容器技术介绍及使用

胡庆宝

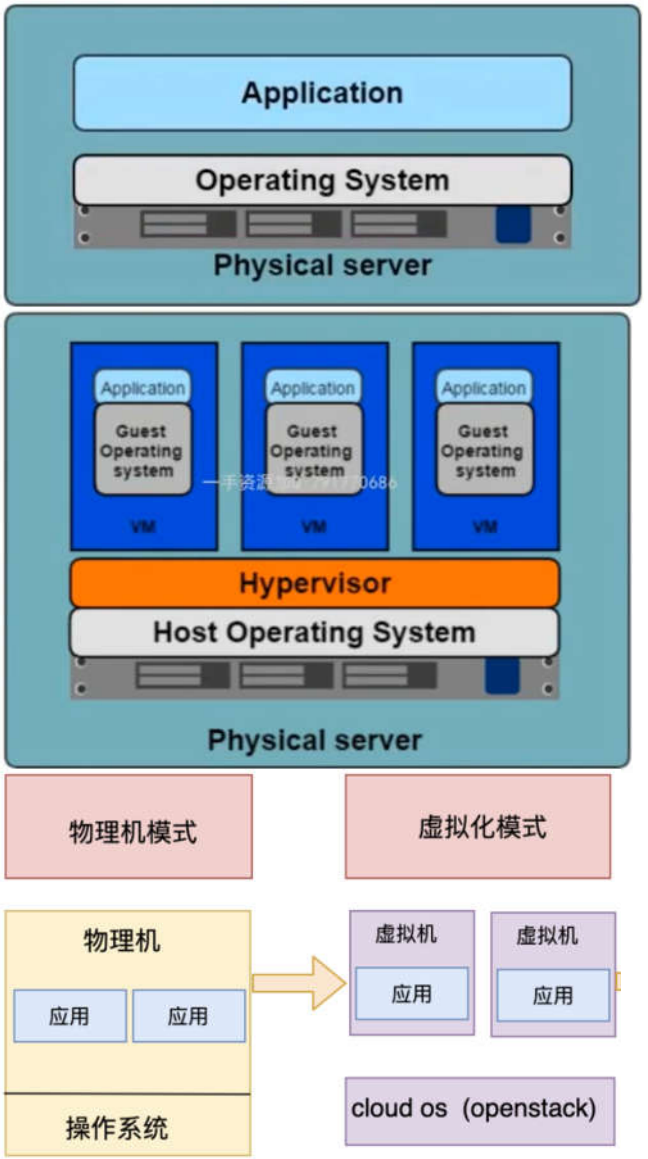
大纲

- 虚拟化技术发展
- 容器技术的意义
- 什么是容器
- 容器环境组件介绍
- 容器编排的意义
- 容器技术在高能物理计算中的应用

虚拟化技术发展(1/4)

- 传统应用部署方式：
  - 物理机 -> 操作系统 -> 应用部署
  - 部署慢（采购硬件、安装系统、装应用）
  - 成本高 & 资源浪费
  - 难以迁移和扩展

- 虚拟化技术出现后：
  - 一个物理机部署多个应用
  - 每个应用独立运行在一个VM中



虚拟化技术发展(2/4)

- 虚拟化的优点
  - 资源池—一个物理机资源分配到不同的VM中
  - 易扩展—加物理机or加虚拟机
  - 易云化—公有云、私有云、混合云

- 虚拟化推动云服务发展
  - 云计算：一个公司买下好多计算机，然后对外出租计算力。
  - 云计算服务体系，分为三层：IaaS（基础设施即服务）、PaaS（平台即服务）、SaaS（软件即服务）

虚拟化技术发展(3/4)

- IaaS：
  - 公司提供机器资源，方便用户部署应用。
- PaaS：
  - 公司提供稳定的单一服务，用户直接使用服务。负载均衡PaaS，数据库PaaS，消息队列PaaS等。
- SaaS：
  - 公司整合多个单一服务，形成具有特定功能整体解决方案，提供给用户使用。智能运维SaaS，信息化服务SaaS，计算服务SaaS等。

用户业务“云原生”部署

虚拟化技术发展(4/4)

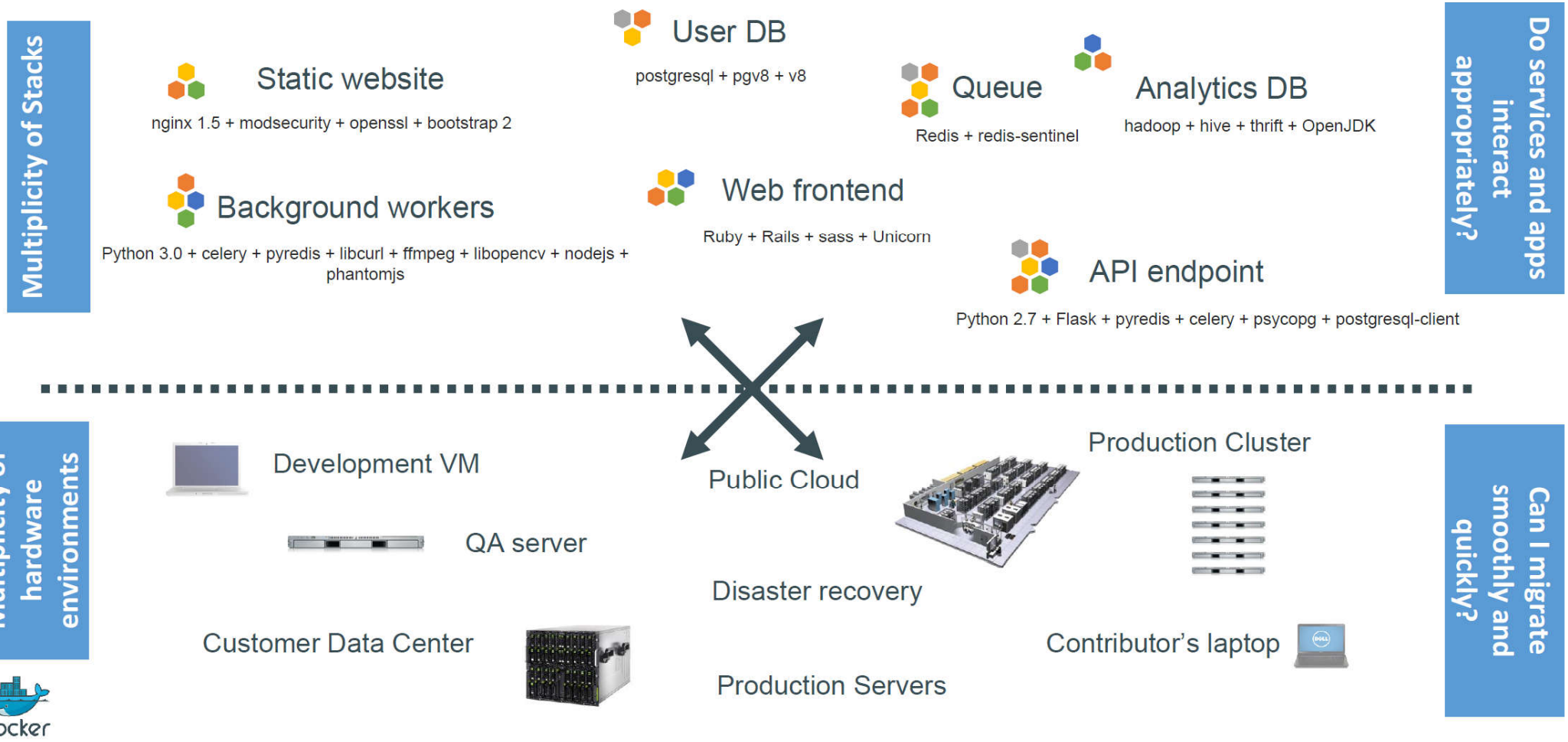
- 虚拟化的局限性
  - 每个VM都是一个完整的操作系统，要给其分配资源。VM数量越多，操作系统本身消耗的资源势必增多。还未部署APP，就已经消耗部分资源。
- PaaS和IaaS解耦合　如何快速消费底层基础架构资源？
  - 同一个公司研发的PaaS和IaaS，配合起来更默契。例如，一家企业选用了A厂的IaaS计算资源，上面的PaaS组件就也得用这家的，时间长了就会发现，所有重要的业务模块都跑在A厂体系内。假如A厂涨价了或者务变差了，想换成B厂的云或同时用AB两厂的云，结果发现我的开发人员大多只熟悉A厂PaaS技术栈开发。A、B两厂技术栈不通用。这时要想切换到B云，不仅很多系统要重新开发，恐怕连懂这些技术的人都要重新招。成本这么大，事实上就很难切换了，这就是所谓的“**上云容易下云难**”

容器技术的意义(1/6)

- 怎样把技术优雅的卖给别人？ 迁移和部署问题
  - 假如你的朋友跟你说：“你烧菜的技术不错啊，十万块钱卖给我吧！”你怎么卖？注意，不是把你烧的菜卖给他，而是把你烧菜的技术卖给他。
  - 满足基础功能的厨房家家都有（硬件资源+操作系统）
  - 仔细想想，起码有三大难度：
    - 首先，你身上好像没有一个能切下来的部分叫做“技术”，如果真的要把烧菜技术卖给朋友，那你只能每天晚上下班后肉身去他家颠勺。（资源依赖）
    - 而且，为了正常发挥手艺，你还需要朋友准备新鲜食材，把油盐酱醋的位置摆得跟你家一模一样，以防出现这边菜都糊了那边还没找到蚝油的情况。（环境依赖）
    - 这还不够，如果你做菜必须用到自己调配的秘制酱料，朋友那不可能有，那你是不是还要在家做好随身带着呢。（能力依赖）

  需要一个“帮助技术输出的技术” 应用开发->应用部署

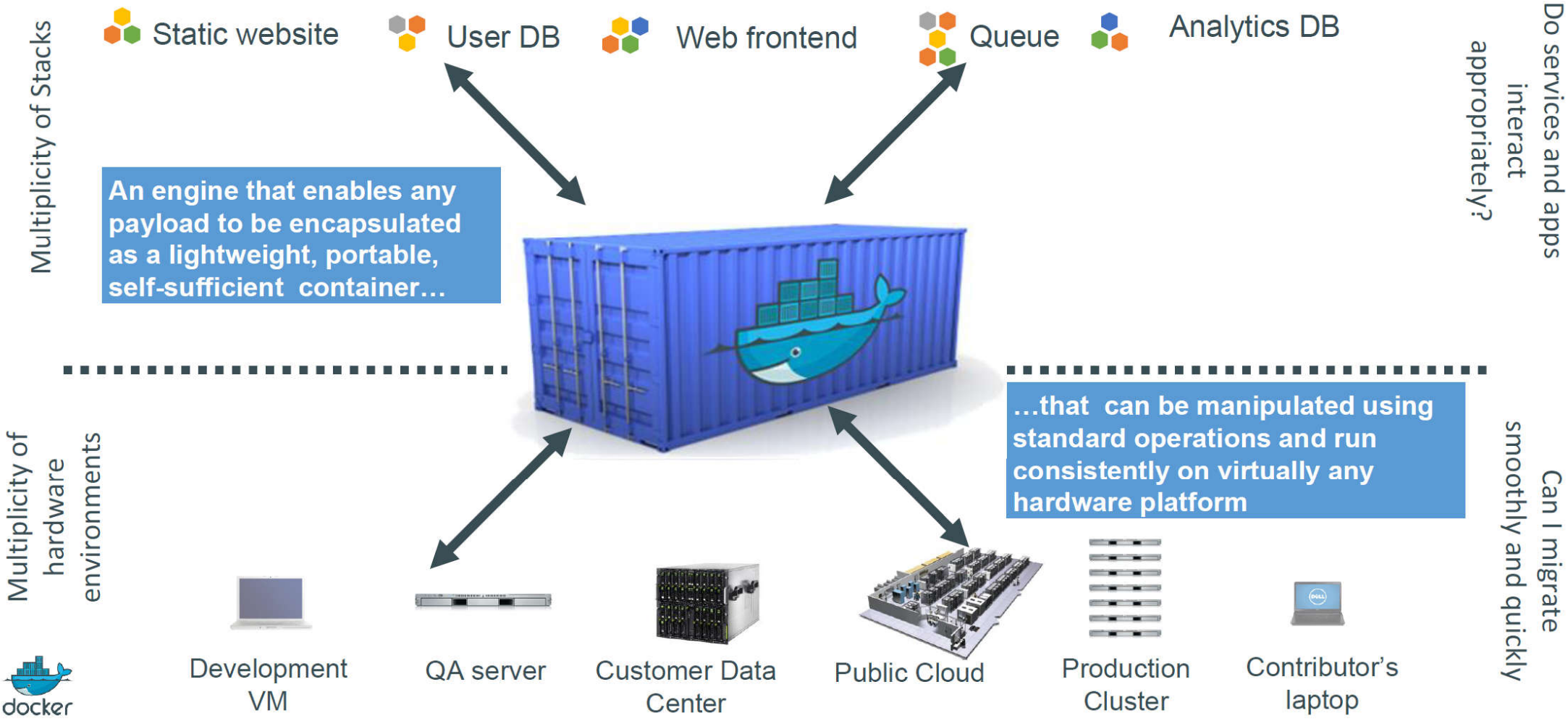
容器技术的意义(2/6)


Multiplicity of Stacks
Static website
nginx 1.5 + modsecurity + openssl + bootstrap 2
User DB
postgresql + pgv8 + v8
Queue
Redis + redis-sentinel
Analytics DB
hadoop + hive + thrift + OpenJDK
Do services and apps interact appropriately?
Background workers
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs
Web frontend
Ruby + Rails + sass + Unicorn
API endpoint
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client
Multiplicity of hardware environments
Development VM
QA server
Customer Data Center
Public Cloud
Disaster recovery
Production Servers
Production Cluster
Contributor's laptop
Can I migrate smoothly and quickly?
docker

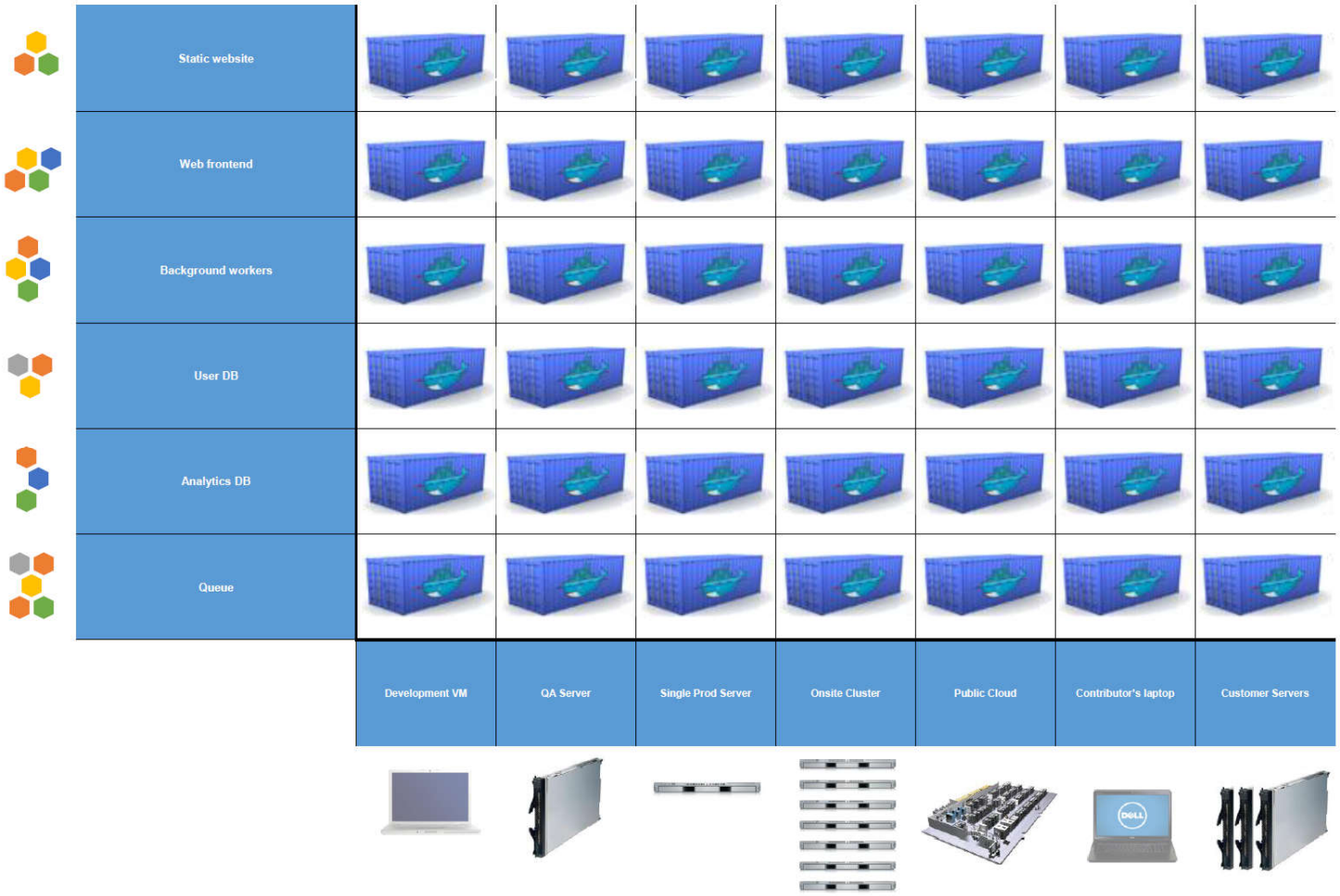
容器技术的意义(3/6)

	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

容器技术的意义(4/6)


Multiplicity of Stacks
Static website
User DB
Web frontend
Queue
Analytics DB
Do services and apps interact appropriately?
An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container…
…that can be manipulated using standard operations and run consistently on virtually any hardware platform
Multiplicity of hardware environments
Can I migrate smoothly and quickly
docker
Development VM
QA server
Customer Data Center
Public Cloud
Production Cluster
Contributor's laptop

容器技术的意义(5/6)

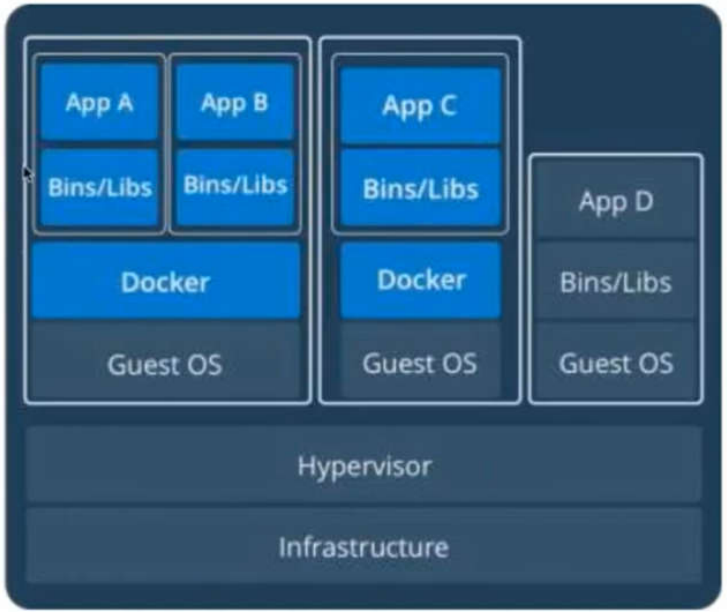

Static website
Web frontend
Background workers
User DB
Analytics DB
Queue
Development VM
QA Server
Single Prod Server
Onsite Cluster
Public Cloud
Contributor's laptop
Customer Servers

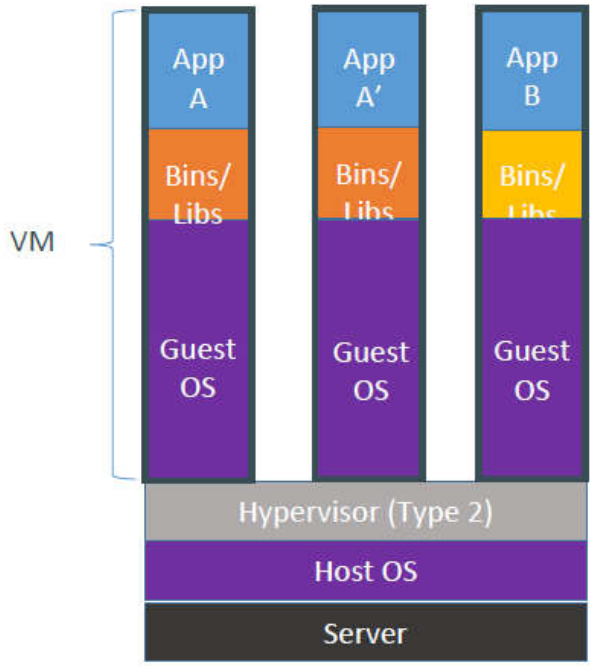
容器技术的意义(6/6)

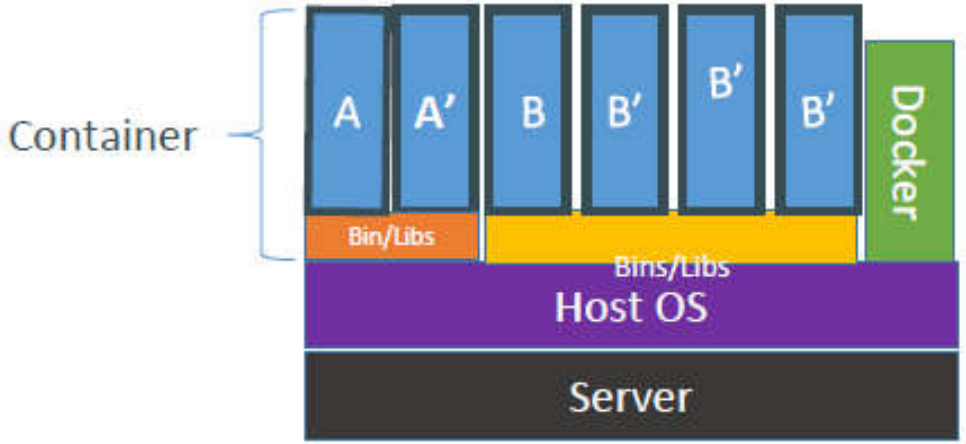
- 容器解决了什么问题
  - 解决了开发和运维之间的矛盾。
  - 在开发和运维之间搭建了一个桥梁
  - 是实现devops的最佳解决方案。

什么是容器(1/2)

- 虚拟机 vs 容器
  - 容器是APP层面的隔离
  - 虚拟化是物理资源层面的隔离

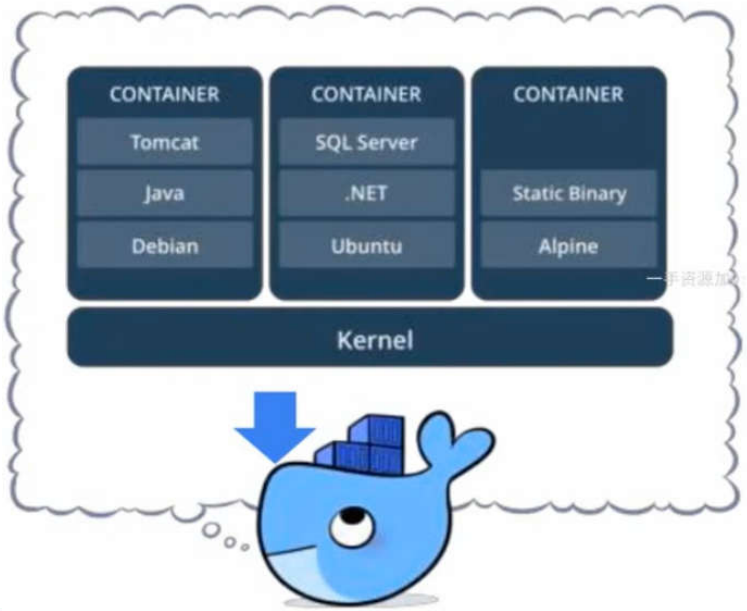


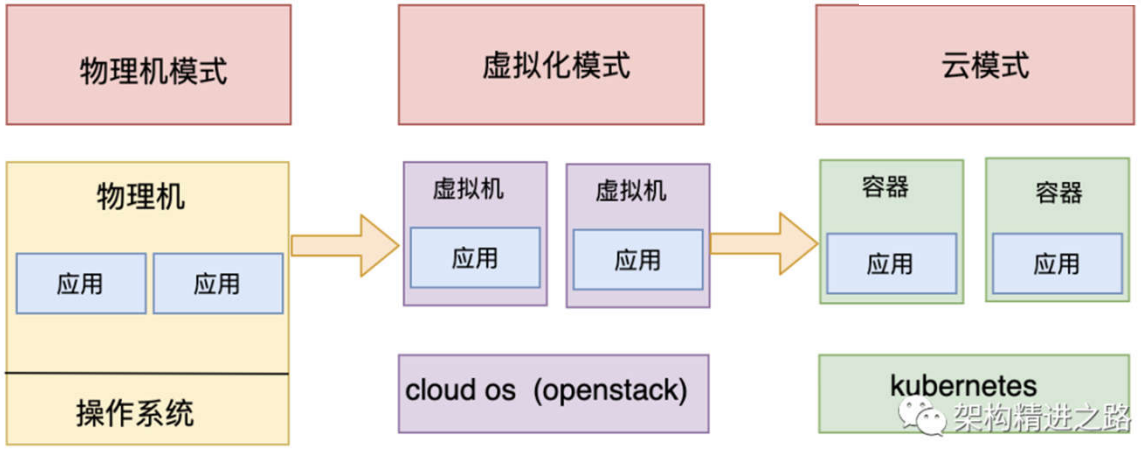




什么是容器(2/2)

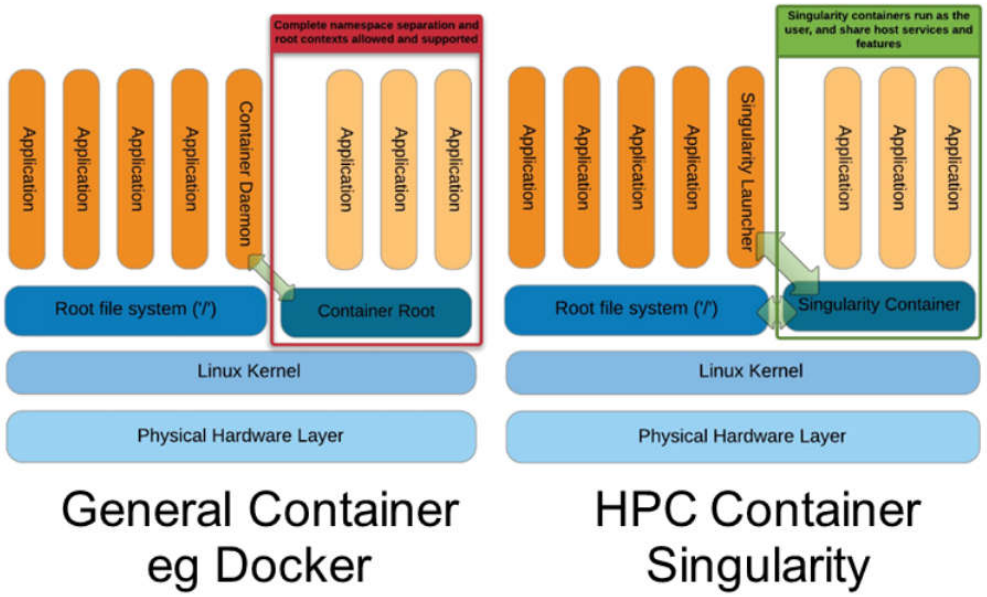
- 对软件和其依赖的标准化打包
- 应用之间相互隔离
- 共享同一个OS Kernel
- 可以运行在很多主流操作系统上
- 轻量级的虚拟化技术





容器环境组件介绍(1/4)

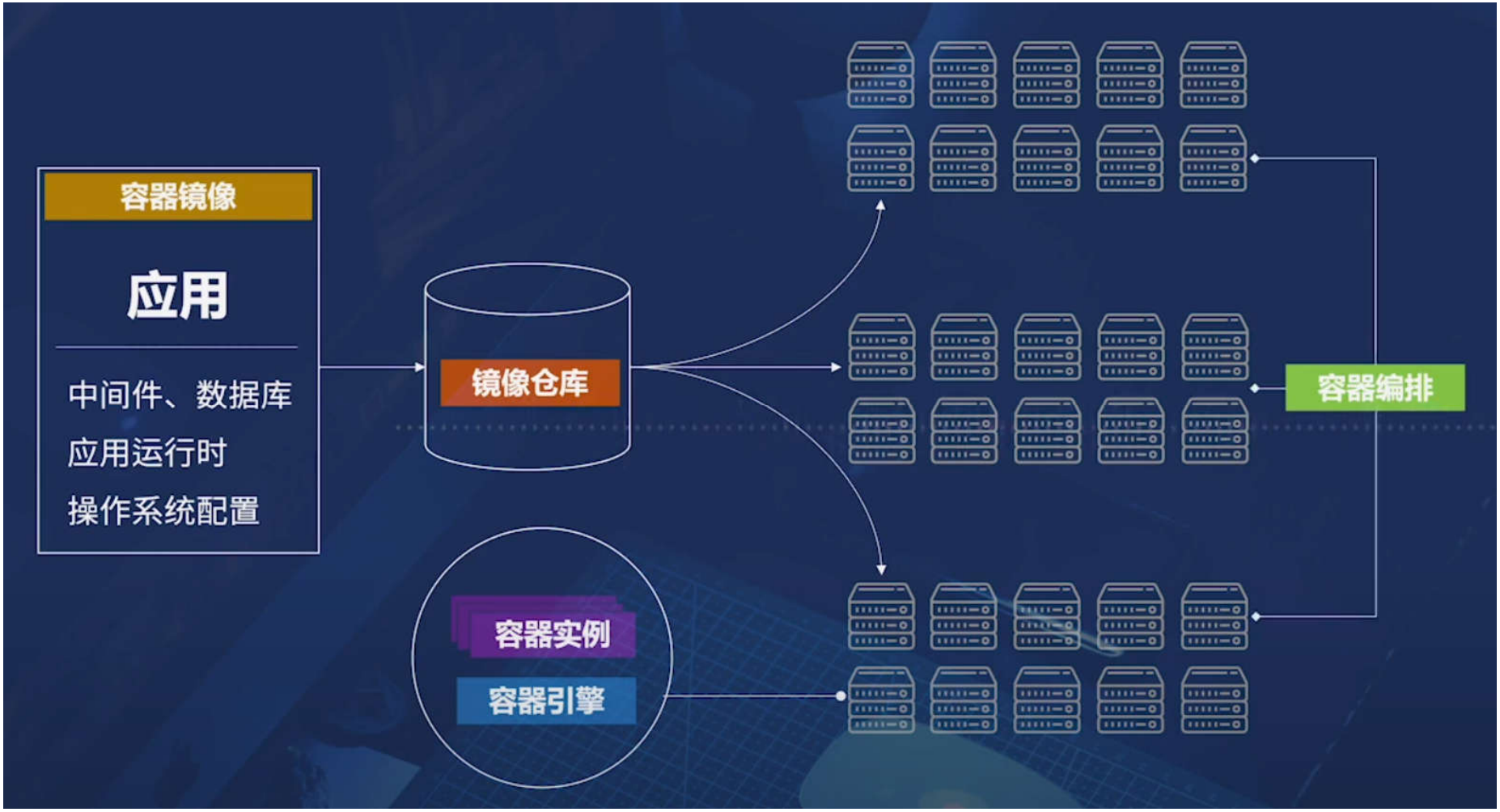
- Docker
  - 注重micro-service
  - 提供方便和稳定部署和服务,相比singluarity隔离的彻底
  - 可以大规模编排
- Singularity
  - 轻量、no-deamon、注重计算
  - 适合HPC，MPI(OpenMPI, MPICH, IntelMPI),GPU,infiniband
  - 用户权限容器内外都一致
  - /dev, /sys and /proc automount



容器环境组件介绍(2/4)


容器镜像
应用
中间件、数据库
应用运行时
操作系统配置
容器实例
应用
中间件、数据库
应用运行时
操作系统配置
容器引擎

容器环境组件介绍(3/4)



容器环境组件介绍(4/4)


容 器
容器镜像
镜像仓库
容器引擎
容器实例
容器编排
Infra
压缩包文件
文件服务器
轻量级
虚拟化工具
受控进程
资源调度器
Apps
应用交付格式
应用的分发
应用的运行环境
应用生命周期管理

什么是容器

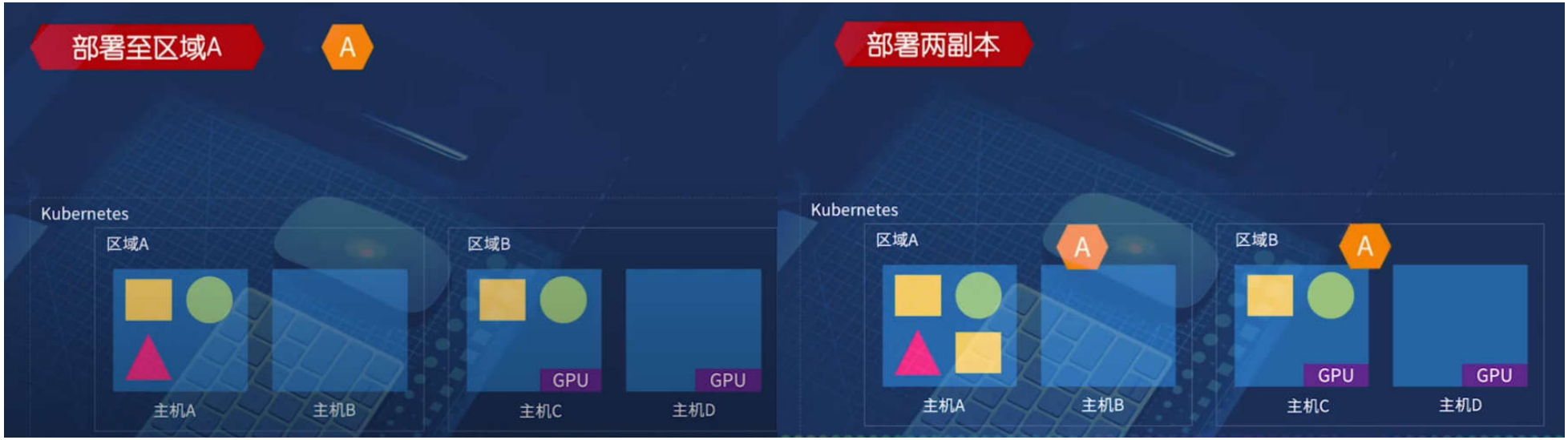
- 容器是轻量级的虚拟化技术
- 容器立足于基础架构，其价值体现在于对应用的支持
- 是当前热门的应用交付手段

容器编排技术(1/6)

- 什么是kubernetes(k8s)
  - 常规定义：容器编排平台
  - 运维视角：资源对接平台
  - 开发视角：分布式系统管理平台
  - 项目视角：开源项目
  - 发展视角：云时代的操作系统

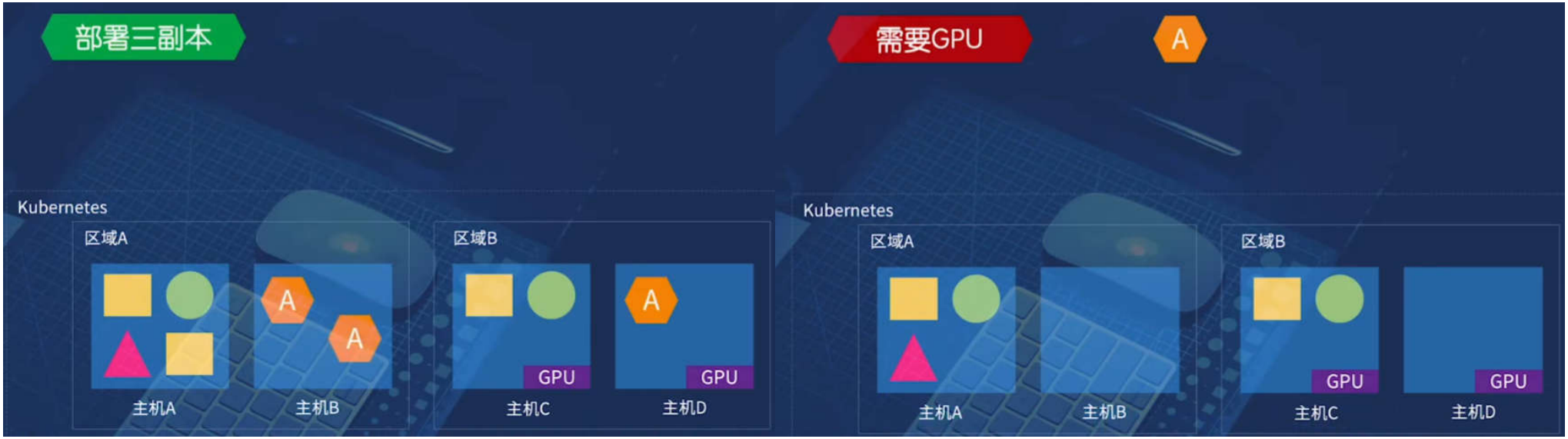
容器编排技术(2/6)

- 什么是容器编排



容器编排技术(3/6)

- 什么是容器编排



容器编排技术(4/6)


172.30.0.2
负载均衡
区域A
区域B
A
10.1.0.11
10.2.0.11
GPU
主机A
主机B
主机C
主机D
Kubernetes
存储

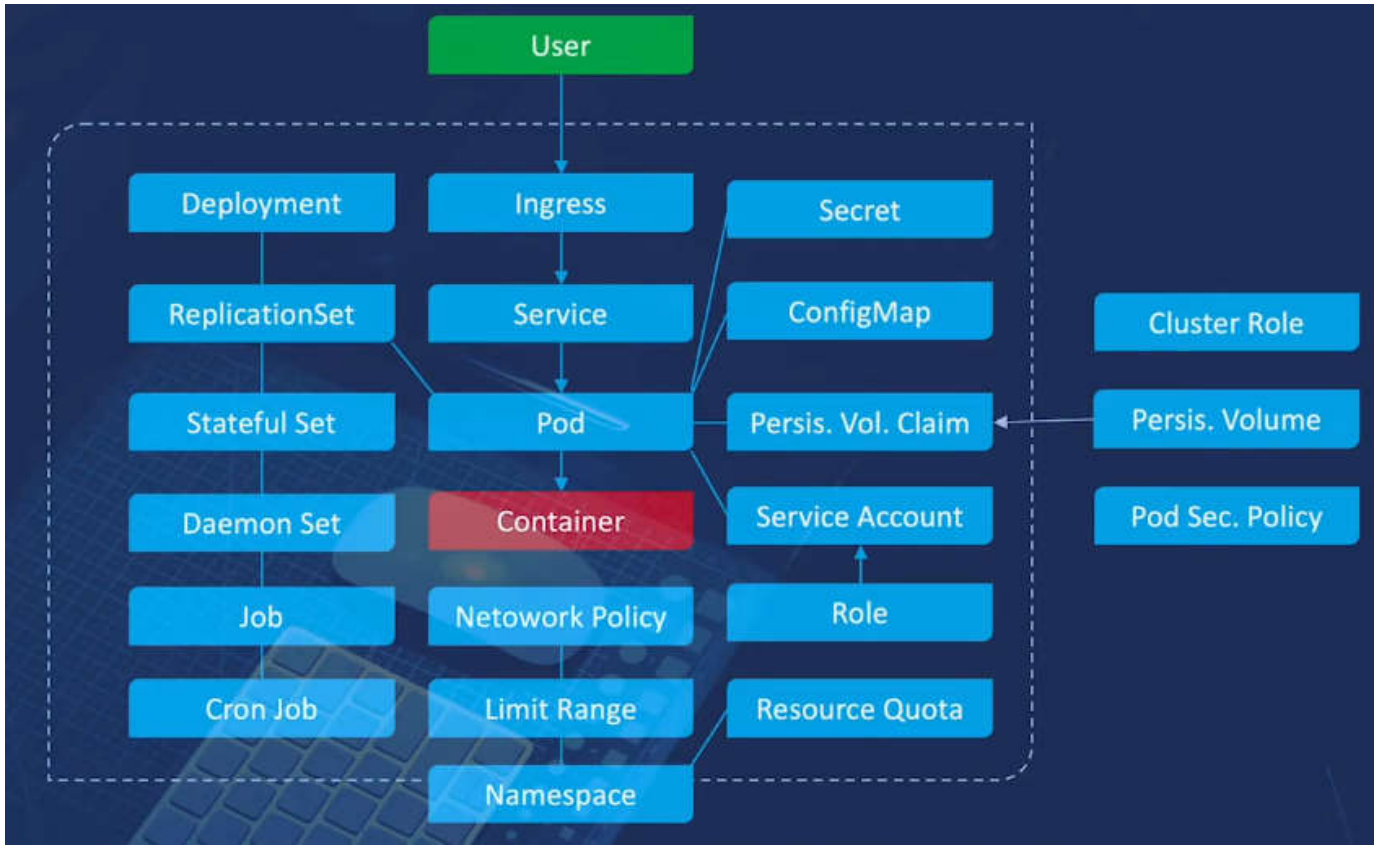
容器编排技术(5/6)

- k8s是面向应用的容器编排平台

- 弹性扩展
- 服务发现
- 健康检查
- 资源管理
- 错误恢复
- 配置管理
- 日志聚合
- 密钥管理
- 监控告警
- 流量转发
- 数据持久化
- 部署调度

容器编排技术(6/6)

- k8s提供一套强大的容器应用管理语言



登陆节点容器环境使用场景

- 个别数据分析需要SL5/6/7等系统环境，登陆节点Centos7操作系统无法满足需求，需要特定容器环境进行数据分析和作业调试。
- 使用说明：
  - http://afsapply.ihep.ac.cn/cchelp/zh/local-cluster/container/
  - 查看支持镜像
    - 命令格式：hep_container images 该指令可以查看当前提供的操作系统容器镜像。

```
$ hep_container images
Hep_container support images:
        SL5 : Scientific Linux 5
        SL6 : Scientific Linux 6
        SL7 : Scientific Linux 7
```

登陆节点容器环境使用场景

- 查看支持用户组/实验组
  - 命令格式：hep_container groups 该指令可以查看容器命令当前支持提供的用户组或者实验组。通过 -g 参数指定用户组或实验组，容器内会挂载对应用户目录和实验目录。不指定-g参数默认采用主组作为用户组或实验组。

```
$ hep_container groups
Hep_container support groups:
  u07|atlas|atlasrun|comet|offline|physics|higgs|ams|cms|dyw|hxmt|polars|juno|argo|lhaaso|
```

- 进入容器环境
  - 命令格式：hep_container shell [container image] 该指令可以在容器内启动一个shell，因此可以在容器外部与容器内部进行交互操作。运行exit则可以退出该shell。

```
$ hep_container shell SL5
Singularity: Invoking an interactive shell within container...
Singularity> cat /etc/redhat-release
Scientific Linux SL release 5.5 (Boron)
Singularity> exit
exit
```

登陆节点容器环境使用场景

- 容器内执行命令
  - 命令格式：hep_container exec [container image] [command] 该指令可以在外部主机上将指定的command运行在指定的容器内。 下例为在lxslc7上以SL5的环境运行SL5命令，并得到结果。

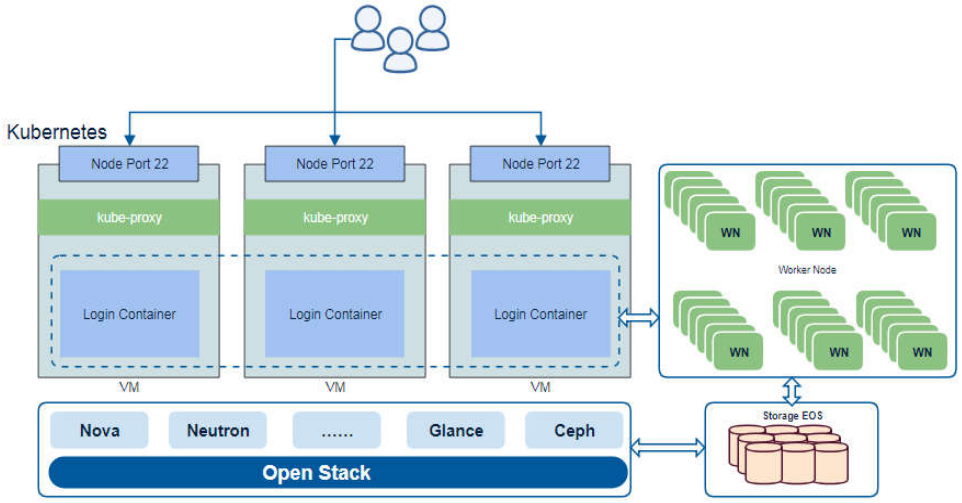
```
$ hep_container exec SL5 cat /etc/redhat-release
Scientific Linux SL release 5.5 (Boron)
```

计算集群容器环境使用场景

- 用户提交作业，并指定作业运行的操作系统。
  - **HepJob工具集**
    - -os：指定作业需要的操作系统版本，不指定时默认为 SL6。假设用户作业希望使用 SL7 系统的资源，则命令格式为
      - hep_sub -os SL7 job.sh

容器虚拟化平台

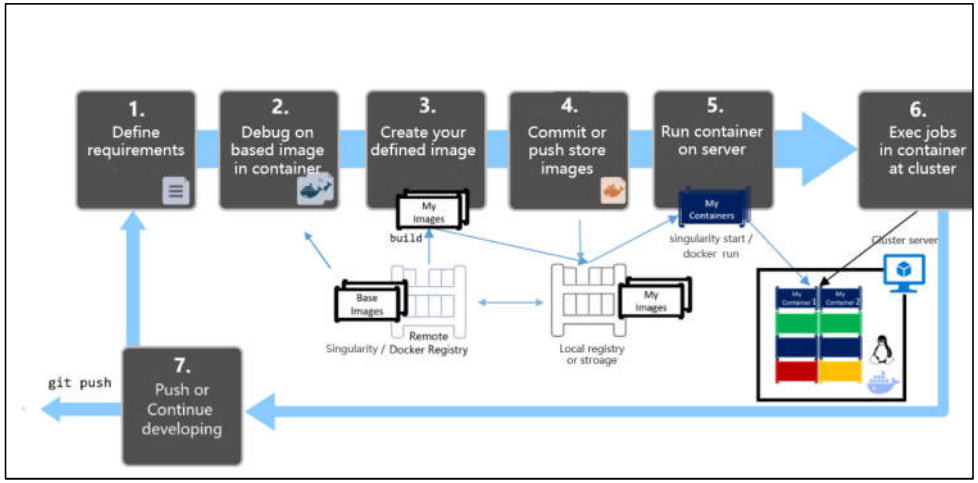
- LHAASO计算系统需求：高海拔、大容量、距离远、无人值守
  - 稻城现场机房
  - 海拔4400米，300平米规模，5000CPU核
  - 首次高海拔数据中心应用
- 传统负载均衡硬件或者LVS，采用容器化方式实现LHAASO稻城站点登陆节点负载均衡
  - 登陆节点根据负载/用户数自动扩展
- 保障站点高可靠、高可用



BES应用@HPC-Tianhe2

- 实现BES应用boss软件从HTC到HPC环境迁移运行
  - 问题：系统版本高、直接编译缺包少库，没有权限，技术支持…
- 容器方案优势：
  - 容器内有安装权限
  - 不受物理机系统影响
  - 容易debug&test
  - 作业运行环境高度一致
  - IHEP容器镜像和测试平台
- 开发制作Boss7.04和Mysql5.0镜像
- 批量模拟作业正式运行

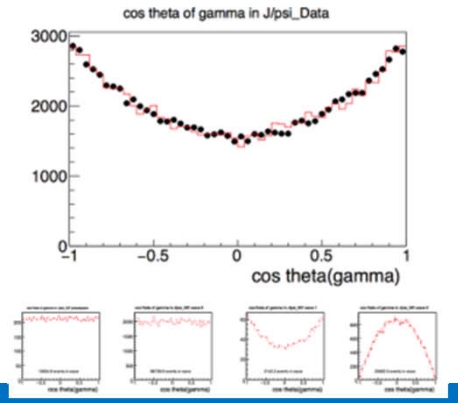
	HTC@IHEP	HPC@TianHe-2
	x86	x86
OS	SL6	Centos7
基础库	/afs,/cvmfs	base
数据库	mysql5.0	mysql5.7
容器工具	singularity/docker	singulairty
技术支持	天	周



分析需求->制定容器跨平台运行方案->容器本地测试环境->-制作容器->天河上运行调试

分波分析应用

- 分波分析（PWA, Partial Wave Analysis）是强子谱学的关键工具。分波分析计算包括作业并行和数据并行，进行数据拟合。
  - 软件环境： CUDA-10.1 • GCC7.3.1/4.8.5 • ROOT v5.34/32
  - 硬件环境： GPU（Tesla V100）， CPU: Xeon(R) 2X8 cores
  - 容器引擎：Docker->Singularity
  - 镜像存储：分布式文件系统/CVMFS(自动同步分发)
    - /cvmfs/containers.ihep.ac.cn/singularity/image/gpuwa/gpuwa_v1.2.sif
  - 作业调度：HTCondor 跨域调度
  - 分布式站点、镜像选择、定义参数….




谢谢聆听