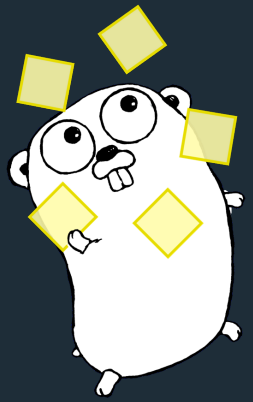
Pivotal

Containers 301

Container Overview

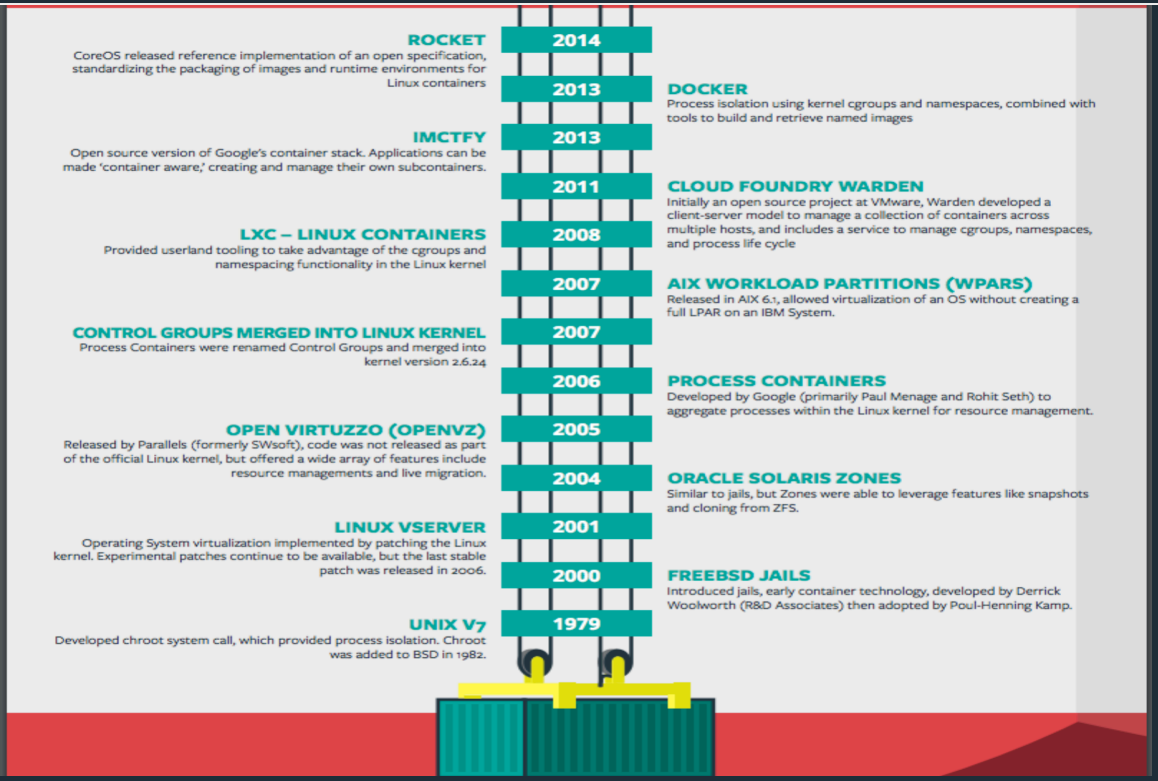
Pivotal

Container Overview



Pivotal

Brief History of Containers



- 2006: Rohit Seth and Paul Menage introduced the concept of Control groups
- 2011: Warden was developed by Pieter Noordius and others at VMware
- 2013: Docker was developed at DotCloud
- 2014: Warden rewritten in Go into Garden by Alex Suraci and others.

Pivotal

VMs vs Containers

- VM's run on a hypervisor
- Containers run on a Linux VM
- Hypervisor provides strong hardware-backed isolation
- OS kernel features provide resource isolation
- Typical VM image is 100s of GB
- Typical container image is 10s of MB
- VMs start in minutes
- Containers start in msecs

Pivotal

Advantages

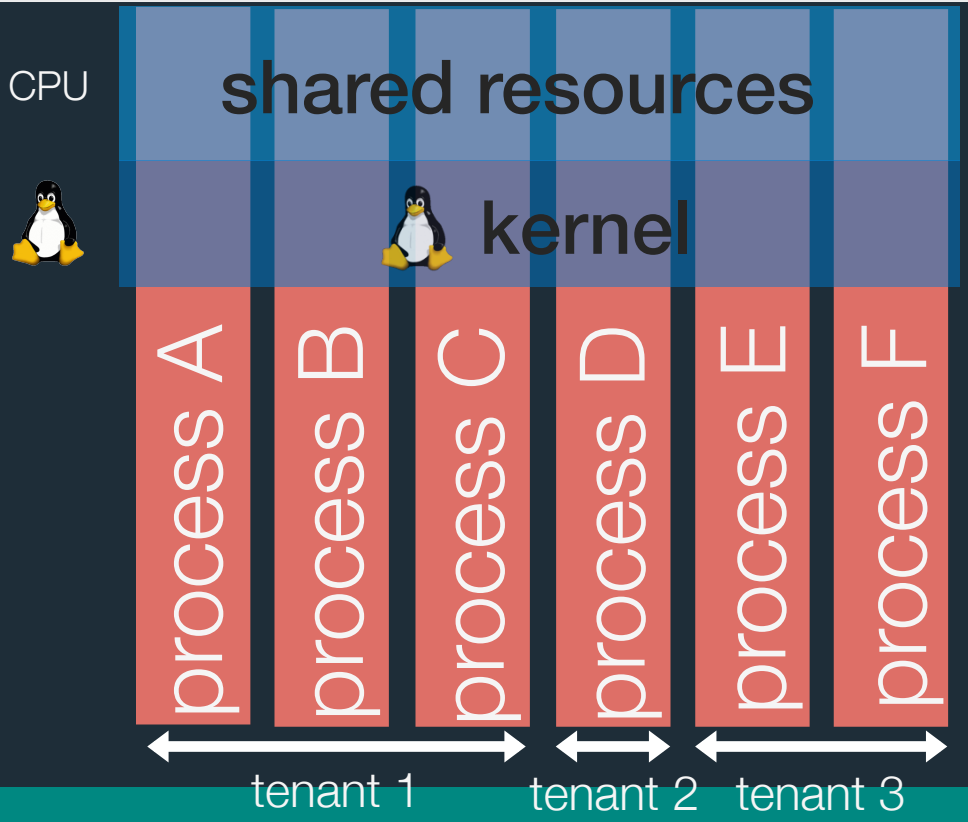
- Containers are really OS level virtualization.
- They are small and so allow for much higher packing density
- They are easy to move around and to replicate
- They do not have any redundant or unnecessary operating system elements from the VMs themselves and so they don't need the care and feeding of a large OS stack.
- They are lightweight and have fast startup times,
- All these attributes makes containers well suited for building hyperscale, highly resilient infrastructure.

Pivotal

Container Fundamentals

Pivotal

Isolation

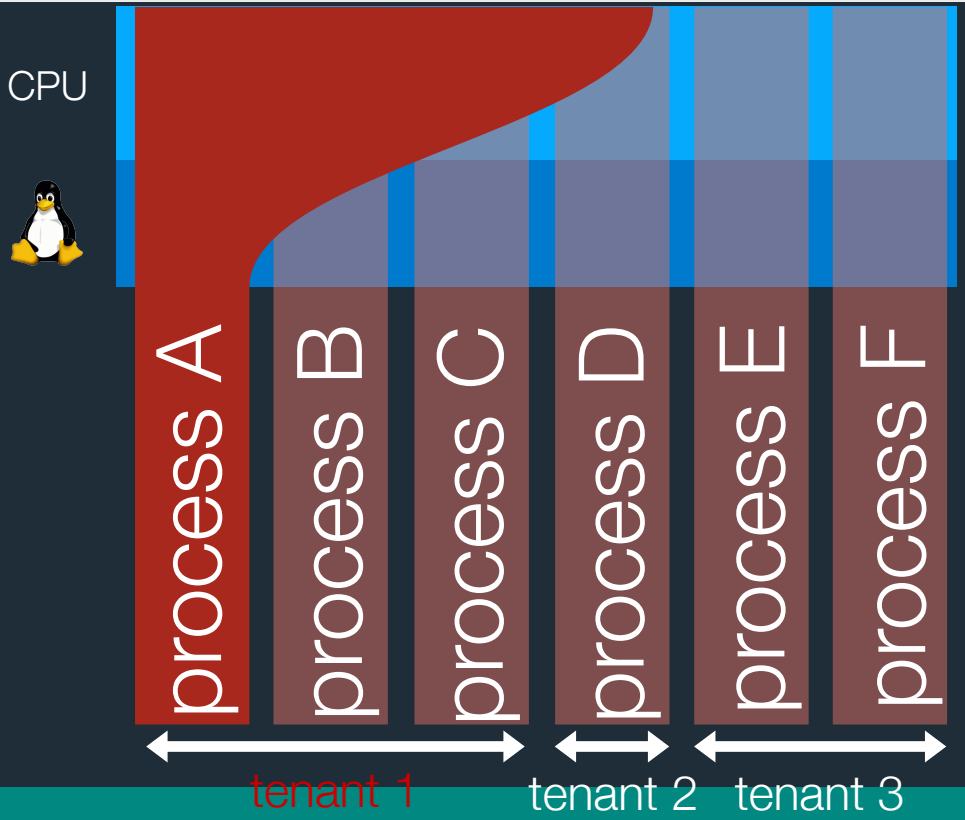


resource isolation

namespace isolation

Pivotal

Isolation

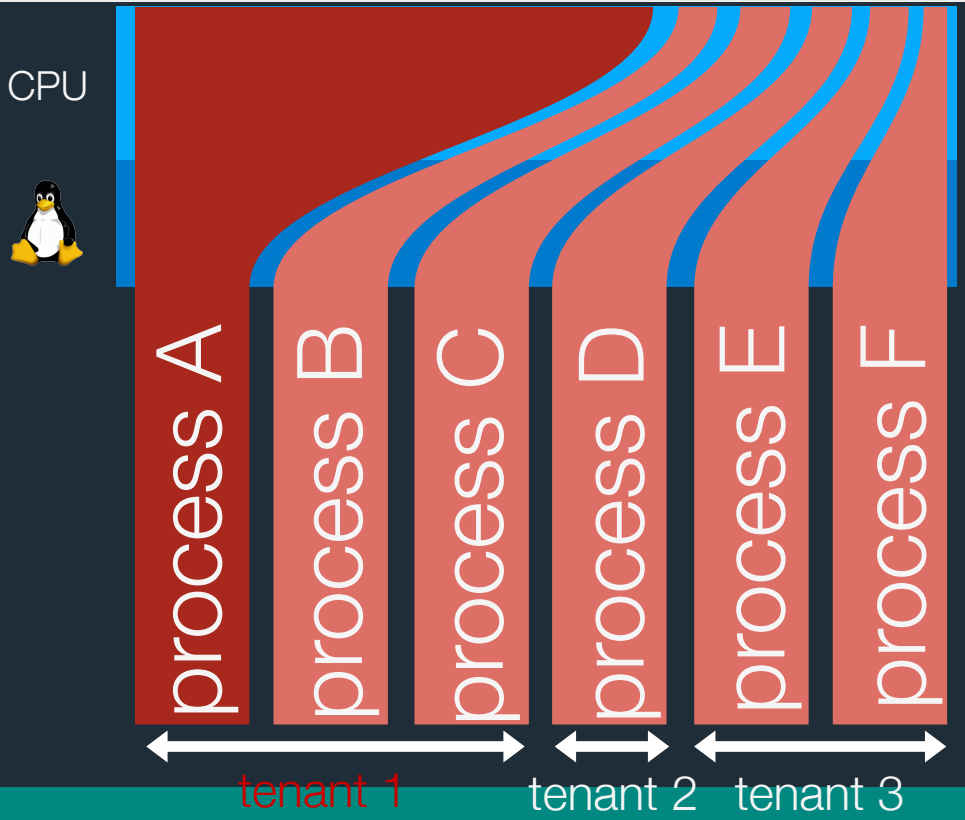


resource isolation

namespace isolation

Pivotal

Isolation

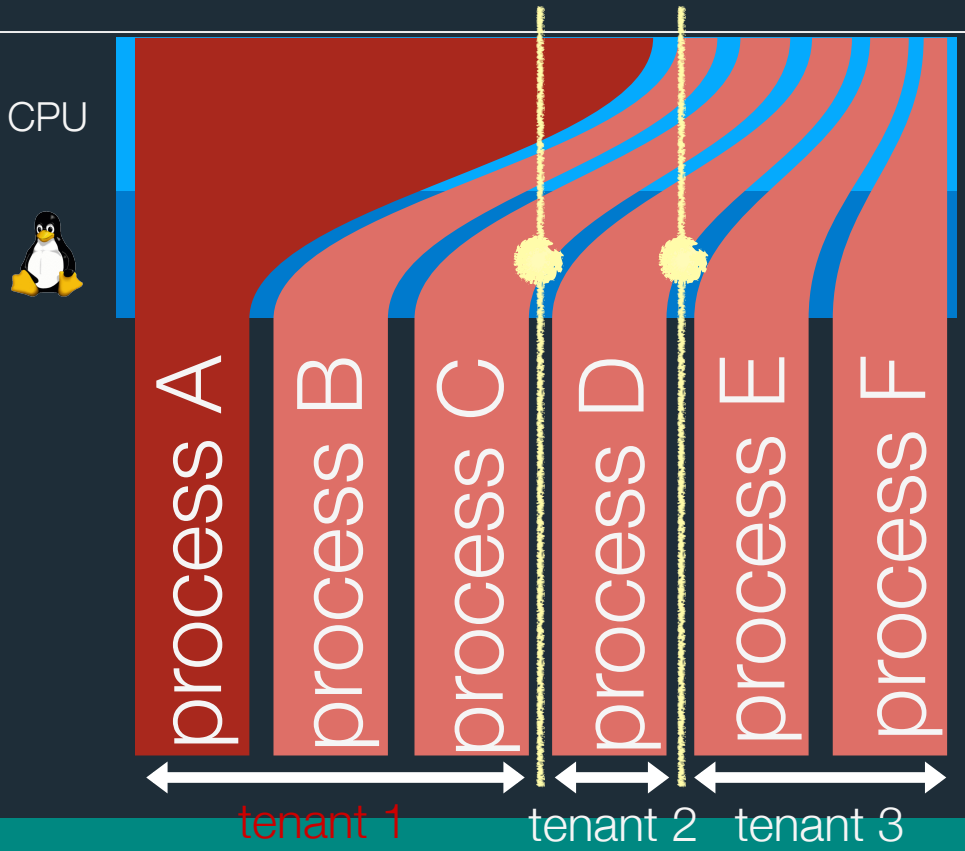


resource isolation

namespace isolation

Pivotal

Isolation



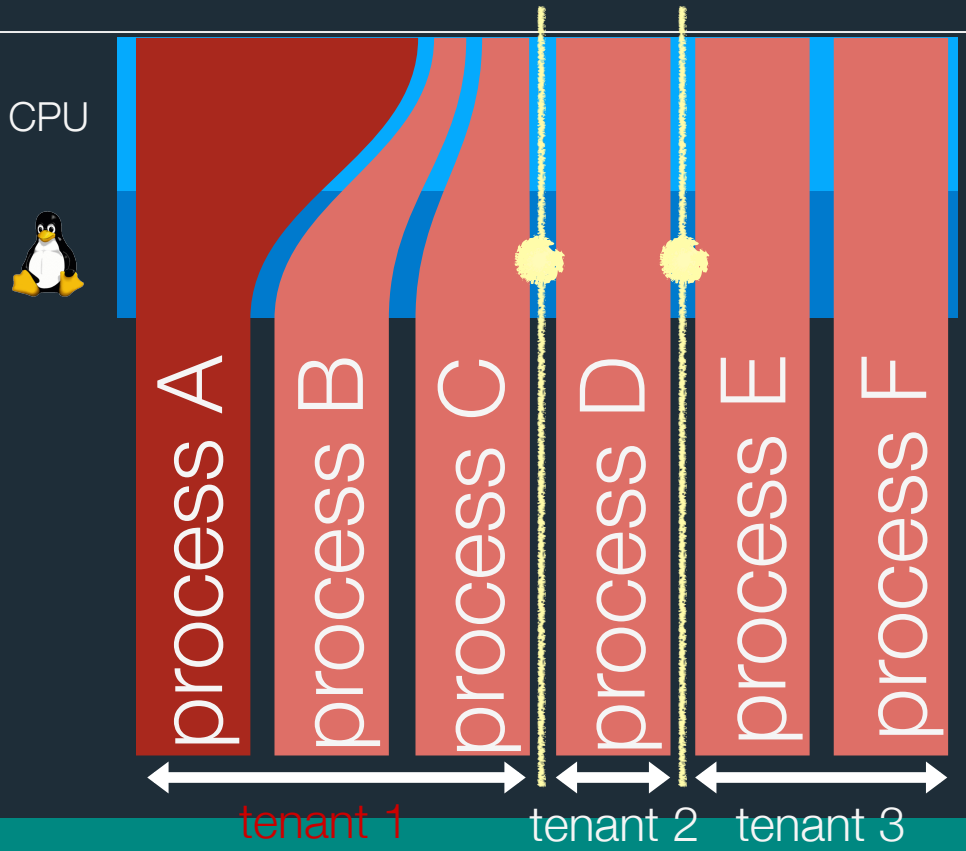
resource isolation

namespace isolation

cgroups

Pivotal

Isolation



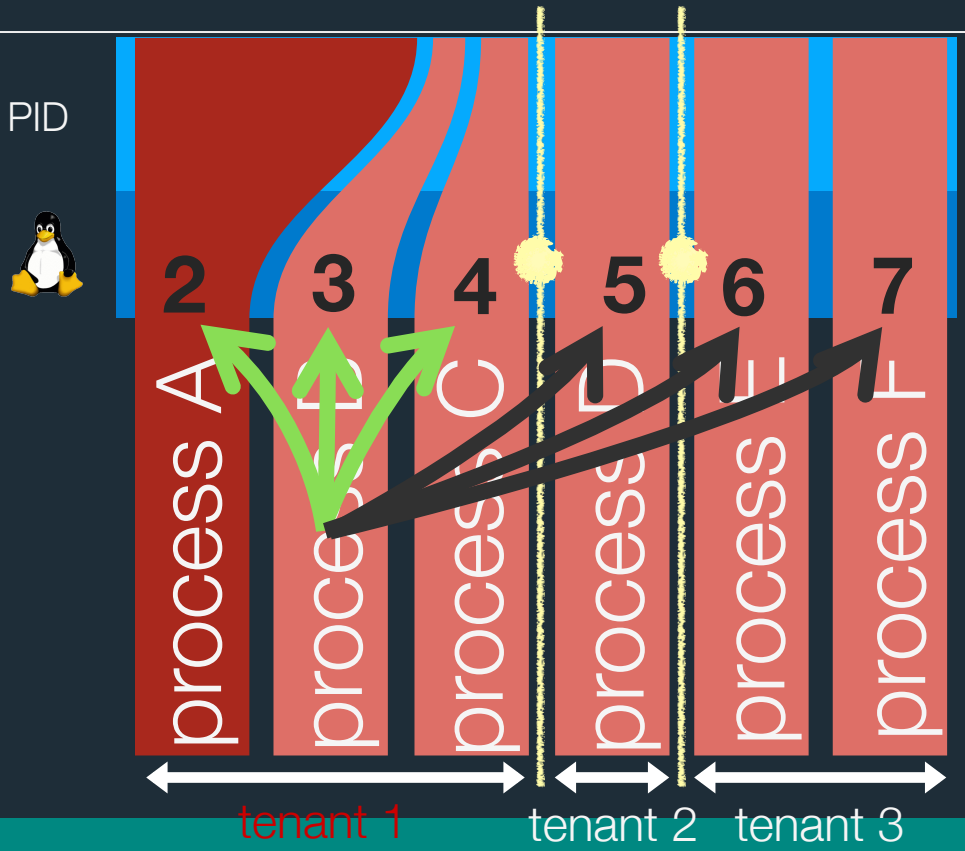
resource isolation

namespace isolation

cgroups

Pivotal

Isolation

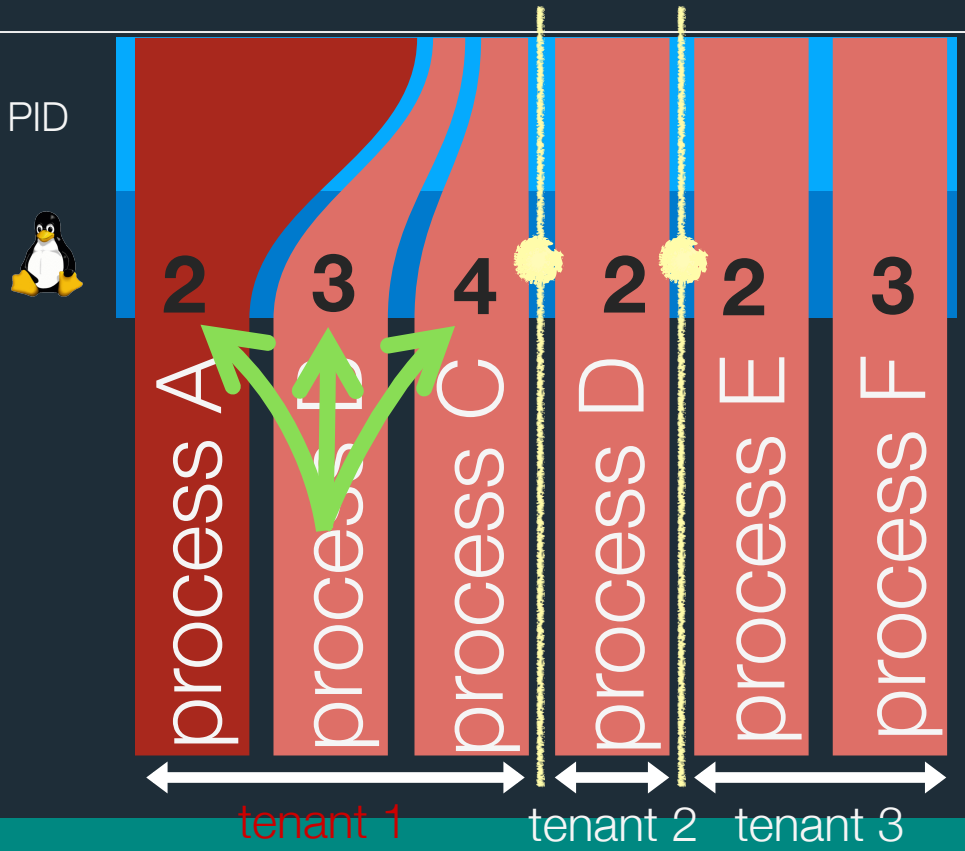


resource isolation

namespace isolation

Pivotal

Isolation



resource isolation

namespace isolation

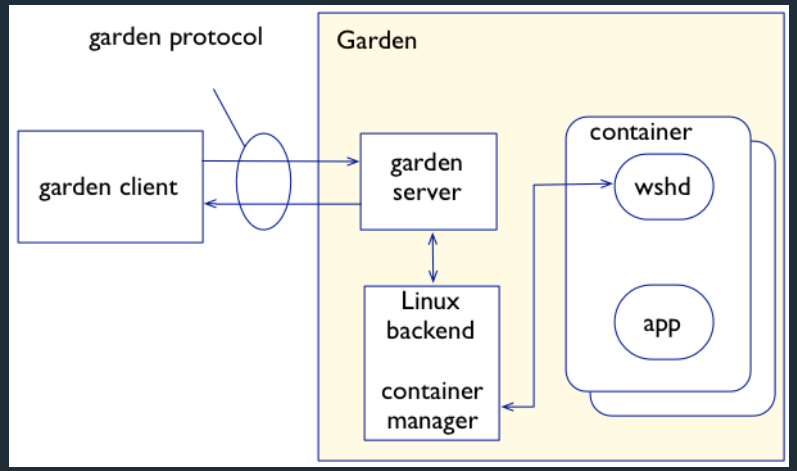
- PID Namespace
- Network
- Mount
- User

Pivotal

Garden

Pivotal

Garden



- Platform agnostic front-end
- Platform specific back-end
- Garden protocol is based on JSON over HTTP
- Rest API for testing
- Service manages lifecycle, provide telemetry

Pivotal

Garden

allows Diego to programmatically say

“make me a container”

“put this in it”

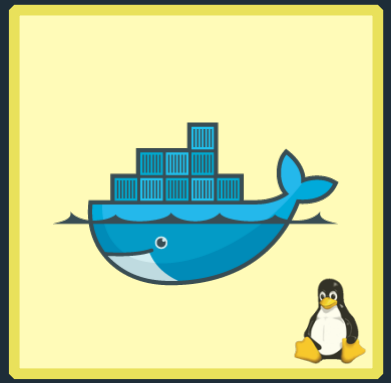
“then run this”

via a platform-agnostic API

Pivotal

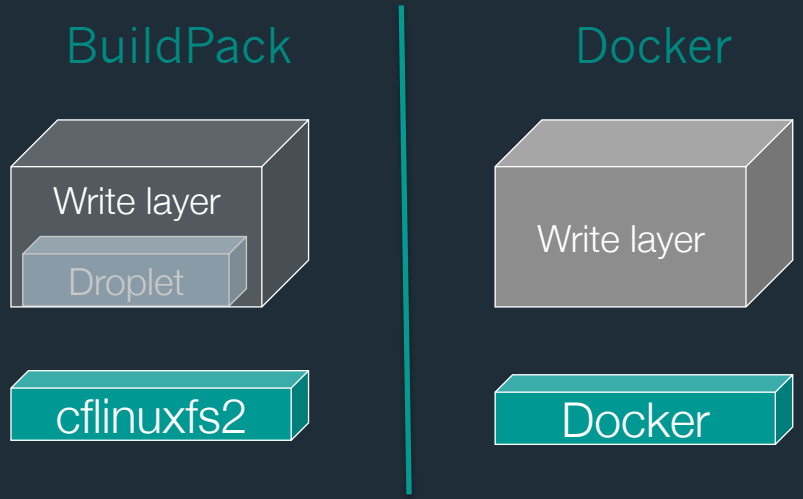
Garden

allows Diego's abstractions to be flexible



Pivotal

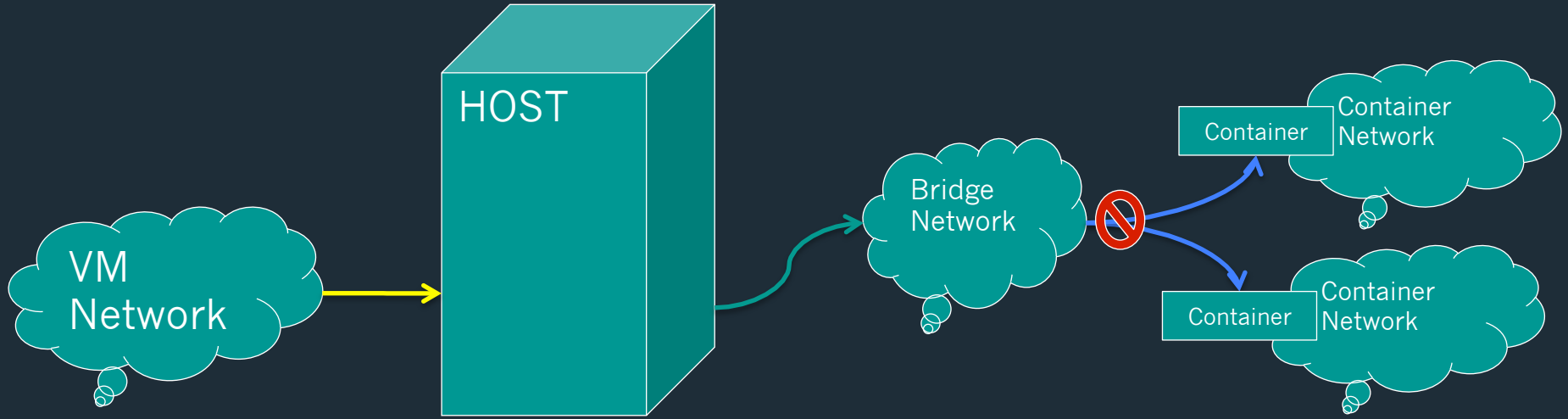
Garden Root Filesystem



- Layered file system in the container namespace
- RootFS changed by Pivot root
- RootFS can be either cflinuxfs2 or from a Docker image
- For buildpack based apps, Droplet is added to the write layer on top of the rootfs
- Write layer is ephemeral

Pivotal

Garden Networking



- HTTP/WS traffic sent to container from the host network interface to container interface
- Traffic is sent to the lowest port on the container
- Containers in a Diego cell belong to a subnet pool (which by default is 10.254.0.0/22)
- Cross Container traffic is blocked by default. Enabling cross container traffic is not recommended in multi-tenant env.

Pivotal

Garden vs Docker

• Strong Multi-tenant capabilities	☑	☒
• Platform agnostic	☑	☒
• Runs Windows back-end	☑	☒
• Container Inspection	☑	☒
• Always runs on trusted RootFs	☑	☒
• Smaller attack surface area	☑	☒
• Prevents shady binaries	☑	☒

Pivotal

OCI

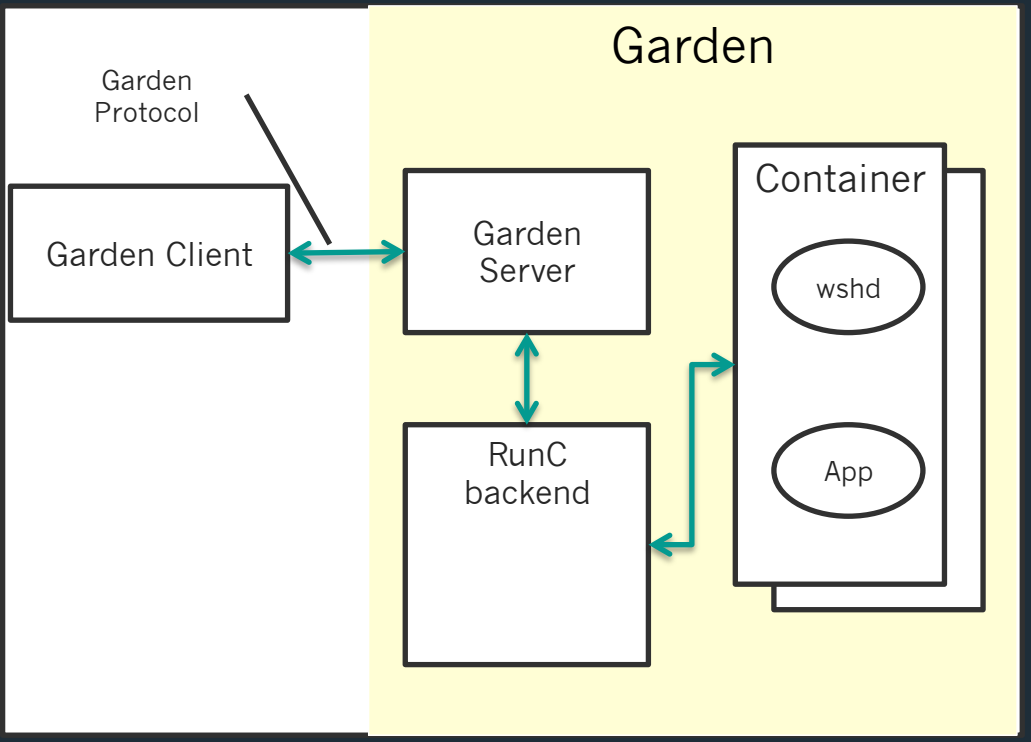
Pivotal

Open Container

- The OCI is run under the auspices of the Linux Foundation
- It defines specs for container format and runtime (OCF – Open Container Format).
- Reference implementation is called RunC
- Initial specs and reference implementation is provided by Docker
- Drivers:
  - A container not bound to higher level constructs such as a particular client or orchestration stack, and
  - A container not tightly associated with any particular commercial vendor or project, and
  - A container portable across a wide variety of operating systems, hardware, CPU architectures, public clouds, etc.

Pivotal

RunC coming to Garden



- RunC is a reference implementation of the Open Container spec
- RunC based linux backend is coming to Garden in 2016
- Docker and Garden will be running the same code - common containerization runtime.

Pivotal

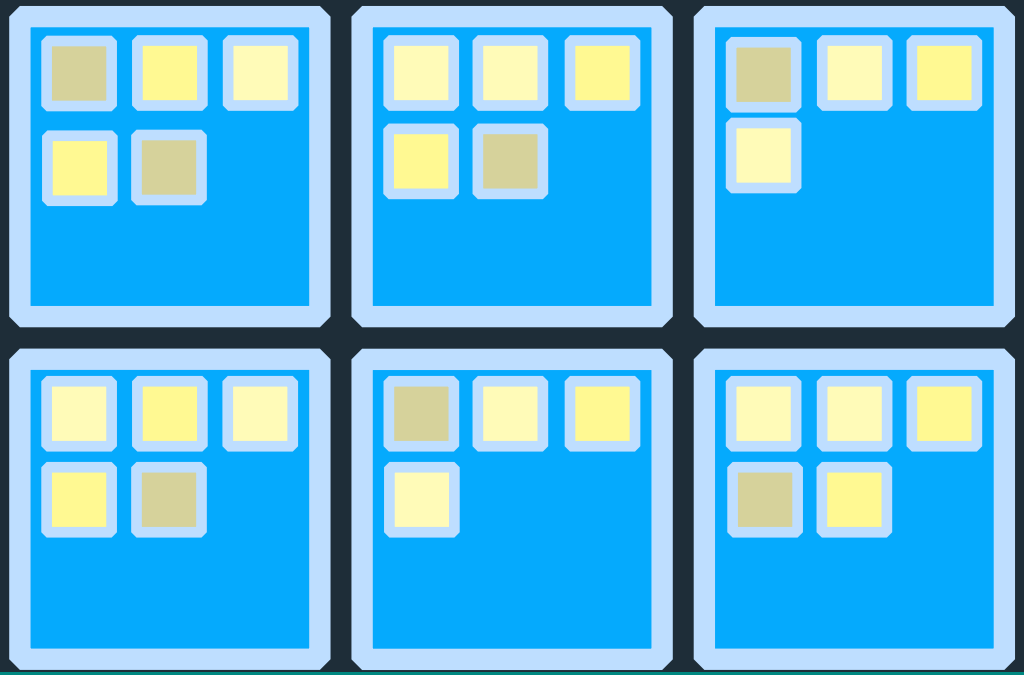
Pivotal

Open.
Agile.
Cloud-Ready.

DIEGO ?

a **distributed system** that **orchestrates containerized workloads**

Cells



BBS
(currently
etcd)

Brain

health-monitor

Pivotal