云+容器
重新定义企业IT架构

阿里云高级技术专家 汤志敏

1 云原生计算与容器技术

云应用平台成熟度模型

智能创新

敏捷应用架构

敏捷交付流程

敏捷基础架构

分钟级创新

秒级伸缩、恢复

13X迭代

50%TCO节省

阿里云容器服务

DevOps

微服务

DUBBO

企业应用

.NET

ORACLE FUSION MIDDLEWARE WEBLOGIC SERVER

智能创新

IoT

容器服务

多集群管理

安全合规

混合云

日志、监控

应用仓库

镜像/解决方案

阿里云

docker

Linux

Windows

专有云

docker

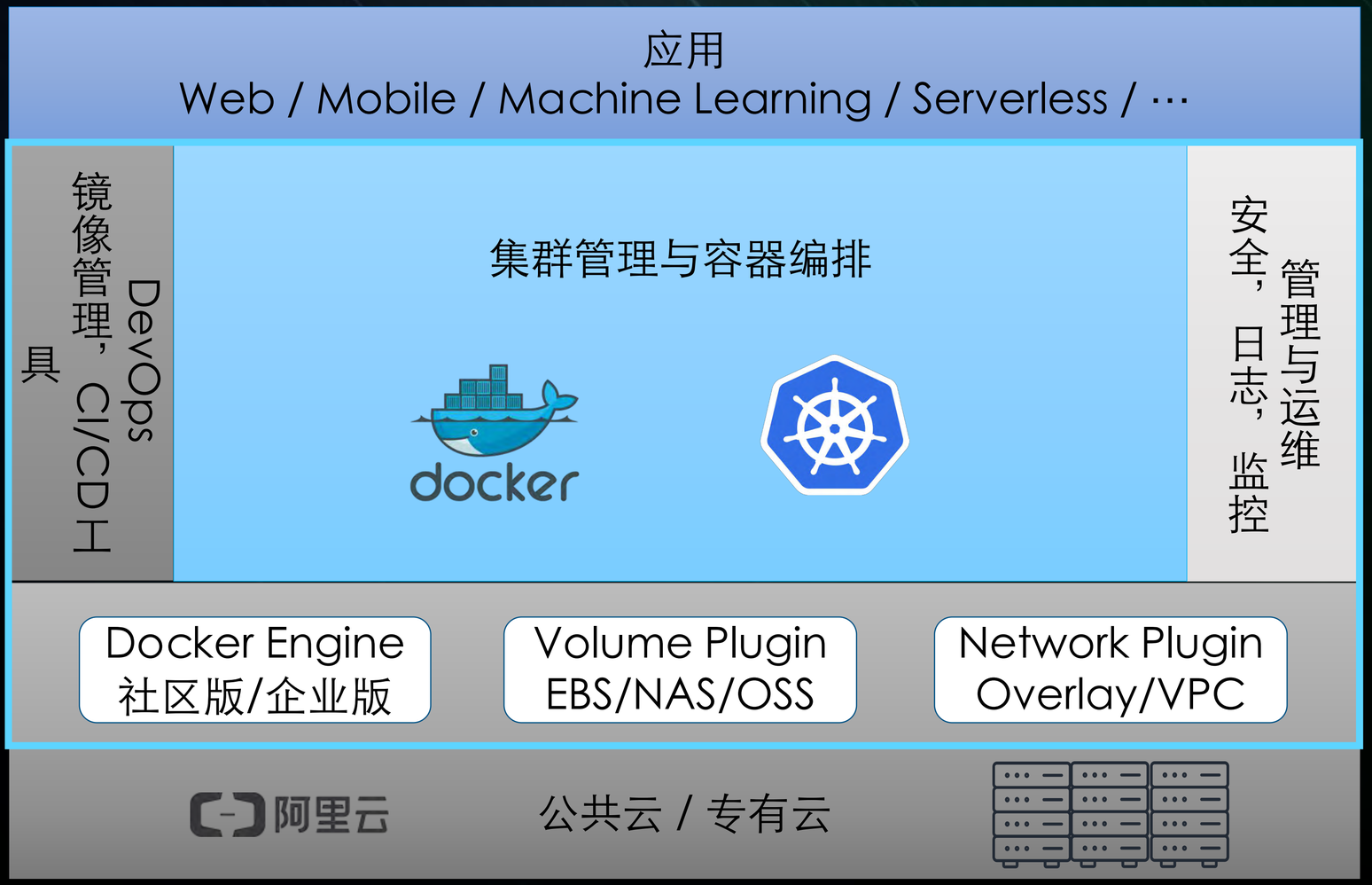
Linux

Windows

物理机

虚拟机

Kubernetes和Docker与云无缝集成



- 将容器技术和IT基础设施(公共云、专有云）集成到一起
- 内建混合云管理
- 内建GPU支持

容器技术发展

萌芽

发展

成熟

标准化

2000

2006

2011

2013

2015

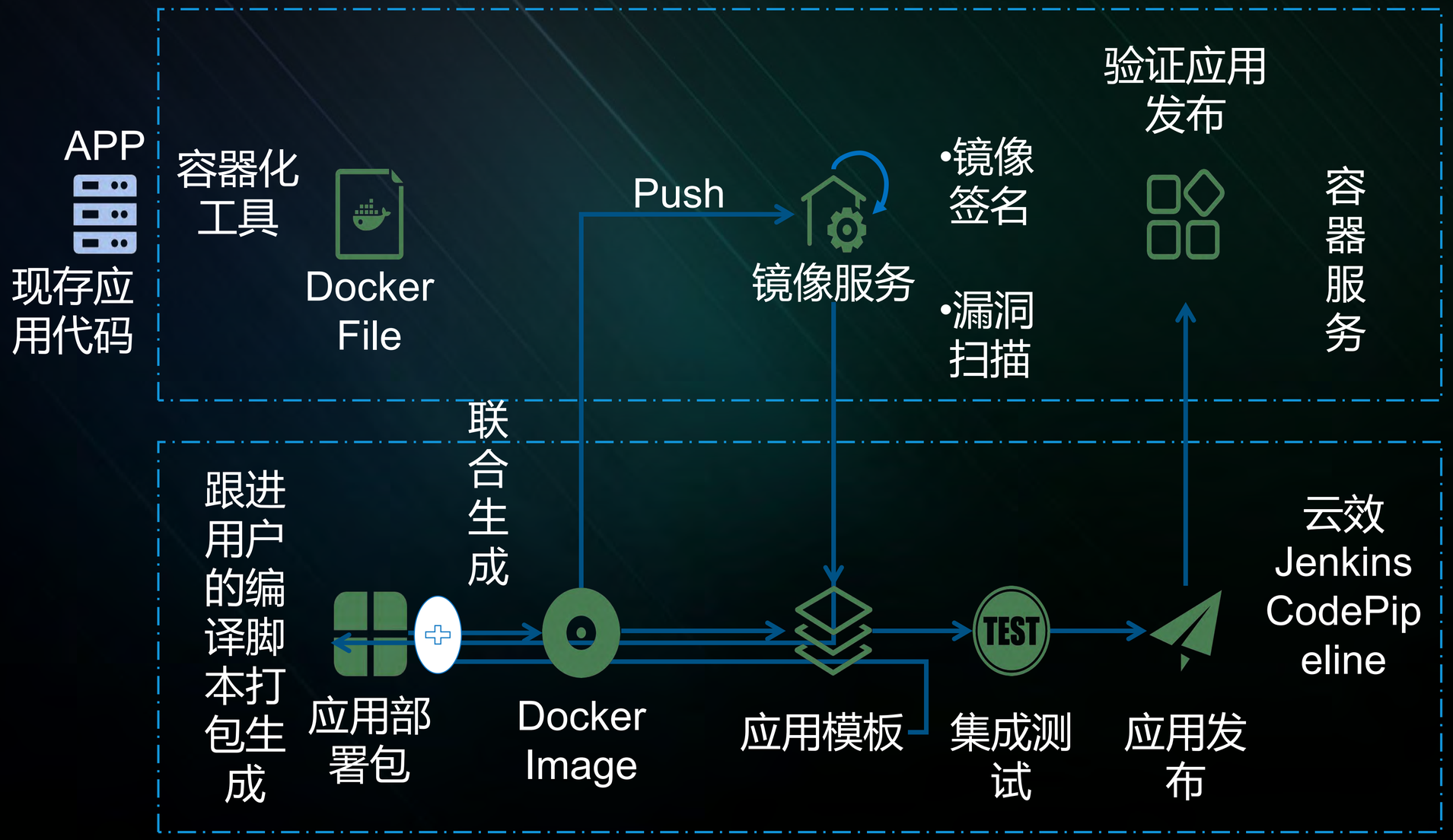
2016

未来将去向何方？

2 应用为王

1. 企业软件供应链
2. 简化应用部署
3. 应用容器化最佳实践

2.1企业软件供应链


APP
现存应用代码
容器化工具
Docker File
Push
镜像服务
•镜像签名
•漏洞扫描
验证应用发布
容器服务
联合生成
跟进用户的编译脚本打包生成
应用部署包
Docker Image
应用模板
TEST
集成测试
应用发布
云效Jenkins CodePipeline

Derrick

让开发人员可以专注在应用的开发上，可快速完成应用Docker化

特点

代码探测，支持常见编程框架

自动生成容器配置

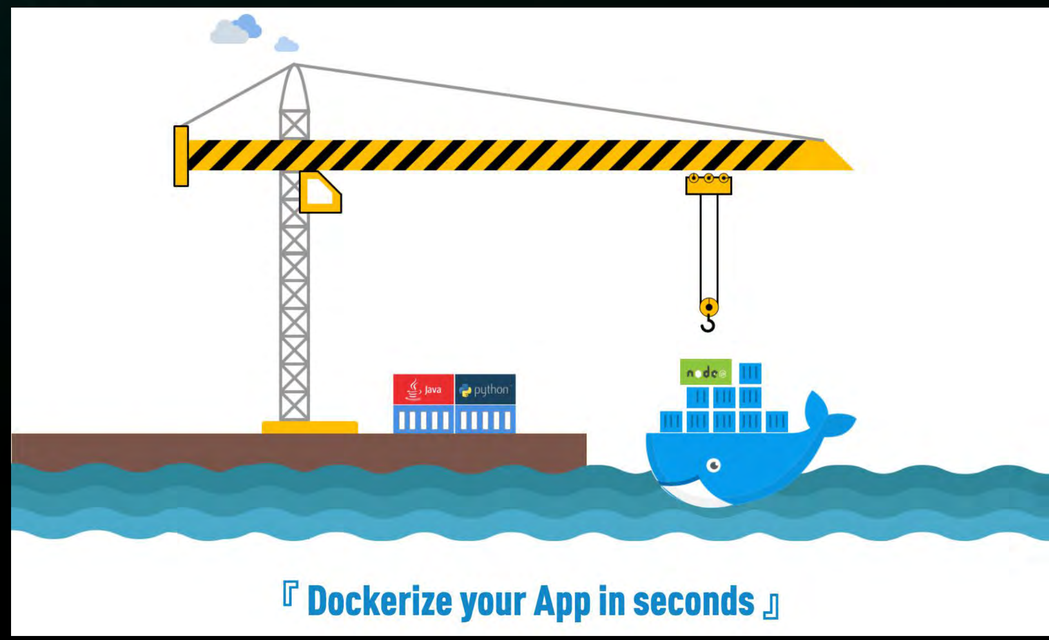
Dockerfile，docker-compose.yml, Kubernets resource

Jenkinsfile

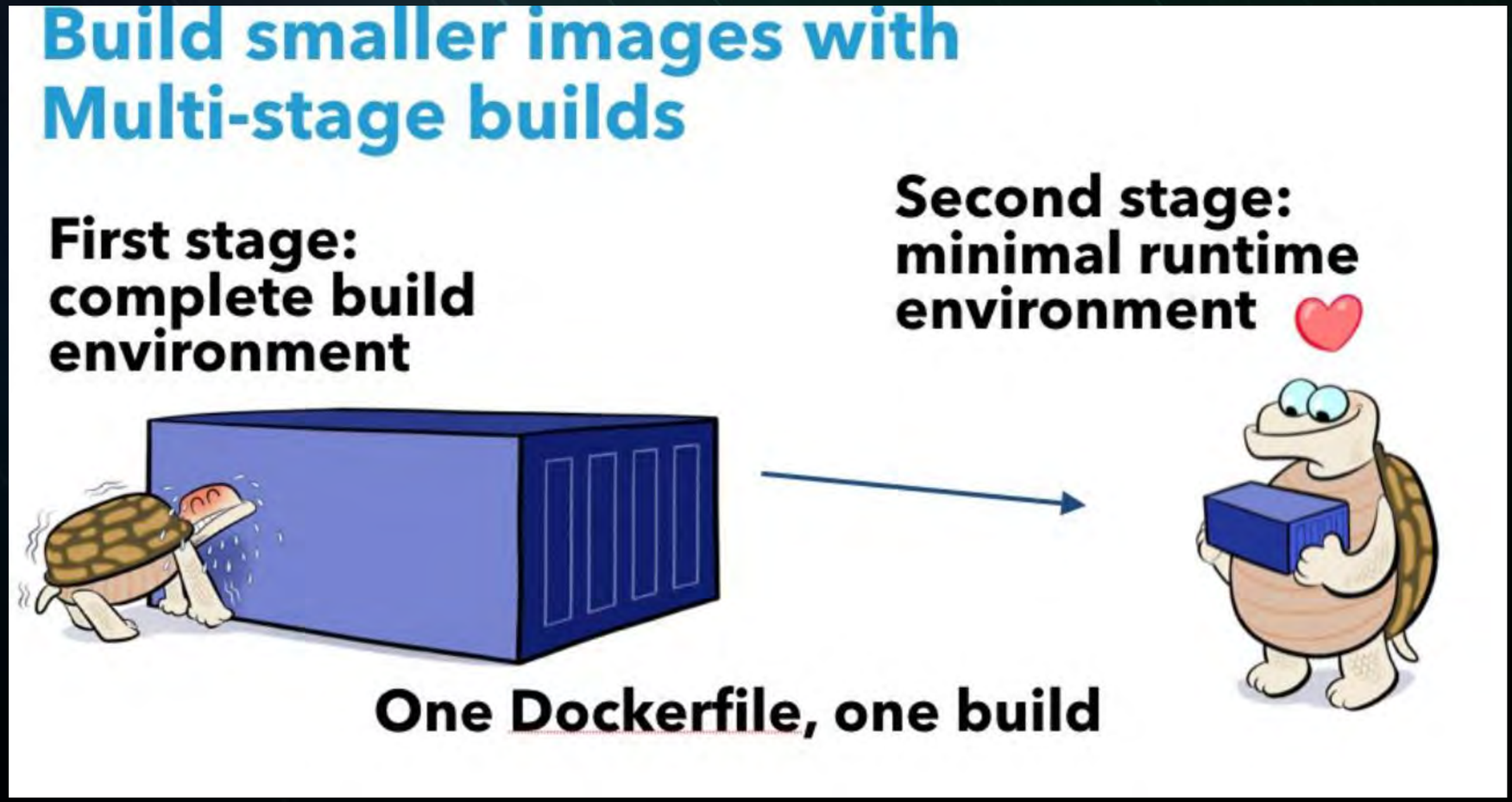
内建最佳实践

多阶段构建等

https://github.com/alibaba/derrick



Multi Stage


Build smaller images with
Multi-stage builds
First stage:
complete build
environment
Second stage:
minimal runtime
environment
One Dockerfile, one build

2.2 简化应用部署

Cloud Provider

Kompose

K8s的云资源适配

Helm/Charts
K8s应用管理

Kompose
Compose转化为
k8s Definiton

CloudProvider

CloudProvider

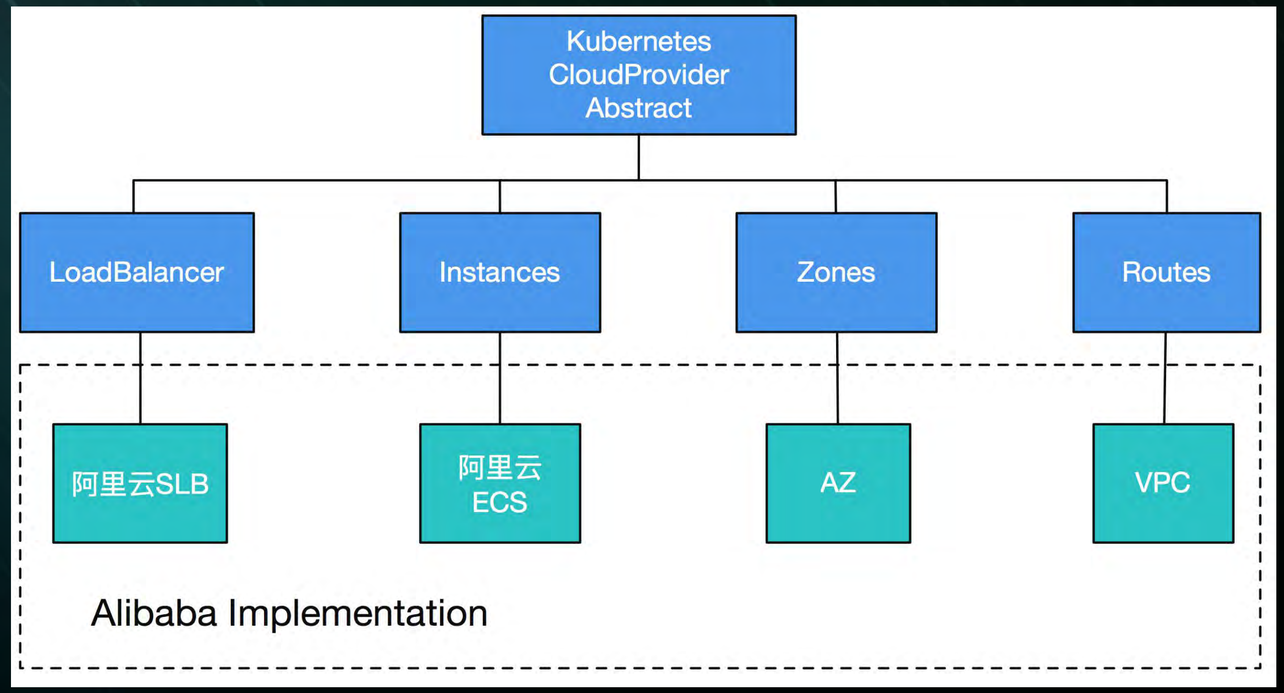
定义了一组接口，不同的云厂商通过实现这组接口来将本厂提供的各种服务集成到kubernetes集群中。

能力集成

网络路由配置、负载均衡动态配置、可用区管理、虚拟机管理

进化

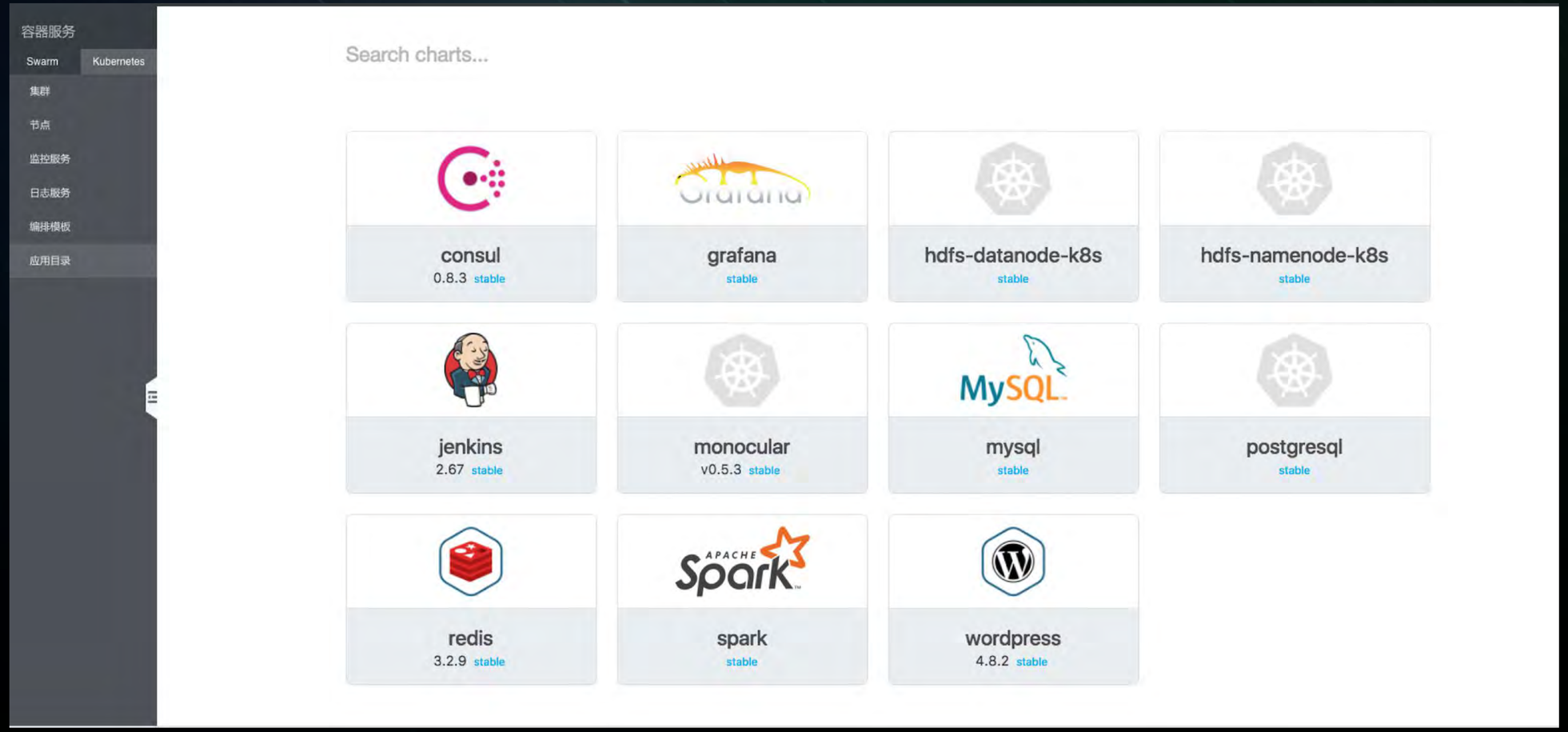
In-tree-cloudprovider
cloud-out-of-tree cloud controller



https://github.com/AliyunContainerService/alicloud-controller-manager

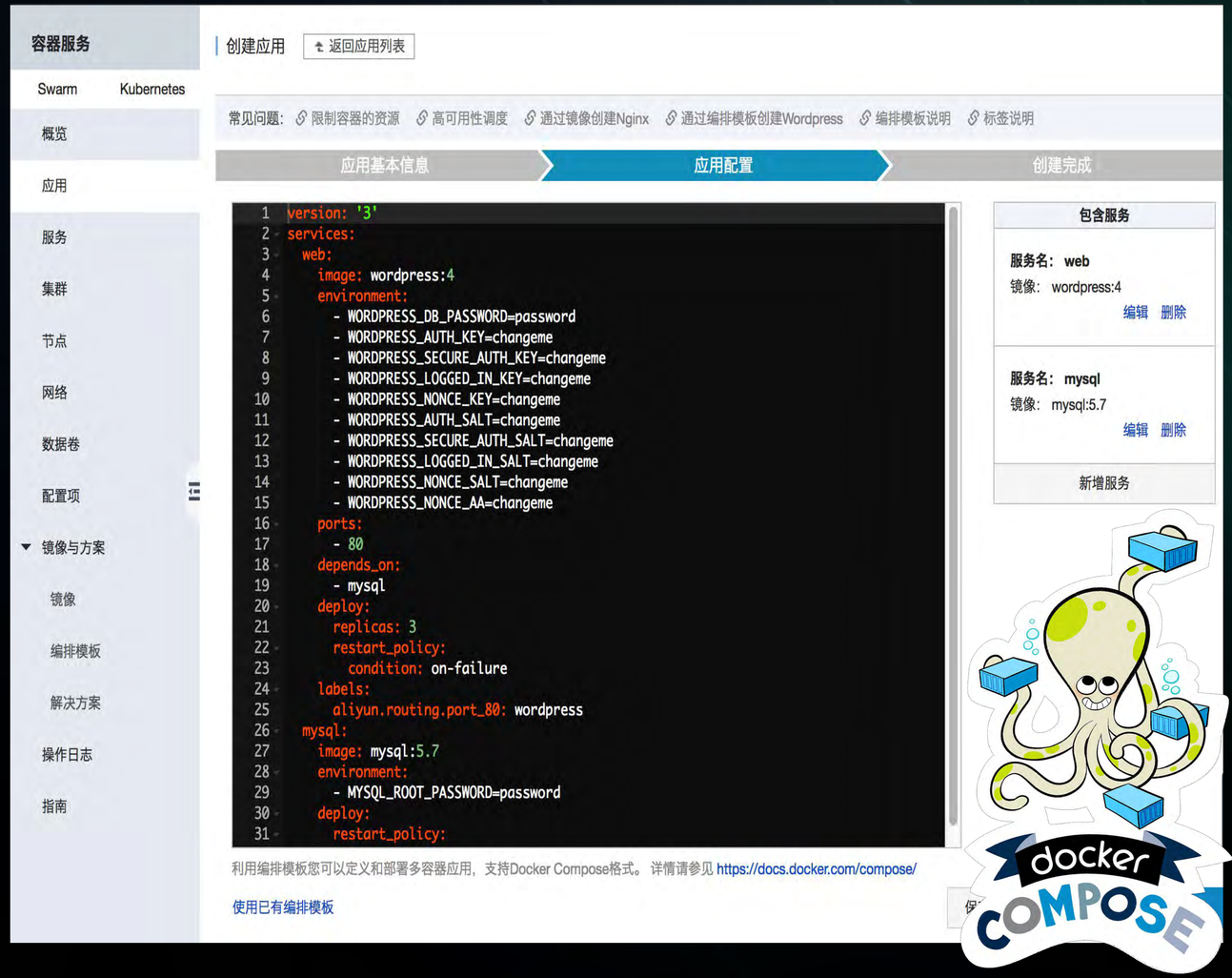
Helm/Charts

软件应用仓库

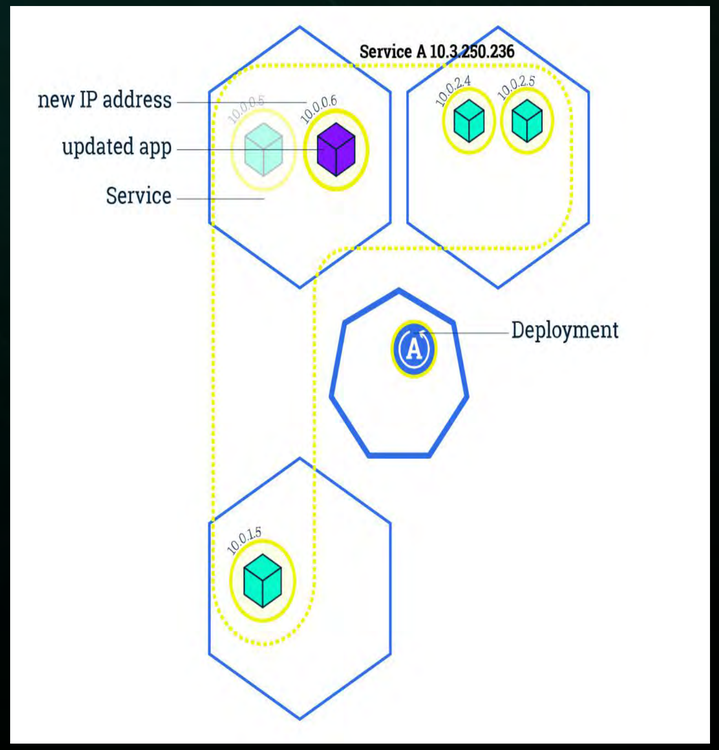


Kompose

支持Compose编排k8s部署：简化、复用



Kubernetes Deployment and Service



2.3 云上应用容器化的最佳实践

如何选择OS

应用容器化的最佳范式

JAVA应用的容器化

透明获取容器资源

应用高可用

OS 选择

- Docker很多特性依赖于操作系统内核版本
- 推荐

  Ubuntu 16.04 LTS： 4.x内核，支持丰富的存储驱动

  CentOS/RHEL 7.4: 3.10内核，向后移植了很多对容器的增强

  Windows Server 2016

    在Swarm集群中，Windows节点应该作为工作节点

应用容器化的最佳范式

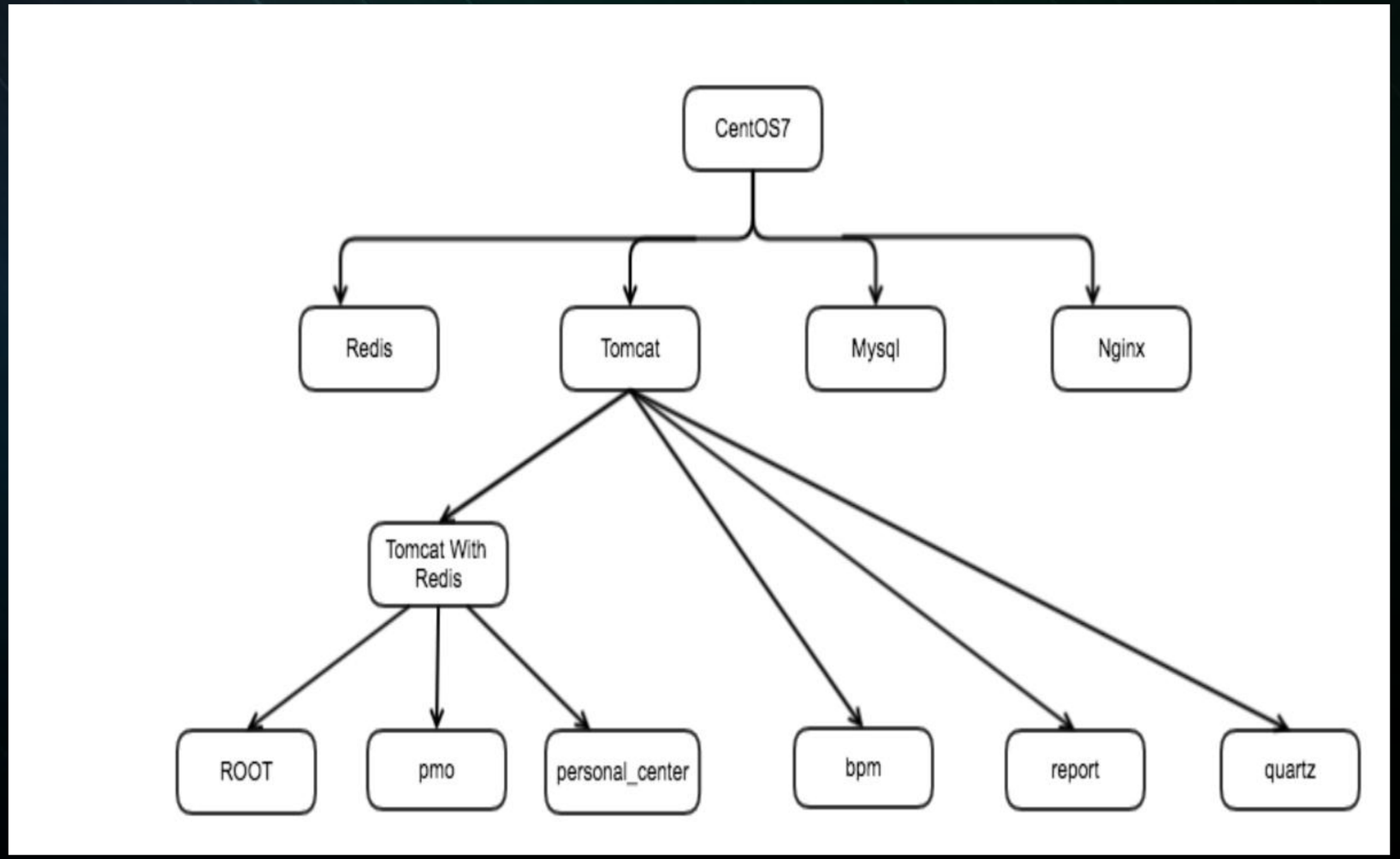
	Virtual Machine	容器	函数
最佳场景	企业核心应用 (E.g. ERP, CRM, …)	互联网应用和企业应用	事件驱动计算 (E.g. 转码, ETL, …)
应用交付	虚机镜像/ 软件配置管理、自动化脚本	Docker 镜像	代码/lib
对应方案	虚拟化	ContainerService/ ContainerInstance	FunctionCompute OpenFaaS/Fission

应用容器化的最佳范式-镜像分层

操作系统

中间件

应用

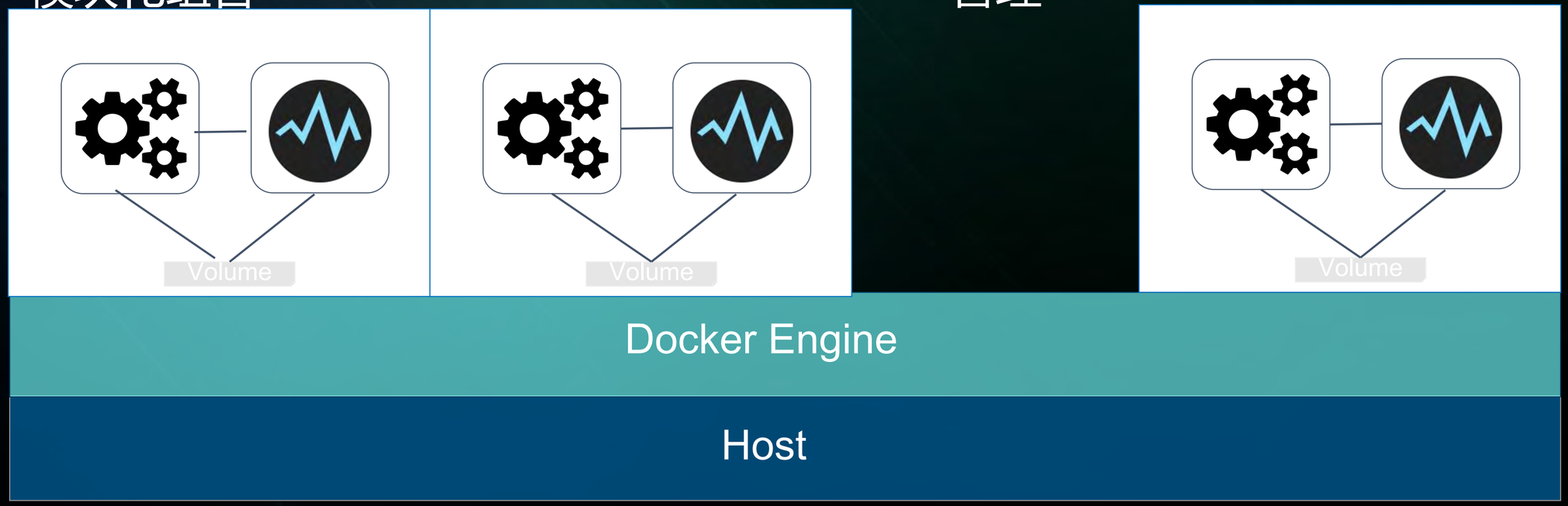


应用容器化的最佳范式-SideCar

镜像化应用交付
模块化组合

容器动态发现注册，如DNS, LB或定制IPAM分配IP

安全 – 减少攻击面
解耦 – 应用运维，独立更新
高效 – 宿主机内部共享监控、日志
可控 – 不可变基础架构， 支持版本管理



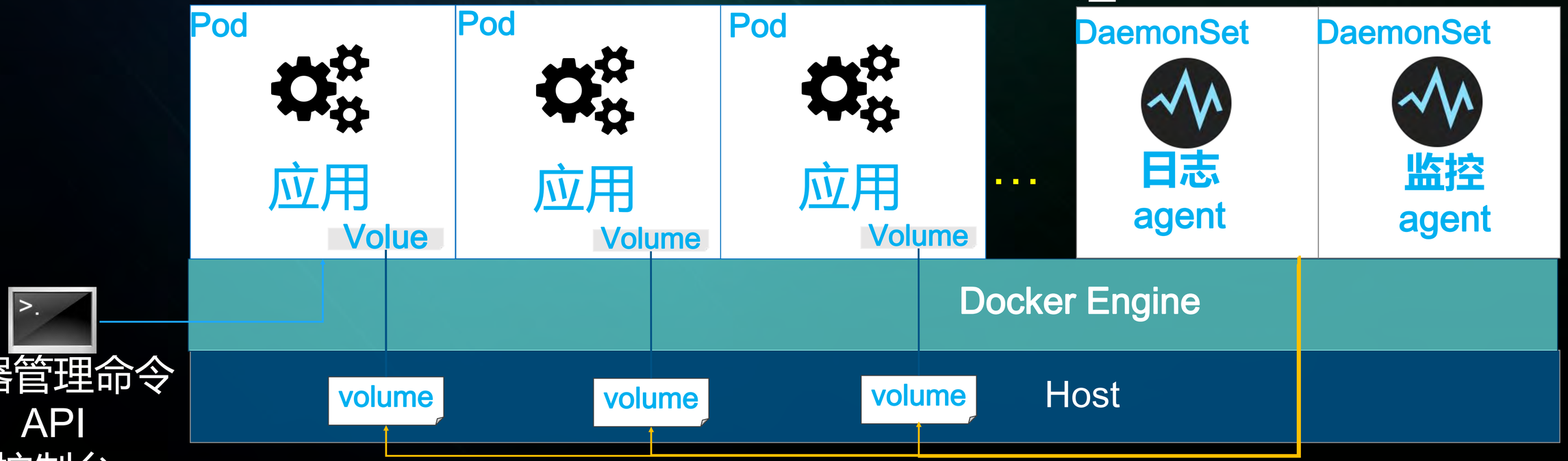
容器管理命令
API
控制台

应用容器化的最佳范式-DaemonSet

镜像化
应用交付

容器动态发现注册，如DNS,LB
或定制IPAM保证Ip不变

安全 – 减少攻击面
解耦 – 应用运维，独立更新
高效 – 宿主机内部共享监控、日
可控 - 不可变基础架构， 支持版
理



容器管理命令
API
控制台

Java应用的容器化迷思

Java应用容器化后相互影响、莫名OOM：早期JVM

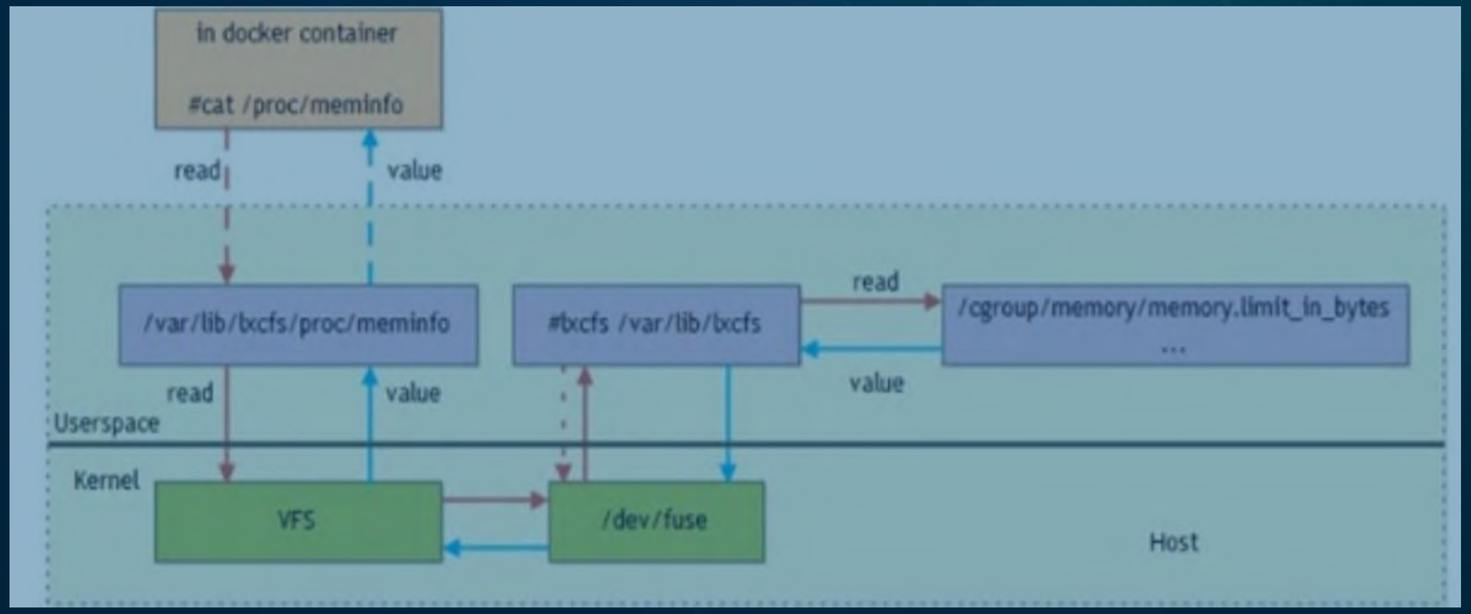
- 缺省堆栈大小取决于宿主机内存大小
- 缺省GC, JIT编译线程取决于宿主机CPU核数

解决方案

- Java SE 8u131+ 和 JDK 9 已经支持对容器环境资源限制的自动感知
  - java -XX:+UnlockExperimentalVMOptions -XX:+UseCGroupMemoryLimitForHeap …
- Docker 1.7 开始将容器cgroup信息挂载到容器中
  - 如可以从 /sys/fs/cgroup/memory/memory.limit_in_bytes 获取内存设置
  - 在容器的应用启动命令中根据Cgroup配置的正确设置 *-Xmx, XX:ParallelGCThreads等参数*

透明获取容器资源

lxcfs

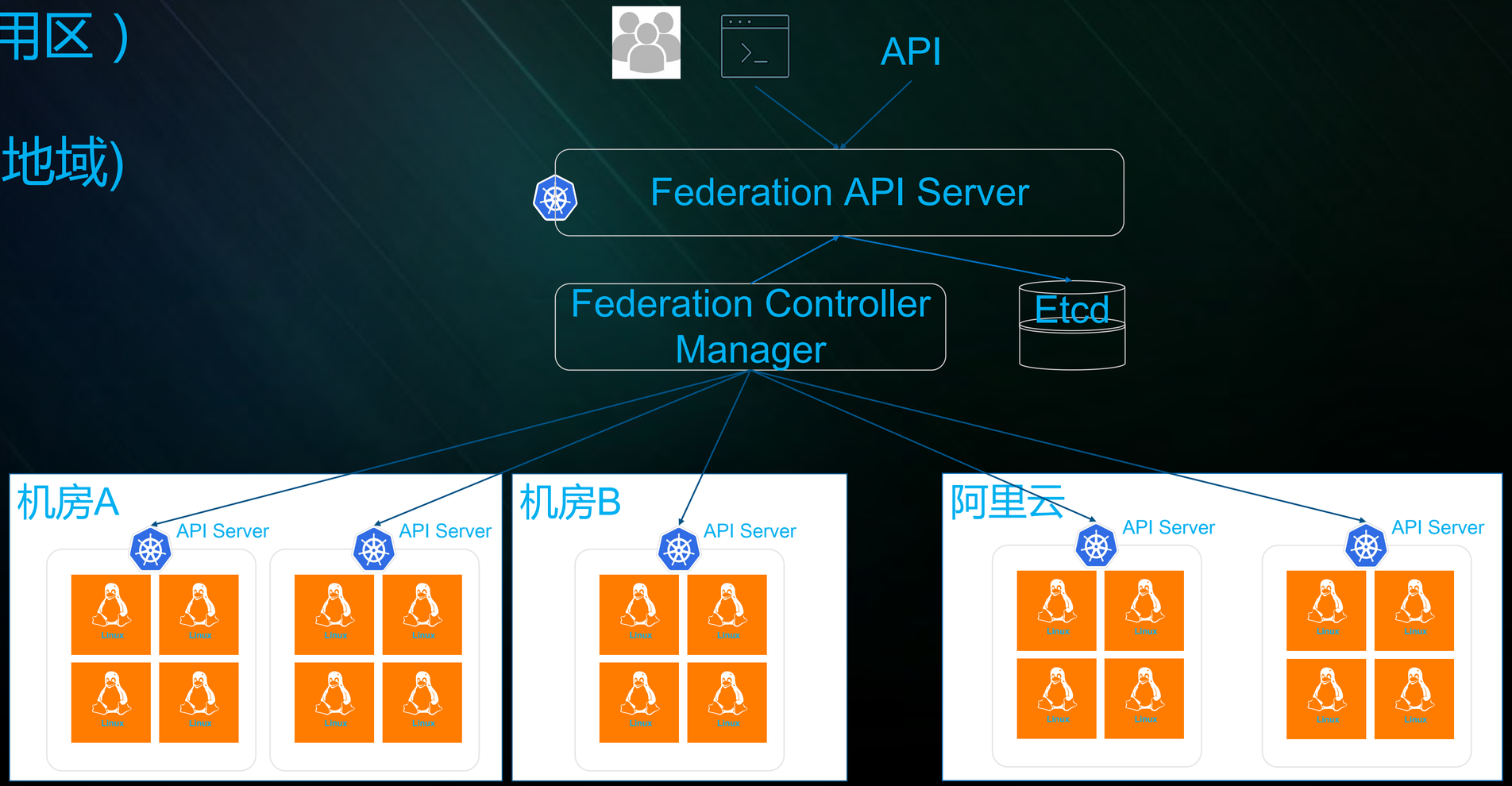


透明挂载lxcfs

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  annotations:
    "initializer.kubernetes.io/lxcfs": "true"
  labels:
    app: helloworld
  name: helloworld
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: helloworld
      name: helloworld
    spec:
      containers:
        - name: helloworld
          image: myapp
          imagePullPolicy: Always
```

应用高可用

- 跨AZ（可用区）
- 跨Region(地域)



感谢聆听

Thanks！