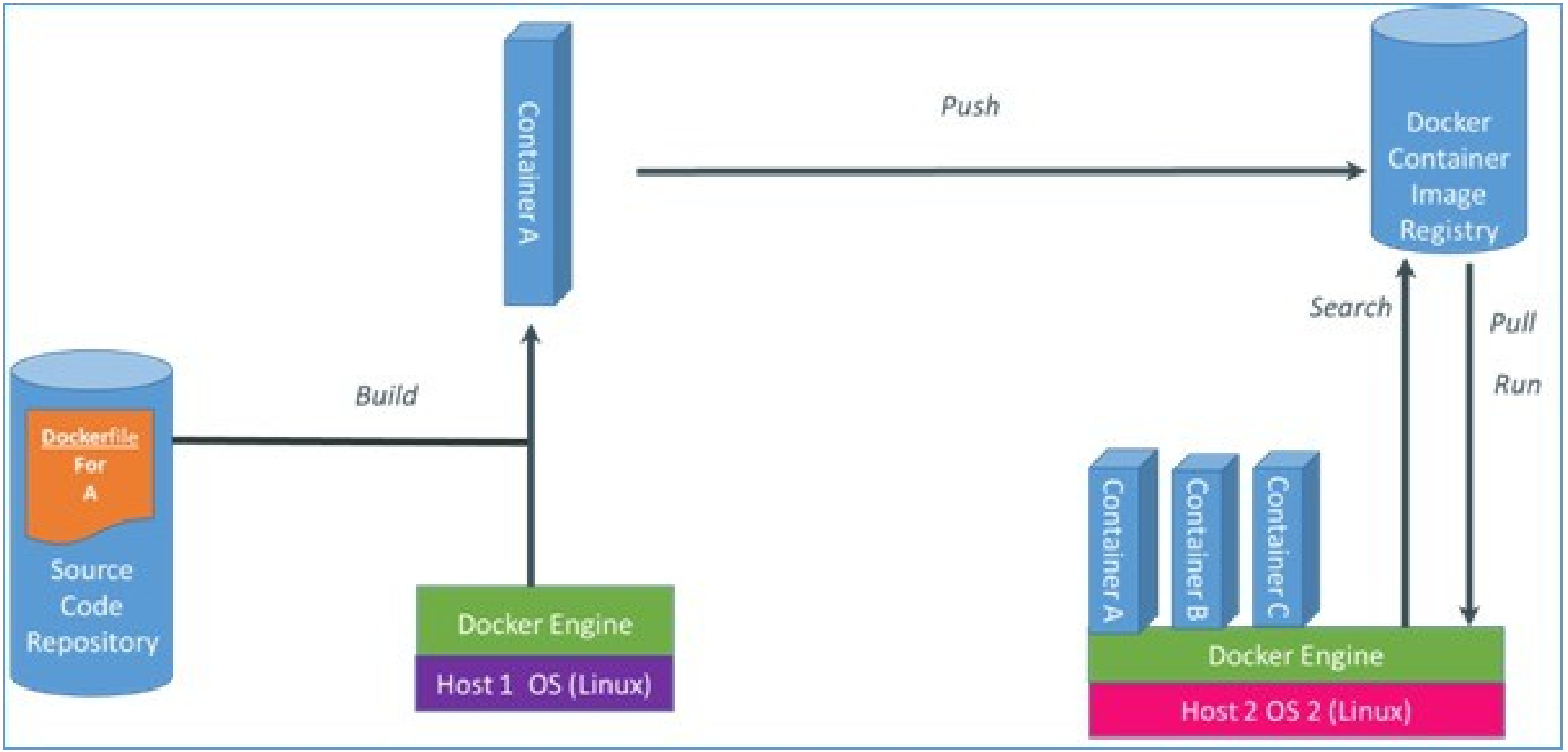
```
FROM fedora
MAINTAINER scollier <scollier@redhat.com>

RUN yum -y update && yum clean all
RUN yum -y install nginx && yum clean all
RUN echo "daemon off;" >> /etc/nginx/nginx.conf
RUN echo "nginx on Fedora" > /srv/www/index.html

EXPOSE 80

CMD [ "/usr/sbin/nginx" ]
```

Docker – registry



http://osv.io/blog/blog/2014/06/19/containers-hypervisors-part-2/

Docker – registry

→ git like semantics

→ pull, push, commit

→ private and public registry

→ https://github.com/dotcloud/docker-registry

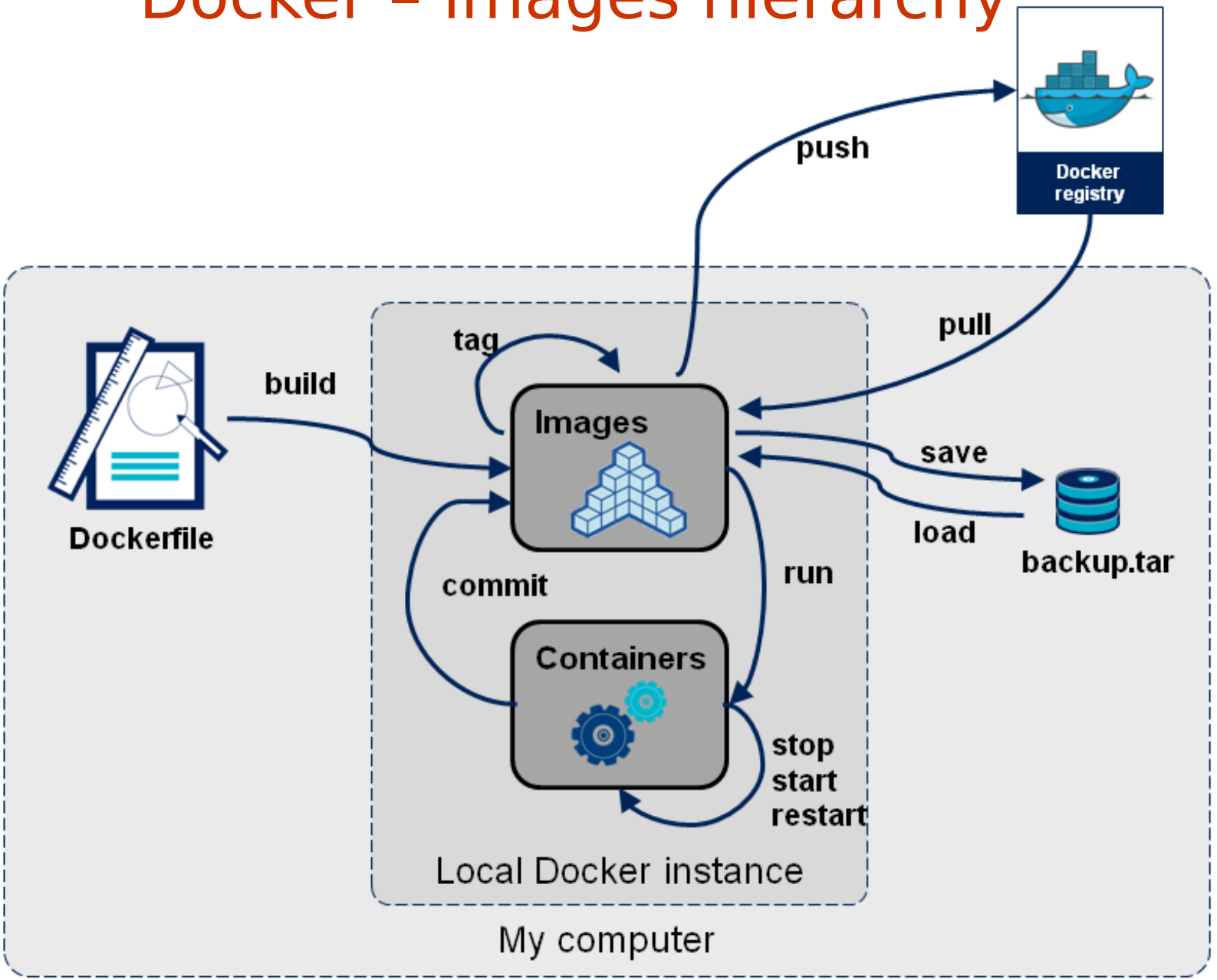
→ yum install docker-registry

$ docker pull

$ docker push

$ docker commit

Docker – images hierarchy



http://blog.octo.com/en/docker-registry-first-steps/

Docker – images hierarchy

base image
-> child image
-> grandchild image

Docker – images hierarchy

base image
-> child image
-> grandchild image

Git’s promise: Tiny footprint with lightning fast performance

Docker – security

→ Isolation via kernel namespaces

→ Additional layer of security: SELinux / AppArmor / GRSEC

→ Each container gets own network stack

→ control groups for resources limiting

f20 policy: https://git.fedorahosted.org/cgit/selinux-policy.git/tree/docker.te?h=f20-contrib

What's there?

seinfo -t -x | grep docker

sesearch -A -s docker_t (and the rest)

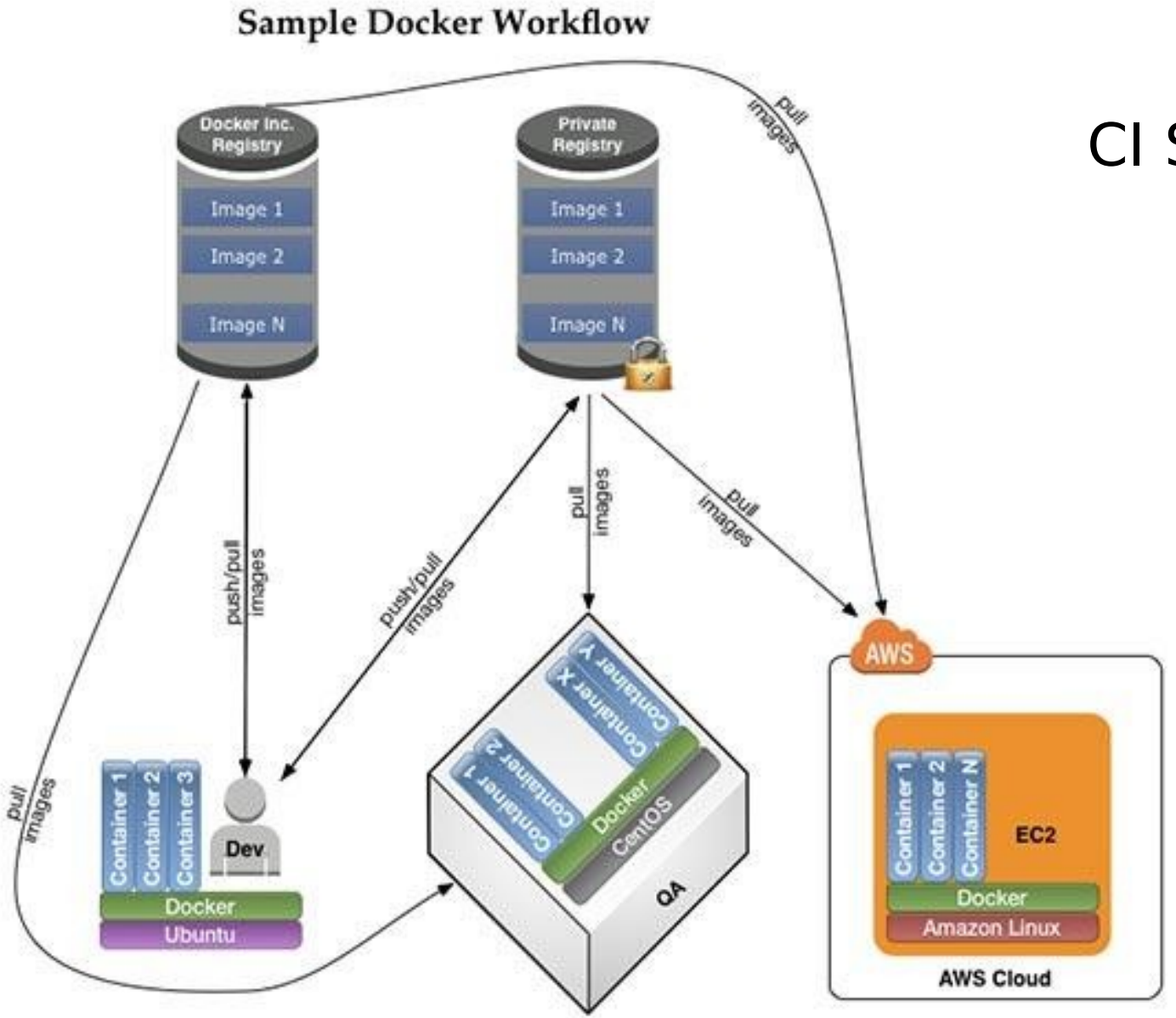
or just unpack docker.pp with semodule_unpackage

Docker – security

http://www.projectatomic.io/blog/2014/08/is-it-safe-a-look-at-docker-and-security-from-linuxcon/

Docker – security

Docker – use cases



CI Stack

http://sattia.blogspot.com/2014/05/docker-lightweight-linux-containers-for.html

Docker – use cases

Continuous Integration

- → local dev
  - → with Docker it's easy to standardize envs
- → deployment
  - → rolling updates (e.g. w/Ansible)
- → testing
  - → unit testing of any commit on dedicated env
  - → don't worry about cleaning up after testing
  - → parrarelized tests across any machines

Docker – use cases

- → version control system for apps
- → microservices
  - → Docker embraces granularity
  - → Services can be deployed independently and faster
  - → parallelized tests across any machines
- → continuous delivery
- → PaaS

Orchestration at scale w/Docker

Orchestration at scale w/Docker

This might be a little problem

Orchestration at scale w/Docker

Orchestration at scale w/Docker

	Big Data	Cloud Platform	IaaS	Data Center OS	Docker OS	Docker Mgmt.	PaaS	Orch. Config Mgmt.
Ansible & Docker								x
Amazon EC2 & Docker		x						
Apache Brooklyn & Docker								x
Apache Hadoop & Docker	x							
Apache Storm & Docker	x							
AppScale & Docker							x	
Atomic Hosts & Docker					x			
Chef & Docker								x
Clocker & Docker								x
Cloud Foundry & Docker	x						x	
CloudStack & Docker			x					
CoreOS & Docker					x			
Deis & Docker							x	
Decker & Docker							x	
Docker & Docker			x		x	x	x	x
Dokku & Docker							x	
Eucalyptus & Docker			x					
Flynn & Docker							x	
Google Compute Platform & Docker		x						
IBM Bluemix & Docker	x						x	
Kubernetes & Docker			x			x	x	x
Mesos, Mesosphere & Docker	x			x		x	x	x
Microsoft Azure & Docker		x						
OpenCamp & Docker		x	x			x	x	
OpenShift & Docker							x	
OpenStack & Docker			x					
Panamax & Docker						x		
Puppet & Docker								x
SaltStack & Docker							x	x
Shipyard & Docker						x		
Stackato & Docker							x	
Tsuru & Docker							x	
VMware & Docker			x					

http://www.cloudssky.com/en/blog/Docker-Is-Not-Enough

Ansible + Docker

&

Docker + Ansible

Docker & Ansible

Ansible docker core module:
http://docs.ansible.com/docker_module.html

```
- hosts: web
  sudo: yes
  tasks:
  - name: run httpd servers
    docker: >
          image=centos
          command="service httpd start"
          ports=8080
          count=5
          memory_limit=32MB
          link=mysql
          expose=8080
          registry=...
          volume=...
```

Docker & Ansible

Building image with Ansible:

```
FROM ansible/centos7-ansible:stable
ADD ansible /srv/example
WORKDIR /srv/example
RUN ansible-playbook web.yml -c local
EXPOSE 80
CMD ["/usr/sbin/nginx"]
```

Docker & Ansible

Building image with Ansible:

```
FROM ansible/centos7-ansible:stable
ADD ansible /srv/example
WORKDIR /srv/example
RUN ansible-playbook web.yml -c local
EXPOSE 80
CMD ["/usr/sbin/nginx"]
```

ansible/web.yml:

```
- name: Install webserver
  hosts: localhost
  tasks:
    - yum: pkg=nginx state=latest
    - shell: echo "ansible" > /usr/share/nginx/html/index.html
```

Docker & Ansible

Yet another demo?

SmartStack

→ automated service discovery and registration framework

→ ideal for SOA architectures

→ ideal for continuous integration & delivery

→ solves “works on my machine” problem

SmartStack

→ automated service discovery and registration framework

→ ideal for SOA architectures

→ ideal for continuous integration & delivery

→ solves “works on my machine” problem

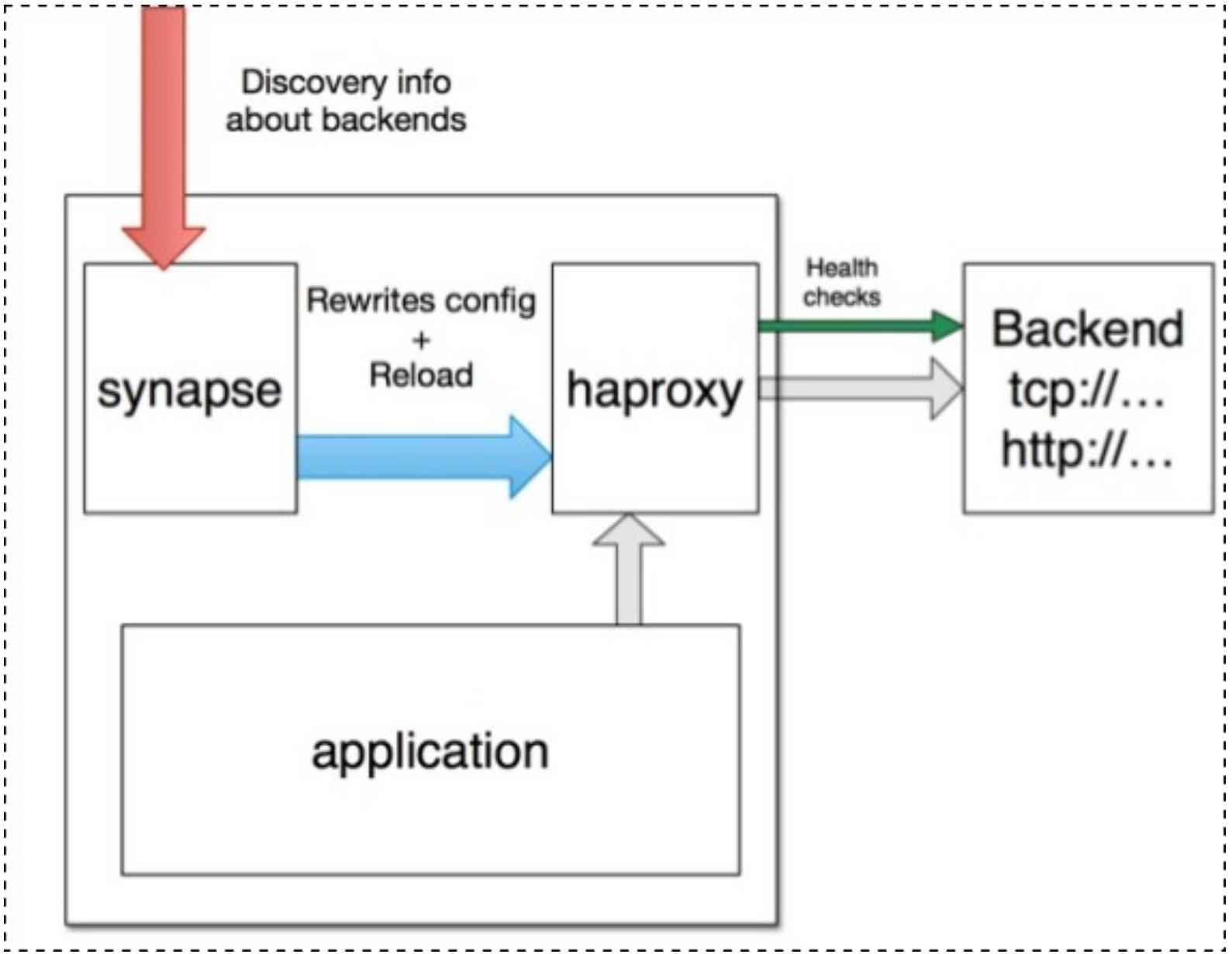
haproxy + nerve + synapse + zookeper = smartstack

SmartStack

Synapse

→ discovery service (via zookeeper or etcd)
→ installed on every node
→ writes haproxy configuration
→ application doesn't have to be aware of this
→ works same on bare / VM / docker
→ https://github.com/airbnb/nerve

SmartStack


Discovery info
about backends
synapse
Rewrites config
+
Reload
haproxy
Health
checks
Backend
tcp://...
http://...
application

SmartStack

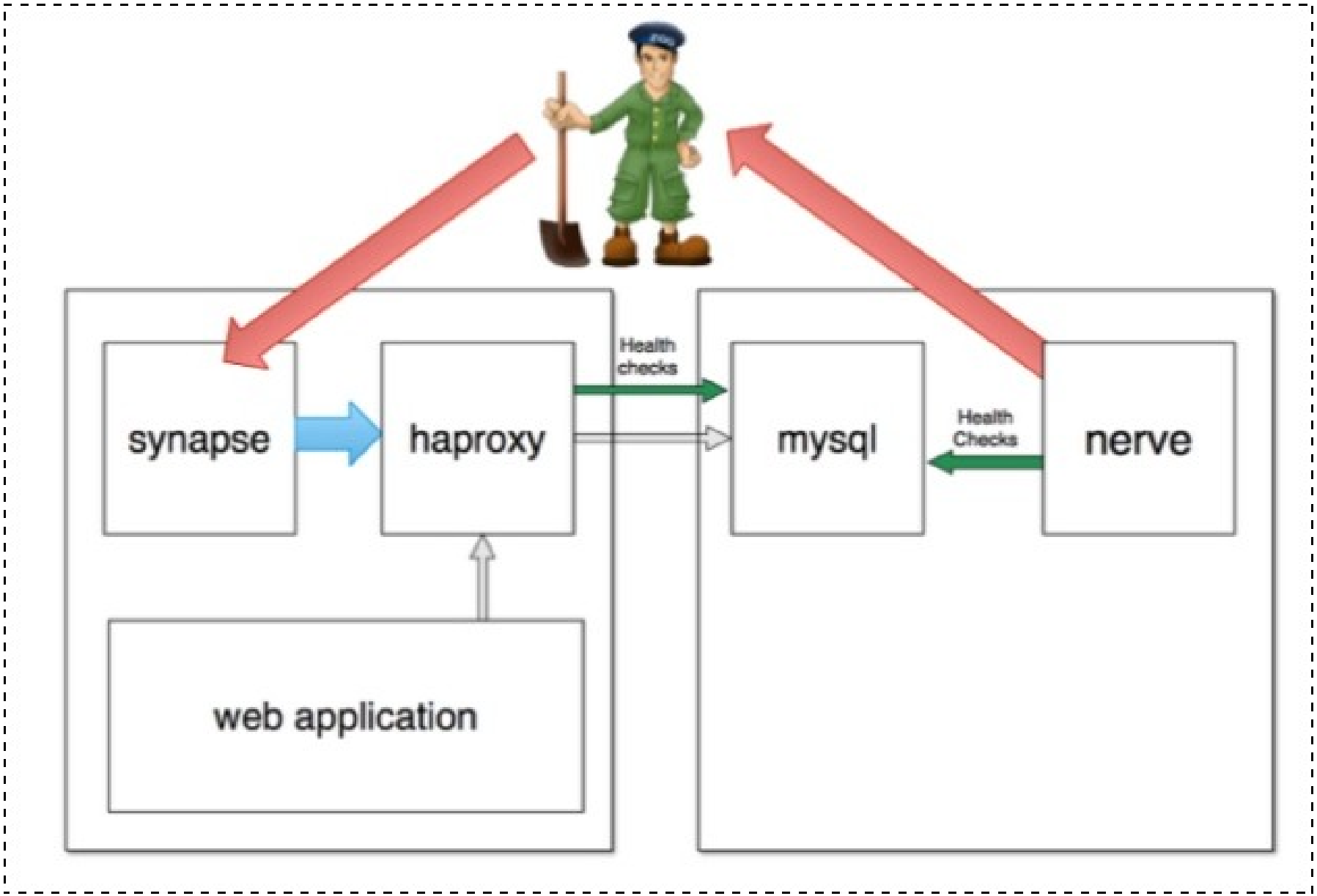
Nerve

→ health checks (pluggable)

→ register service info to zookeper (or etcd)

→ https://github.com/airbnb/synapse

SmartStack


synapse
haproxy
Health checks
mysql
Health Checks
nerve
web application

SmartStack

Smartstack + Docker = <3

Smartstack + Docker = <3

but also remember about Consul
(come to #dockerkrk 2 meetup!)

Wanna learn Docker?

http://dockerbook.com/

Freenode #docker
#KrkDocker meetups (http://www.meetup.com/Docker-Krakow-Poland/)
https://github.com/docker/docker

sources?

→ docker.io documentation

→ dockerbook.com

→ slideshare!

→ zounds of blogposts (urls provided)

→ and some experience ;)

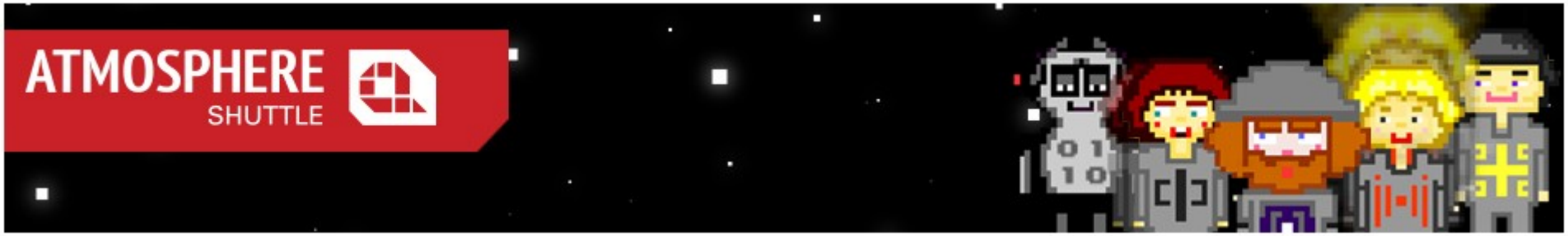
Looking for a job?

- Software Engineer (java)

- Information Security Manager

- Product Analyst

Catch me: maciek@lasyk.info



Linux containers & Devops

Maciej Lasyk

Atmosphere Shuttle #02 – Wrocław

2015-04-17