

Podman and Kubernetes: Running Linux Containers without a Docker daemon



Overview

- Containers?
- Docker
- OCI - standard
- runC
- containerd
- podman
- buildah
- CRI - common
- CNI - plugins
- boot2podman
- Kubernetes
- Pod?
- Minikube
- CRI-O - runtime

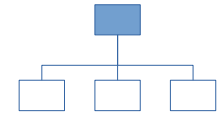


Containers?

- 1982: Chroot (Unix)
- 2000: Jails (FreeBSD)
- 2002: Zones (Solaris)
- 2005: OpenVZ (Linux)
- 2008: LXC (also Linux)

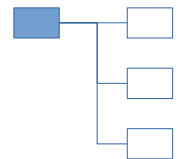
File Tree

/ (root)



Process Tree

1 (init)



User

0 (root)

Group

0 (root)

~~Virtual Private Servers~~

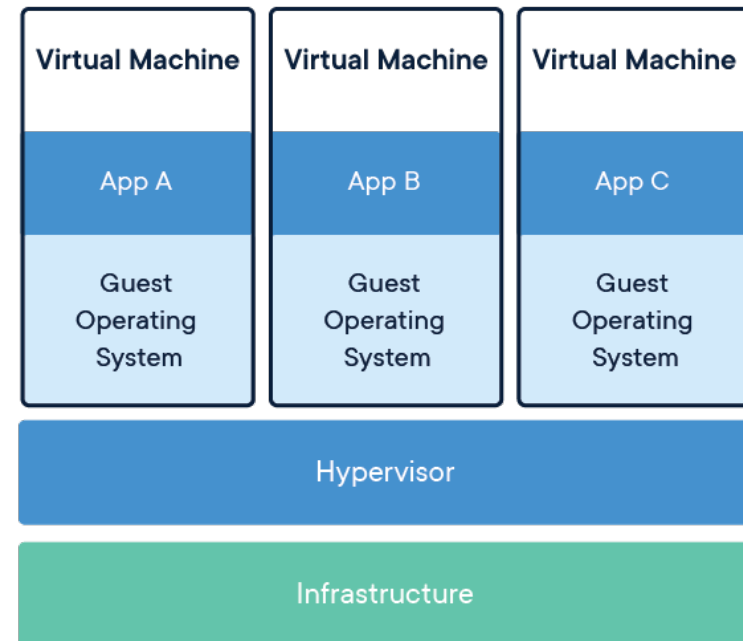
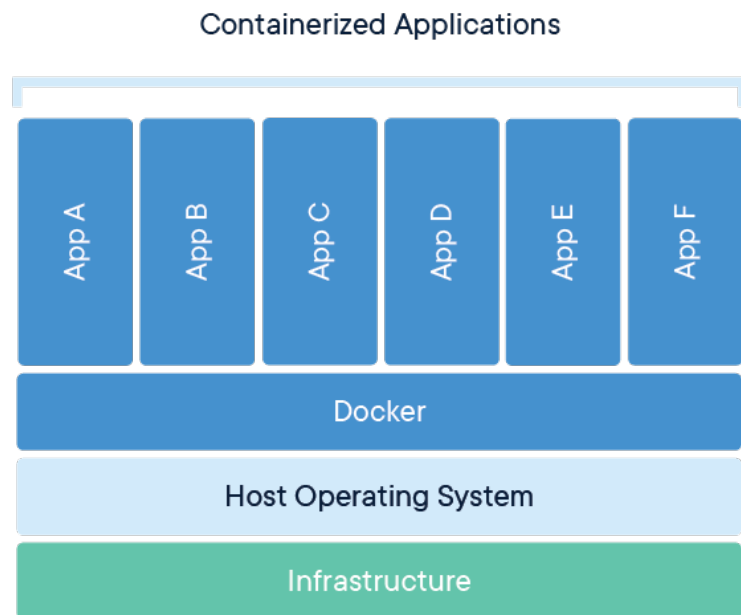
~~Virtual Environments~~

Now called: **Containers**



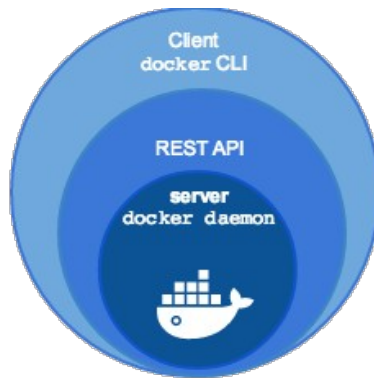
Containers vs. Virtual Machines

“A container is a standard unit of software that packages up code and all its dependencies ...”



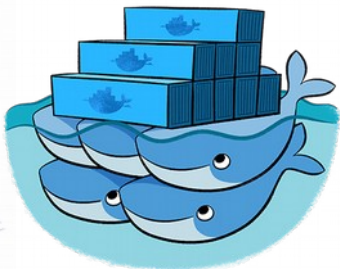
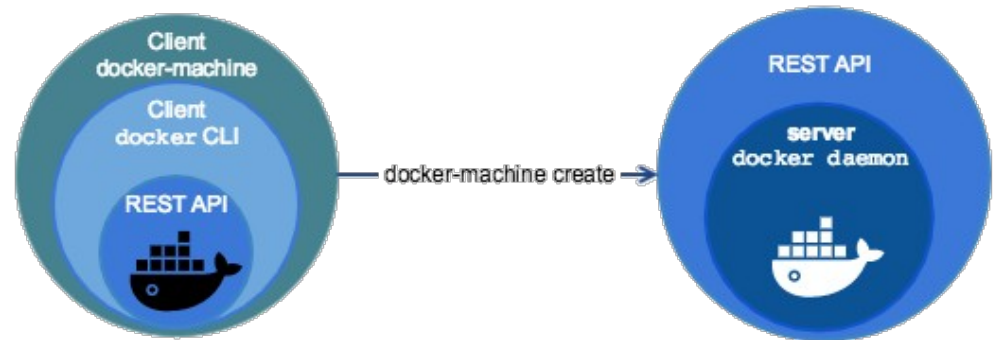
Docker

- 2013: Docker announced
- <https://www.docker.com/>
- Docker Engine
 - Client (docker)
 - Server (daemon)



Docker

- (Boot2Docker)
- Docker Machine
- Docker Swarm
- Docker Compose



OCI (Open Container Initiative)

- 2015: OCI Announced
- <https://www.opencontainers.org/>
- Runtime specification
- Image specification



OPEN CONTAINER
INITIATIVE

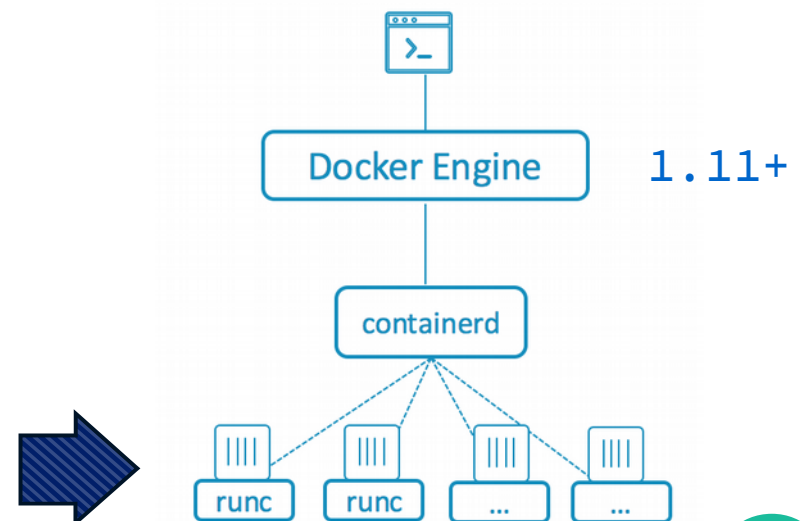


runC

”runC is a lightweight, portable container runtime. It includes all of the plumbing code used by Docker to interact with system features related to containers.”

<https://github.com/opencontainers/runc>

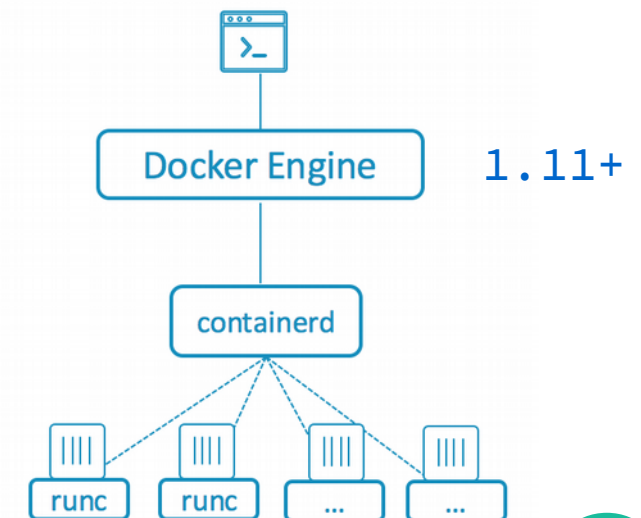
<https://blog.docker.com/2015/06/runc/>



containerd

“Containerd is a daemon providing a GRPC API to manage containers on the local system. Containerd leverages runC to [...] provide advanced functionality.”

<https://github.com/docker/containerd>



<https://blog.docker.com/2015/12/containerd-daemon-to-control-runc/>



podman

- Running containers – similar to docker
- <https://podman.io/>

```
$ sudo podman ps
```

```
$ sudo podman images
```

```
$ sudo podman run -d busybox sleep 3600
```

```
$ sudo podman stop --latest
```



podman



buildah

- Building images – similar to Dockerfile
- <https://buildah.io/>

```
# buildah images  
  
# buildah containers  
  
# newcontainer=$(buildah from scratch)  
  
# scratchmnt=$(buildah mount $newcontainer)
```

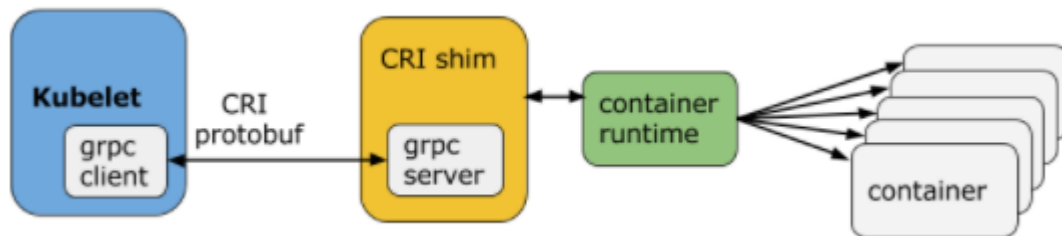


buildah



CRI (Container Runtime Interface)

- Common is a utility within CRI-O that is used to monitor the containers, handle logging from the container process, serve attach clients and detects Out Of Memory (OOM) situations.



CNI (Container Network Interface)

- CNI consists of a specification and libraries for writing plugins to configure network interfaces in Linux containers, along with a number of supported plugins.

host-local
bridge dhcp
static
sample vlan
macvlan ptp flannel
portmap ipvlan
bandwidth tuning
loopback
host-device



boot2podman

- **Boot2podman is a lightweight Linux distribution made specifically to run Linux containers.**

Virtual Machine

- Linux
- Windows
- Darwin (macOS)



Base: Tiny Core Linux

- Kernel (4.14)
- Initrd (9.4M!)
- Multiple Arch
 - Pure64 = x86_64
- Desktop (Tiny)
 - X11 with FLWM
- Packages (.tcz)
- Containers



15M CorePure64-9.0.iso
28M TinyCorePure64-9.0.iso



Add: boot2podman

- Custom kernel
 - vmlinuz64
- Add-on initrd
 - corepure64.gz
 - boot2podman.gz
- Build Tools
 - Compile TC
 - Go compiler



boot2
podman



podman-machine

- Machine lets you create servers with Podman, then configures the Podman clients.

```
$ podman-machine create box
$ podman-machine ssh box

tc@box:~$ sudo podman

$ eval $(podman-machine env box)
$ pypodman version
```



Demo - Podman



Kubernetes

Kubernetes is an open-source container orchestration system for automating deployment, scaling, and management of containerized applications.

<https://kubernetes.io>

```
$ kubectl get pods
```



Google open-sourced the Kubernetes project in 2014.

“Kubernetes builds upon a decade and a half of experience that Google has with running production workloads at scale, combined with best-of-breed ideas and practices from the community. “

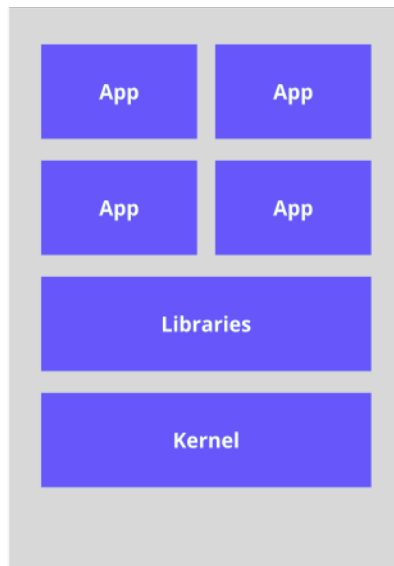


Kubernetes

- Why containers ? (Containers-as-a-Service, CaaS)

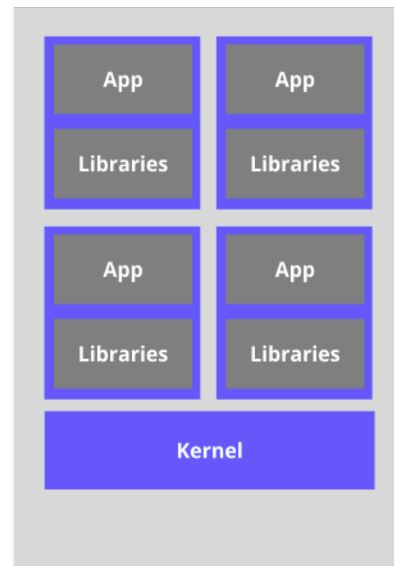
“Kubernetes provides a container-centric management environment. It orchestrates computing, networking, and storage infrastructure on behalf of user workloads. This provides much of the simplicity of Platform as a Service (PaaS) with the flexibility of Infrastructure as a Service (IaaS), and enables portability across infrastructure providers. “

The old way: Applications on host

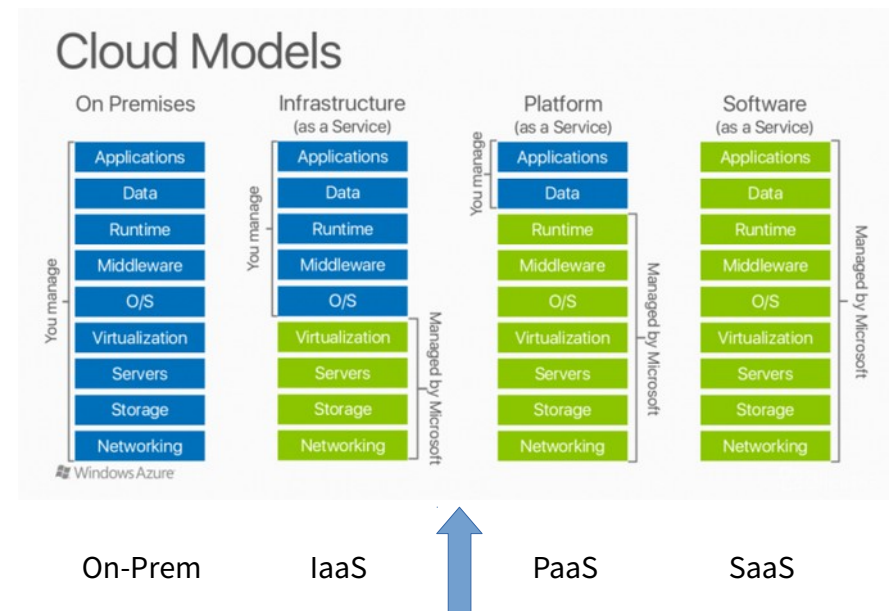


Heavyweight, non-portable
Relies on OS package manager

The new way: Deploy containers



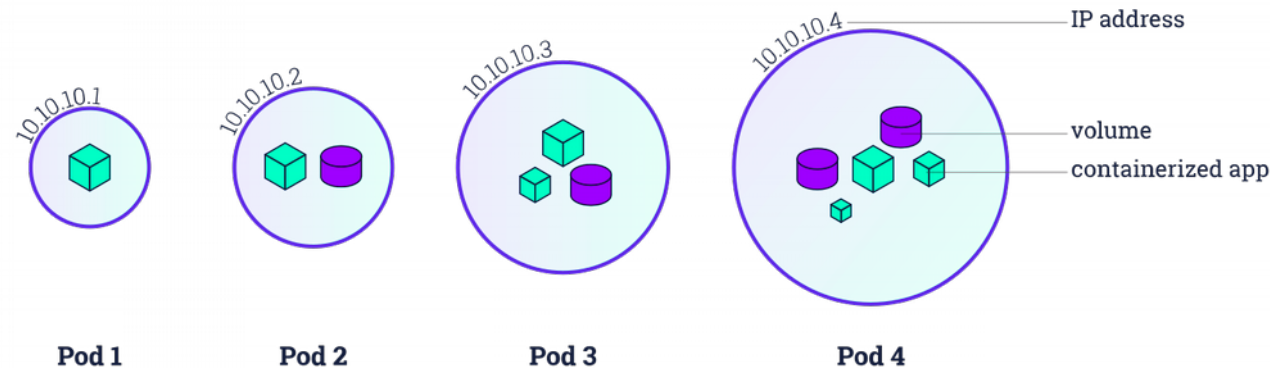
Small and fast, portable
Uses OS-level virtualization



<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/#why-containers>

Pod?

A pod (as in a pod of seals or peas) is a group of one or more containers, with shared storage/network, and a specification for how to run the containers.



Minikube

“Minikube runs a single-node Kubernetes cluster inside a VM on your laptop for users looking to try out Kubernetes or develop with it day-to-day. ”

<https://github.com/kubernetes/minikube>

```
$ minikube start
```



<https://kubernetes.io/blog/2016/07/minikube-easily-run-kubernetes-locally/>



CRI-O

- CRI-O is an implementation of the Kubernetes CRI to enable using OCI (Open Container Initiative) compatible runtimes. It is a lightweight alternative to using Docker as the runtime for Kubernetes.

```
$ minikube start --container-runtime=cri-o
```



<https://www.redhat.com/en/blog/introducing-cri-o-10> (2017/10)



Demo - Kubernetes



Questions?

Anders F Björklund
github.com/afbjorklund

For more info on boot2podman:
<https://boot2podman.github.io/>

