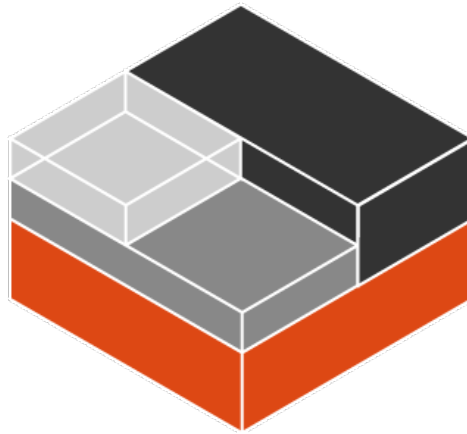


An Introduction to Kubernetes



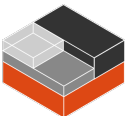
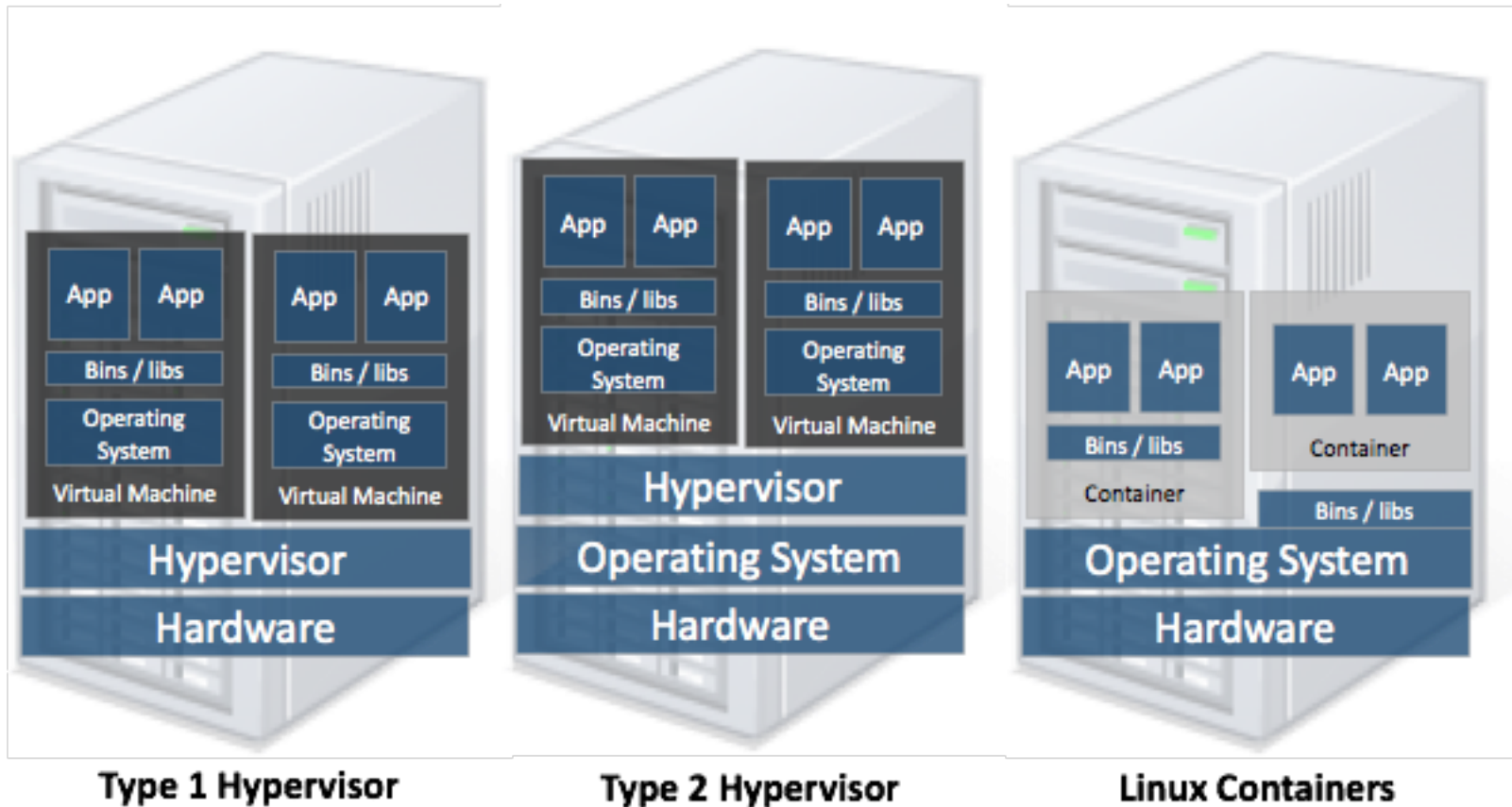
Agenda

- Linux Containers
- Docker
- Kubernetes
- Kubernetes Architecture
- Kubernetes Demo



Linux Containers

Linux Containers



Linux Containers

An operating system–level virtualization method for running multiple isolated Linux systems (containers) on a single control host.

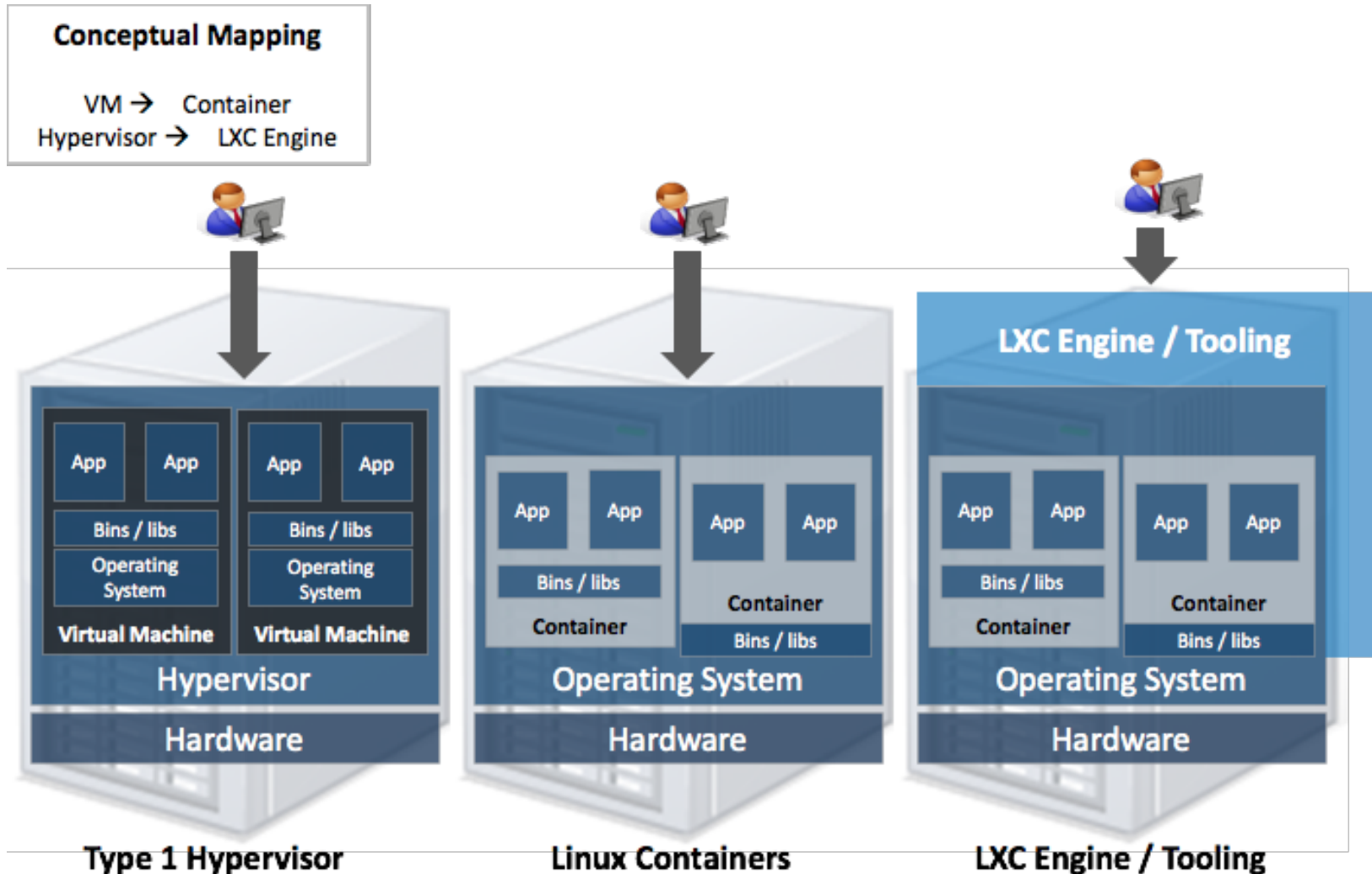


Linux Kernel Features used by Linux Containers



- Namespaces
(mnt, pid, net, ipc, uts/hostname, user ids)
- cgroups
(cpu, memory, disk, i/o - resource management)
- AppArmor, SELinux
(security/access control)
- seccomp
(computation isolation)
- chroot
(file system isolation)



LXC Engine: A Hypervisor for Containers



More about Linux Containers

  <https://linuxcontainers.org>

Projects. The interesting stuff.

LXC

LXC is the well known set of tools, templates, library and language bindings. It's pretty low level, very flexible and covers just about every containment feature supported by the upstream kernel.

LXC is production ready with LXC 1.0 getting 5 years of security updates and bugfixes (until April 2019).

[More »](#)

LXD

LXD is the new LXC experience. It offers a completely fresh and intuitive user experience with a single command line tool to manage your containers. Containers can be managed over the network in a transparent way through a REST API. It also works with large scale deployments by integrating with OpenStack.

LXD was announced in early November 2014 and is still under very active development.

[More »](#)

CGManager

CGManager is our cgroup manager daemon. It's designed to allow nested unprivileged containers to still be able to create and manage their cgroups through a DBus API.

CGManager has been used by default with LXC in Ubuntu since April 2014 and since by other distributions as they start needing working unprivileged containers.

[More »](#)

LXCFS

Userspace (FUSE) filesystem offering two main things:

- Overlay files for `cpuinfo`, `meminfo`, `stat` and `uptime`.
- A `cgroupfs` compatible tree allowing unprivileged writes.

It's designed to workaround the shortcomings of `procfs`, `sysfs` and `cgroupfs` by exporting files which match what a system container user would expect.

[More »](#)





Docker

- A platform for managing Linux Containers
- Began as an open-source implementation of the deployment engine which powers dotCloud
- Started in March, 2013
- Provided an easy to use API and powerful container image management features
- Attracted the community very fast

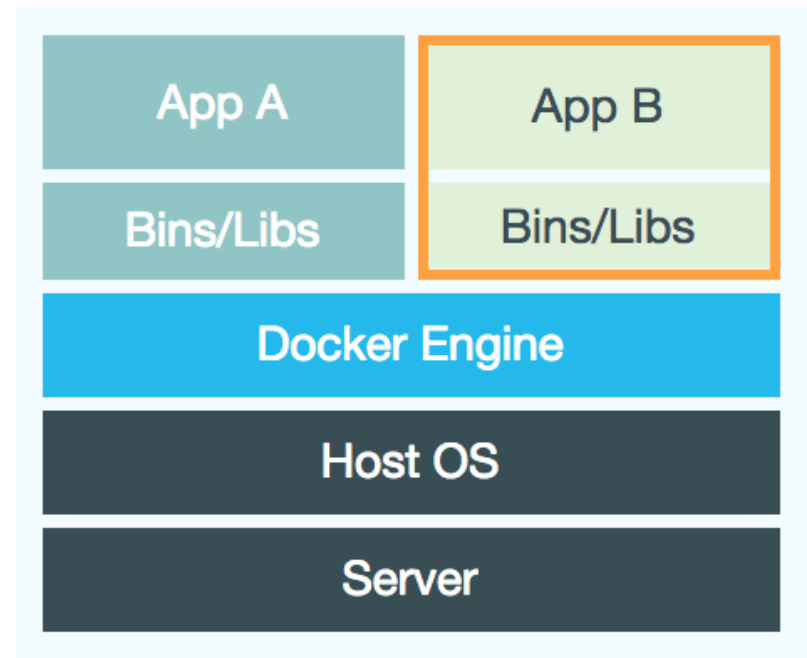
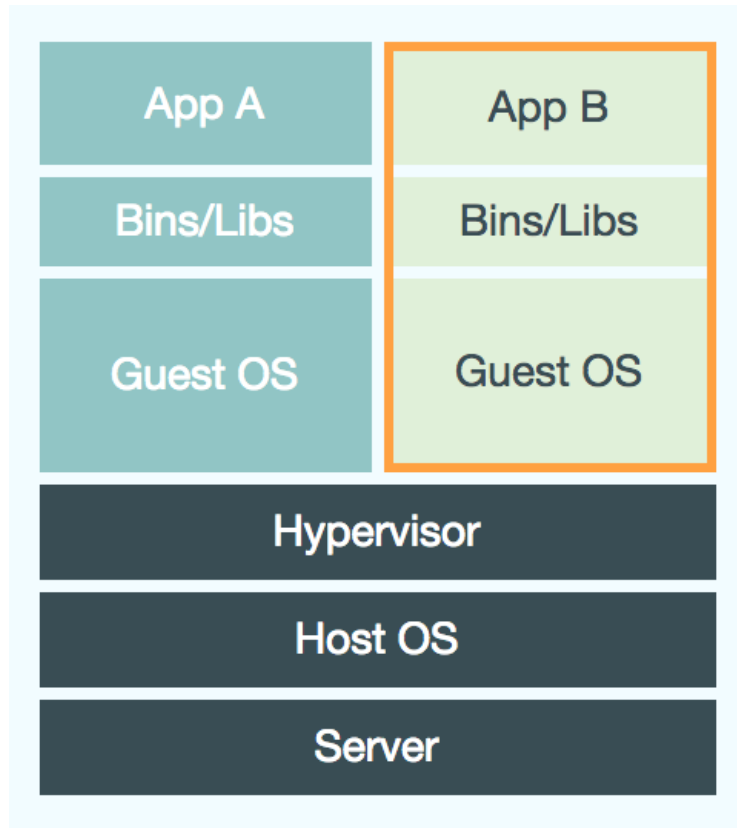


Docker is built on

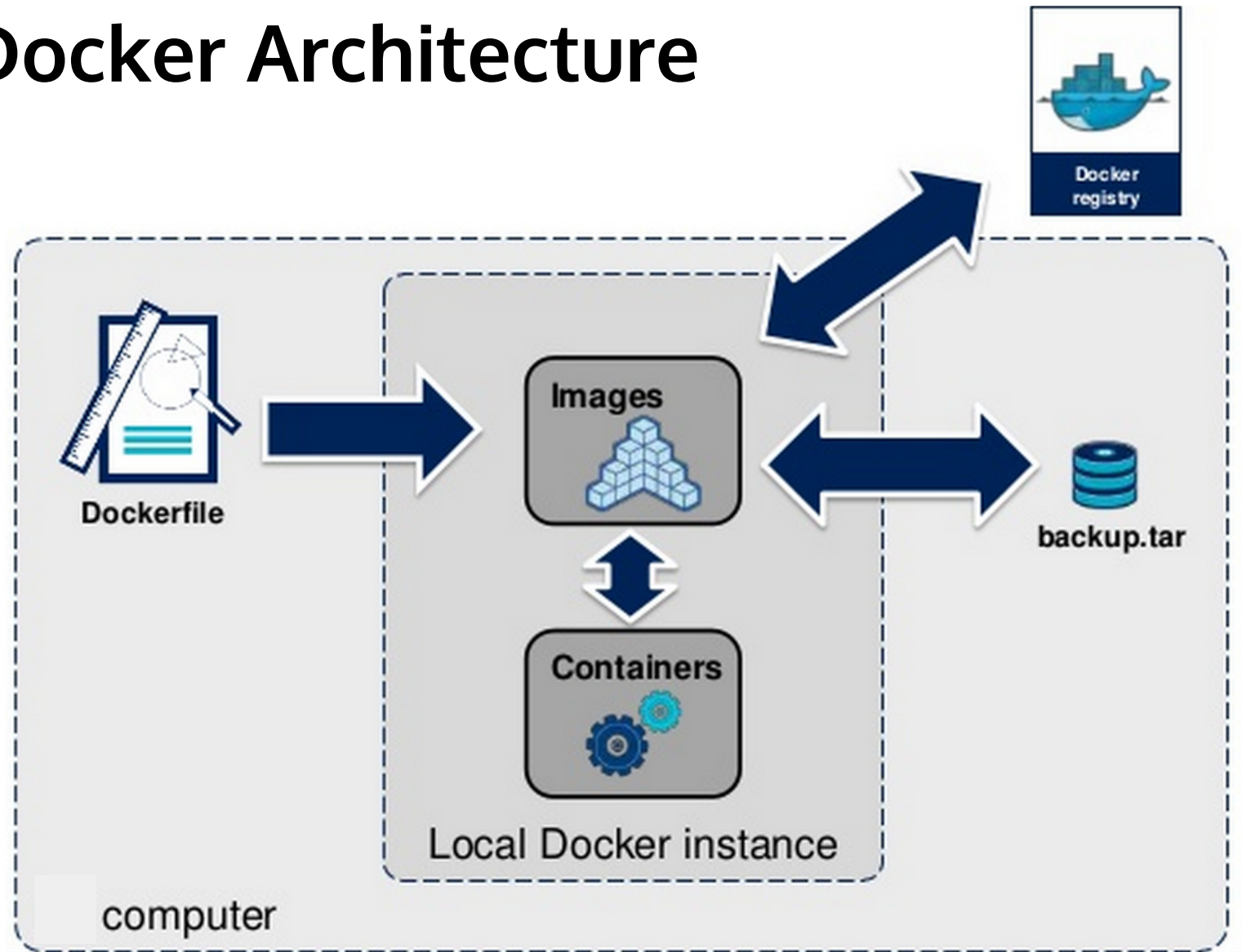
- **cgroup** and **namespacing** capabilities of the Linux kernel
- **Go** programming language
(written in Go)
- **Docker Image** Specification
(for container image management)
- **Libcontainer** Specification
(namespaces, filesystem, resources, security, etc)



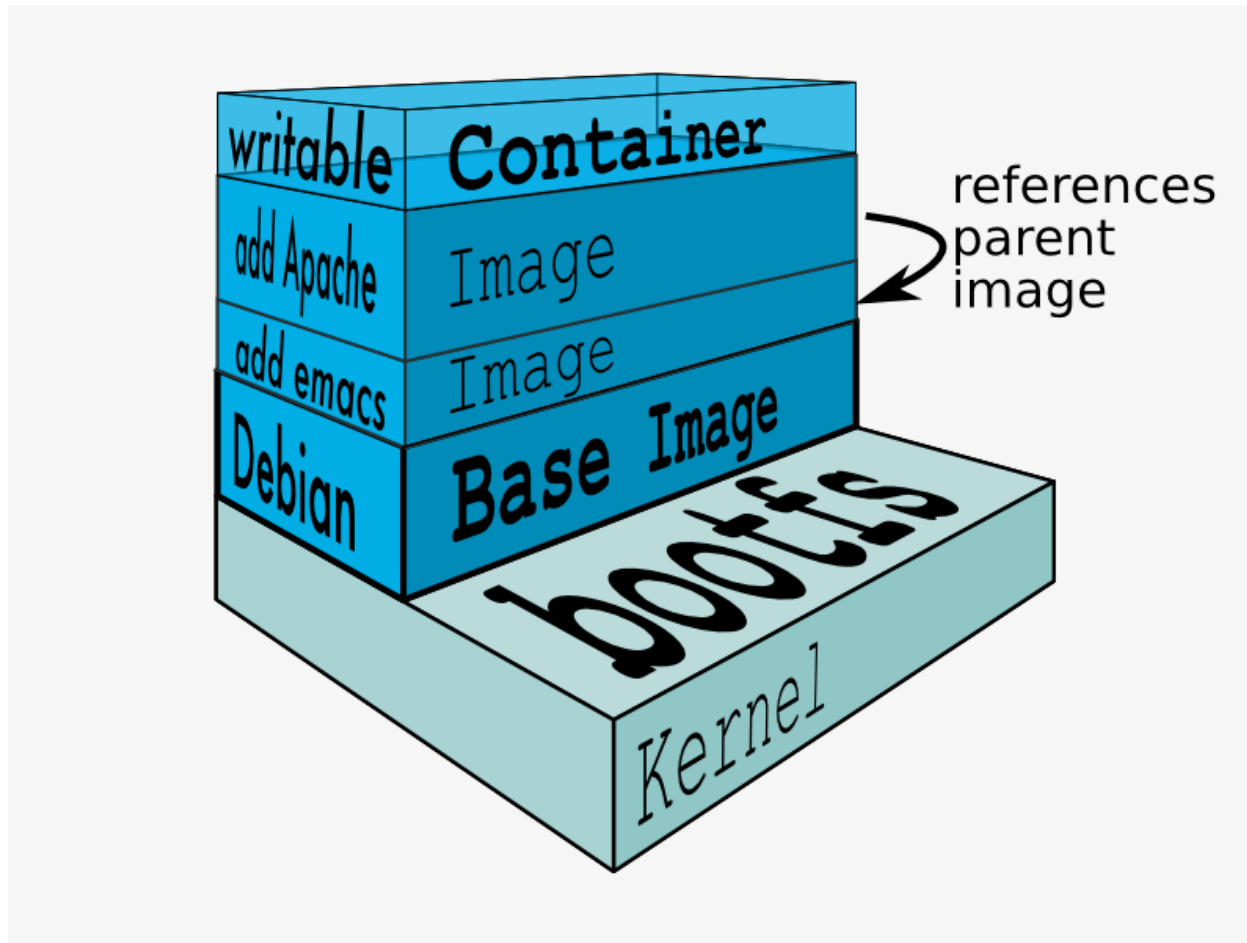
Virtual Machines Vs Docker



Docker Architecture



Docker Image Structure



Docker - Hello World

Get one base Docker image

```
>docker pull ubuntu
```

List Docker images available

```
>docker images
```

```
ubuntu      12.10      6006e6343fad  5 months ago  172.2 MB
ubuntu      quantal    6006e6343fad  5 months ago  172.2 MB
ubuntu      13.10      d2099a5ba6c5  5 months ago  180.2 MB
ubuntu      saucy      d2099a5ba6c5  5 months ago  180.2 MB
ubuntu      14.04      5cf8fd909c6c  5 months ago  274.3 MB
```

Run hello world

```
>docker run ubuntu:14.04 echo "hello world"
```



Detached mode

Run hello world in detached mode (-d)

```
>docker run -d ubuntu sh -c "while true; do echo  
hello world; sleep 1; done"
```

Get container's ID

```
>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
fb392aee70fc	ubuntu:14.04	sh -c 'while true; d	6 seconds ago	Up 2 seconds

Attach to the container

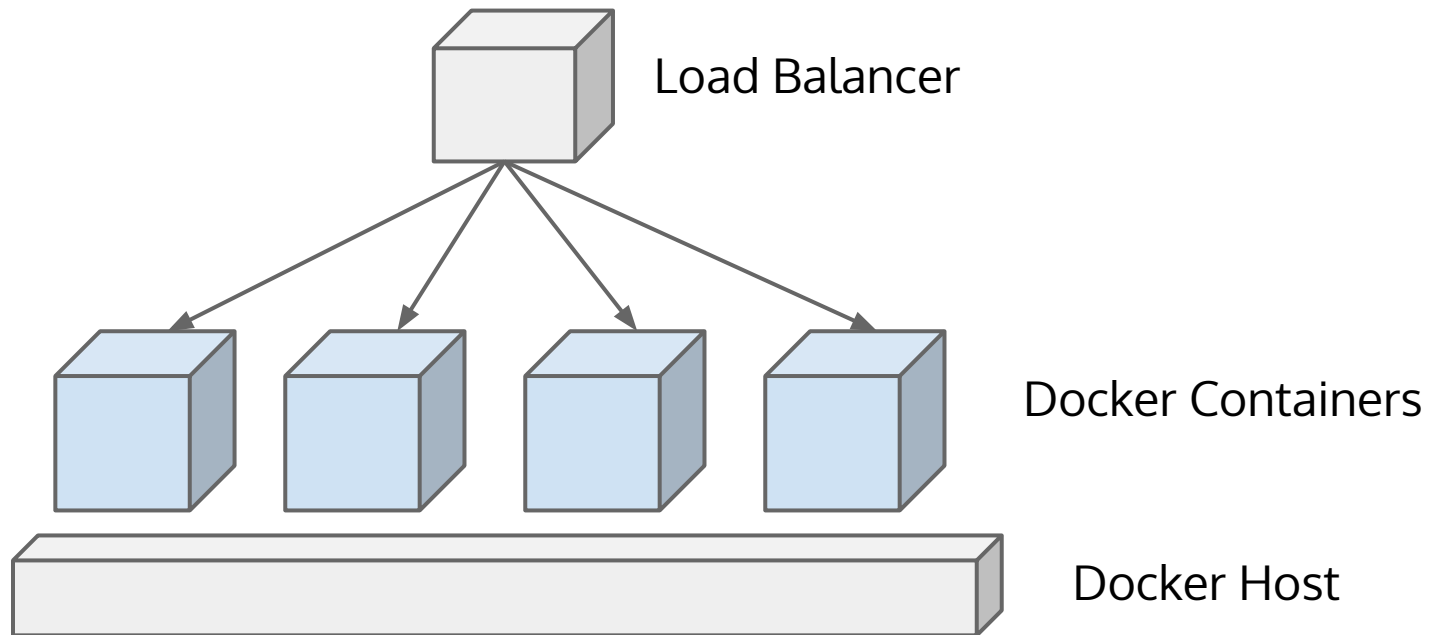
```
>docker attach <container-id>
```

Stop/start/restart the container

```
>docker stop <container-id>
```

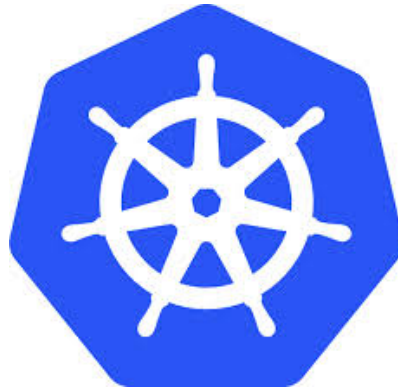


Problems with standalone Docker



- Running a server cluster on a set of Docker containers, on a single Docker host is vulnerable to single point of failure!





Kubernetes

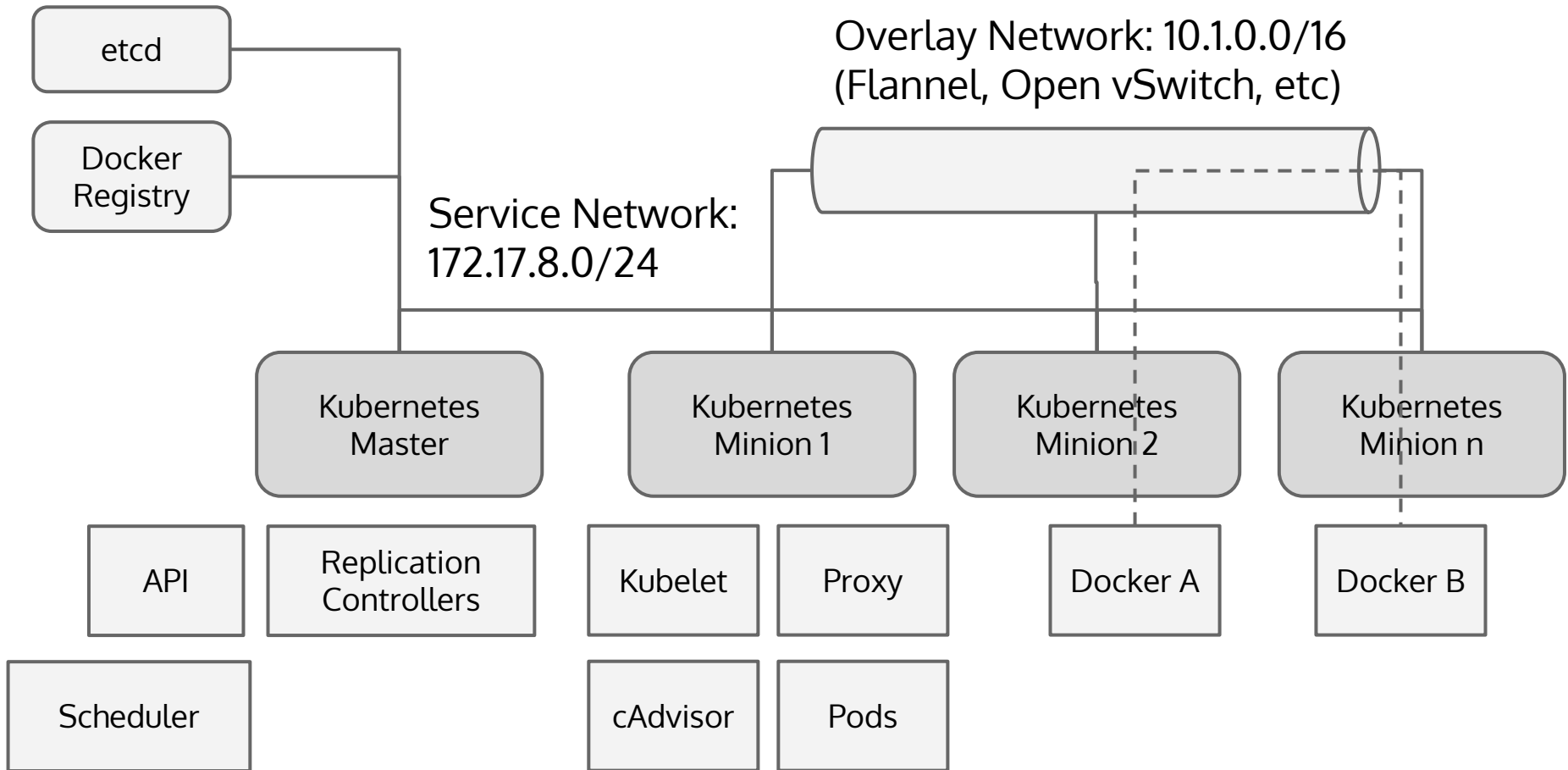
Kubernetes

- Kubernetes is a platform for hosting Docker containers in a clustered environment with multiple Docker hosts
- Provides container grouping, load balancing, auto-healing, scaling features
- Project was started by Google
- Contributors == Google, CodeOS, Redhat, Mesosphere, Microsoft, HP, IBM, VMWare, Pivotal, SaltStack, etc

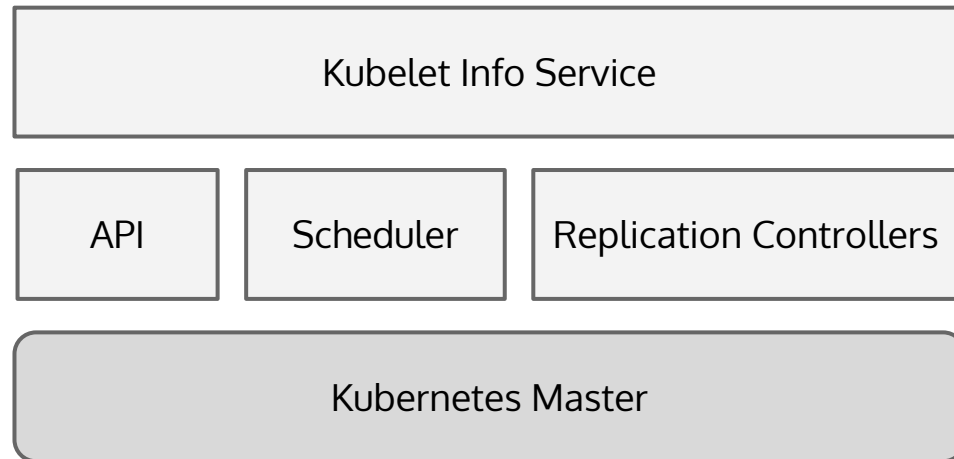
Key Concepts of Kubernetes

- **Pod** - A group of Containers
- **Labels** - Labels for identifying pods
- **Kubelet** - Container Agent
- **Proxy** - A load balancer for Pods
- **etcd** - A metadata service
- **cAdvisor** - Container Advisor provides resource usage/performance statistics
- **Replication Controller** - Manages replication of pods
- **Scheduler** - Schedules pods in worker nodes
- **API Server** - Kubernetes API server

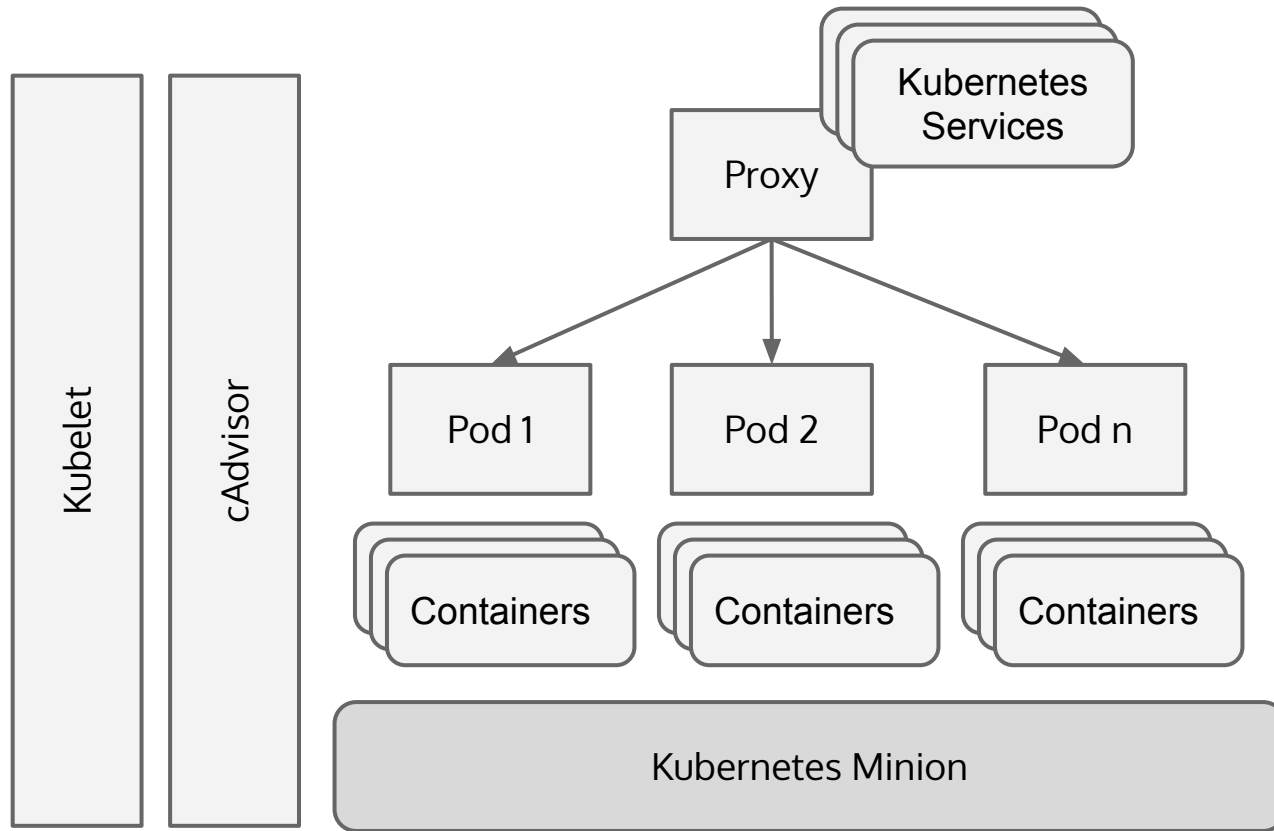
Kubernetes Architecture



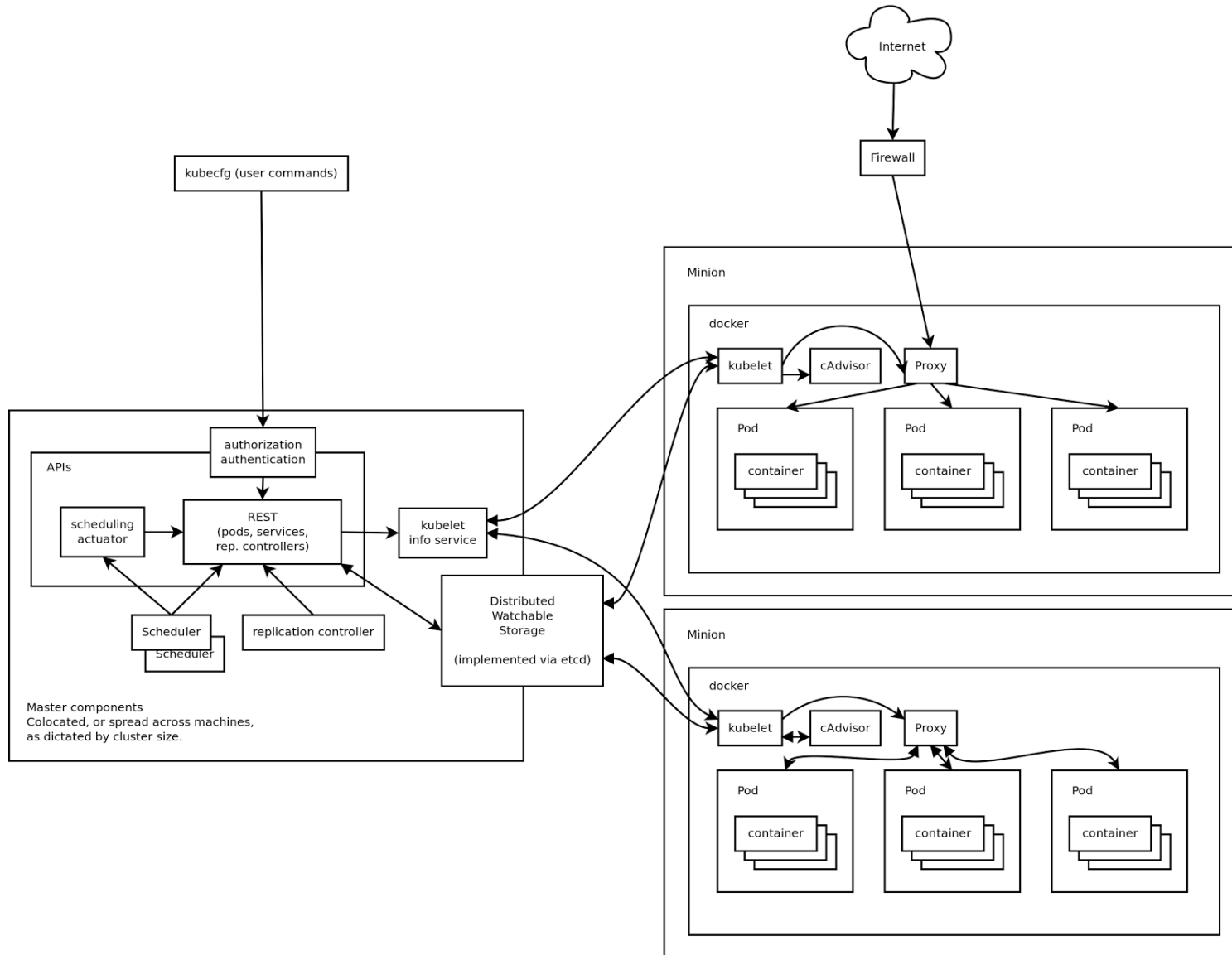
Kubernetes Master



Kubernetes Minion (Worker Node)



Kubernetes Component Architecture



Kubernetes Demo

References

- <http://en.wikipedia.org/wiki/Virtualization>
- <http://en.wikipedia.org/wiki/Hypervisor>
- <http://en.wikipedia.org/wiki/LXC>
- <http://www.cs.ucsb.edu/~rich/class/cs290-cloud/papers/lxc-namespace.pdf>
- <http://en.wikipedia.org/wiki/Cgroups>
- <http://en.wikipedia.org/wiki/AppArmor>
- http://en.wikipedia.org/wiki/Security-Enhanced_Linux
- <http://www.lorien.ch/server/chroot.html>

References

- SELinux for Everyday Users, PaulWay
- <http://en.wikipedia.org/wiki/Seccomp>
- <http://en.wikipedia.org/wiki/Chroot>
- Linux Container Brief for IEEE WG P2302, Boden Russell
- <http://kubernetes.io/>
- <https://www.youtube.com/watch?v=Fcb4aoSAZ98>
- <http://www.slideshare.net/enakai/architecture-overview-kubernetes-with-red-hat-enterprise-linux-71>