

# 如何打造百万级规模的容器技术

孙宏亮/阿里巴巴集团  
2018 中国·上海



# 目录

- 1 阿里巴巴容器现状
- 2 PouchContainer 技术特性
- 3 PouchContainer 开源现状



# 阿里巴巴容器现状

## 规模：

- 覆盖集团大部分BU
- 2017年双11百万级容器
- 在线业务100%容器化

## 覆盖场景：

- 运行模式
- 编程语言
- 技术栈

## 覆盖业务：

- 蚂蚁&交易&中间件
- B2B/CBU/ICBU/1688/村淘
- 合一集团（优酷）
- 菜鸟&高德&UC（接入中）
- 集团测试环境
- 广告（阿里妈妈）
- 阿里云专有云输出
- .....



# 阿里巴巴容器现状

- 本意育儿袋，隐喻贴身呵护应用
- 始于2011年
- 基于LXC
- 阿里内部容器技术产品，并于当年上线
- 2015年初开始吸收Docker镜像功能
- 容器结合阿里内核，大幅提高隔离性
- 大规模部署于阿里集团内部
- opensource :

<https://github.com/alibaba/pouch>

## PouchContainer



# 阿里容器演进之路

容器的要素--阿里内部运维和应用视角

- 有独立IP
- 能够ssh登陆
- 独立的的文件系统
- 资源隔离—使用量和可见性

手工Hack实现容器要素

- 虚拟网卡，网桥
- sshd
- Chroot (pivot\_root)
- CGroup，Namespace



PouchContainer

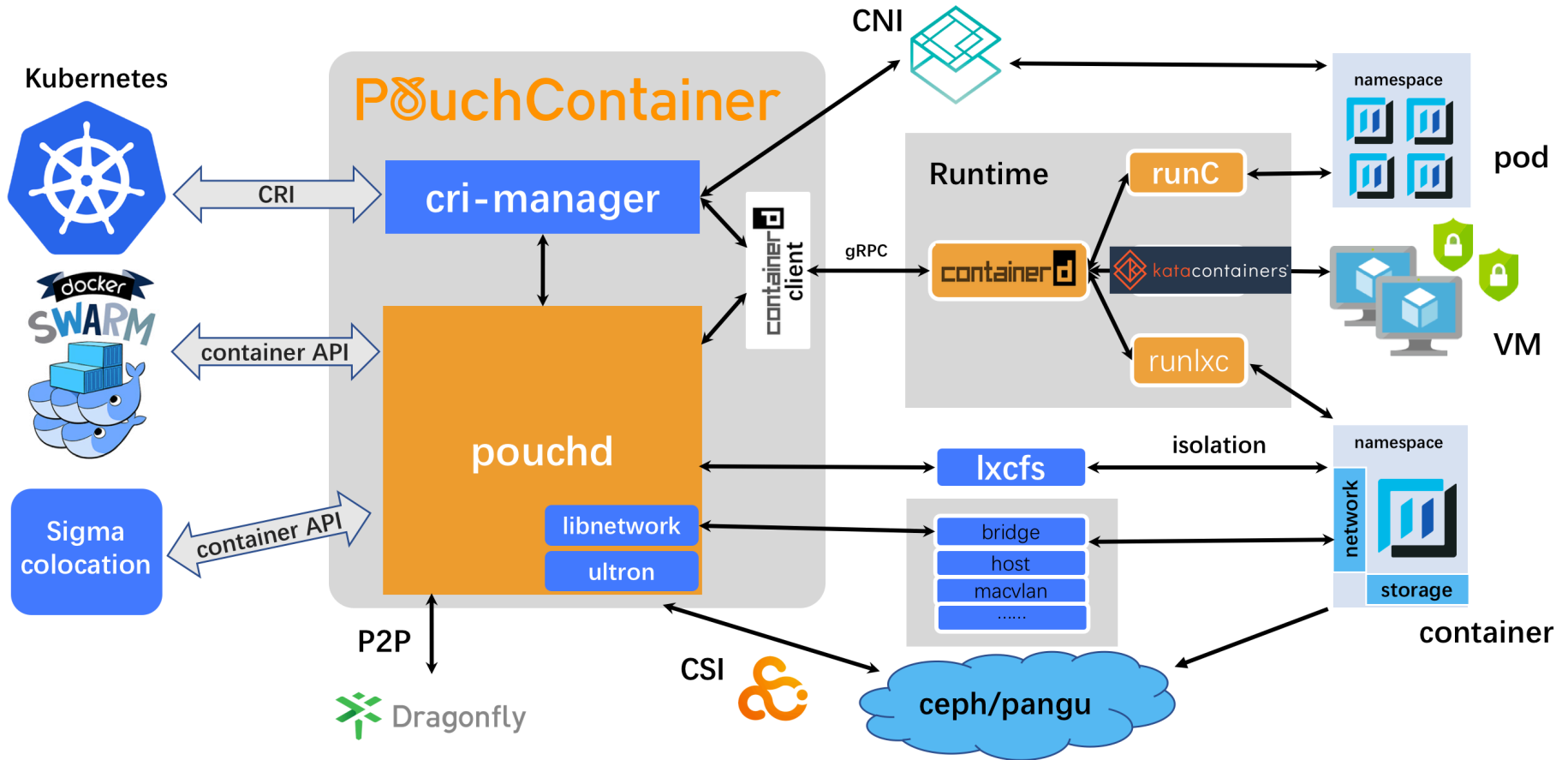
阿里容器技术

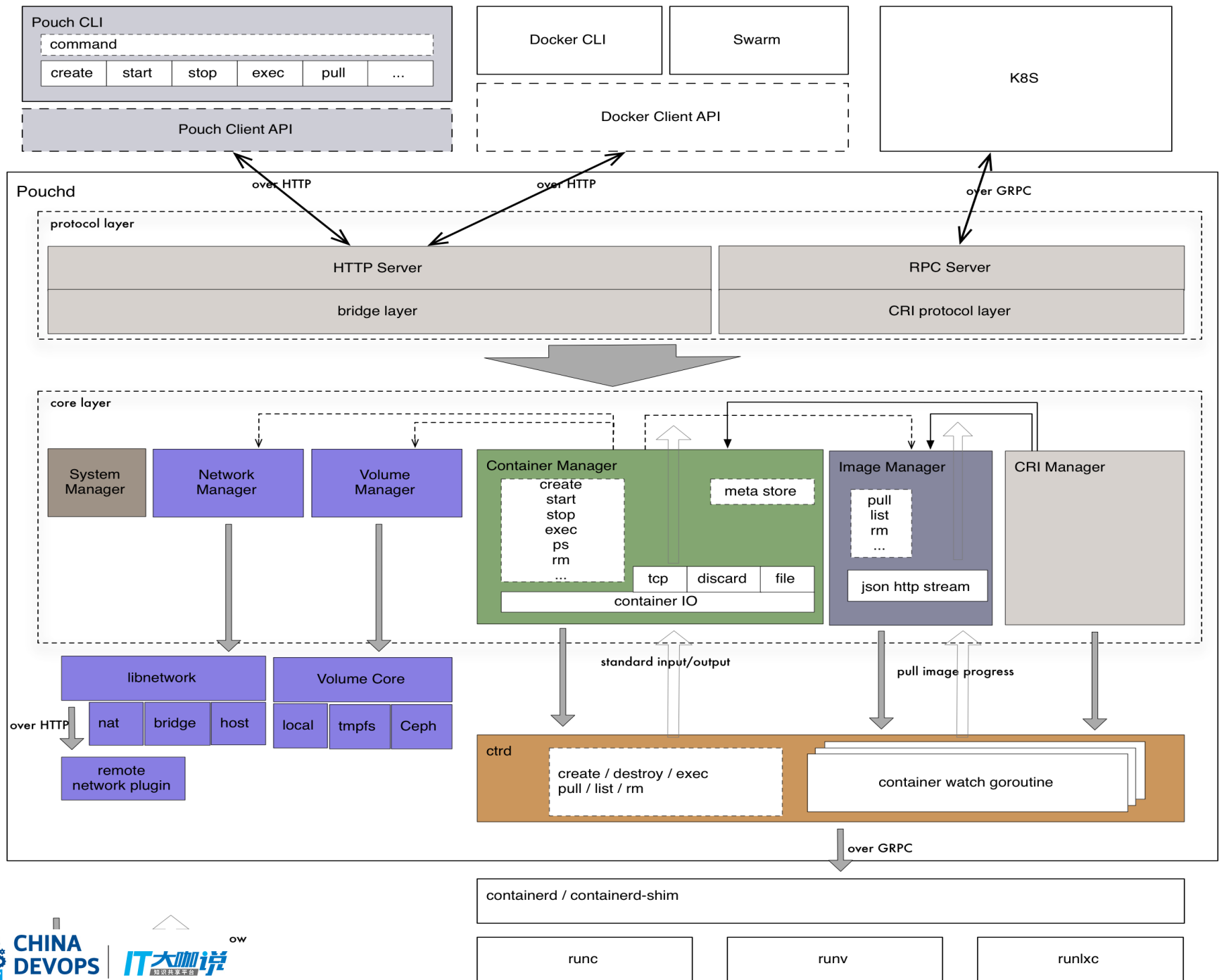
引入Docker镜像技术

- 引入LXC ( [Linux Container](#) )
- 内核可见性隔离Patch
- 内核磁盘空间配额Patch



# PouchContainer 架构





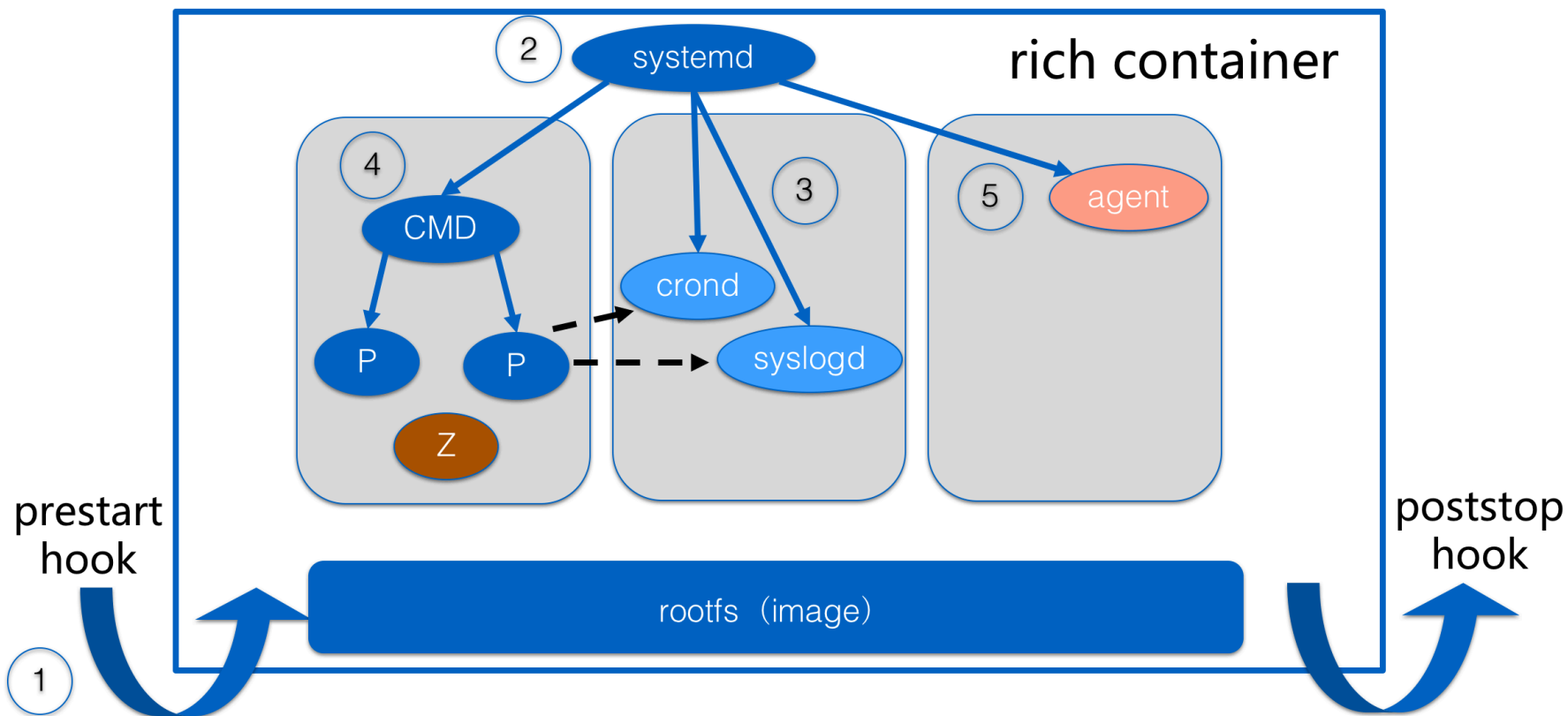
# 富容器

- 容器内运行init进程，PID = 1
- 满足运维域视角（应用运维、基础设施运维）
- 容器内运行系统服务，满足业务需求
- 极强的应用适配性，快速容器化存量业务
- 阿里集团应用100%容器化的重要前提
- 容器内资源多维度隔离（alkernel支持）





# 富容器



兼容容器镜像  
-保障交付效率

兼容运维体系  
-保障运维能力



# 强隔离性

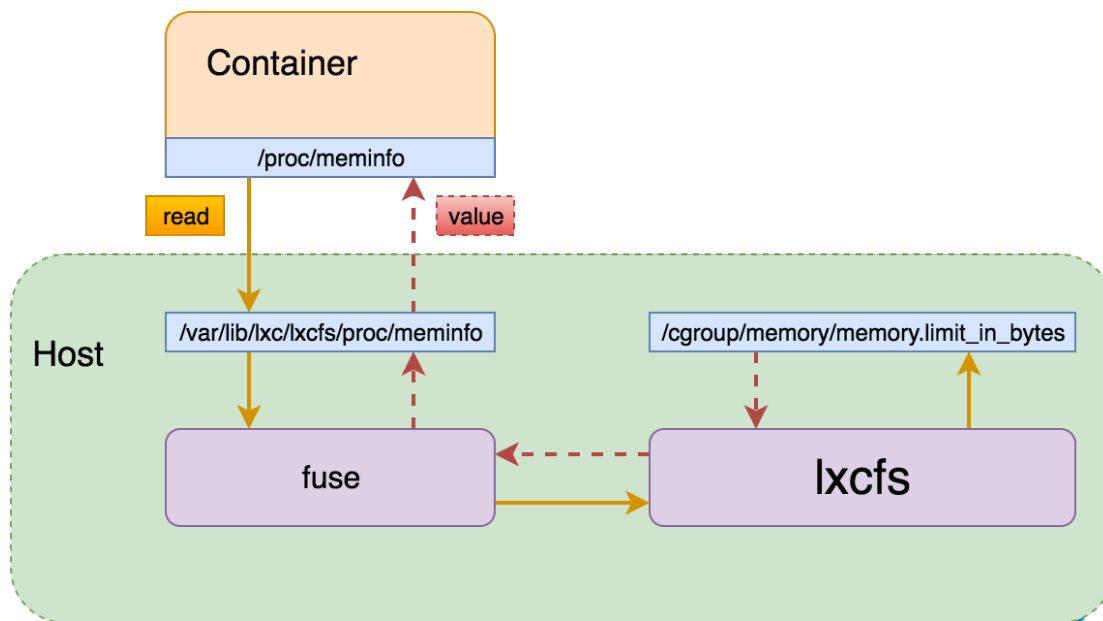
- 传统容器的隔离维度：namesapce , cgroup
- 更优的容器可见性隔离：内核patch , lxcfs
- 额外隔离维度：磁盘，网络等：diskquota
- 基于Hypervisor的强容器隔离
  - runV
  - clear container
  - katacontainer



# 资源视角隔离

使用场景：

- 容器内部进程不能感知自身的资源限额，只能感知宿主机的总资源
- Java应用判断内存资源大小动态分配堆栈大小，莫名OOM
- Java中间件通过CPU核来创建线程数
- /proc



# 资源视角隔离

## 不使用LXCFS

```
$ pouch run -m 200m registry.hub.docker.com/library/ubuntu:16.04 free -h
```

	total	used	free	shared	buff/cache	available
Mem:	2.0G	103M	1.2G	3.3M	684M	1.7G
Swap:	2.0G	0B	2.0G			

## 使用LXCFS

```
$ pouch run -m 200m --enableLxcfs registry.hub.docker.com/library/ubuntu:16.04 free -h
```

	total	used	free	shared	buff/cache	available
Mem:	200M	876K	199M	3.3M	12K	199M
Swap:	2.0G	0B	2.0G			



# 磁盘限额diskquota

- DiskQuota是一种限制文件系统磁盘空间使用的技术；
- 控制磁盘使用量的功能(Volume/容器rootfs)；
- 基于块设备的方式是可以直接控制磁盘的使用量 ( size/inode )；

DiskQuota功能在内核支持的版本情况：

	user/group quota	project quota
ext4	> 2.6	> 4.5
xfs	> 2.6	> 3.10



# 磁盘限额 diskquota

## 1. rootfs设置quota , 通过--disk-quota的参数指定

```
# pouch run -ti --disk-quota 10g registry.hub.docker.com/library/busybox:latest df -h
```

Filesystem	Size	Used	Available	Use%	Mounted on
overlay	10.0G	24.0K	10.0G	0%	/
tmpfs	64.0M	0	64.0M	0%	/dev
shm	64.0M	0	64.0M	0%	/dev/shm
tmpfs	64.0M	0	64.0M	0%	/run
tmpfs	64.0M	0	64.0M	0%	/proc/kcore
tmpfs	64.0M	0	64.0M	0%	/proc/timer_list
tmpfs	64.0M	0	64.0M	0%	/proc/sched_debug
tmpfs	1.9G	0	1.9G	0%	/sys/firmware
tmpfs	1.9G	0	1.9G	0%	/proc/scsi



# 磁盘限额 diskquota

```
# pouch volume create -n volume-quota-test -d local -o mount=/data/volume -o size=10g
```

```
Name: volume-quota-test
```

```
Scope:
```

```
Status: map[mount:/data/volume sifter:Default size:10g]
```

```
CreatedAt: 2018-3-24 13:35:08
```

```
Driver: local
```

```
Labels: map[]
```

```
Mountpoint: /data/volume/volume-quota-test
```

```
# pouch run -ti -v volume-quota-test:/mnt registry.hub.docker.com/library/busybox:latest df -h
```

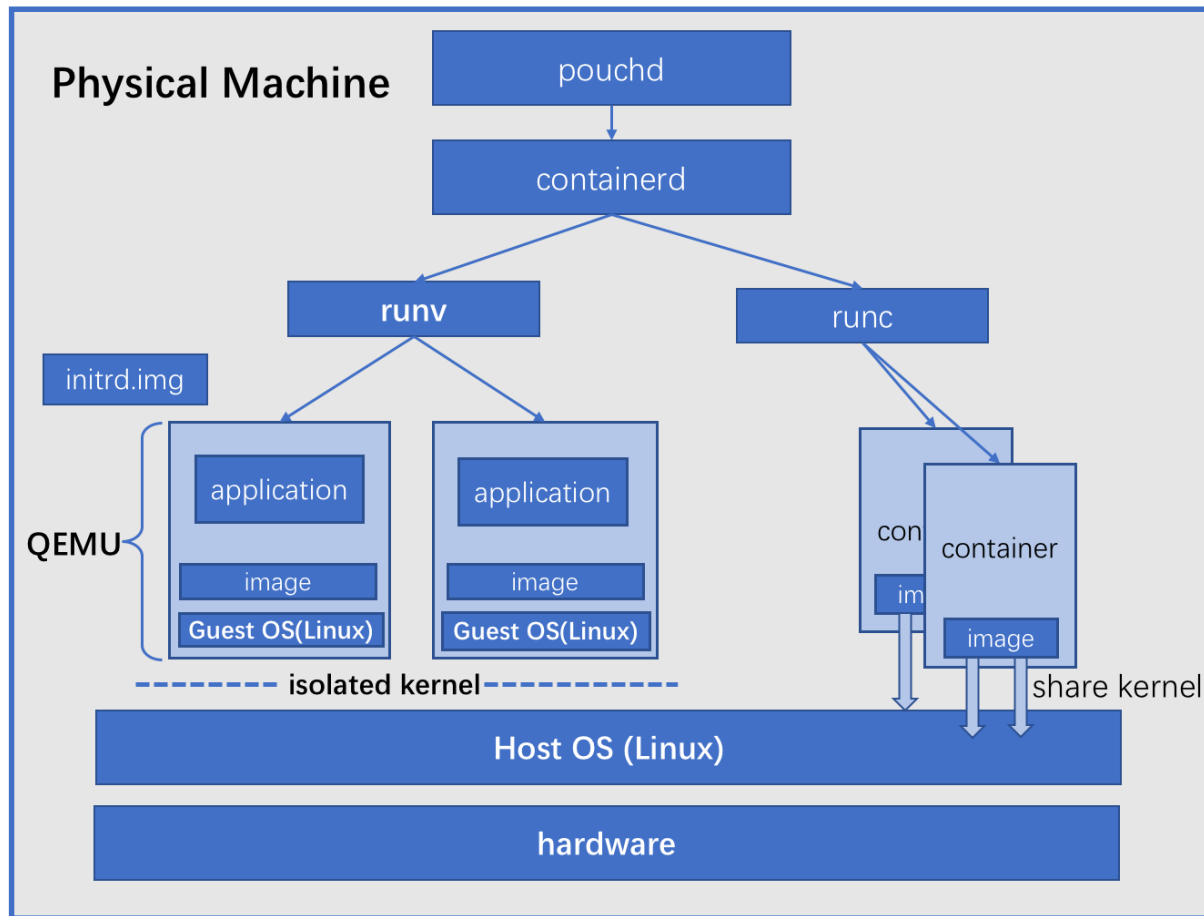
Filesystem	Size	Used	Available	Use%	Mounted on
overlay	20.9G	212.9M	19.6G	1%	/
tmpfs	64.0M	0	64.0M	0%	/dev
shm	64.0M	0	64.0M	0%	/dev/shm
tmpfs	64.0M	0	64.0M	0%	/run
/dev/sdb2	10.0G	4.0K	10.0G	0%	/mnt
tmpfs	64.0M	0	64.0M	0%	/proc/kcore
tmpfs	64.0M	0	64.0M	0%	/proc/timer_list
tmpfs	64.0M	0	64.0M	0%	/proc/sched_debug
tmpfs	1.9G	0	1.9G	0%	/sys/firmware
tmpfs	1.9G	0	1.9G	0%	/proc/scsi

# Hypervisor 容器

runV  
QEMU

兼容容器镜像  
-保障交付效率

提供隔离的内核  
-保障容器安全





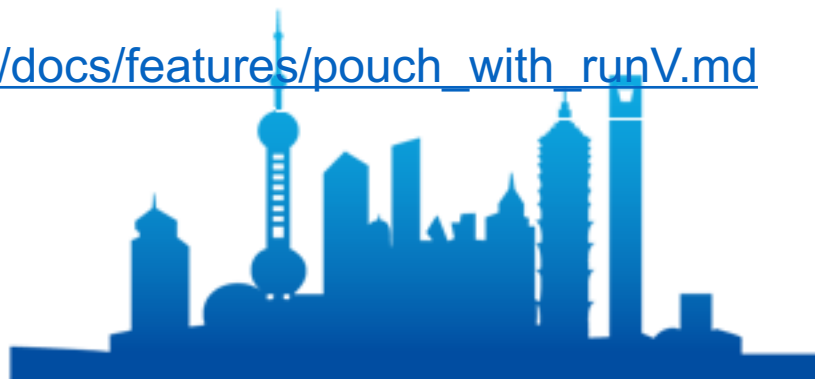
# Hypervisor 容器

## 多容器运行时统一管理

```
$ pouch create --name hypervisor --runtime runv docker.io/library/busybox:latest
container ID: 95c8d52154515e58ab267f3c33ef74ff84c901ad77ab18ee6428a1ffac12400d, name: hypervisor
$
$ pouch ps
```

Name	ID	Status	Image	Runtime
hypervisor	95c8d5	created	docker.io/library/busybox:latest	runv
4945c0	4945c0	stopped	docker.io/library/busybox:latest	runc
1dad17	1dad17	stopped	docker.io/library/busybox:latest	runv
fab7ef	fab7ef	created	docker.io/library/busybox:latest	runv
505571	505571	stopped	docker.io/library/busybox:latest	runc

[https://github.com/alibaba/pouch/blob/master/docs/features/pouch\\_with\\_runV.md](https://github.com/alibaba/pouch/blob/master/docs/features/pouch_with_runV.md)

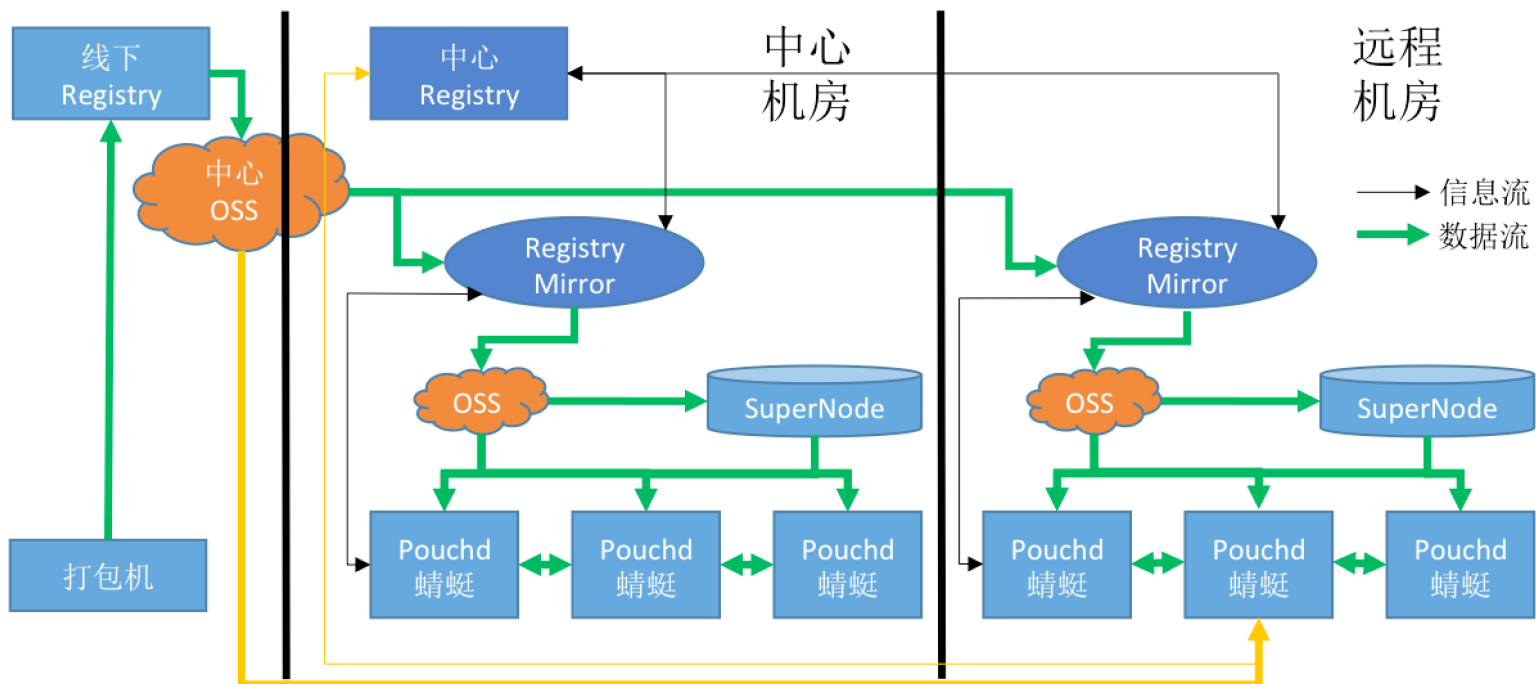


# 支持2.6内核

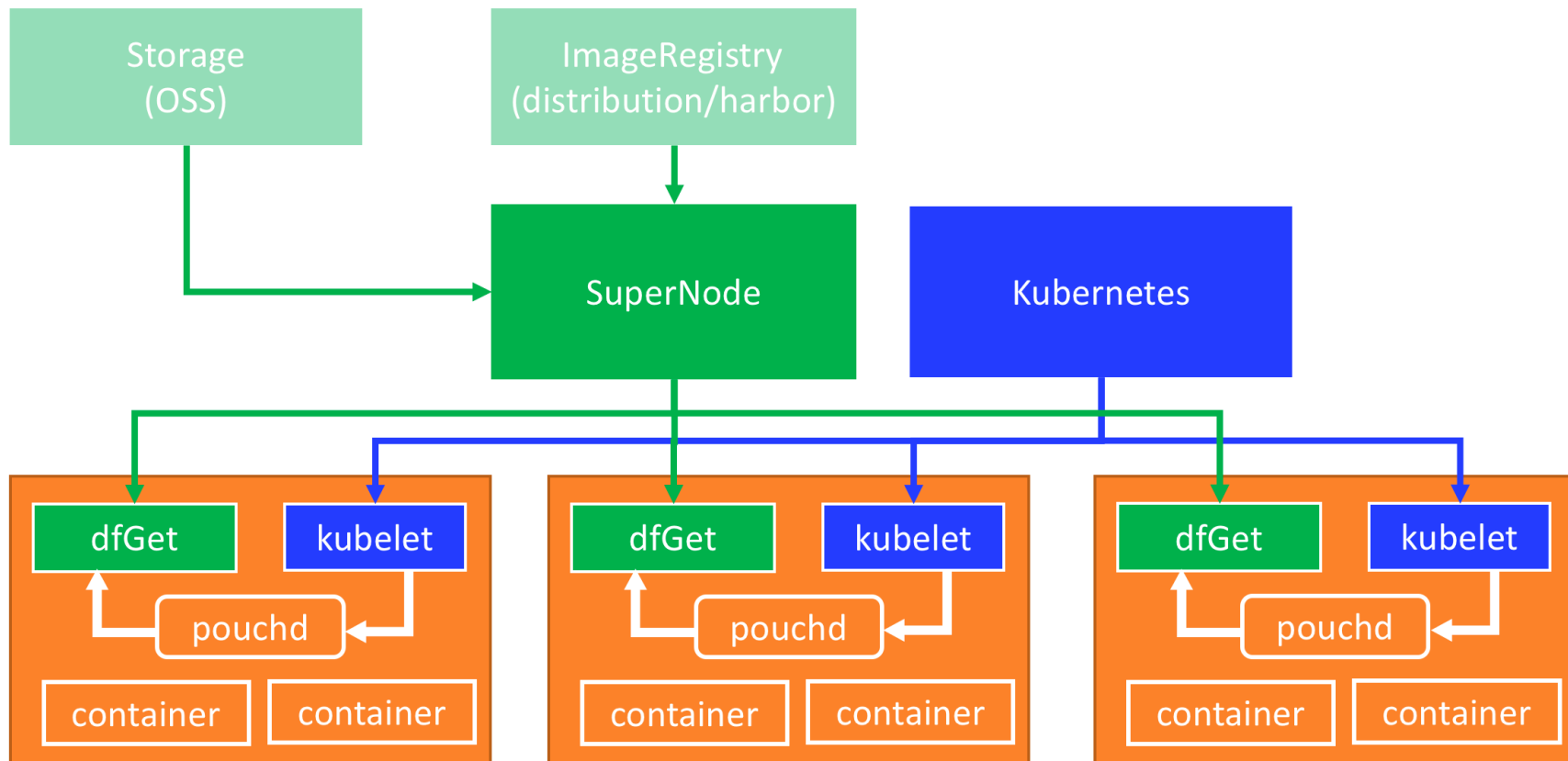
```
1. allensun.shl@katacontainer:/ (ssh)

[root@katacontainer ~]#
```

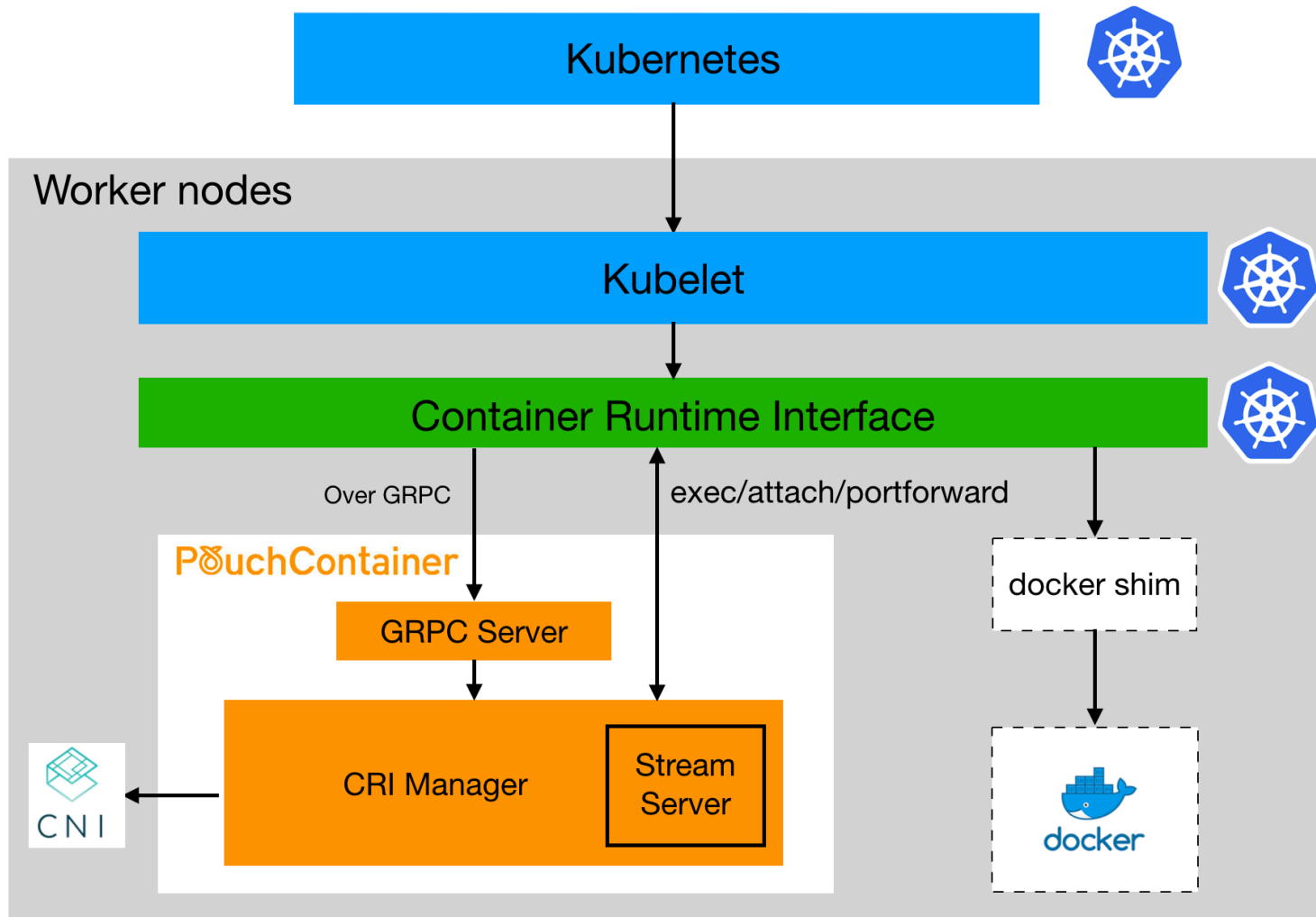
# P2P 镜像分发



# 生态架构图

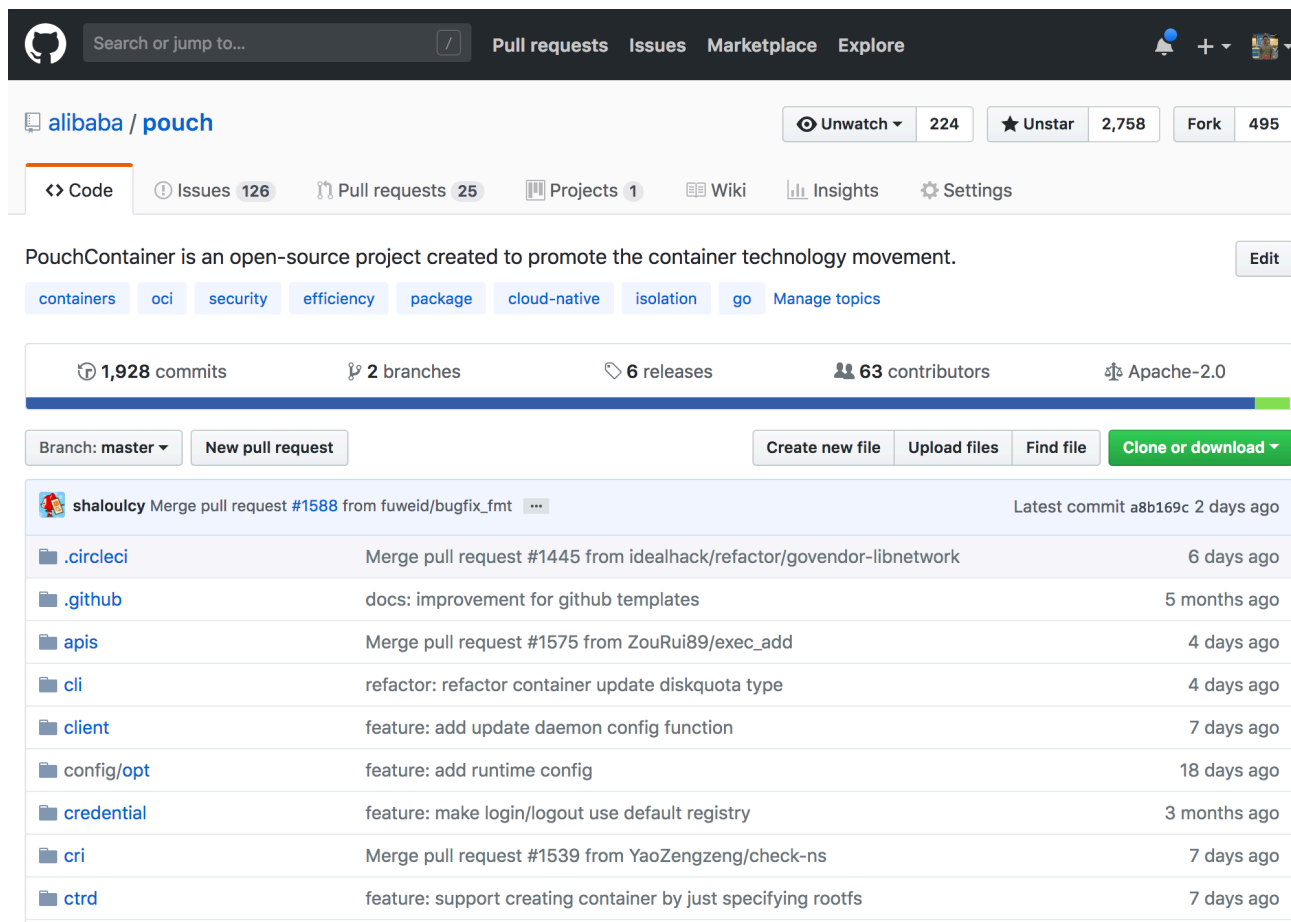


# 原生支持Kubernetes



# 开源发展

- 2.7k star
- 63位贡献者
- 1位协作机器人
- 文档
- 测试



The screenshot shows the GitHub repository page for `alibaba/pouch`. The repository has 2.7k stars, 63 contributors, and 1,928 commits. It is an open-source project created to promote the container technology movement. The page displays a list of recent pull requests and commits, including:

- shaloulcy Merge pull request #1588 from fuweid/bugfix\_fmt (Latest commit a8b169c 2 days ago)
- .circleci Merge pull request #1445 from idealhack/refactor/govendor-libnetwork (6 days ago)
- .github docs: improvement for github templates (5 months ago)
- apis Merge pull request #1575 from ZouRui89/exec\_add (4 days ago)
- cli refactor: refactor container update diskquota type (4 days ago)
- client feature: add update daemon config function (7 days ago)
- config/opt feature: add runtime config (18 days ago)
- credential feature: make login/logout use default registry (3 months ago)
- cri Merge pull request #1539 from YaoZengzeng/check-ns (7 days ago)
- ctrd feature: support creating container by just specifying rootfs (7 days ago)

欢迎参与PouchContainer开源社区



# THANKS

Website :  
[chinadevopsdays.org/](http://chinadevopsdays.org/)

Global Website:  
[www.devopsdays.org/events/2018-shanghai/](http://www.devopsdays.org/events/2018-shanghai/)

Official Email:  
[organizers-shanghai-2018@devopsdays.org](mailto:organizers-shanghai-2018@devopsdays.org)



Official Wechat

