



RED HAT®  
ENTERPRISE LINUX®

# Containers and Docker

Presenter: Dan Walsh

@rhatdan, Blog: [danwalsh.livejournal.com](http://danwalsh.livejournal.com),

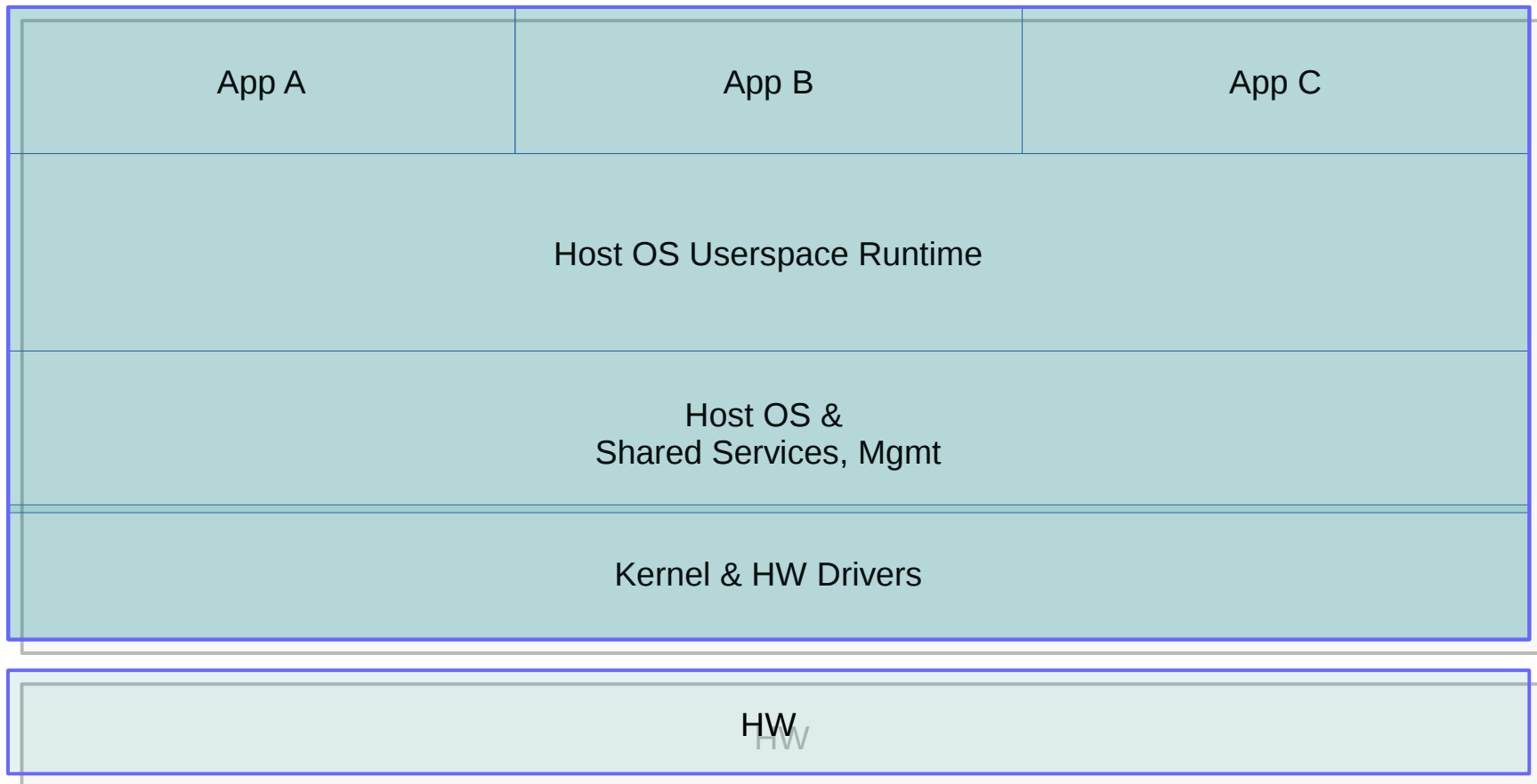
[dwalsh@redhat.com](mailto:dwalsh@redhat.com)

# Evolution of The Operating System

## RHEL 4

### Traditional Enterprise Operating System

Multiple Applications per machine + Single userspace runtime

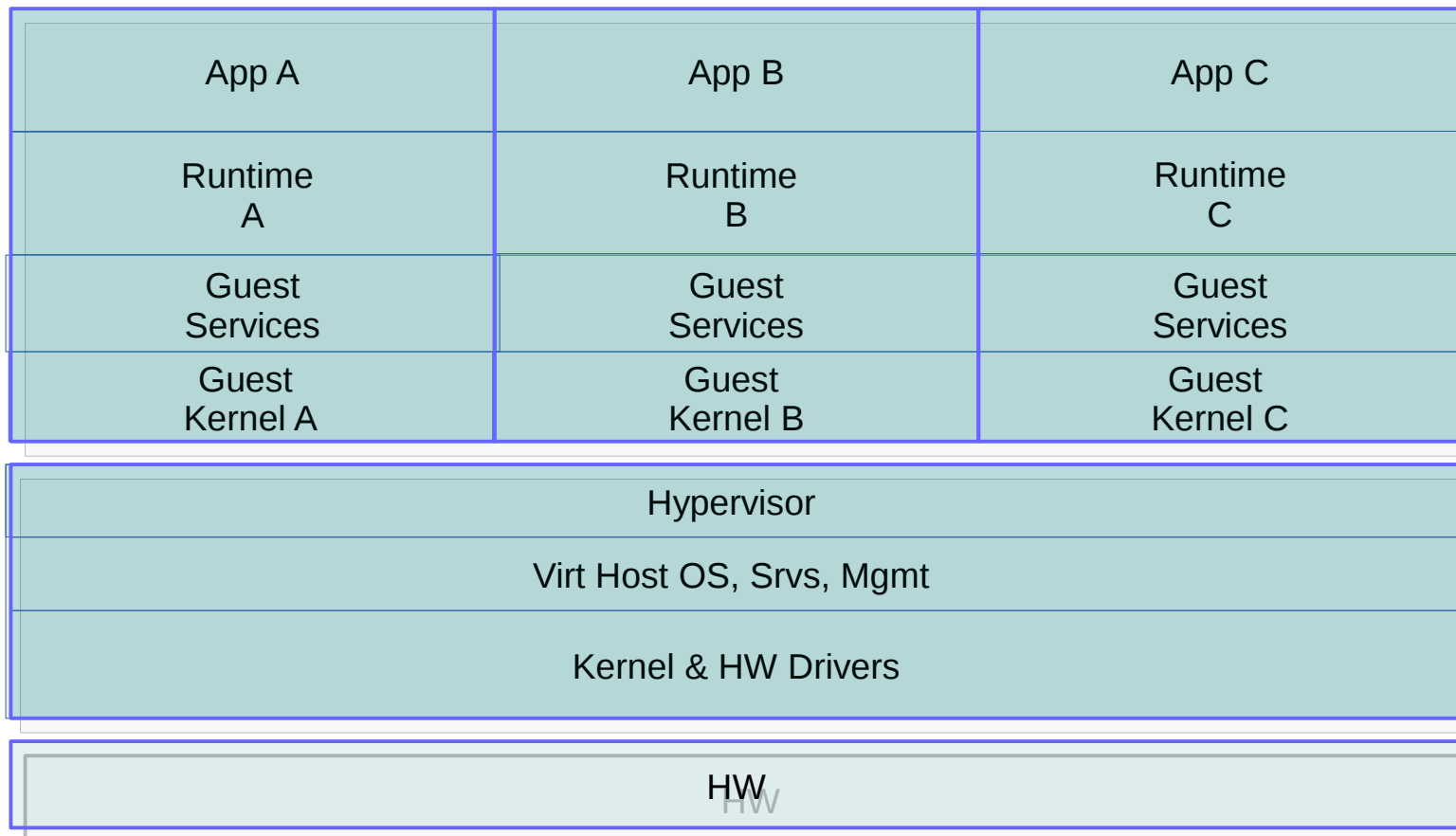


# Evolution of Operating System

## RHEL 5 & RHEL 6

### Virtualization & IaaS Cloud

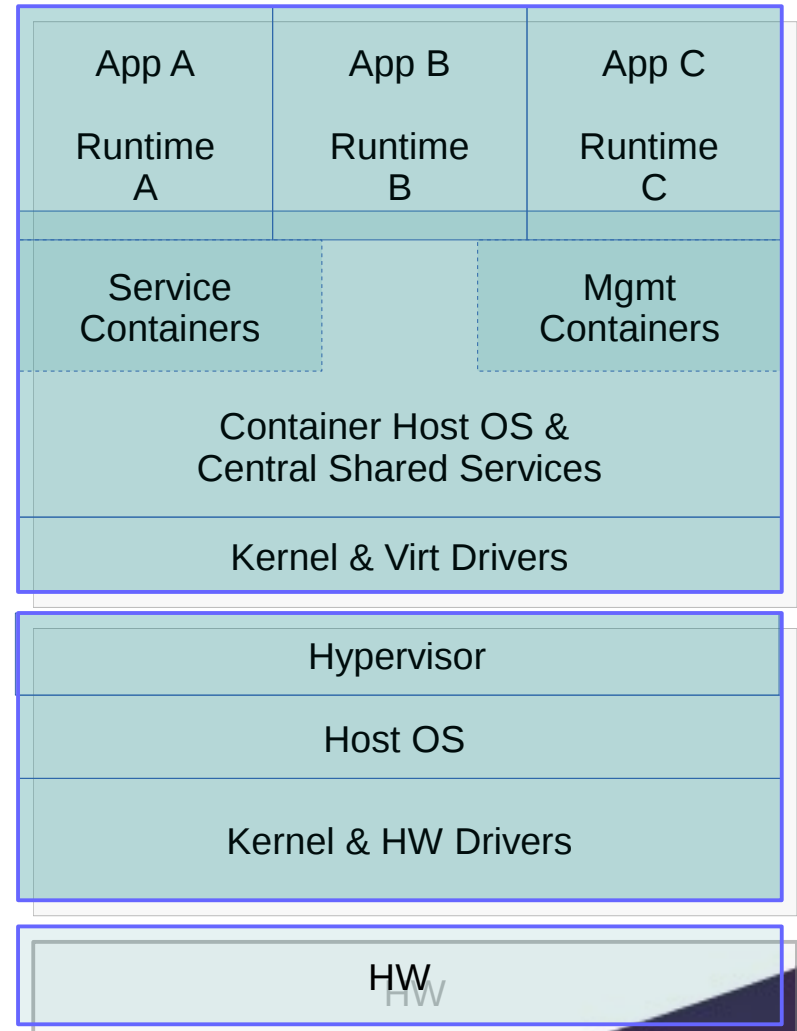
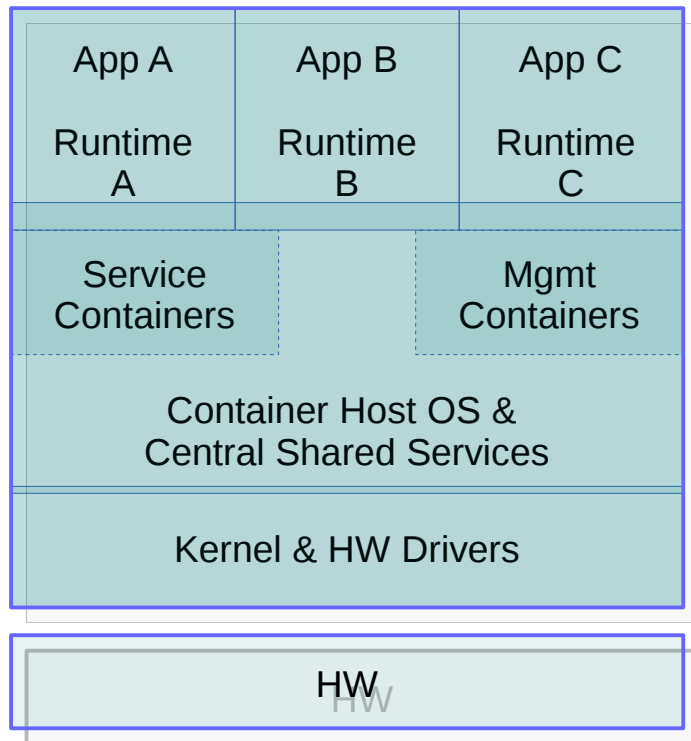
Applications run inside a guest – full separation of host and guest



# Evolution of Operating System RHEL 7

## Light-weight Application Isolation

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



# The kernel knows SQUAT about CONTAINERS



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



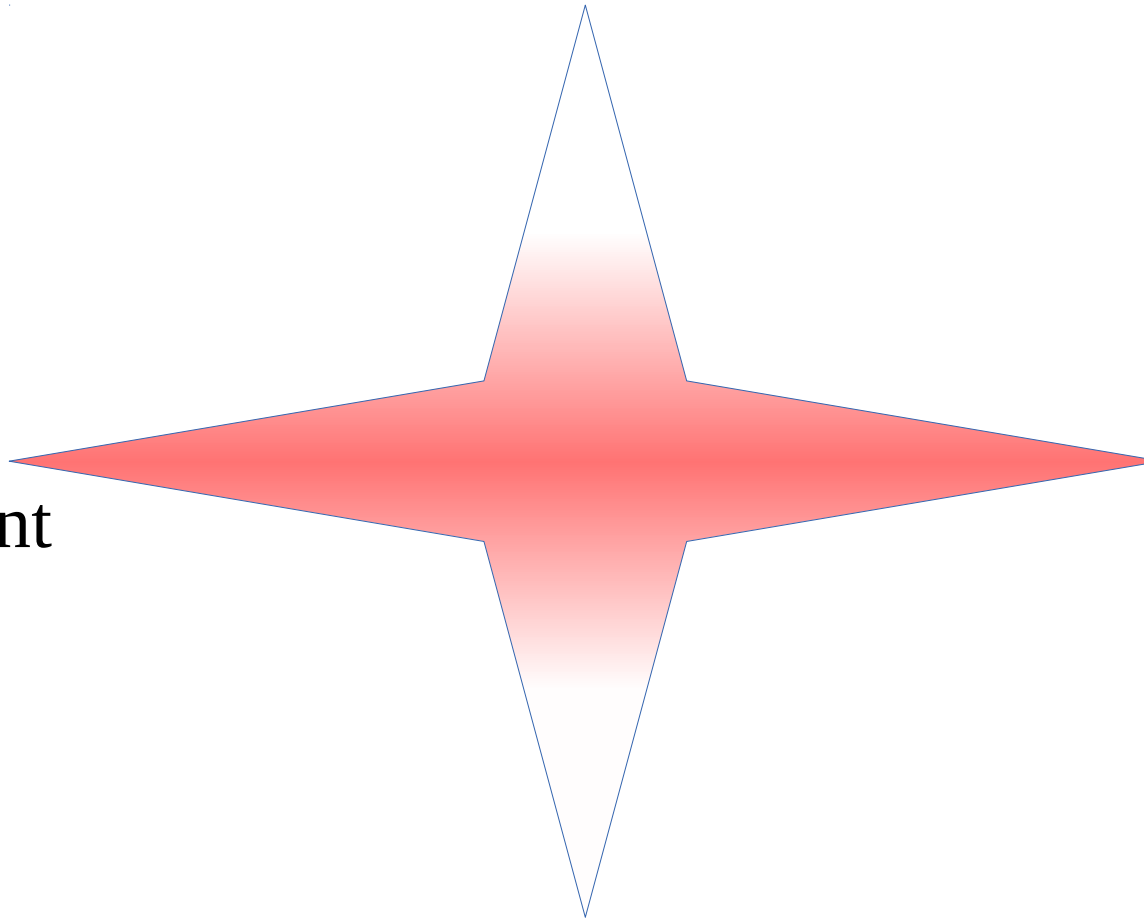
# Key elements of Linux Containers

Process Isolation

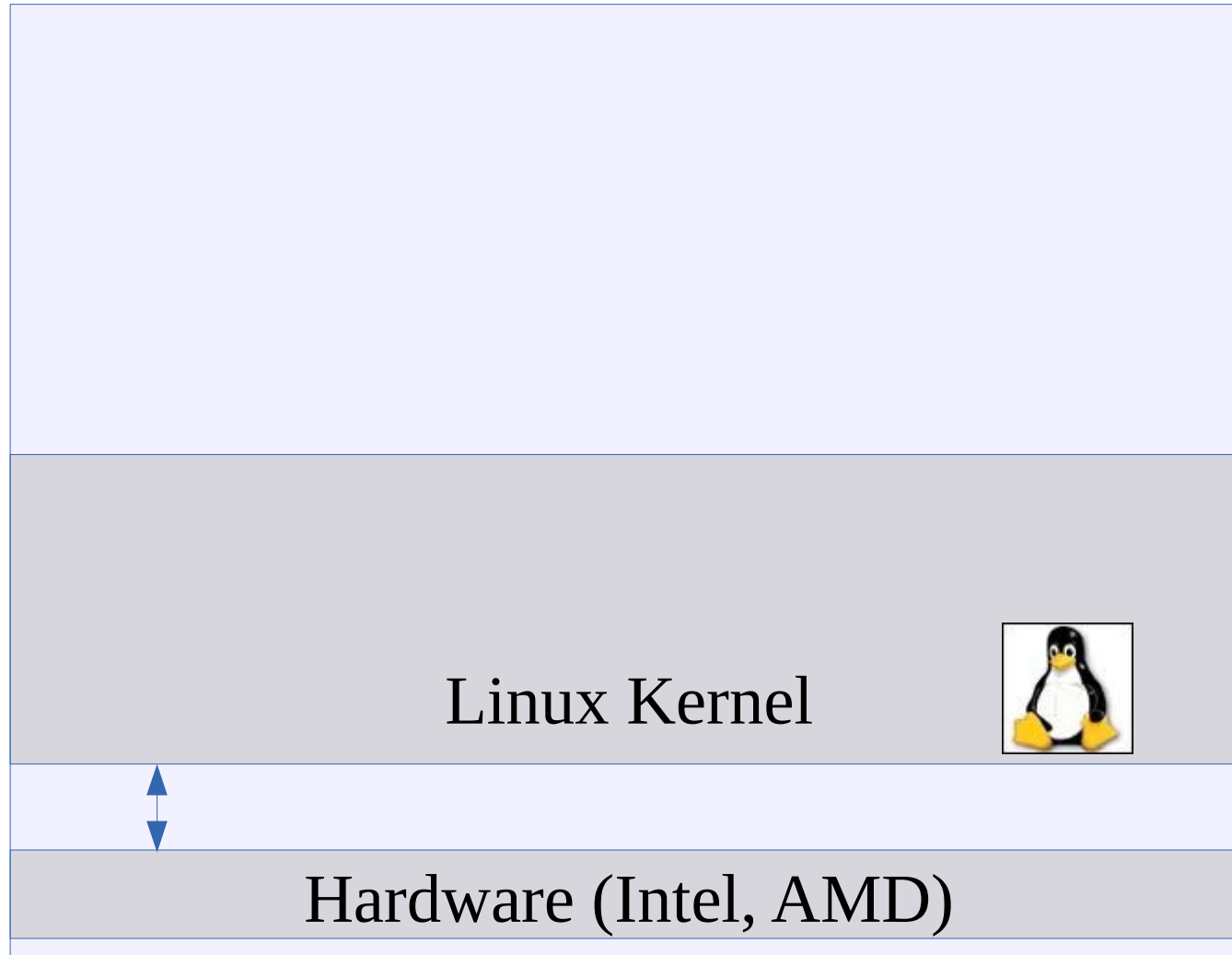
Resource  
Management

Security

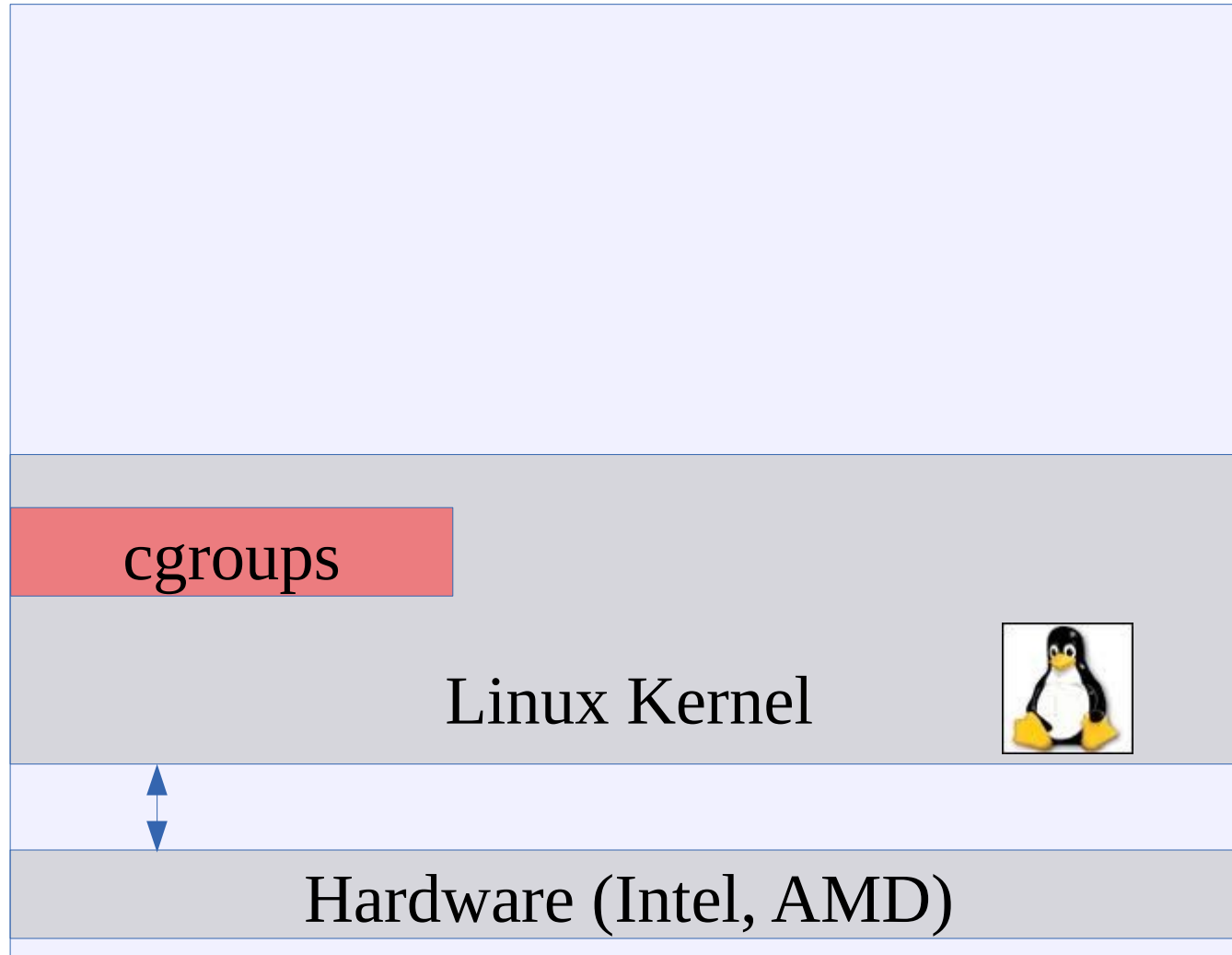
Management



# Red Hat Enterprise Linux Container Architecture

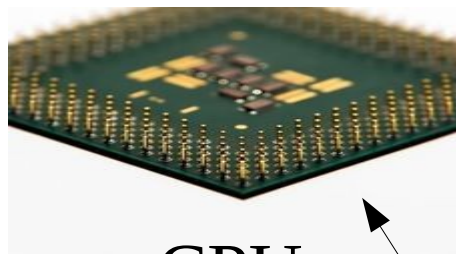
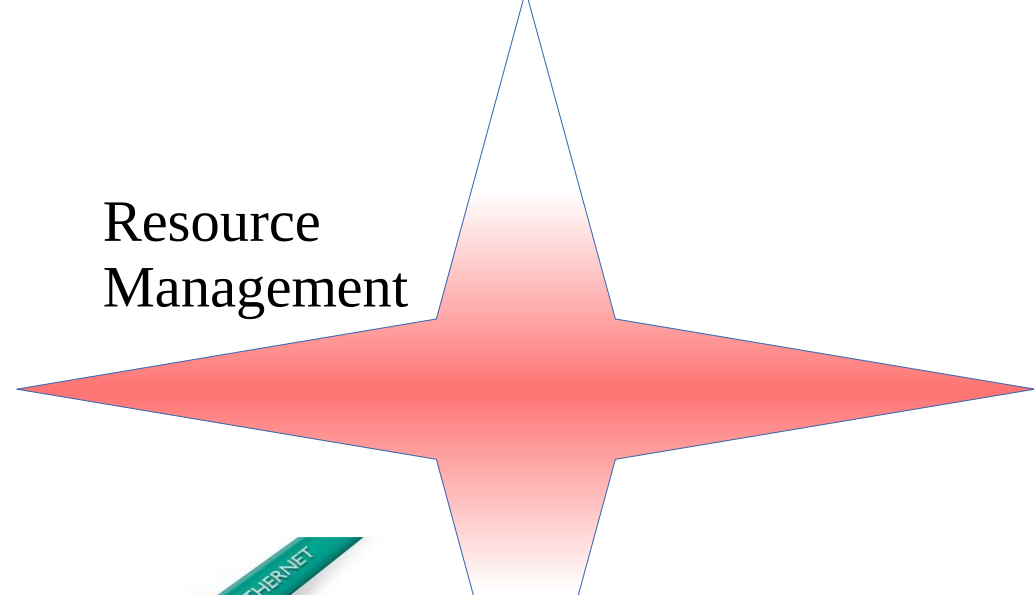


# Red Hat Enterprise Linux Container Architecture



# Cgroups

Resource Management



CPU

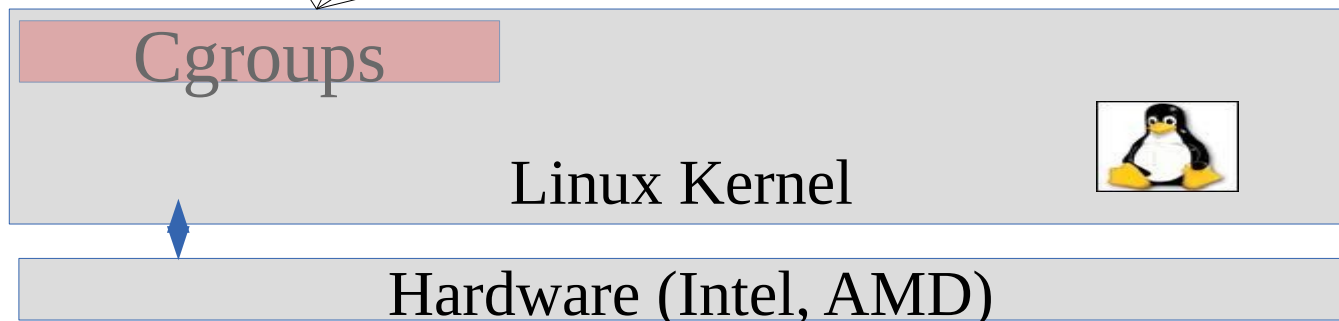
Memory



Network



Block IO



# Systemd management of cgroups

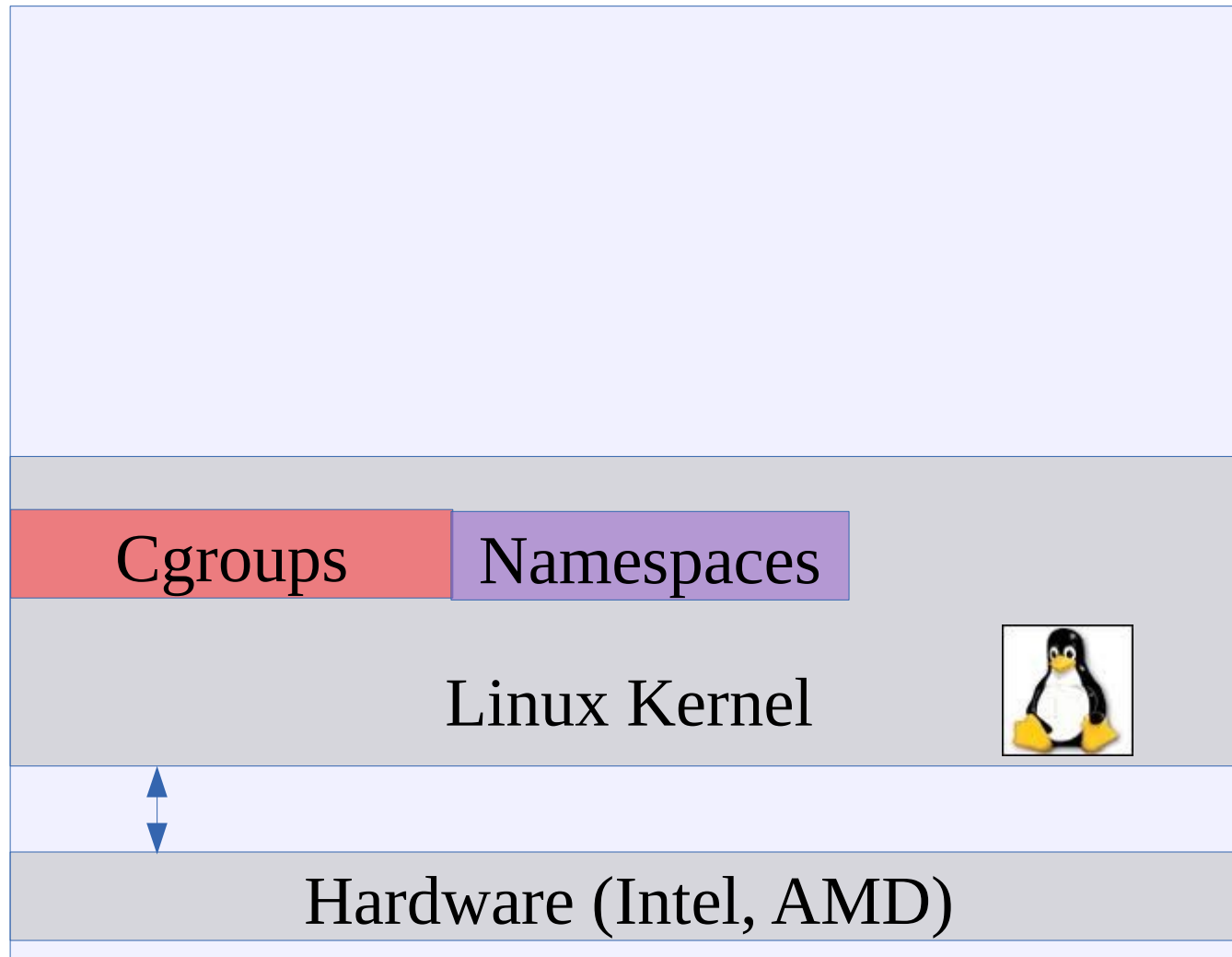
- Systemd unit file simple way to manage cgroups
  - `man systemd.resource-control`
    - `CPUShares=100`
    - `MemoryLimit=500M`
  - `systemctl set-property foobar.service CPUShares=777`
  - `systemd-run -p CPUShares=100 updatedb`



# Libcgroup available but being retired



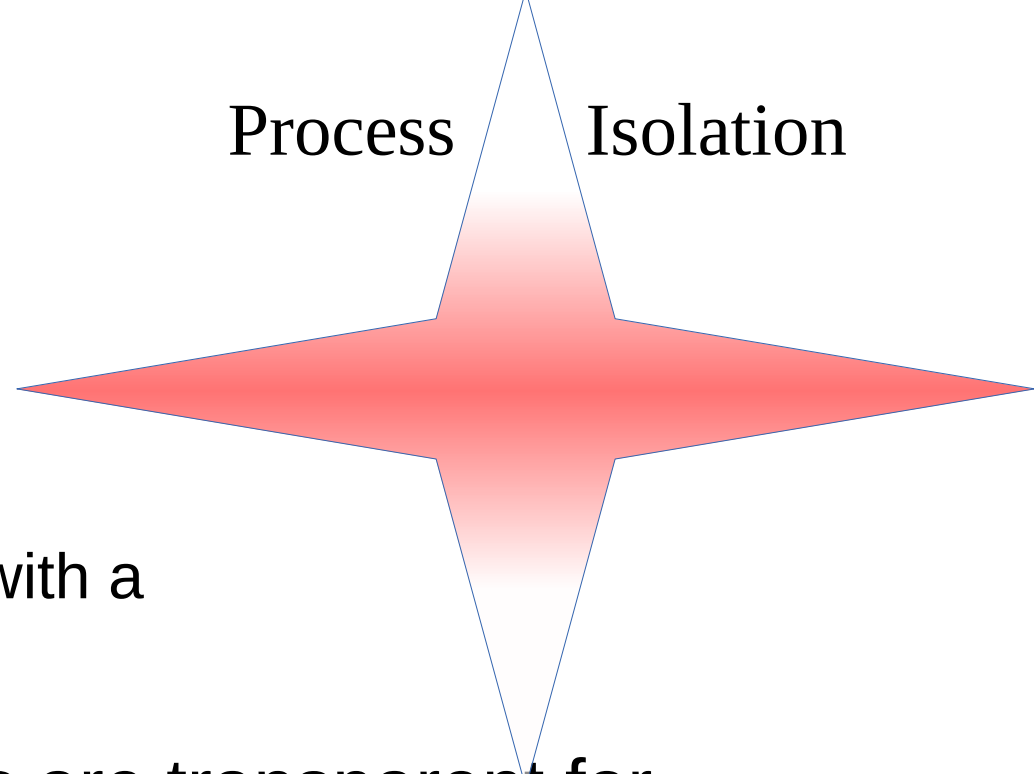
# Red Hat Enterprise Linux Container Architecture



# 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



# Systemd management of namespaces

- Systemd unit file simple way to use namespaces
  - PrivateTmp
  - PrivateNet
  - PrivateDevice (Future)
- systemd-nspawn
  - Run chroot OS within container

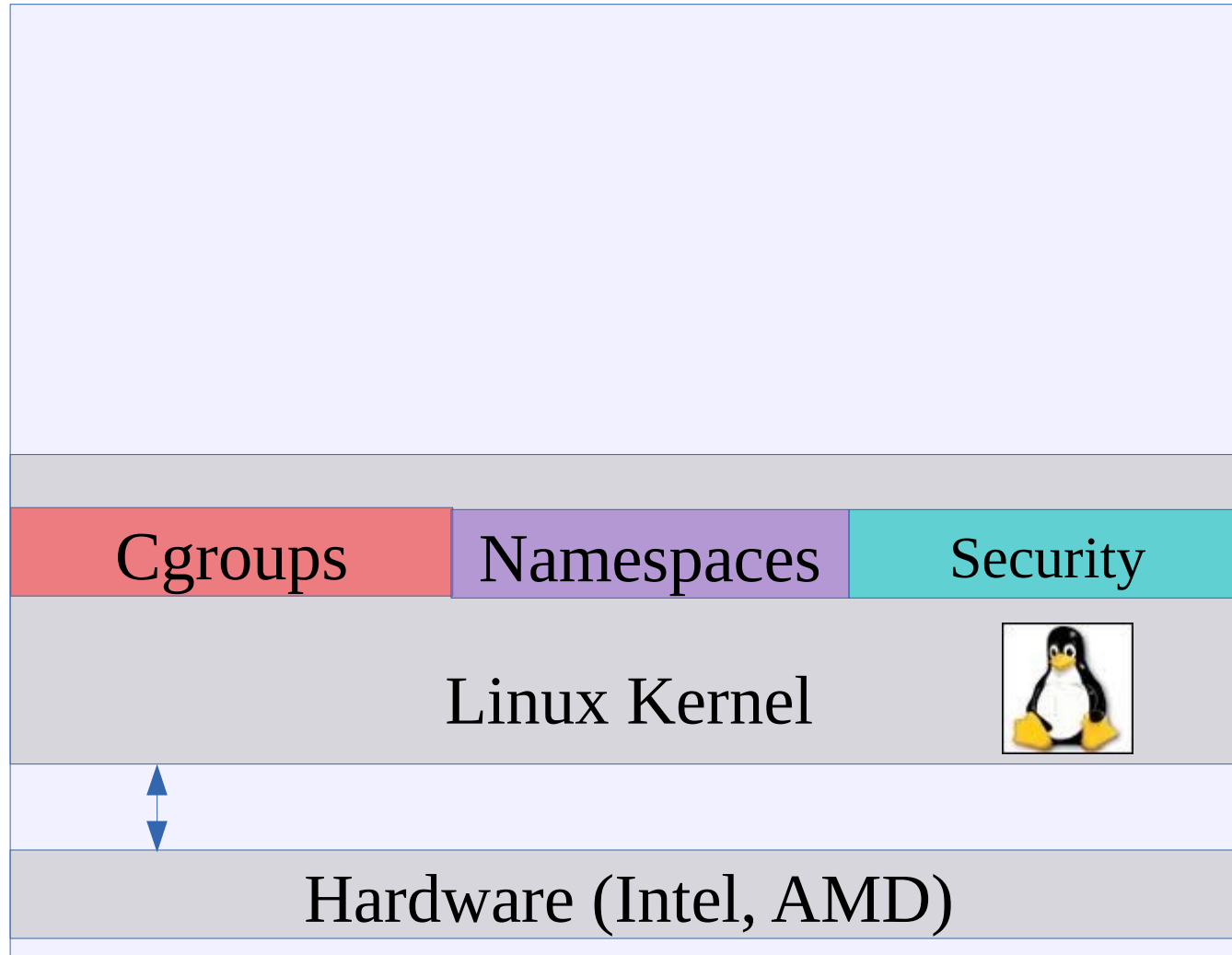


# Other Namespace Managers

- unshare tool
  - Allows you to experiment with namespaces
  - Potentially could use it with scripts
- SELinux Sandbox
  - Sandbox desktop applications
    - `sandbox -X evince foobar.pdf`
- libvirt-lxc
  - You can build containers using `virsh`
- Build your own
  - `man clone`
  - `man unshare`



# Red Hat Enterprise Linux Container Architecture



# Containers do NOT Contain!!!

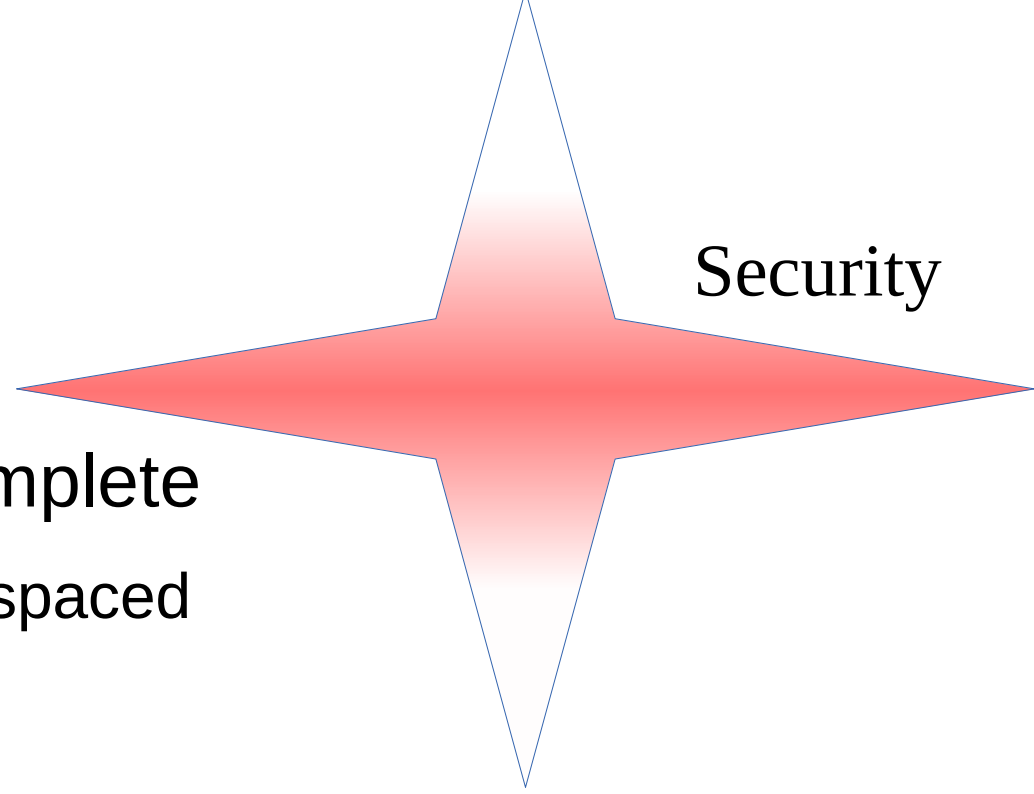


Les conteneurs  
ne contiennent pas!!!



# 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



# Links for Docker Security

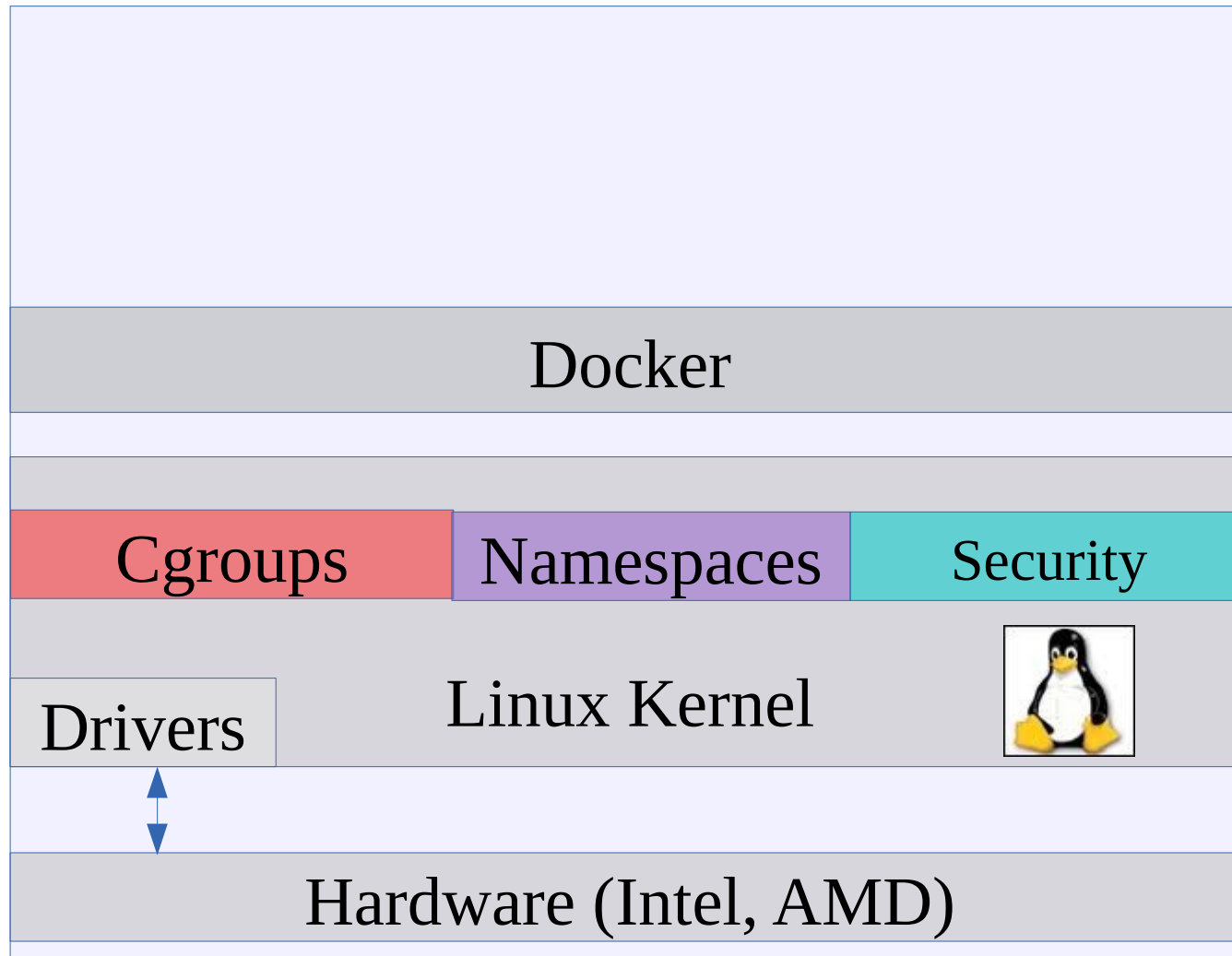
<http://opensource.com/business/14/7/docker-security-selinux>

<https://opensource.com/business/14/9/security-for-docker>

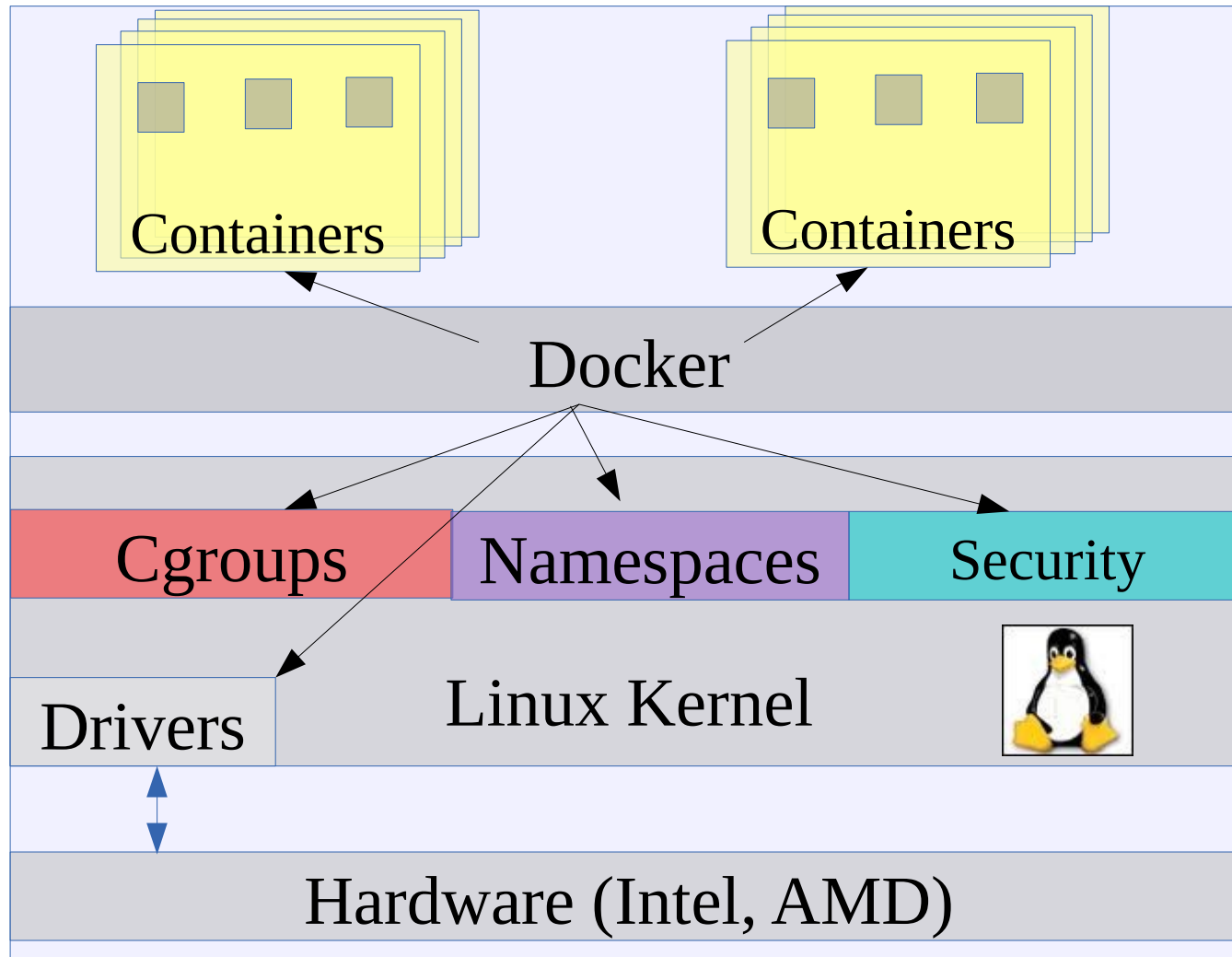
<https://www.youtube.com/watch?v=zWGFqMuEHdw>



# Red Hat Enterprise Linux Container Architecture



# Red Hat Enterprise Linux Container Architecture





September 19, 2013

# RED HAT AND DOCKER COLLABORATE

We are thrilled to [announce](#) the collaboration between [Docker](#) and [Red Hat](#).

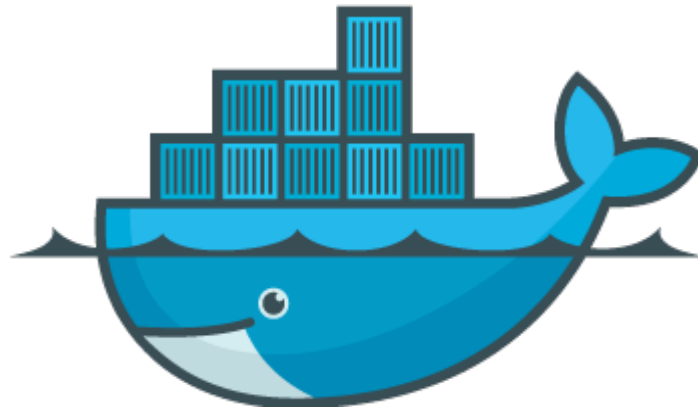
Collaboration with Red Hat is important for a number of reasons, including:

- 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

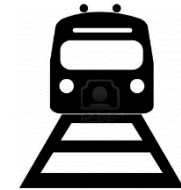
**First, it is critically important for us to make Docker work seamlessly with Red Hat Enterprise Linux and related Linux distributions, such as Fedora.**

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, and we believe this initial collaboration will ultimately benefit Docker users

# 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.

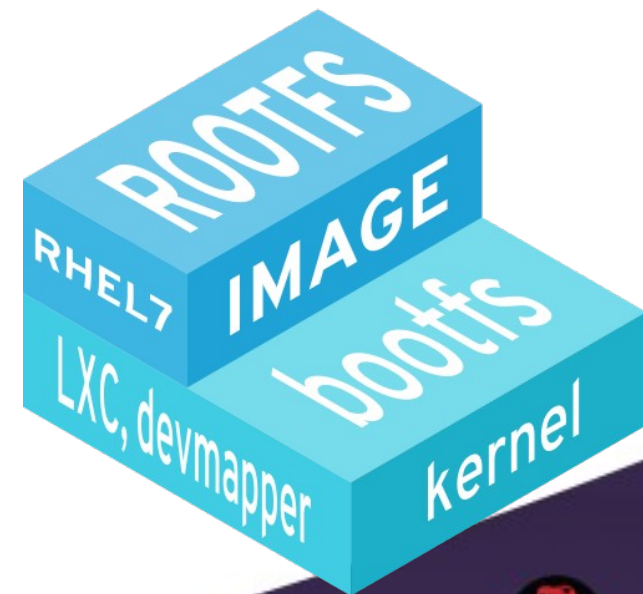


Docker as a packaging tool for shipping software may be a game changer.



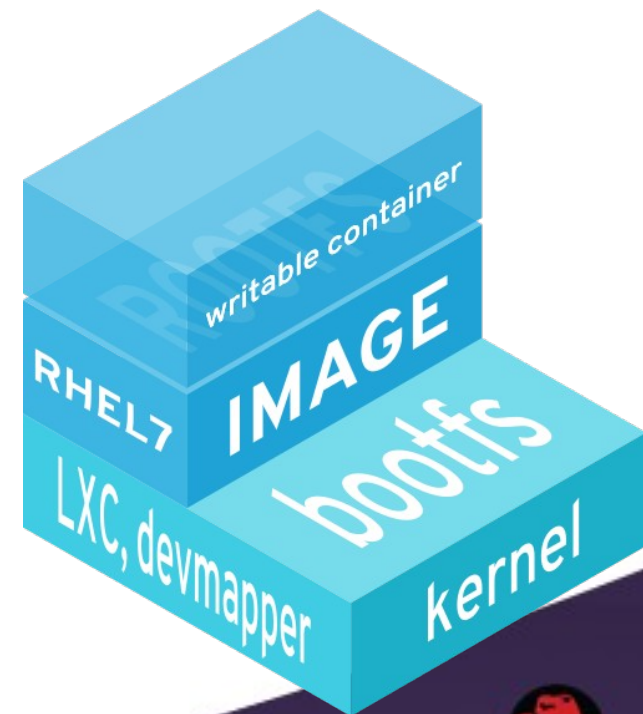
# Docker glossary

- 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

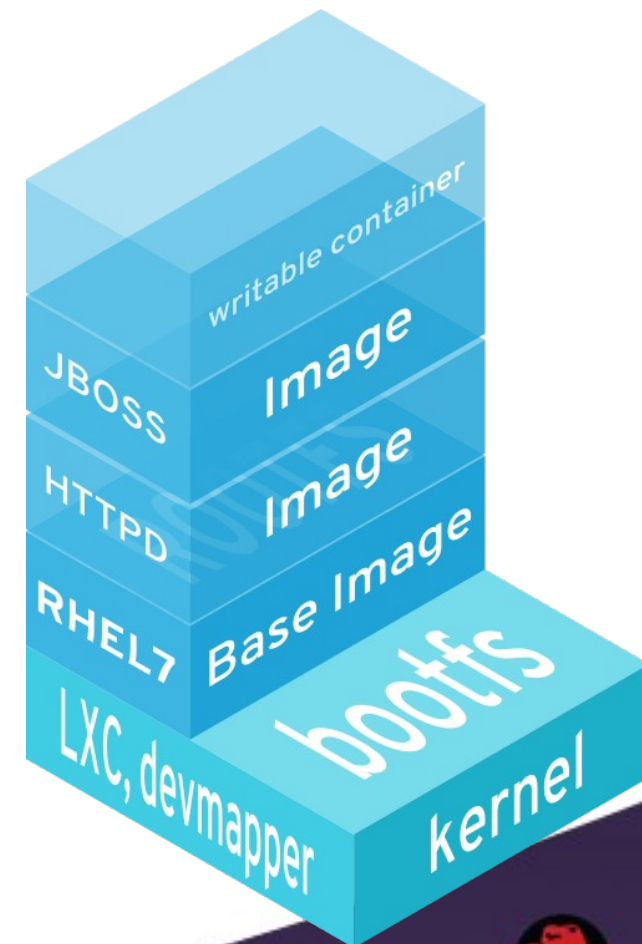
- Container
  - Running processes
  - Based on a particular image
  - Typically a single process
  - Isolated from host system
  - Cheap
  - Can write to filesystem
  - Commit creates new Image



# 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
```



# Docker 101

- Working with existing containers.

```
$> docker start httpd_container
```

```
$> docker stop httpd_container
```

```
$> docker kill httpd_container
```

```
$> docker commit httpd_container rhel7/httpd
```

- List running containers

```
$> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0718f7c6c53e	rhel6:latest	/bin/sh	4 seconds ago	Up 2 seconds		nostalgic_stallman
393c8c9908dd	redhat/rhel7.0:0	"sleep 6000"	About a minute ago	Up About a minute		drunk_brattain
233fc2fab09dd	httpd_image	/usr/sbin/httpd	1 day	Up 1 day	0.0.0.0:20->20/tcp	httpd_container



# Dockerfile

- Simple instructions
- Feels like scripting

```
FROM rhel7
```

```
RUN yum install -y postgresql && mkdir -p /data/db
```

```
EXPOSE 27017
```

```
VOLUME ["/data/db"]
```

```
CMD "postgresql"
```

- Easy to make an image

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



# 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
```

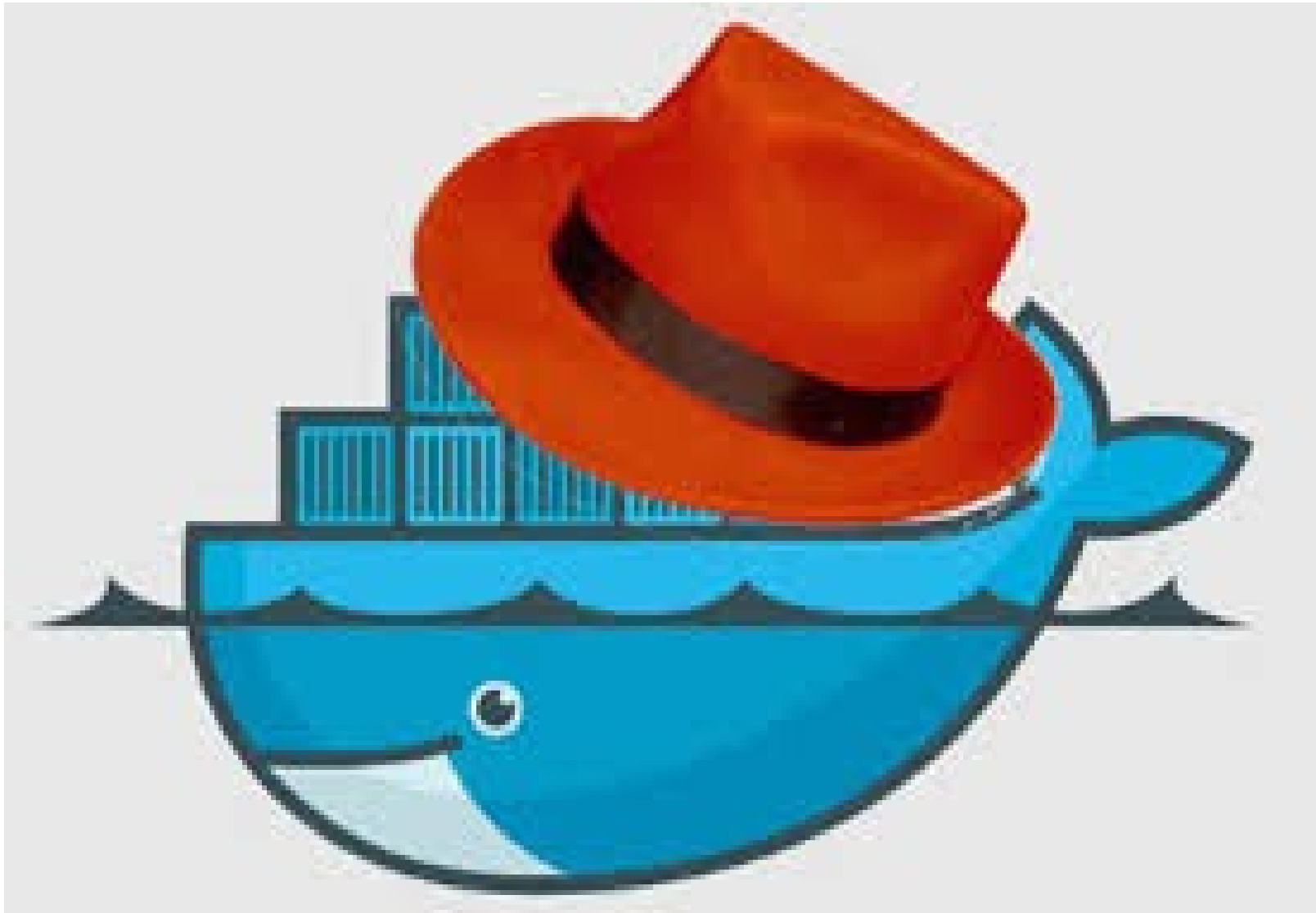
```
/usr/share/fedora-dockerfiles/postgres/Dockerfile
```

```
/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/](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](http://yahoo.com) or [AltaVista.com](http://AltaVista.com)  
and google it?



I found it on [rpmfind.net](http://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.

# 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.

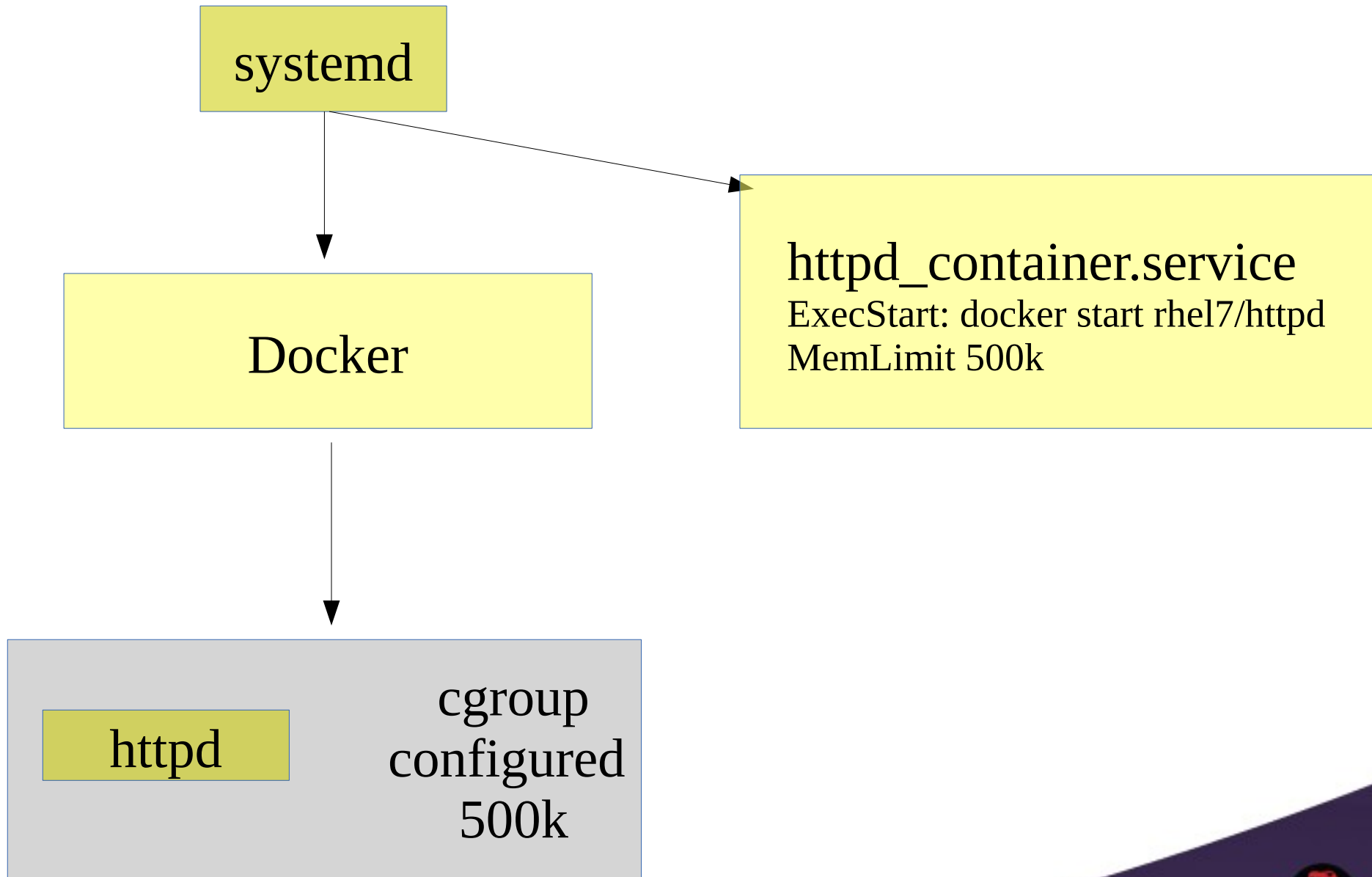


# Systemd integration with Docker

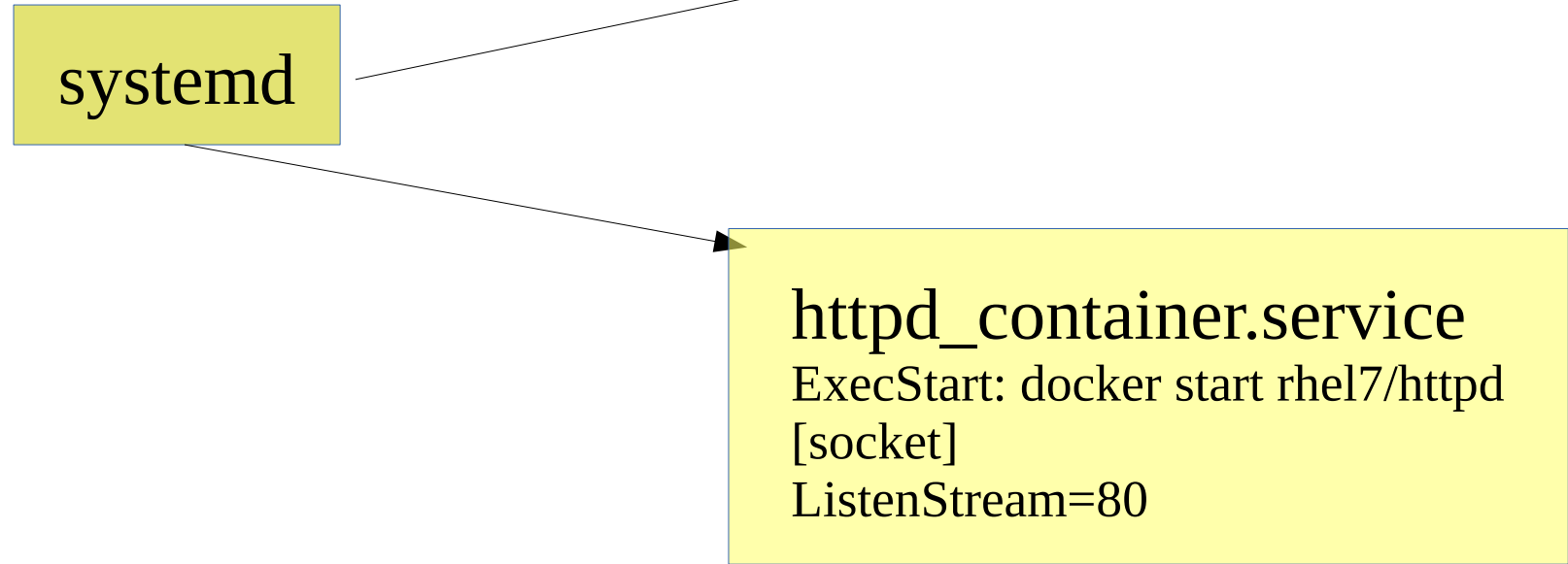
- Manage application containers same as services
  - Docker container applications started on demand
  - Socket Activation
  - Cgroup Integration
- Journald logging
  - Stdout/stderr of container automatically logged to host.
  - Syslog messages automatically logged to host.



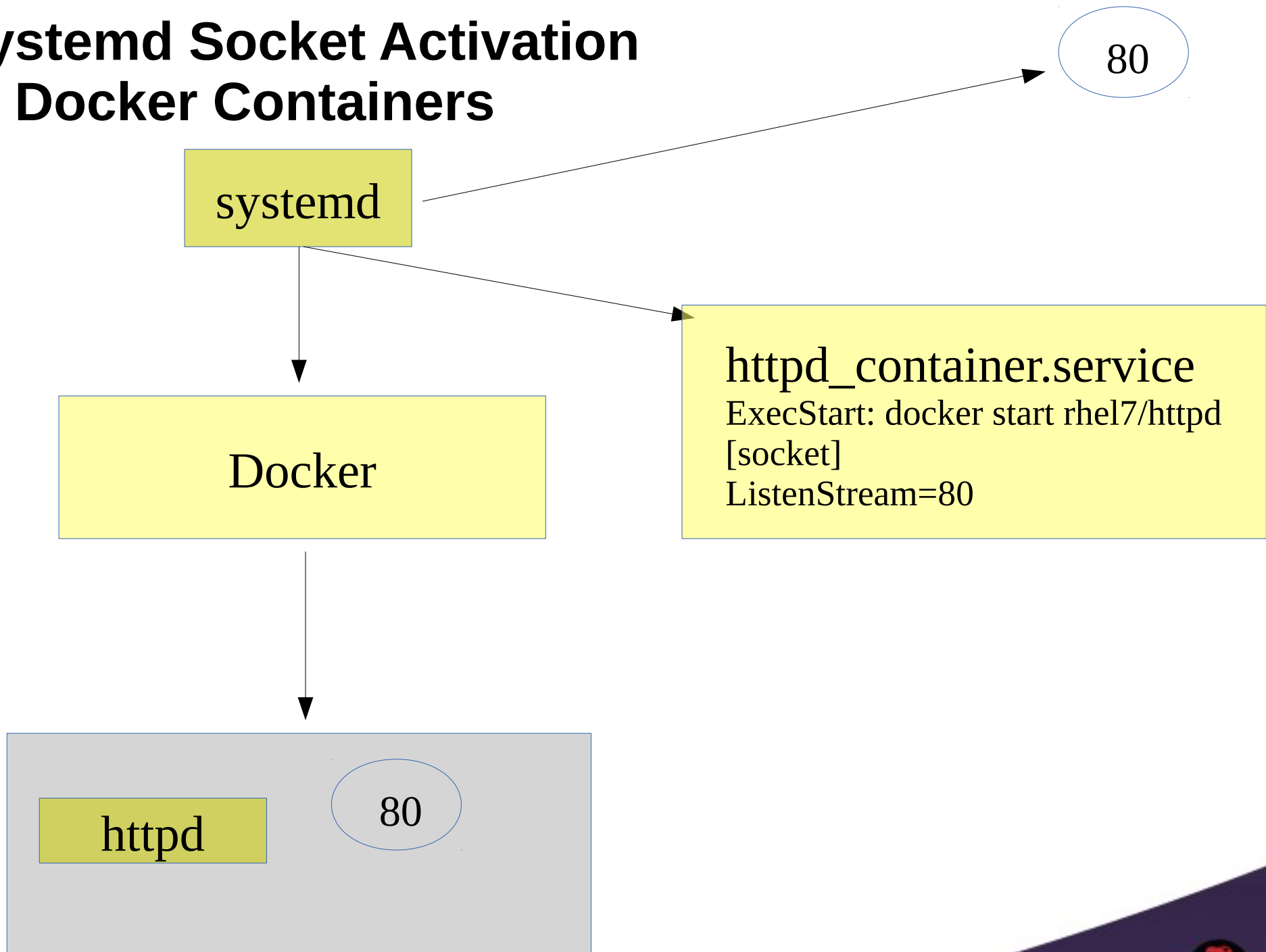
# Systemd Cgroup Configuration passed to Docker



# Systemd Socket Activation of Docker Containers



# Systemd Socket Activation of Docker Containers



# httpd\_container.service

[Unit]

Description=httpd\_container

After=docker.service

Requires=docker.service

[Service]

TimeoutStartSec=0

ExecStartPre=-/usr/bin/docker kill httpd\_container

ExecStartPre=-/usr/bin/docker rm httpd\_container

ExecStartPre=/usr/bin/docker pull httpd\_image

ExecStart=/usr/bin/docker run -d --name httpd\_container httpd\_image

ExecStop=-/usr/bin/docker stop httpd\_container

MemLimit=500k

CPUShares=1000

[Install]

52 WantedBy=multi-user.target



# 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



# Further information on Containers.

- Get Started with Docker Containers in RHEL 7
  - <http://red.ht/1vcxBVv>
- What is Docker and how is it integrated in RHEL 7?
  - <http://red.ht/1Dwtvxf>
- RHEL7 Resource Management & Linux Containers
  - CH 6. USING DOCKER
    - <http://red.ht/1uhQXvc>
- RHEL7 Release Notes:
  - CH 7. LINUX CONTAINERS WITH DOCKER FORMAT
    - <http://red.ht/1zjfOPU>



A night sky filled with stars and the Milky Way galaxy. In the foreground, there are silhouettes of a large tree on the left, a satellite dish in the center, and a line of trees on the right. The text "Thank-you!" is centered in the sky.

**Thank-you!**