redhat

RED HAT ENTERPRISE LINUX

Dockah, Dockah, Dockah

Presenter: Dan Walsh

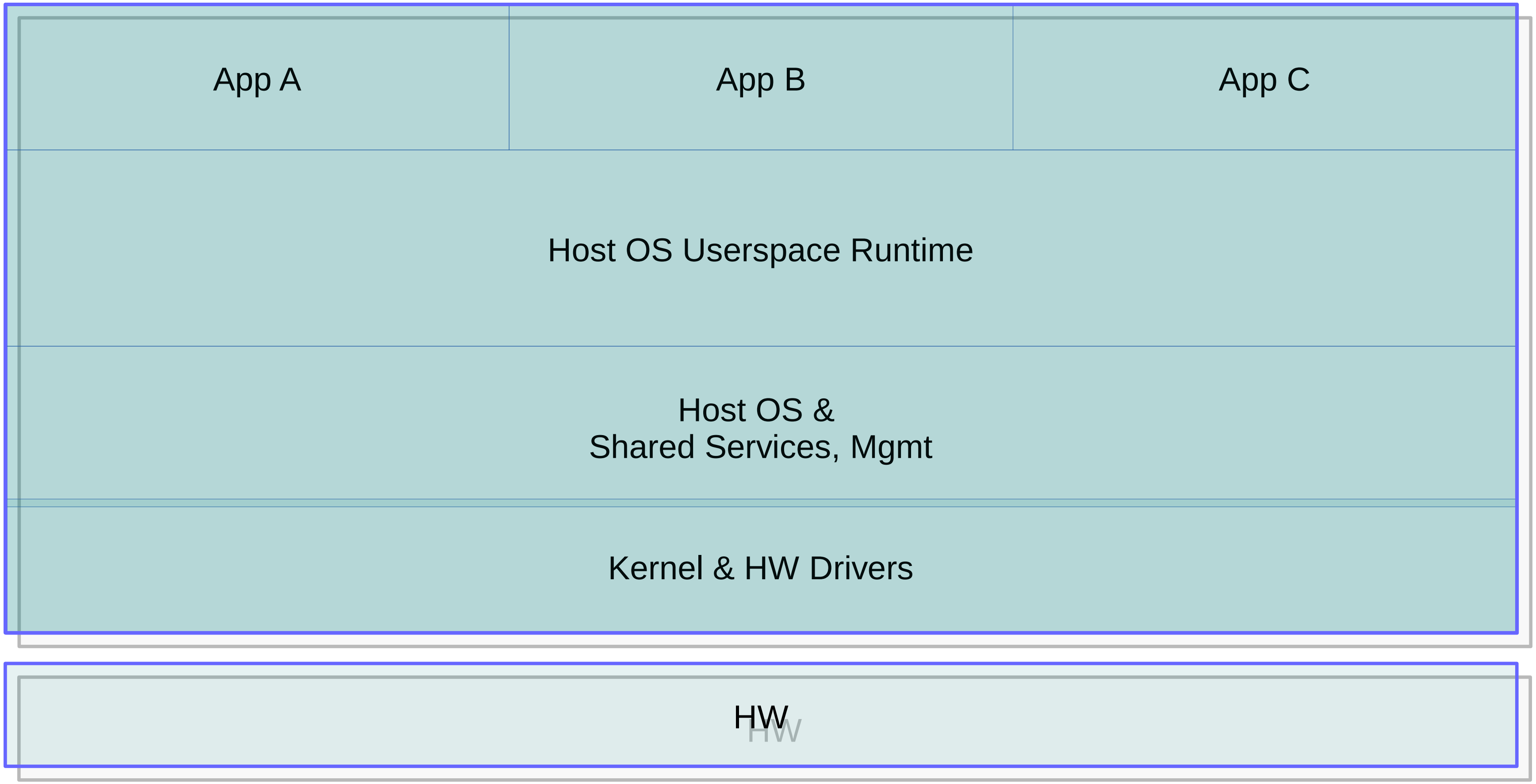
@rhatdan, Blog: danwalsh.livejournal.com,

dwalsh@redhat.com

Evolution of The Operating System
RHEL 4

Traditional Enterprise Operating System

Multiple Applications per machine + Single userspace runtime



App A

App B

App C

Host OS Userspace Runtime

Host OS &
Shared Services, Mgmt

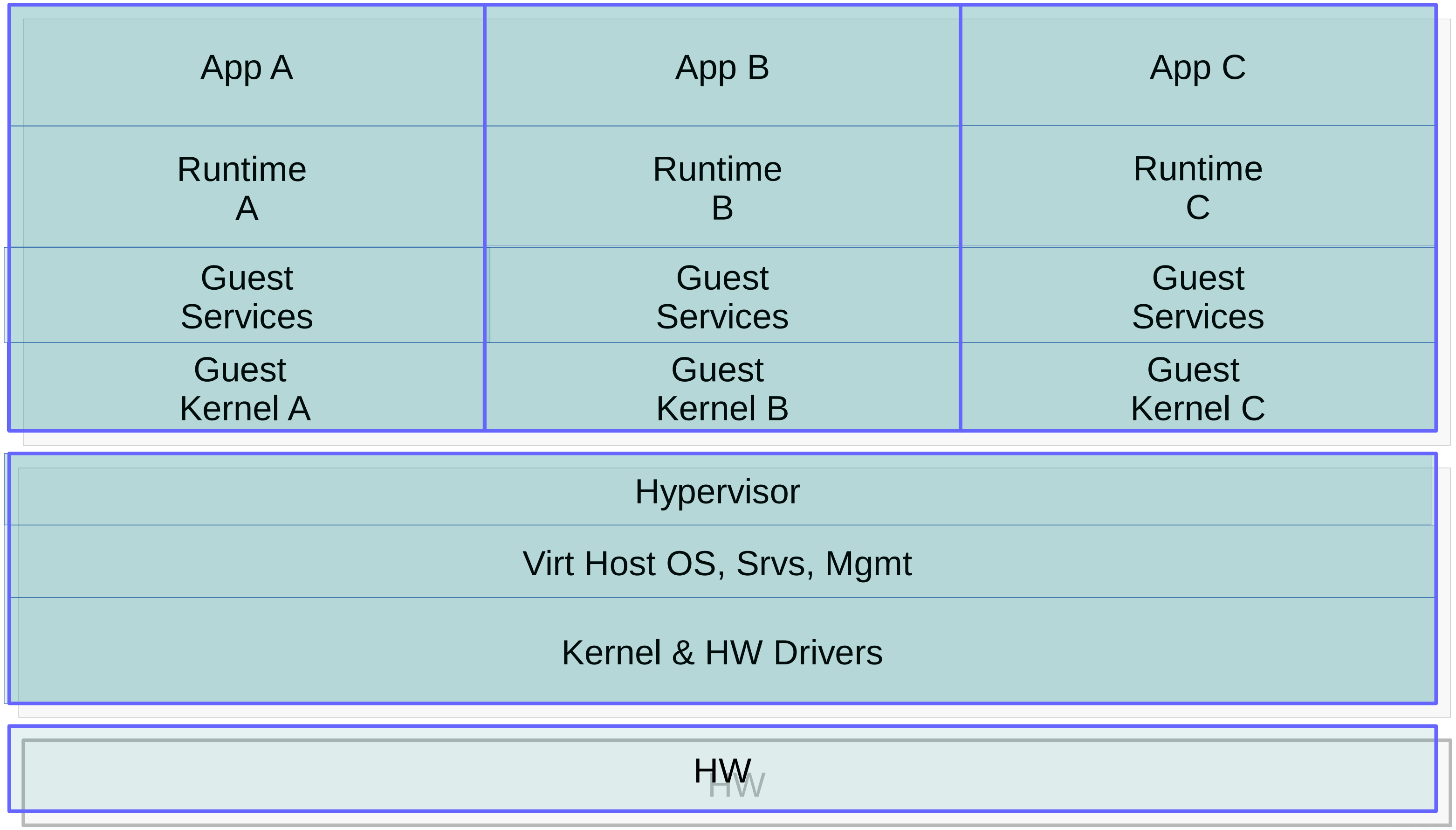
Kernel & HW Drivers

HW

Evolution of Operating System
RHEL 5 & RHEL 6

Virtualization & IaaS Cloud

Applications run inside a guest – full separation of host and guest



App A

Runtime A

Guest Services

Guest Kernel A

App B

Runtime B

Guest Services

Guest Kernel B

App C

Runtime C

Guest Services

Guest Kernel C

Hypervisor

Virt Host OS, Srvs, Mgmt

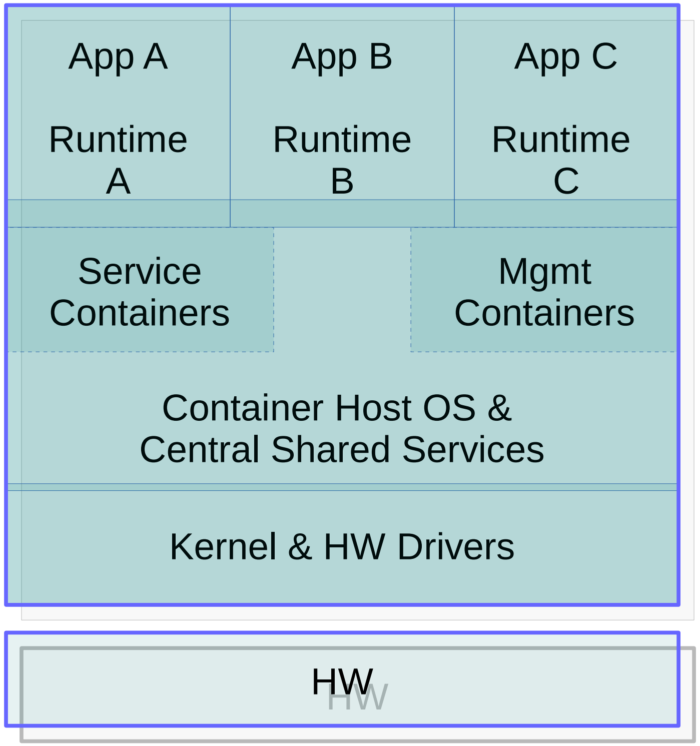
Kernel & HW Drivers

HW

Evolution of Operating System
RHEL 7

Light-weight Application Isolation

Application runs inside a container
Container deployed on bare metal or Virt/Cloud



App A

Runtime A

App B

Runtime B

App C

Runtime C

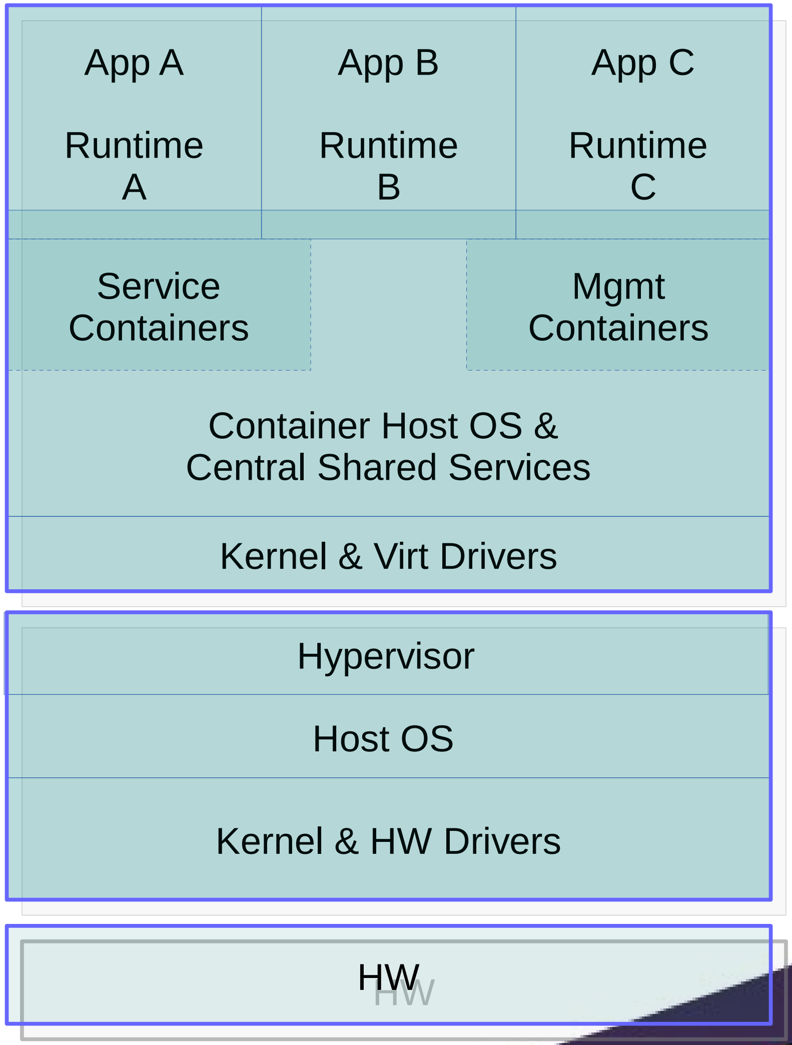
Service Containers

Mgmt Containers

Container Host OS & Central Shared Services

Kernel & HW Drivers

HW



App A

Runtime A

App B

Runtime B

App C

Runtime C

Service Containers

Mgmt Containers

Container Host OS & Central Shared Services

Kernel & Virt Drivers

Hypervisor

Host OS

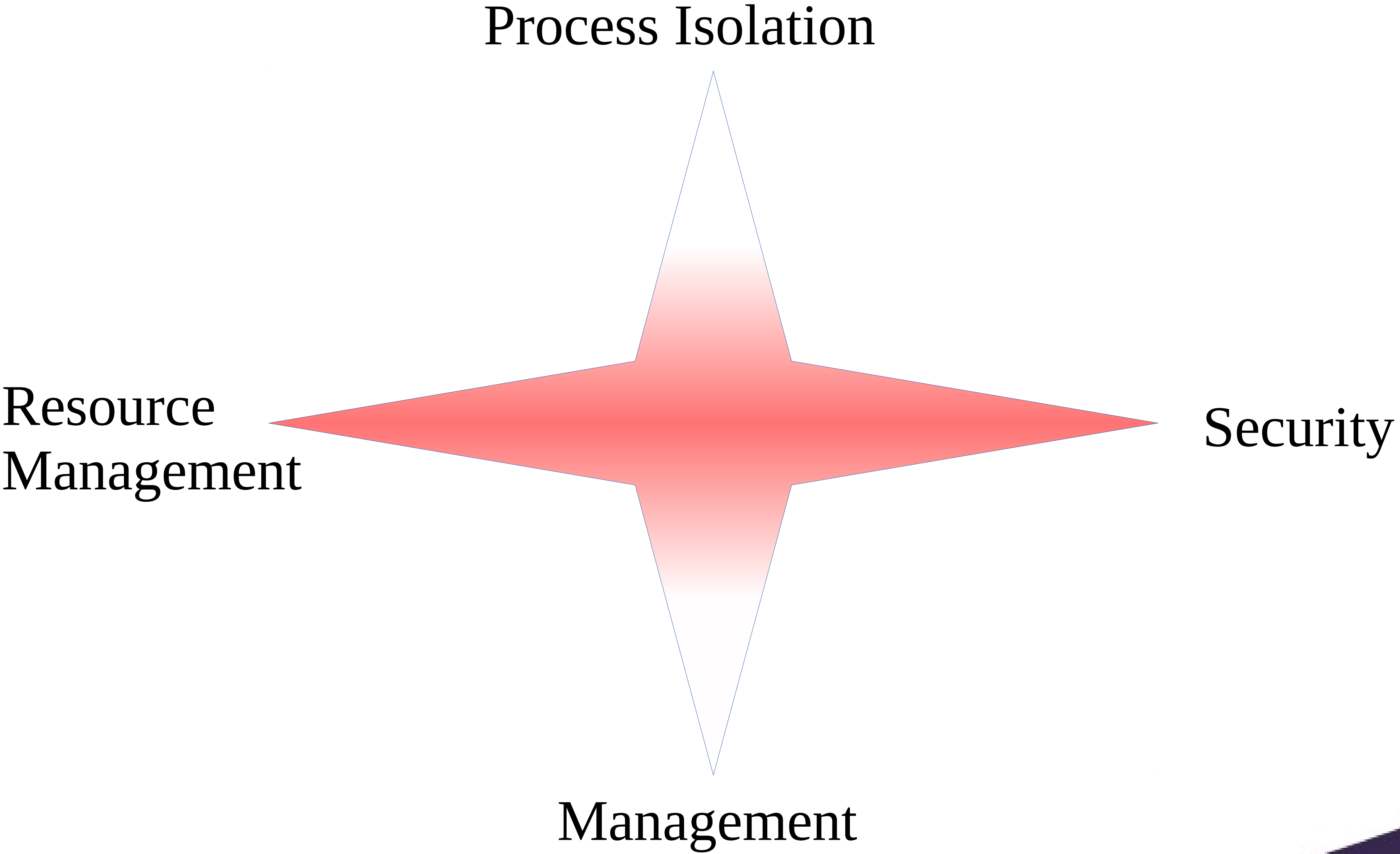
Kernel & HW Drivers

HW

The kernel knows SQUAT about CONTAINERS

Containers are a userspace concept that takes advantage of several Kernel Subsystems

Key elements of Linux Containers



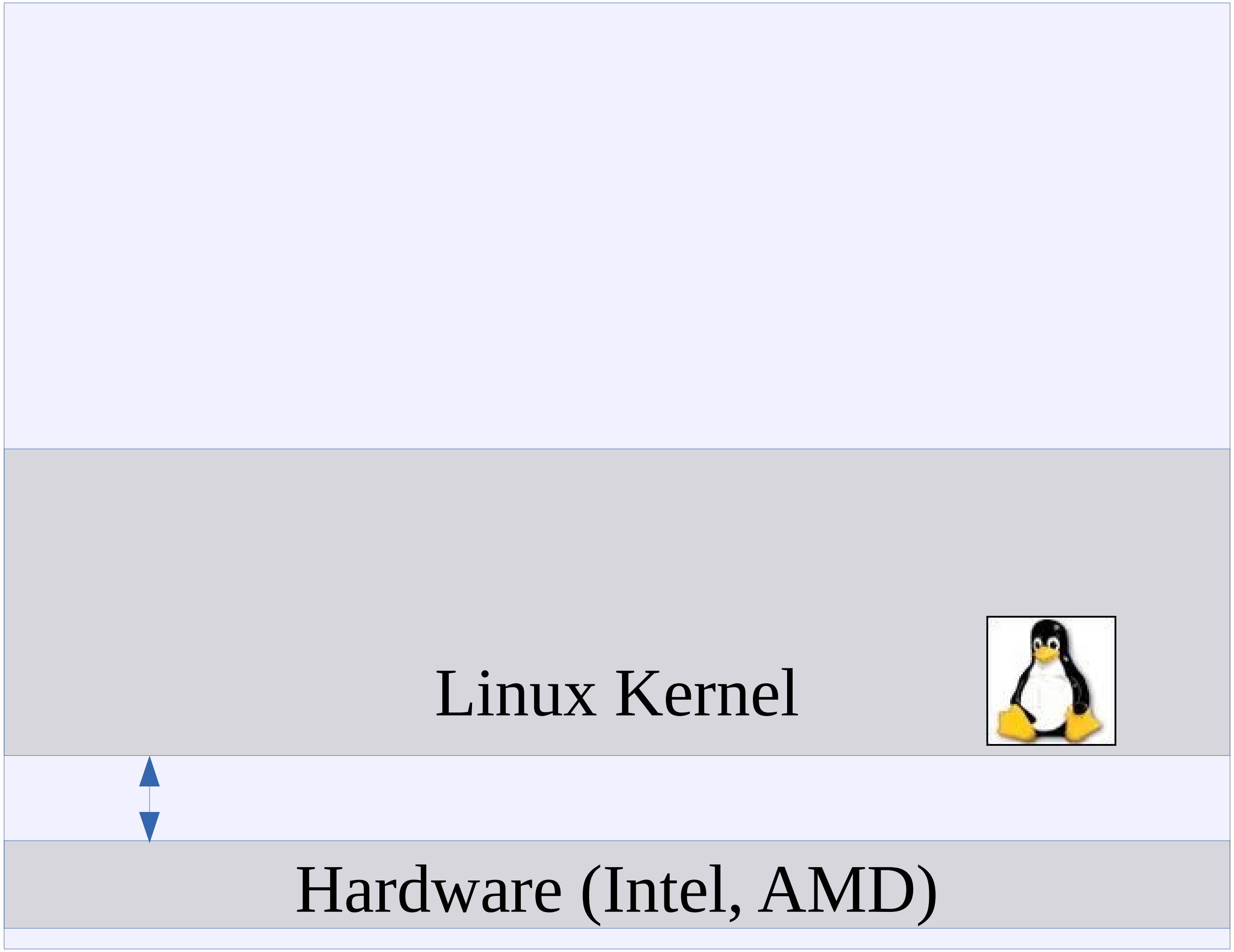
Process Isolation

Resource Management

Security

Management

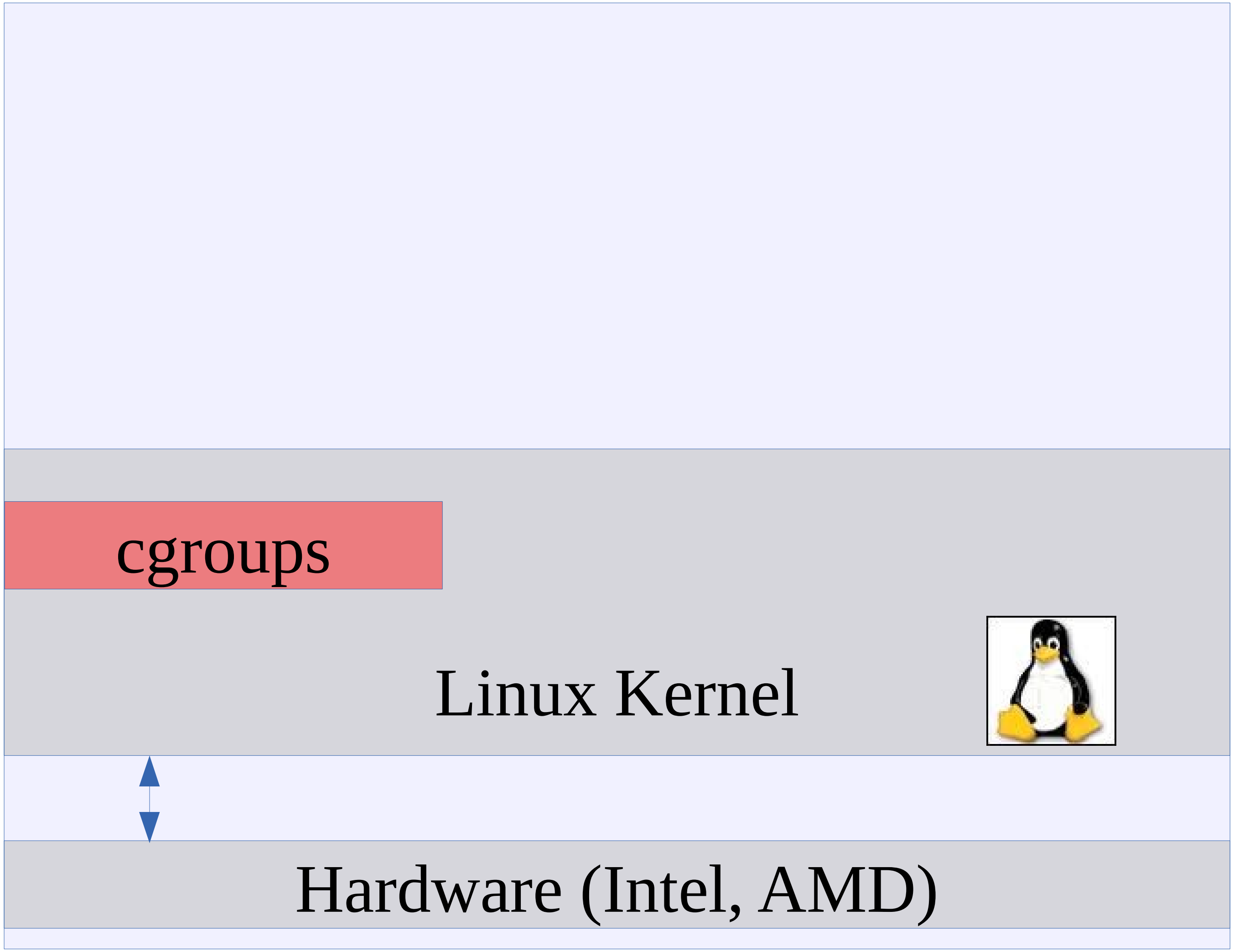
Red Hat Enterprise Linux Container Architecture



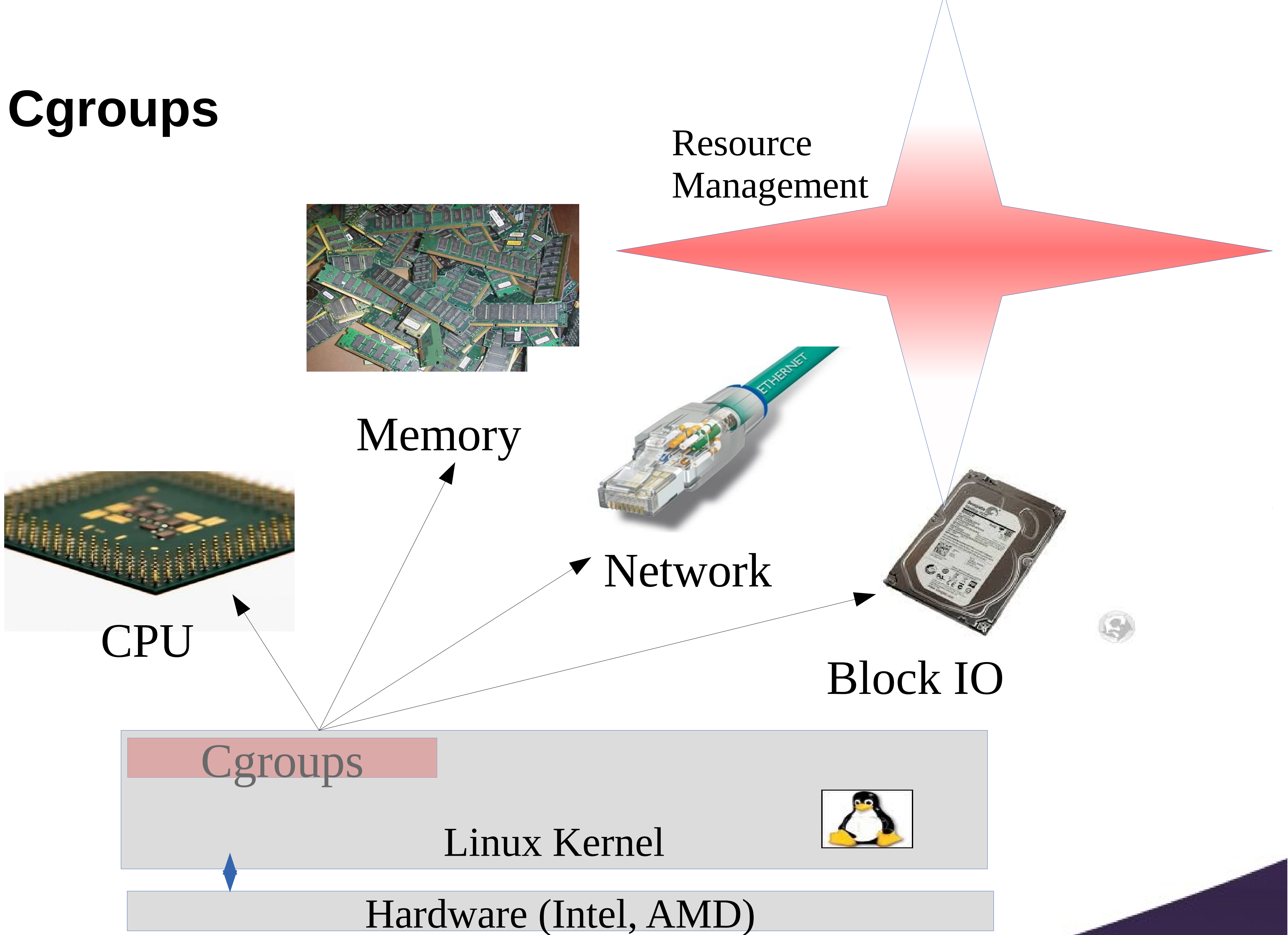
Linux Kernel

Hardware (Intel, AMD)

Red Hat Enterprise Linux Container Architecture



Cgroups



Resource Management

Memory

CPU

Network

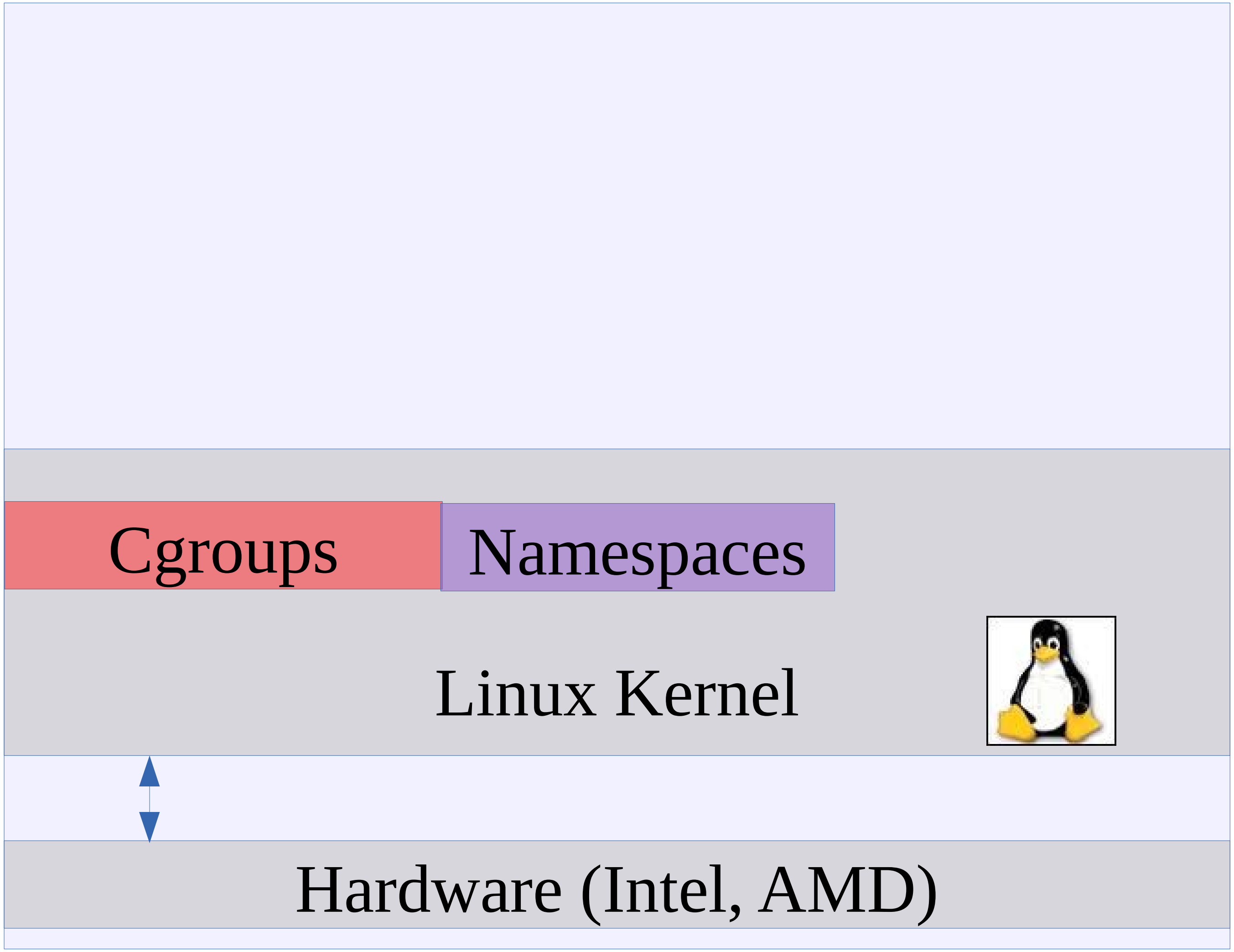
Block IO

Cgroups

Linux Kernel

Hardware (Intel, AMD)

Red Hat Enterprise Linux Container Architecture

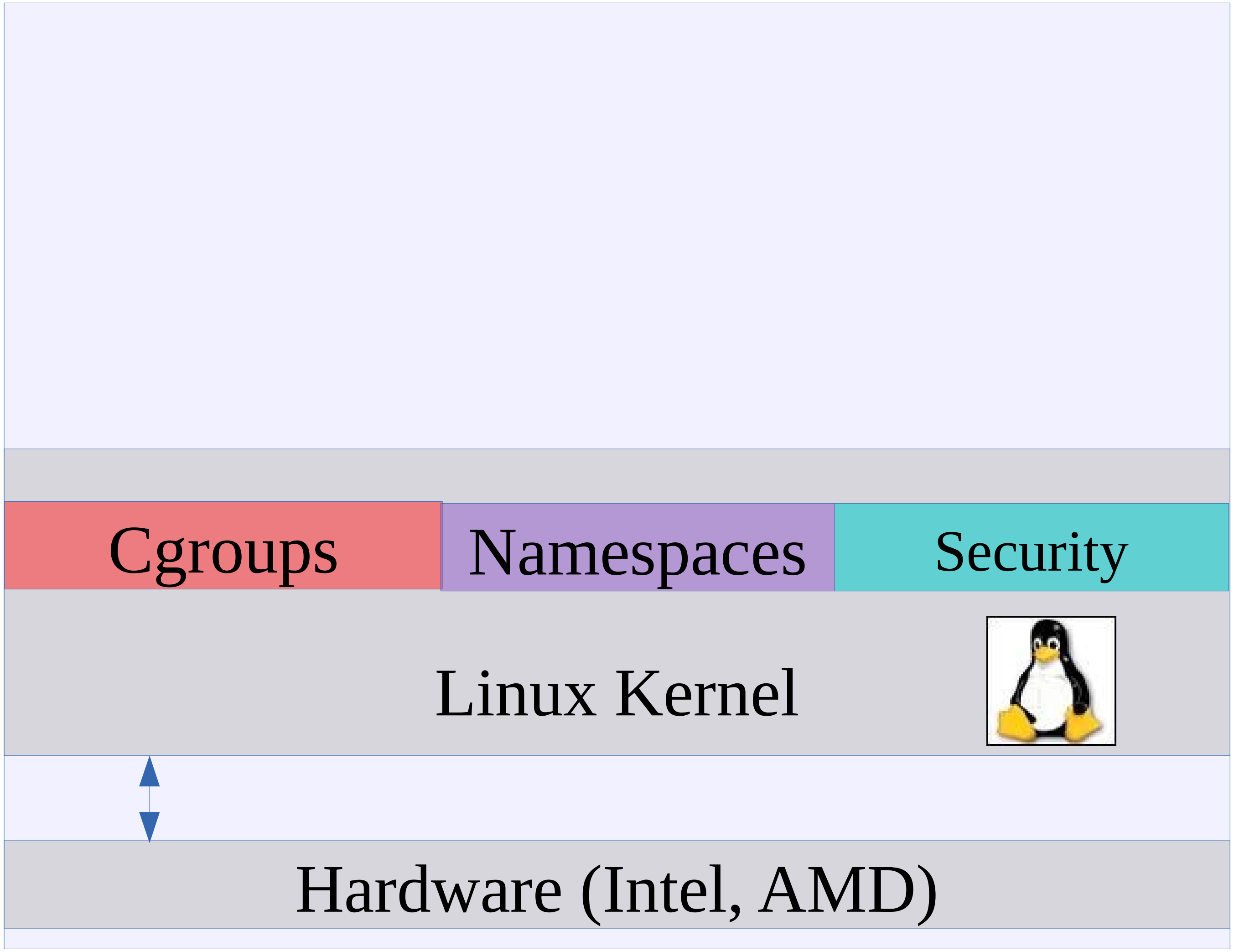


Namespaces

Process Isolation

- Isolate processes
  - Create a new environment with a
  - Subset of the resources
- Once set up, namespaces are transparent for processes
- Can be used in custom and complex scenarios
- Supported Namespaces
  - ipc, pid, mnt, net, uts
  - Future Red Hat Enterprise Linux 7: user

Red Hat Enterprise Linux Container Architecture



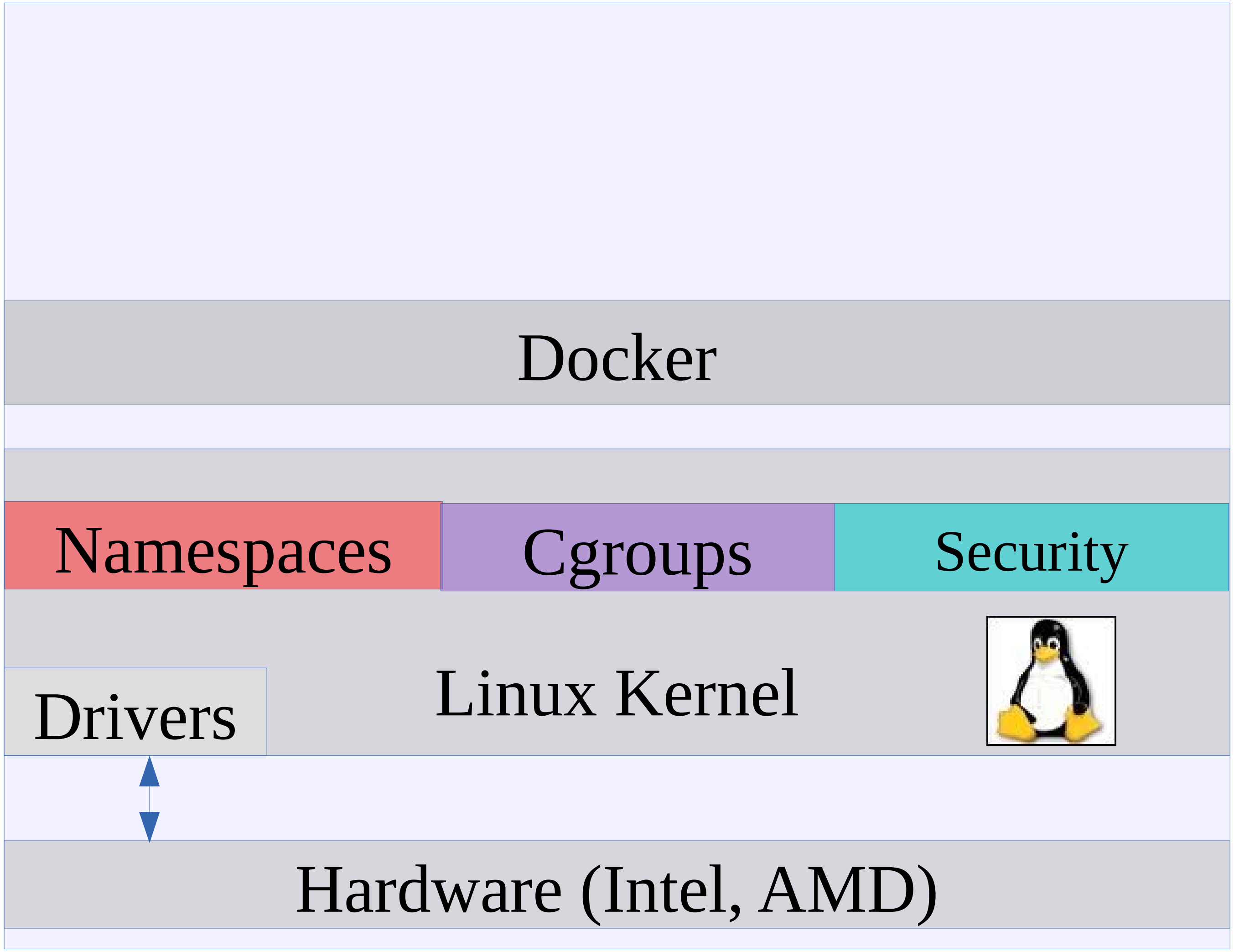
Containers do NOT Contain!!!

Security Isolation

Security

- Linux Containerization not complete
  - Not everything in Linux is namespaced
- SELinux sVirt
  - Container tooling uses sVirt
    - Type Enforcement
    - MCS Separation
- Capabilities
- Future User Namespaces

Red Hat Enterprise Linux Container Architecture



Docker

Namespaces

Cgroups

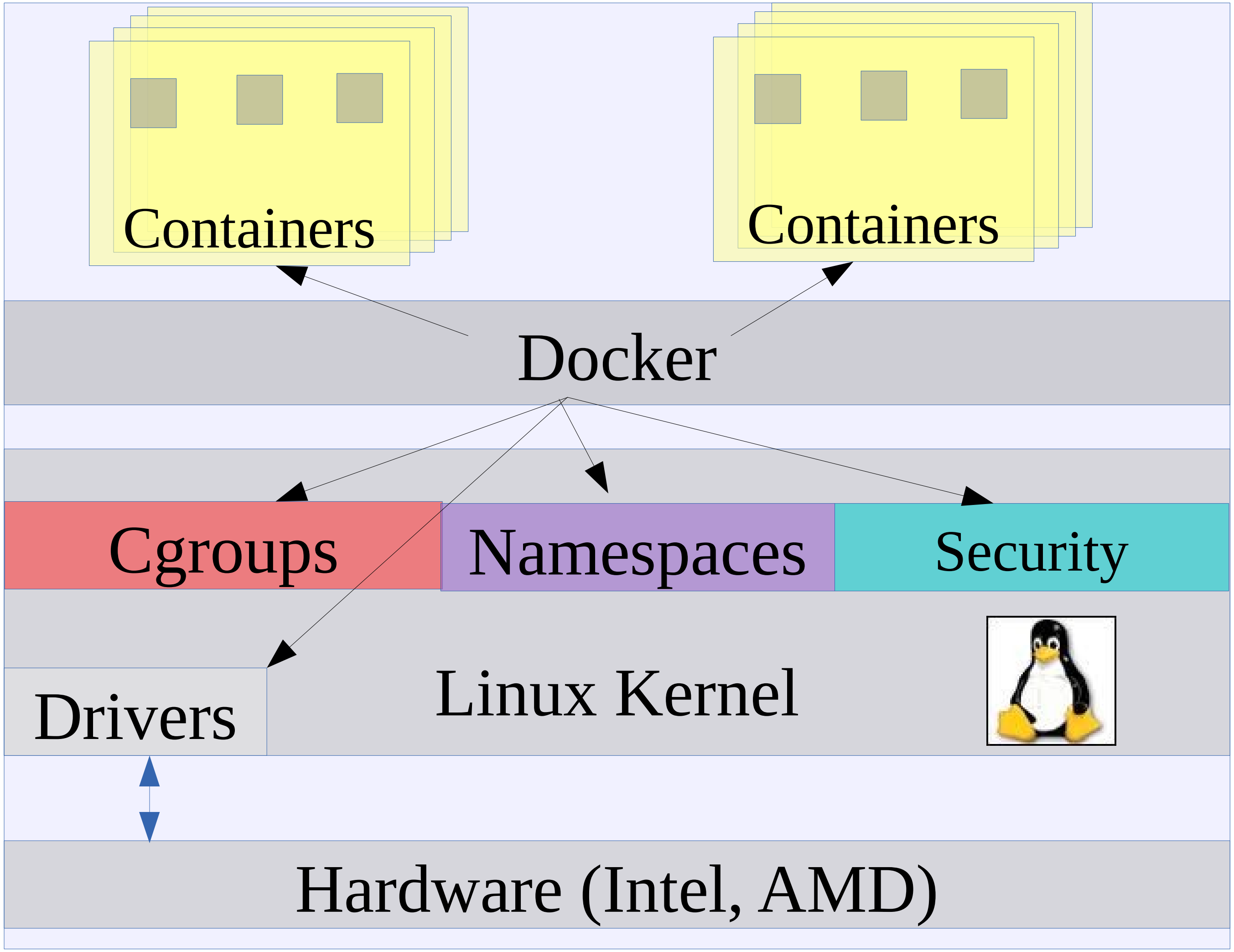
Security

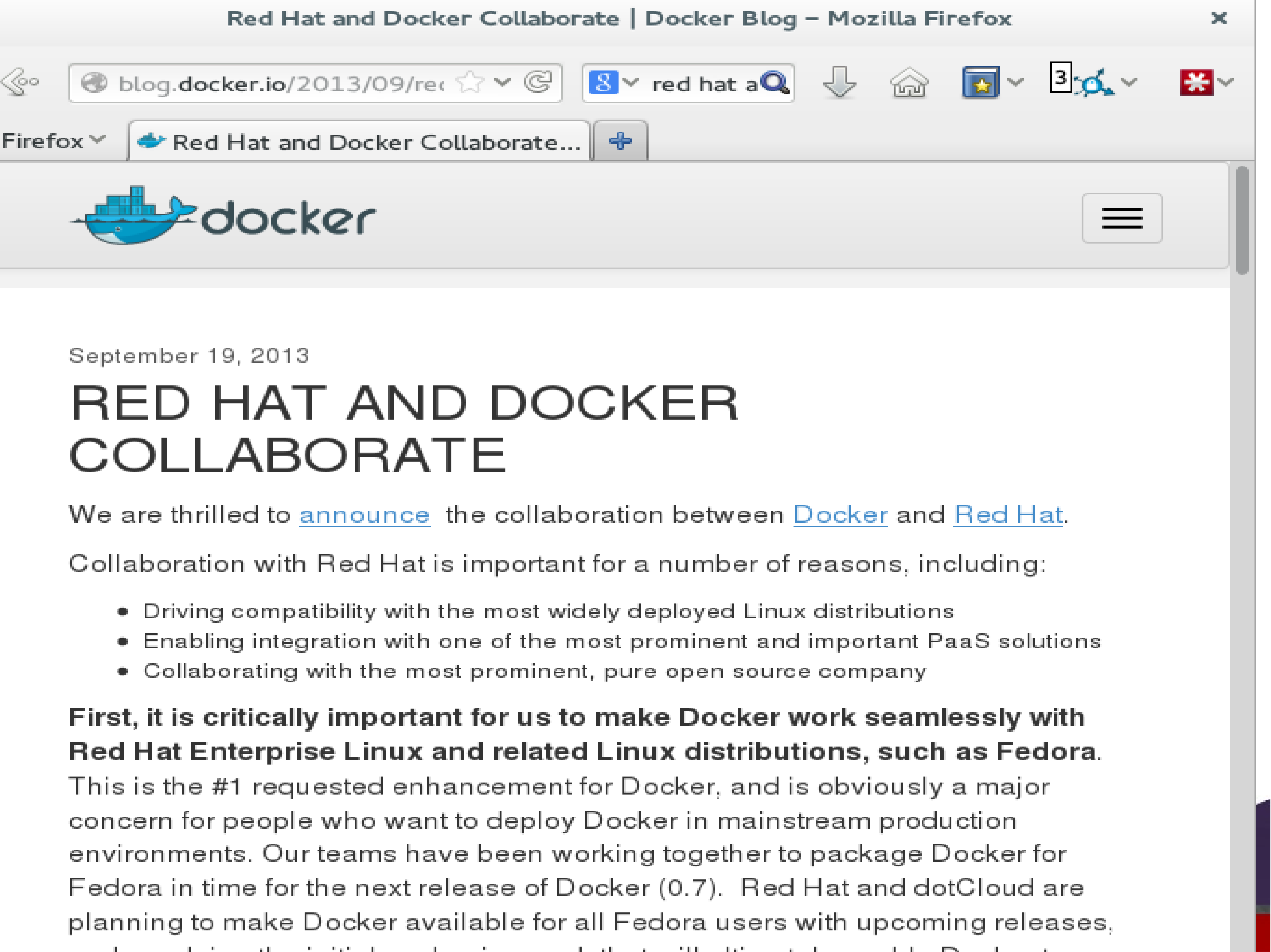
Linux Kernel

Drivers

Hardware (Intel, AMD)

Red Hat Enterprise Linux Container Architecture




Red Hat and Docker Collaborate | Docker Blog – Mozilla Firefox
blog.docker.io/2013/09/re
red hat a
Firefox
Red Hat and Docker Collaborate...
docker

September 19, 2013

RED HAT AND DOCKER COLLABORATE

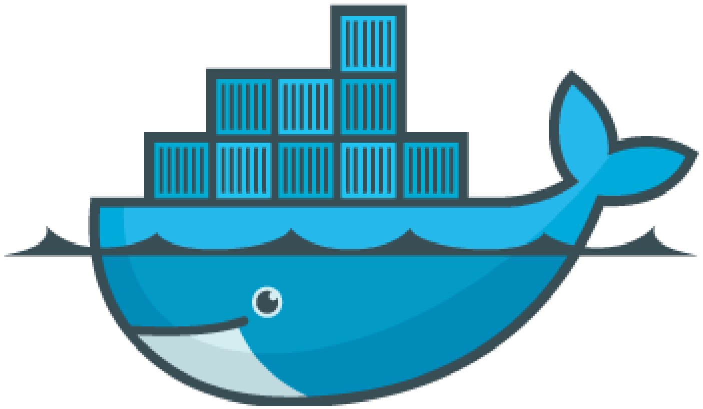
We are thrilled to announce the collaboration between Docker and Red Hat.

Collaboration with Red Hat is important for a number of reasons, including:

- Driving compatibility with the most widely deployed Linux distributions
- Enabling integration with one of the most prominent and important PaaS solutions
- Collaborating with the most prominent, pure open source company

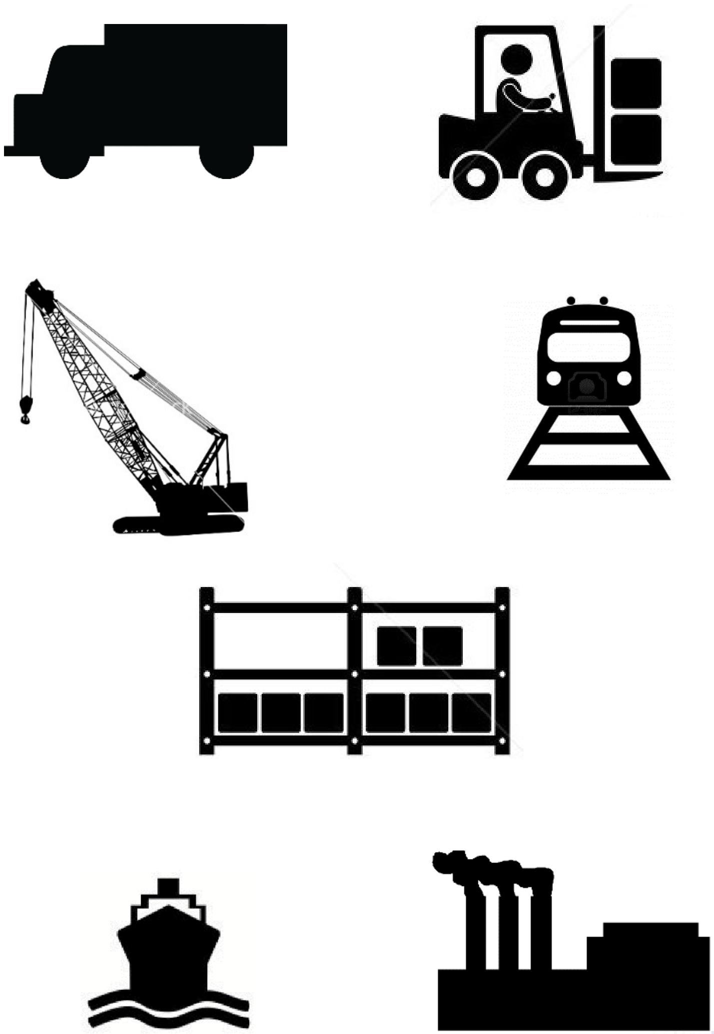
First, it is critically important for us to make Docker work seamlessly with Red Hat Enterprise Linux and related Linux distributions, such as Fedora. This is the #1 requested enhancement for Docker, and is obviously a major concern for people who want to deploy Docker in mainstream production environments. Our teams have been working together to package Docker for Fedora in time for the next release of Docker (0.7). Red Hat and dotCloud are planning to make Docker available for all Fedora users with upcoming releases,

Intro to Docker



Pre-1960 shipping industry

X



Solution: Shipping container

Separation of concerns

- User cares about packing the inside
- Shipper cares about moving the container

Standardized interface

Docker containers

Standardized interface for software container

Developer concerns
Code
Libraries
Services
Configuration
Data

All servers look the same

Ops concerns
Moving containers
Starting/Stopping containers
Logging
Monitoring
Network configuration

All containers look the same

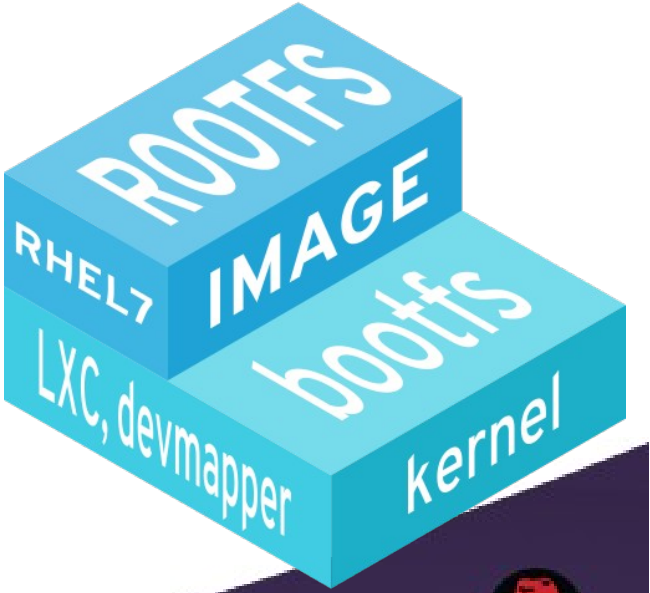
Isolation

Docker as a CLI for containers interesting but not that significant, we have had container type technology since RHEL5.

Docker as a packaging tool for shipping software may be a game changer.

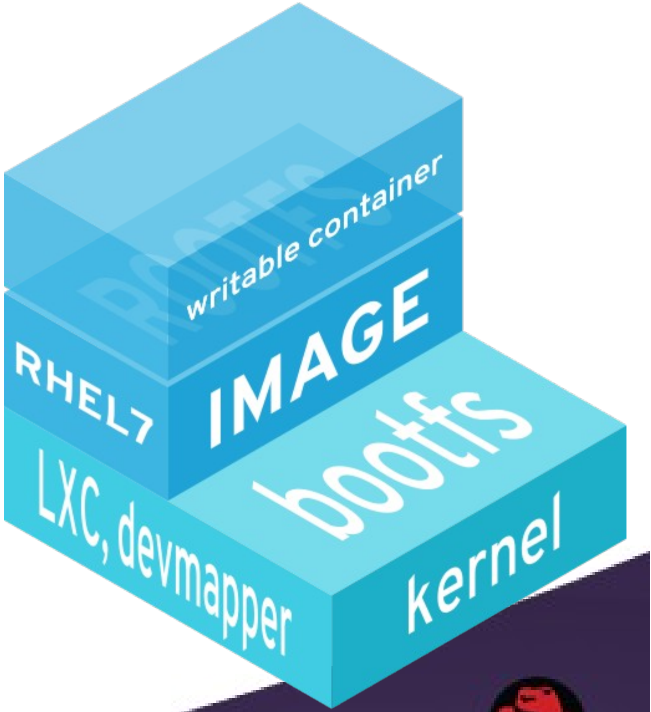
Docker glossary

- Image
  - Read-only template for a container
  - Includes all files required for application to run
  - Has additional metadata
    - Exposed network ports
    - Binary to start



Docker glossary

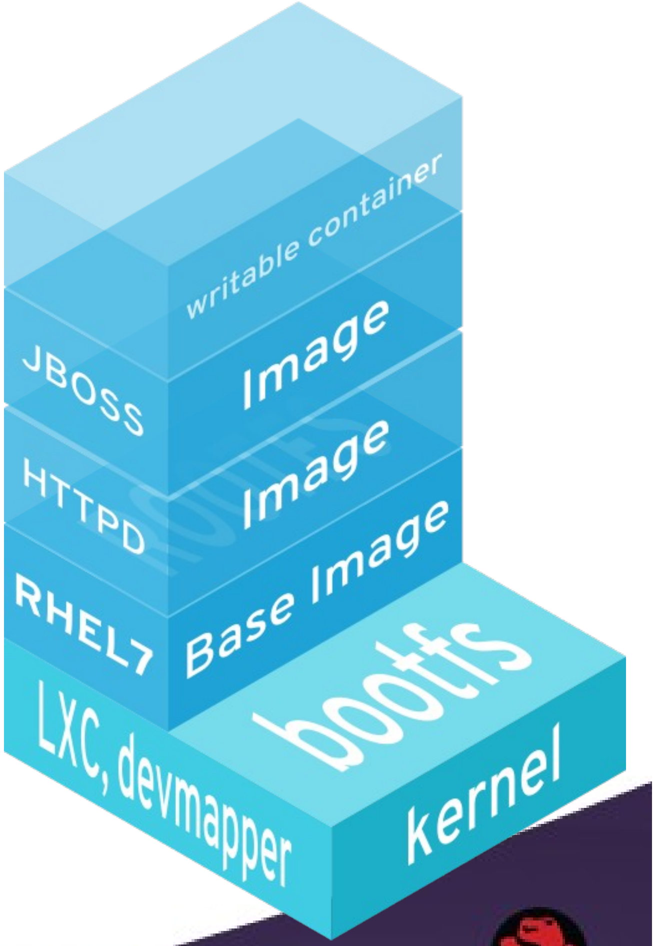
- Container
  - Running processes
  - Based on a particular image
  - Typically a single process
  - Isolated from host system
  - Cheap
  - Can write to filesystem
  - Commit creates new Image



Docker glossary

Layers

- Images are based on a parent
- The layers stack on top
- Files in base layers are shared between Images
- Each commit creates a layer
- Base image has no parent



Docker 101

- Hello, World!

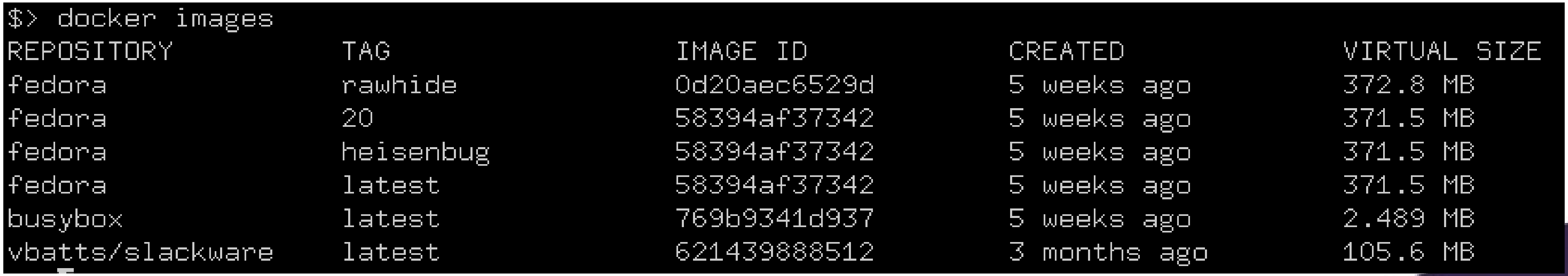
```
$> docker run rhel7 echo “Hello, World!”
```

- Fetch an image

```
$> docker pull rhel6
```

- List images

```
$> docker images
```



```
$> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
fedora              rawhide             0d20aec6529d        5 weeks ago         372.8 MB
fedora              20                  58394af37342        5 weeks ago         371.5 MB
fedora              heisenbug           58394af37342        5 weeks ago         371.5 MB
fedora              latest              58394af37342        5 weeks ago         371.5 MB
busybox             latest              769b9341d937        5 weeks ago         2.489 MB
vbatts/slackware    latest              621439888512        3 months ago        105.6 MB
```

Dockerfile

- Simple instructions
- Feels like scripting

```
FROM rhel7

RUN yum install -y mongodb-server && mkdir -p /data/db

EXPOSE 27017

VOLUME [“/data/db”]

CMD “mongod”
```

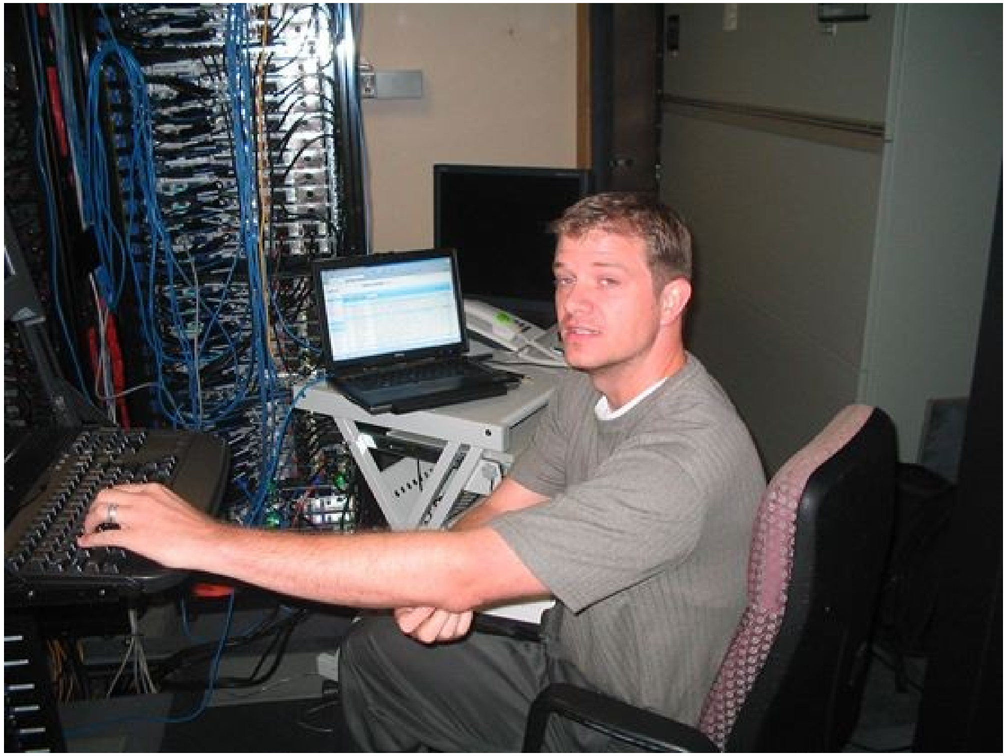
- Easy to make an image

```
$> docker build -t MY_MONGO .
```

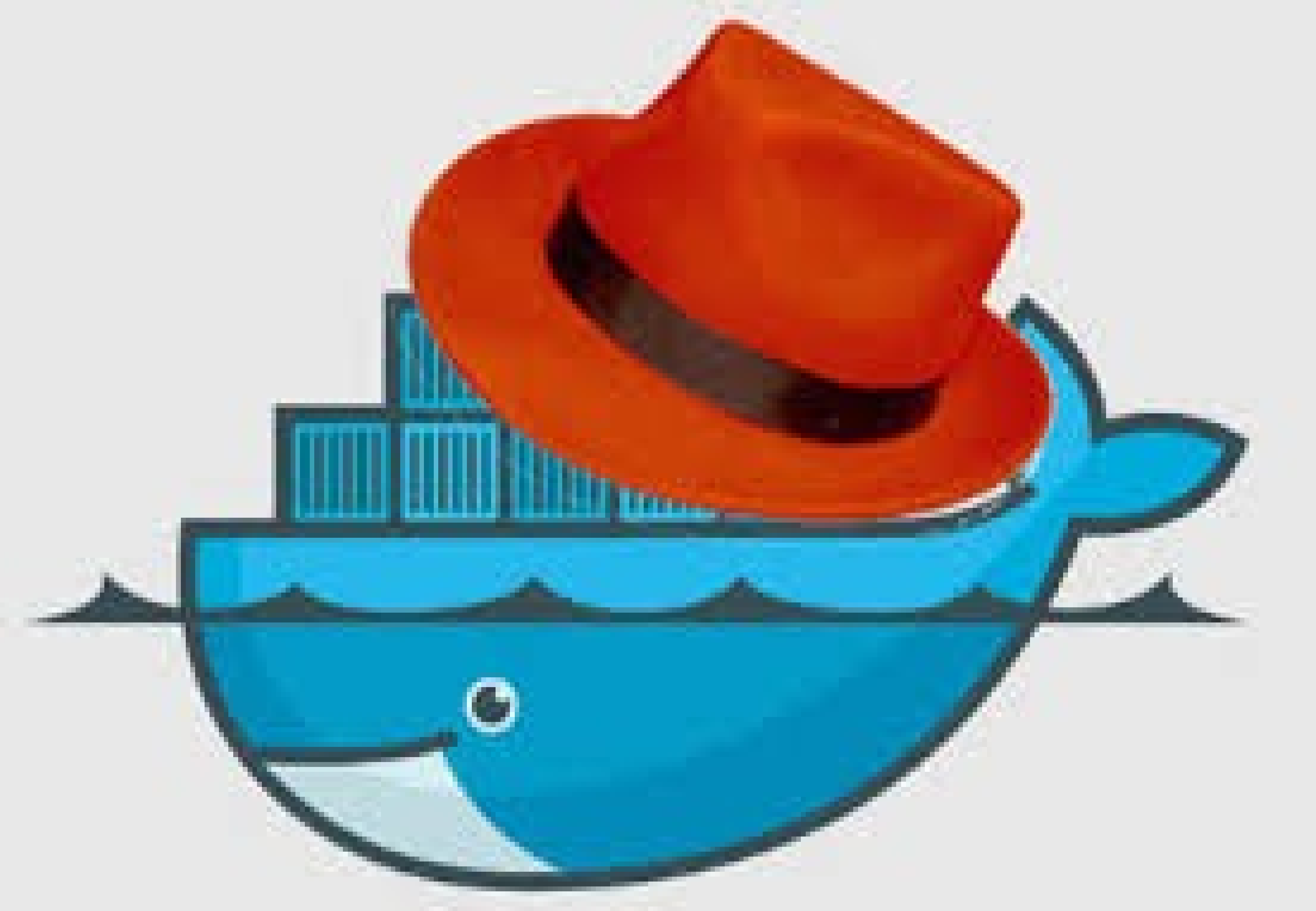
Dockerfile

- Scott Collier

```
rpm -q fedora-dockerfiles -l| grep /Dockerfile

/usr/share/fedora-dockerfiles/apache/Dockerfile

/usr/share/fedora-dockerfiles/couchdb/Dockerfile

/usr/share/fedora-dockerfiles/firefox/Dockerfile

/usr/share/fedora-dockerfiles/memcached/Dockerfile

/usr/share/fedora-dockerfiles/mongodb/Dockerfile

/usr/share/fedora-dockerfiles/mysql/Dockerfile

/usr/share/fedora-dockerfiles/nginx/Dockerfile

/usr/share/fedora-dockerfiles/nodejs/Dockerfile

/usr/share/fedora-dockerfiles/postgres/Dockerfile

/usr/share/fedora-dockerfiles/rabbitmq/Dockerfile

/usr/share/fedora-dockerfiles/ssh/Dockerfile
```



Red Hat Enhancements of docker



http://www.theregister.co.uk/2014/03/11/red_hat_docker_linux/

Who remembers Linux prior to Red Hat Enterprise Linux?

Linux 1999

Go to yahoo.com or AltaVista.com and google it?

I found it on rpmfind.net, download and install.

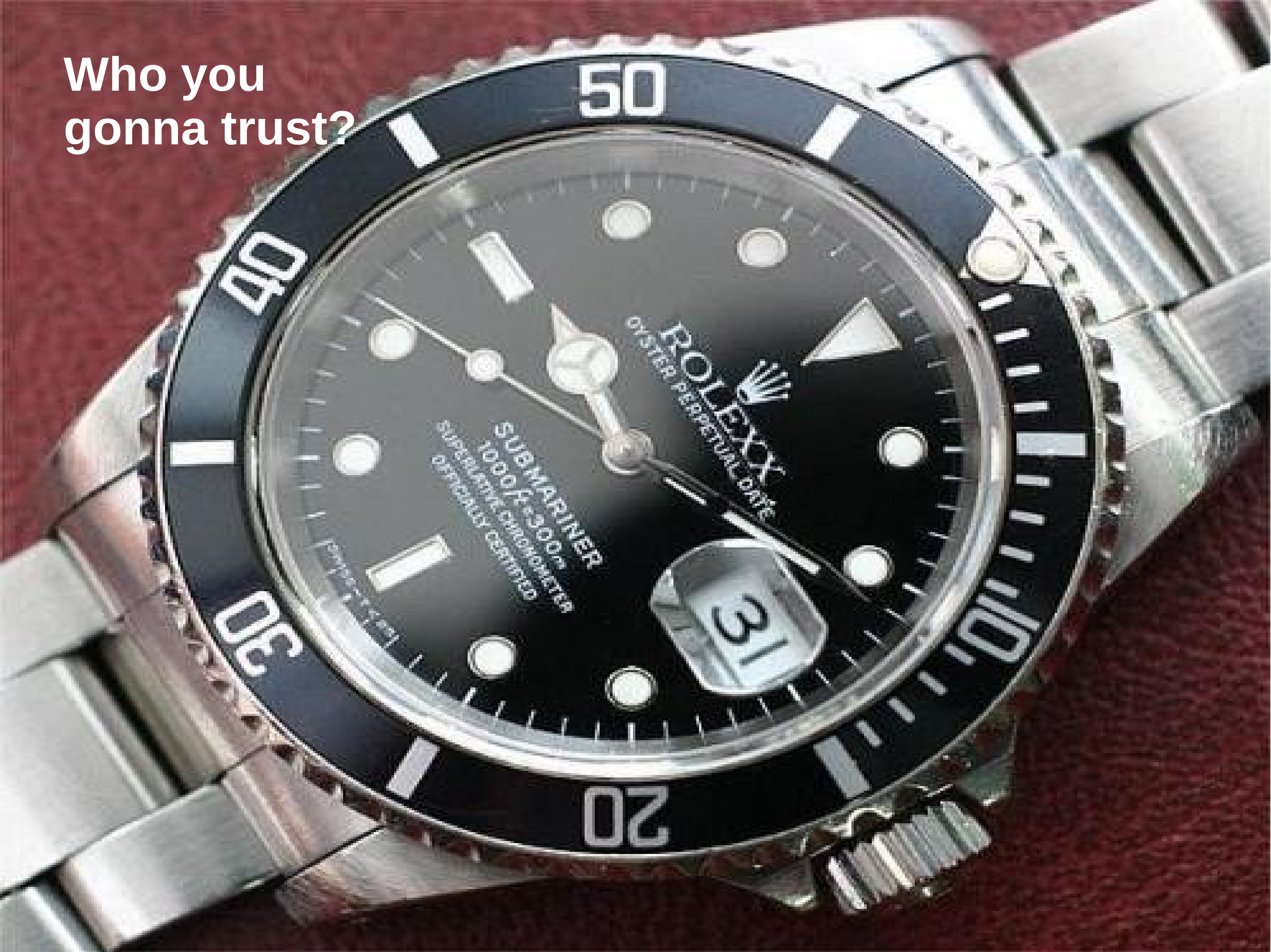
Hey I hear there is a big Security vulnerability in Zlib.

How many copies do you have on your system???

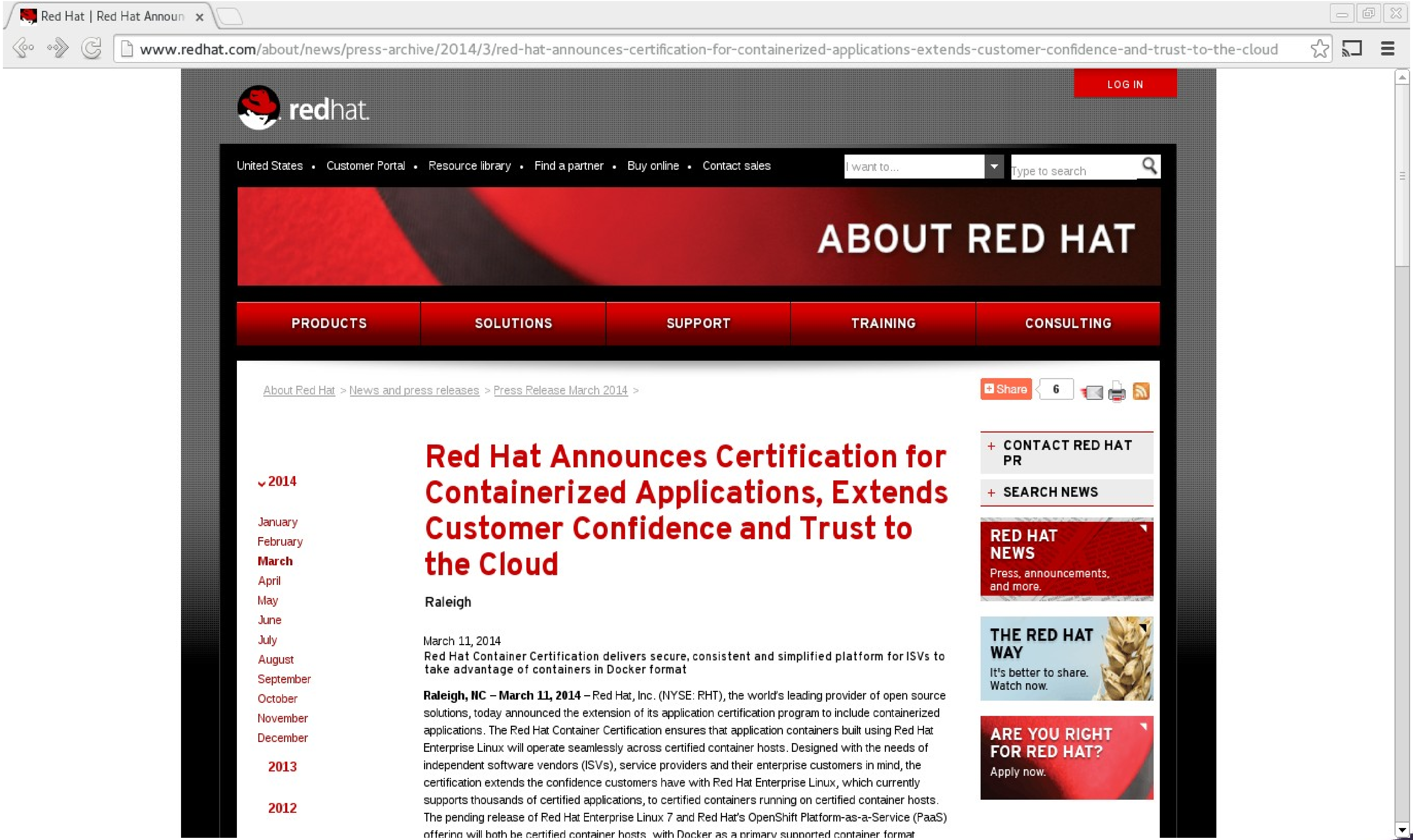
Bundling: Static Builds vs Shared Libraries

- A large part of the application developers dilemma:
  - What is part of the app and what is part of the dependency layer provided by OS?
  - What features can we depend on from the OS vs what should be “vendored” into the app?
- Shared Libraries:
  - RHEL and Linux in general depend on the use of shared libraries to ease security and feature updates
- Static Builds:
  - Vendors like to include (static link) as much as possible, but it leaves them open to vulnerabilities in unpatched code.

Who you gonna trust?



Red Hat Certified Images



www.redhat.com/about/news/press-archive/2014/3/red-hat-announces-certification-for-containerized-applications-extends-customer-confidence-and-trust-to-the-cloud

About Red Hat > News and press releases > Press Release March 2014 >

Red Hat Announces Certification for Containerized Applications, Extends Customer Confidence and Trust to the Cloud

Raleigh

March 11, 2014

Red Hat Container Certification delivers secure, consistent and simplified platform for ISVs to take advantage of containers in Docker format

Raleigh, NC – March 11, 2014 – Red Hat, Inc. (NYSE: RHT), the world's leading provider of open source solutions, today announced the extension of its application certification program to include containerized applications. The Red Hat Container Certification ensures that application containers built using Red Hat Enterprise Linux will operate seamlessly across certified container hosts. Designed with the needs of independent software vendors (ISVs), service providers and their enterprise customers in mind, the certification extends the confidence customers have with Red Hat Enterprise Linux, which currently supports thousands of certified applications, to certified containers running on certified container hosts. The pending release of Red Hat Enterprise Linux 7 and Red Hat's OpenShift Platform-as-a-Service (PaaS) offering will both be certified container hosts, with Docker as a primary supported container format

link

Red Hat Images

- RHEL6 and RHEL7 base images
- Potentially RHEL5 Base image?
- Packaged images?
  - httpd?,mariadb?, postgresql? FreeIPA?
- Layered Product Images
  - Jboss? Gluster? Openstack? ...

Red Hat Certified Images

- Partner Images
  - Third Party packagers
  - Build layered images on top of RHEL base images.

The Red Hat Container Certification ensures that application containers built using Red Hat Enterprise Linux will operate seamlessly across certified container hosts.

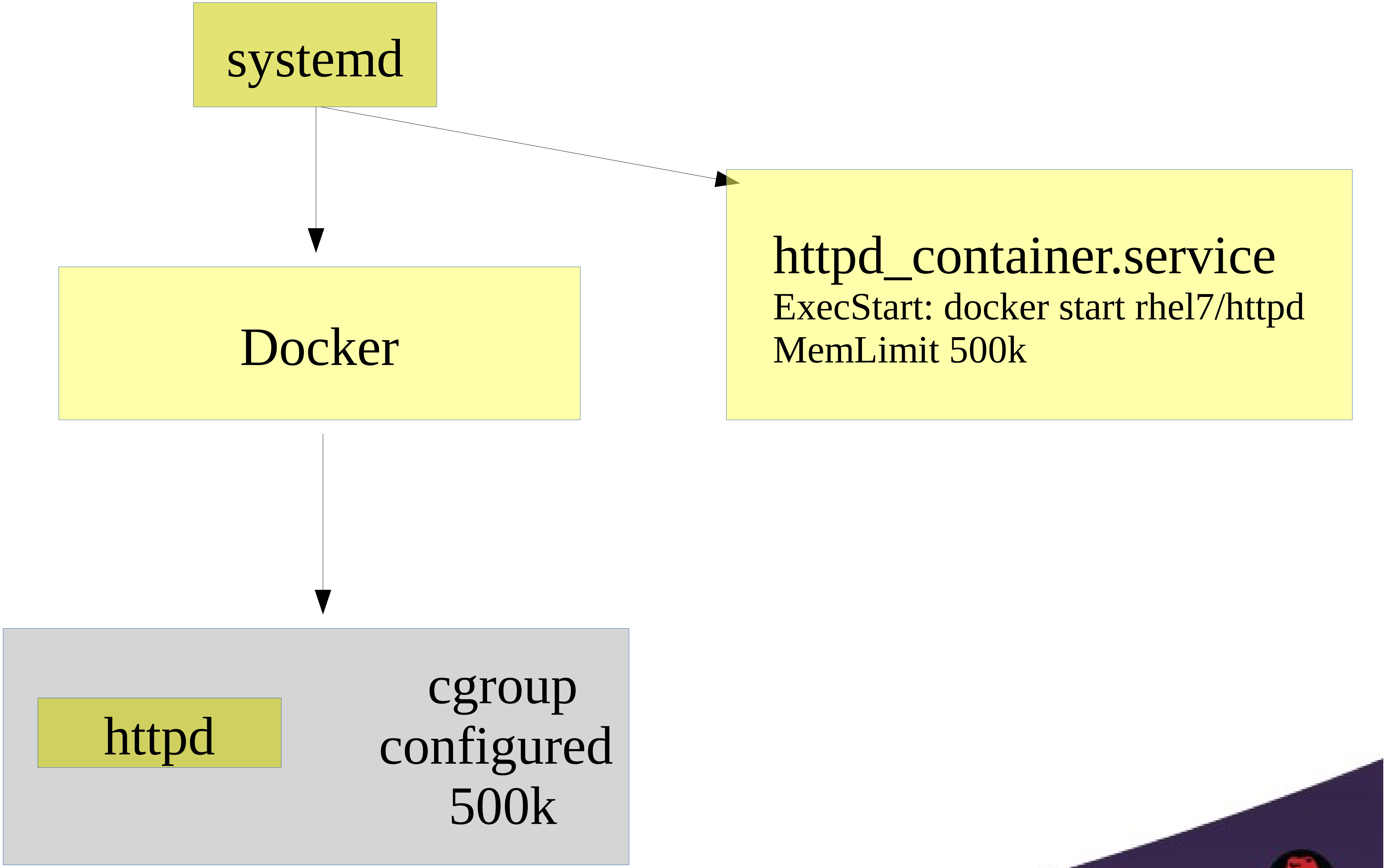
Docker == Static Builds

- Docker bundles userspace.
  - What happens when a Docker APP has a CVE?
  - You want to avoid application base image sprawl
  - Red Hat will update images with latest fixes
  - Customer apps will get fixes via subscription
- Customer apps based on RHEL images need simple rebuild.
  - docker build myapp

Systemd integration with Docker

- Manage application containers same as services
  - Docker container applications started on demand
  - Socket Activation
  - Cgroup Integration
- Journald logging
  - Stdout/stderr of container automatically logged to host.
  - Syslog messages automatically logged to host.

Systemd Cgroup Configuration passed to Docker



systemd

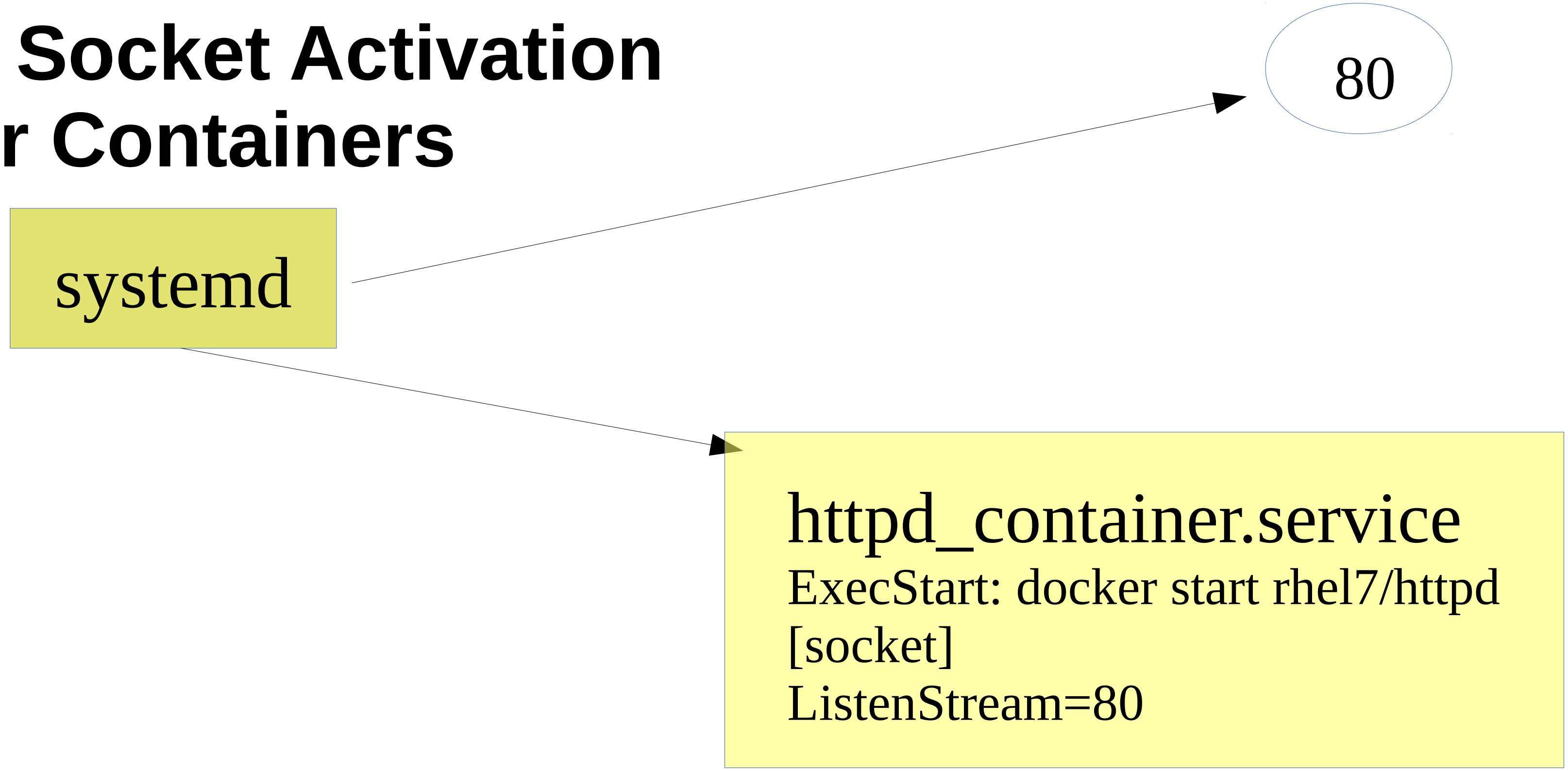
Docker

httpd_container.service
ExecStart: docker start rhel7/httpd
MemLimit 500k

httpd

cgroup
configured
500k

Systemd Socket Activation of Docker Containers



80

systemd

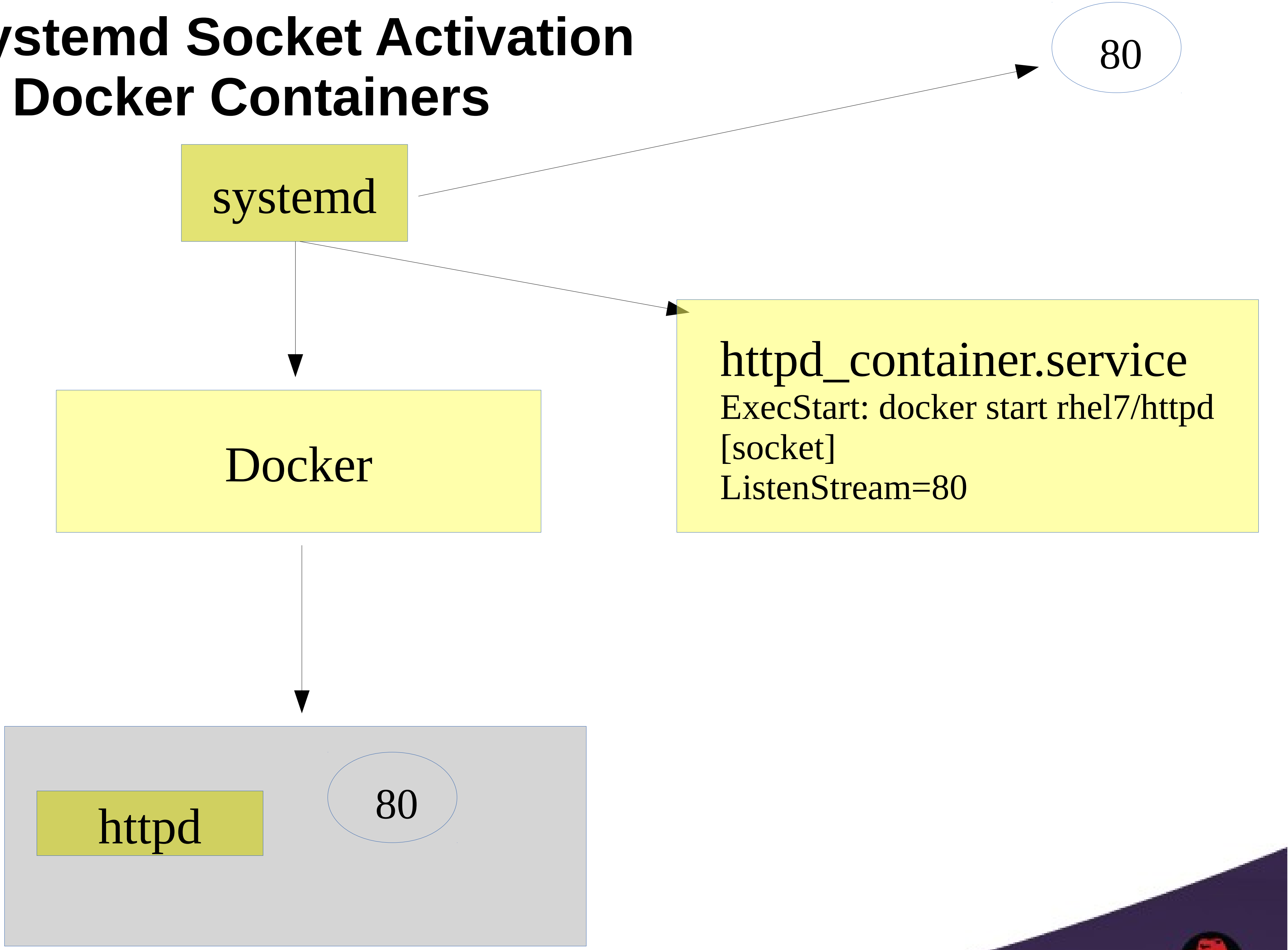
httpd_container.service

ExecStart: docker start rhel7/httpd

[socket]

ListenStream=80

Systemd Socket Activation of Docker Containers



ExecStart: docker start rhel7/httpd
[socket]
ListenStream=80

RHEL Security Integration

- Update RHEL images on CVE
  - Security response team
- SELinux integration
  - Containers will automatically be labeled based on sVirt
- libseccomp
- Auditing
  - Proper auditing of container events
    - Start/Stop

Thank-you!