

# Docker基础和应用场景

[www.huawei.com](http://www.huawei.com)

# 内容提要

- 背景

- 原理

- 基础

- 容器/镜像/仓库/网络

- 应用实践

- 模拟虚拟机

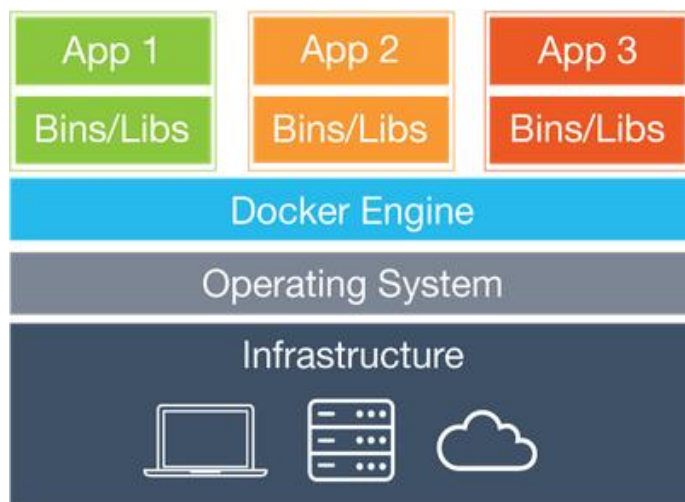
- 集群部署

# 背景-优势

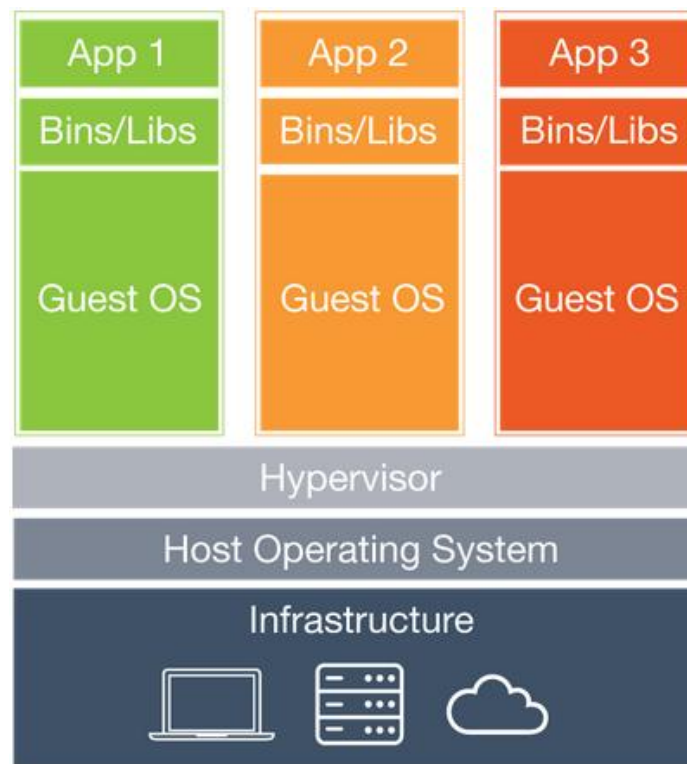
- 轻量化
- 资源隔离与控制
- 构建与部署效率
- 可移植性

# 背景-类比

- **Container VS VM**

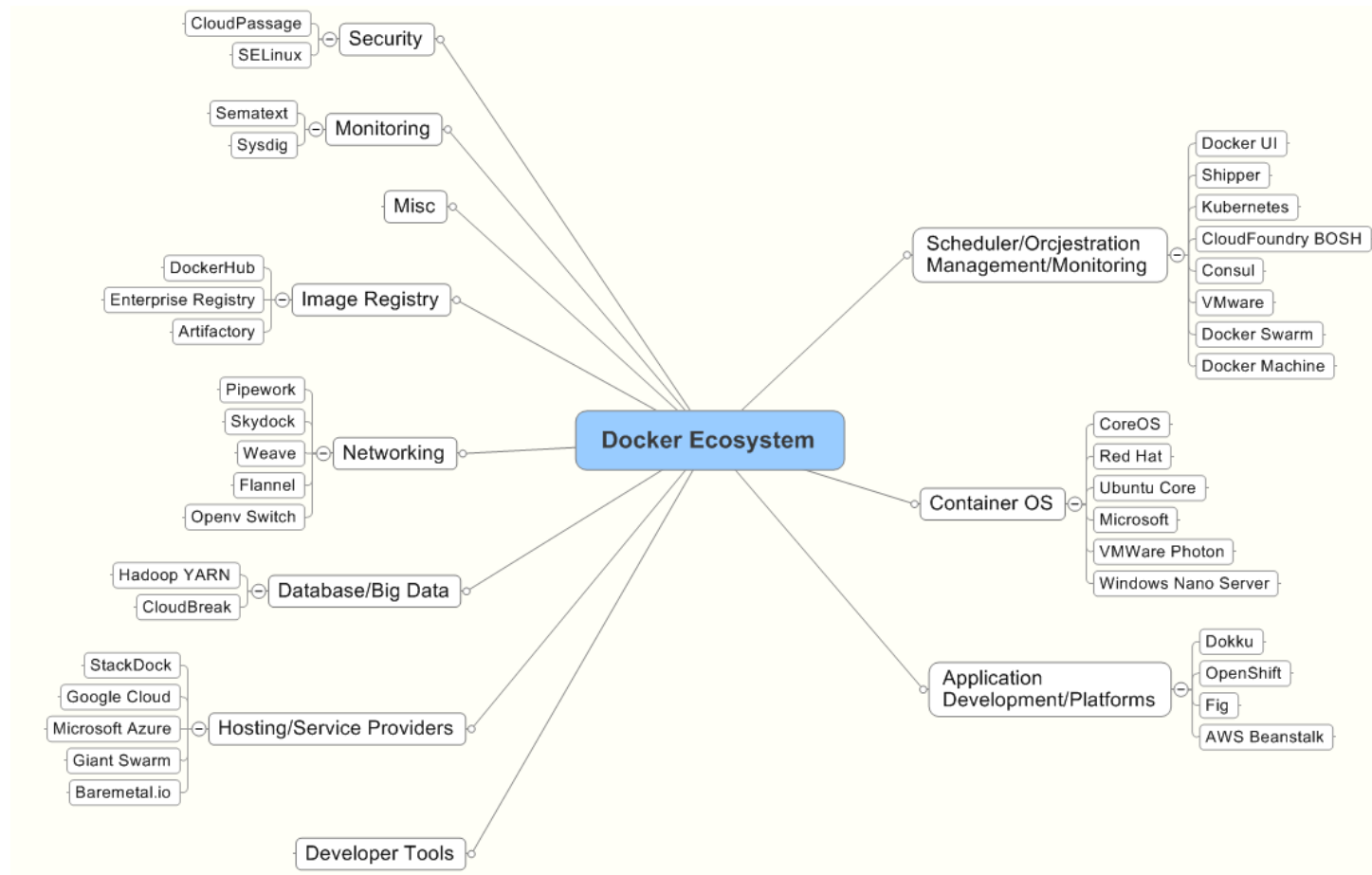


Container



Virtual Machine

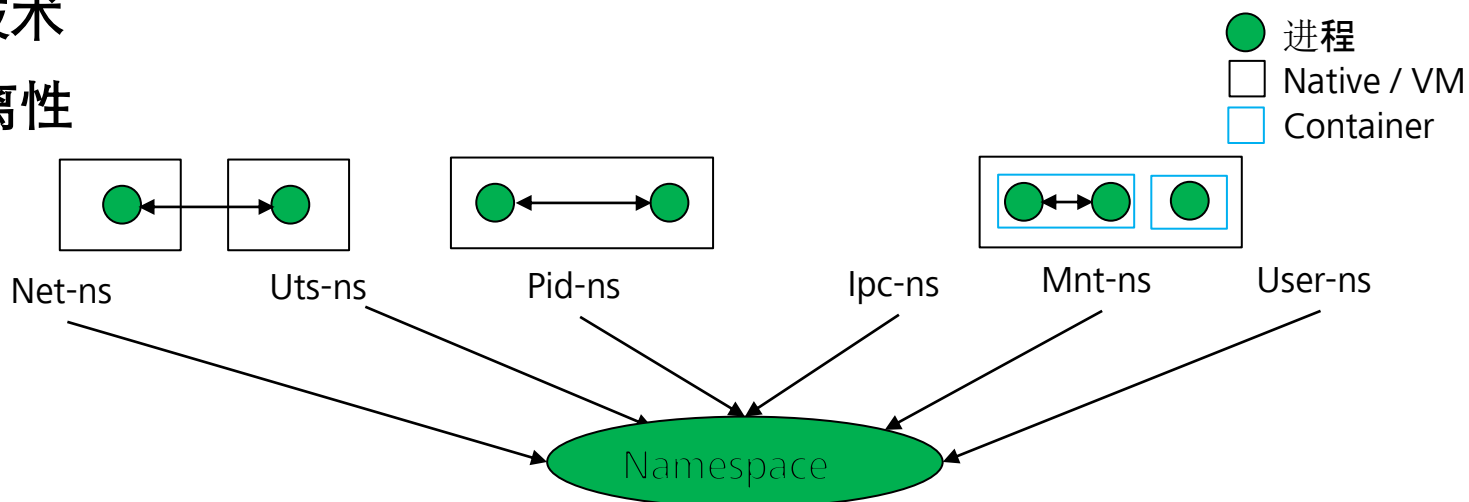
# 背景-生态圈



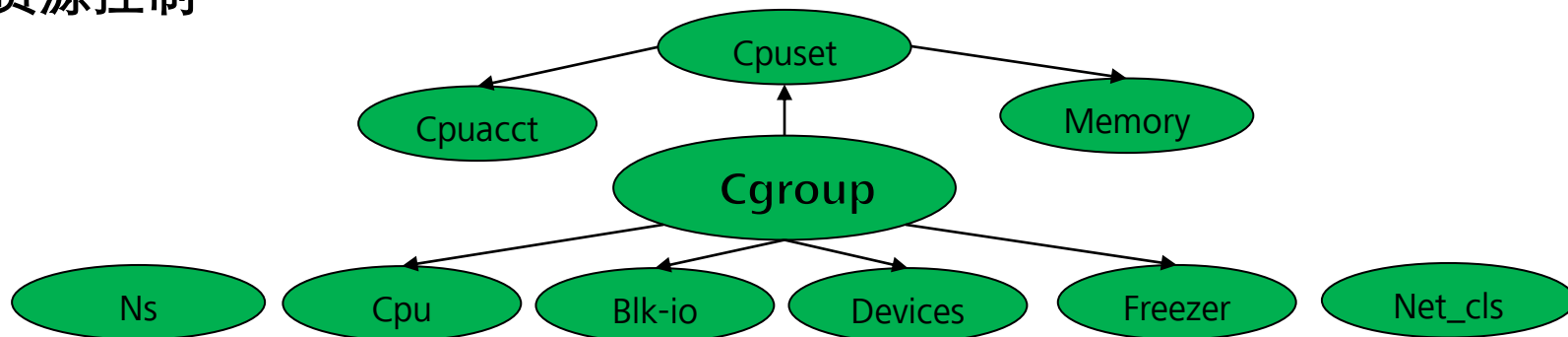
# 原理-基础

## 内核技术

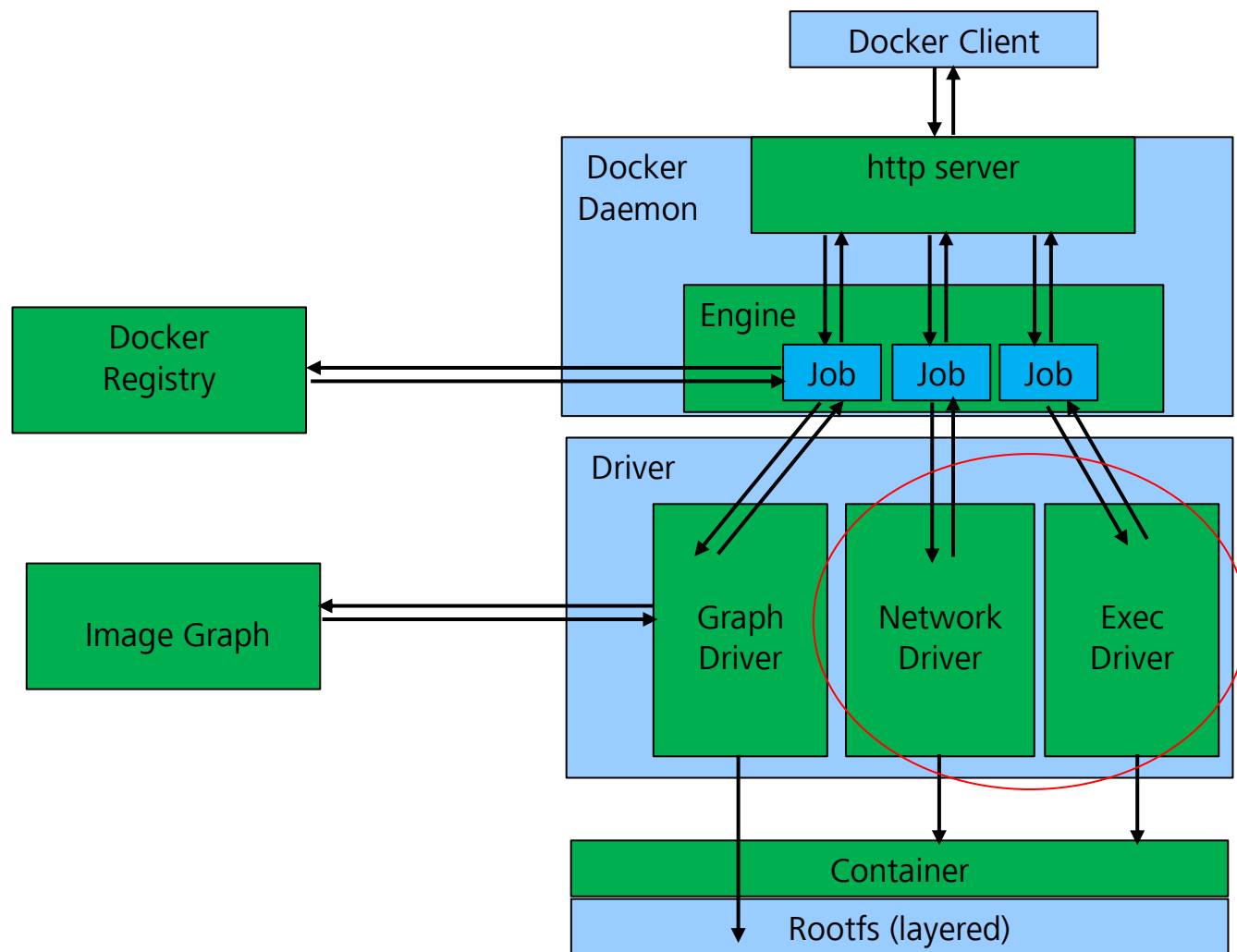
### ➤ 隔离性



### ➤ 资源控制



# 原理-流程



# 基础-容器

➤ 容器：独立运行一个或一组进程的运行态环境

进程：环境和资源隔离的核心对象

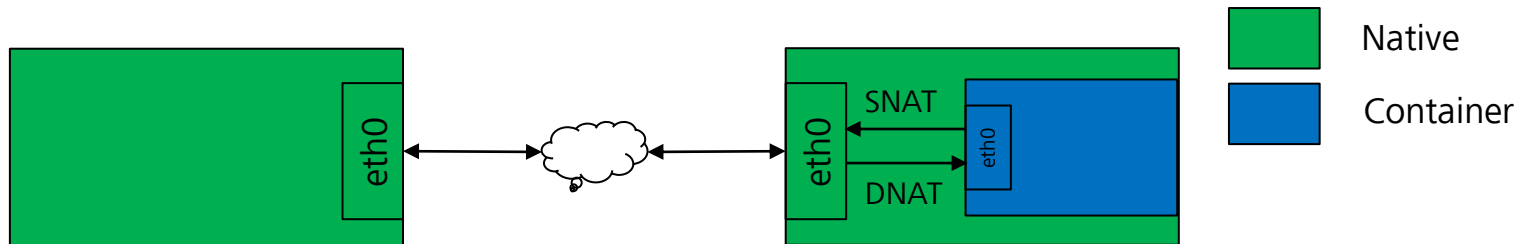
➤ 操作

运行 `docker run -d -args image:tag $Command`

访问： `docker exec -ti $ContainerID $ Command`

启停： `docker start / stop / rm $ContainerID`

➤ 通信

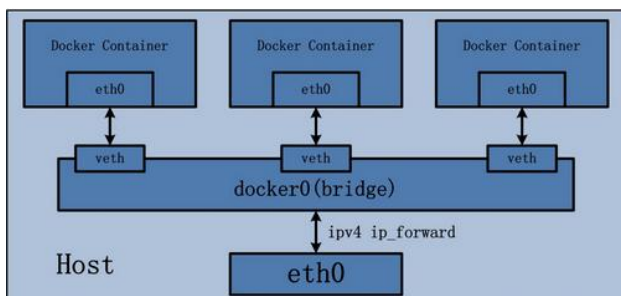




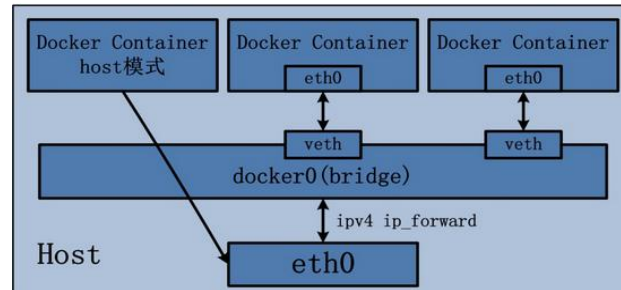
# 基础-网络

## 网络模式

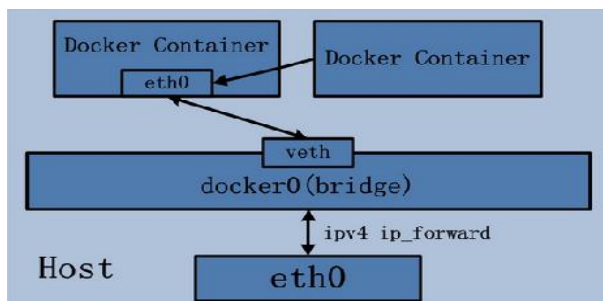
**docker run -net=\$arg Repository:tag \$Command**



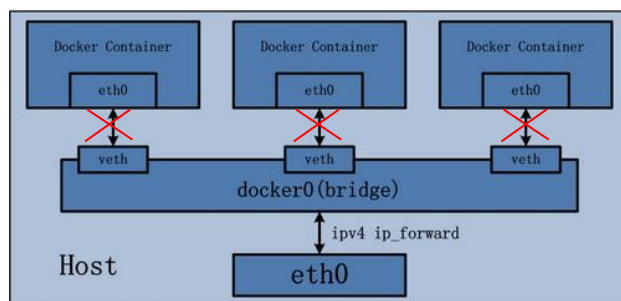
arg=bridge



arg=host



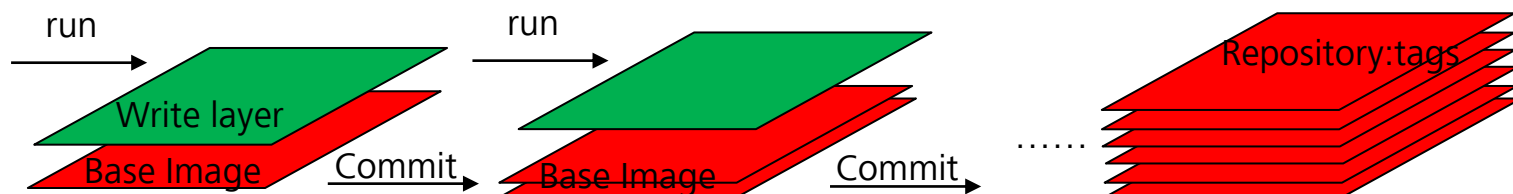
arg=container:\$ContainerID



arg=None

# 基础-镜像

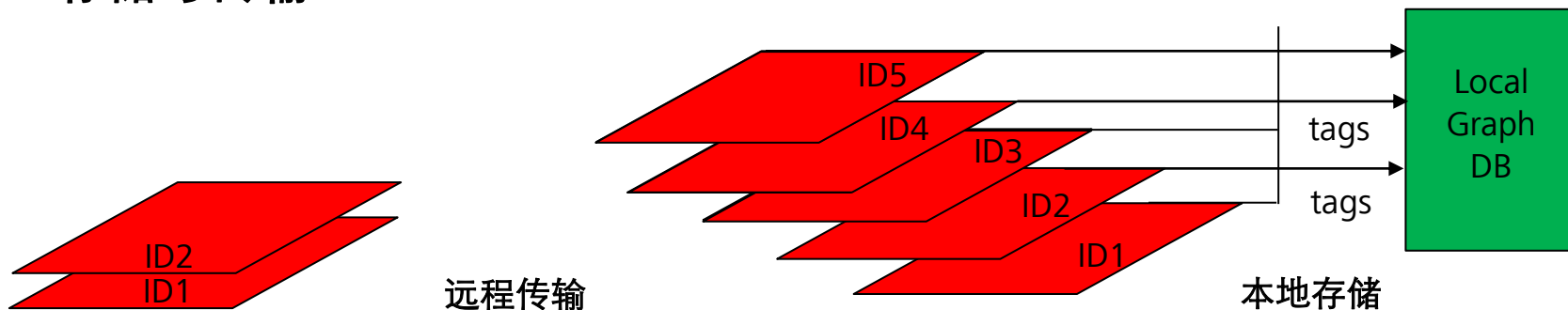
- 镜像: 提供进程可执行的环境操作类集合



**docker commit \$ContainerID Repository:tag**

**docker build -t Repository:tag .** (.表示Dockerfile所在目录)

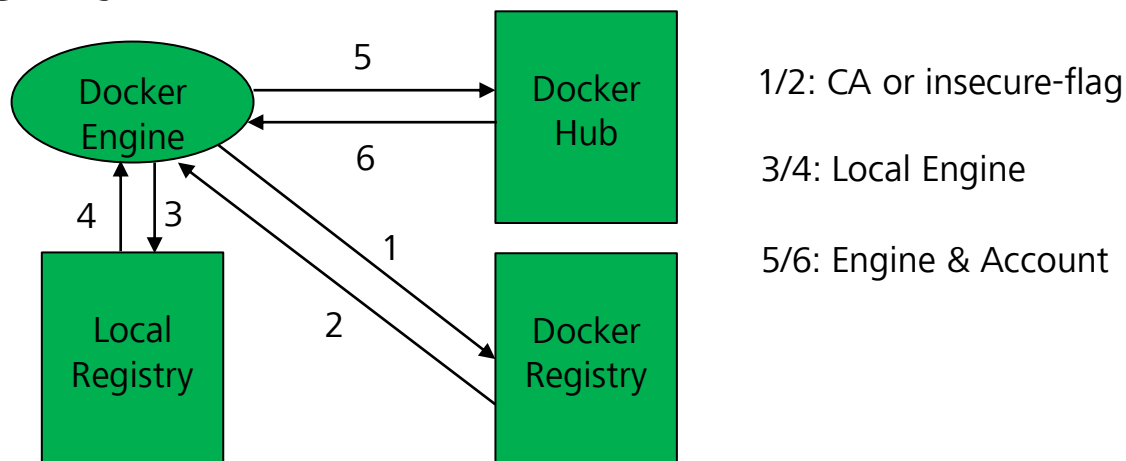
- 存储与传输



# 基础-仓库

## ➤ Docker 仓库

- Hub
- Public /Private registry



**docker push / pull Repository:tag**

**docker push 10.67.164.106:5001/test/suse:latest**

# 基础-资源控制

## ➤ CPU配额

指定使用核

`docker run -d --cpuset=0 10.67.164.106:5001/test/limit:v2.0` #指定运行核数

## • 指定cpu占用份额(share)

`docker run -d -c=1024 10.67.164.106:5001/test/limit:v2.0` (share 2/3)

`docker run -d -c=512 10.67.164.106:5001/test/limit:v2.0` (share 1/3)

## ➤ 内存控制

`docker run -d -m 128m 10.67.164.106:5001/library/oss3.0:mgrnode`  
`/usr/sbin/sshd -D`

# 实践1-demo

## ➤ 制作镜像 (docker file)

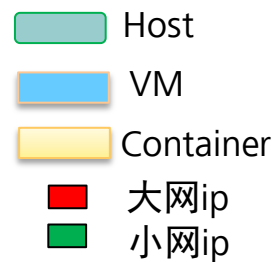
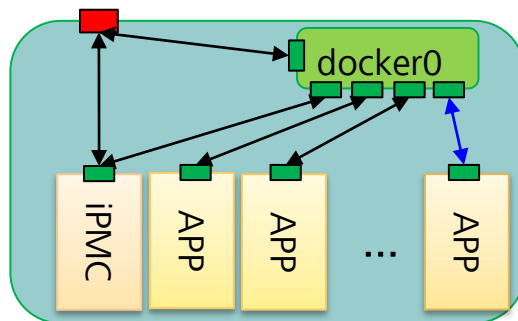
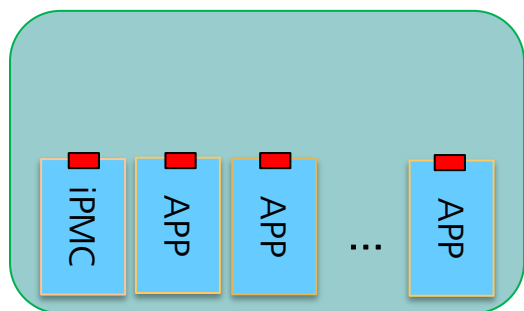
FROM 10.67.164.106:5001/bsp/oss3.0:new	#援引基础镜像
ADD python2.6.8 /opt/	#拷贝文件到容器
WORKDIR /opt	#指定后续命令工作目录
RUN ["/bin/bash", "-c", "/opt/install.sh"]	#执行脚本或命令
ENTRYPOINT /usr/sbin/sshd -D	#指定启动进程

## ➤ 部署私有仓库

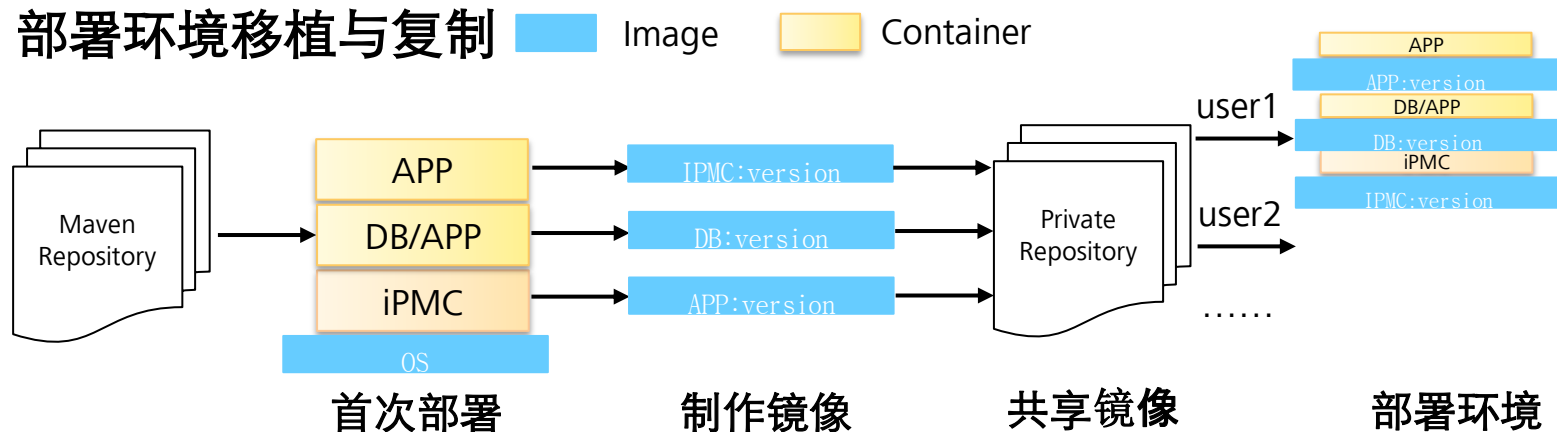
```
docker pull 10.67.164.106:5001/library/registry:v3.0 #拉取私有仓库服务镜像
docker run -d -p $server_port:5000 --restart=always --name registry
10.67.164.106:5001/library/registry:v3.0 #运行私有仓库服务
docker run -d -p 8090:8080 -e REG1=http://$server_ip:$server_port/v1/
10.67.164.106:5001/docker-registry-ui #运行私有仓库简易UI 服务
```

# 实践2-类虚拟机

## 部署集群网管-BSP

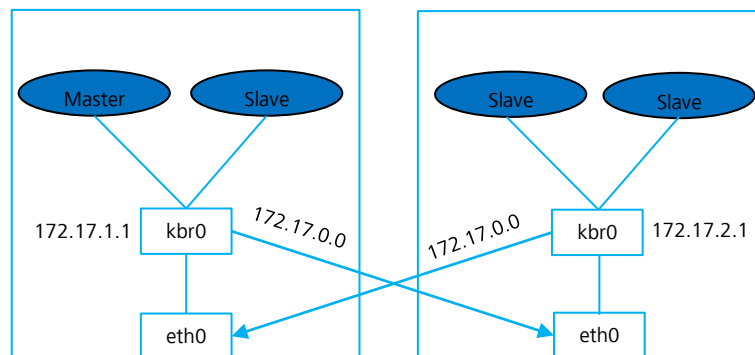
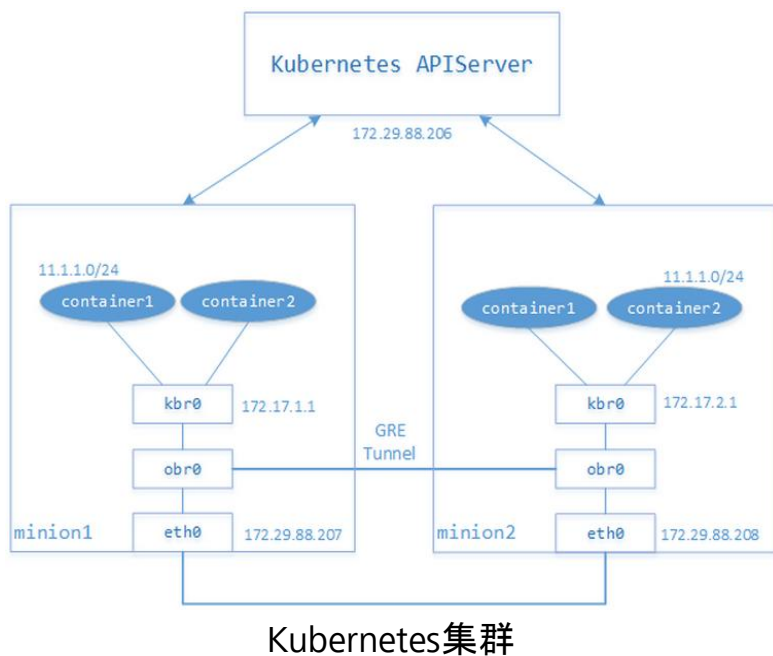


## 部署环境移植与复制



# 实践3-跨机通信

路由表、网络隧道(VPN/OVS/flannel等)



路由添加+iptables

# 实践建议

## ➤ 微服务化

- 单个容器运行单一进程
- 使用挂载或容器卷

## ➤ Dockerfile

- 精简baseimage
- 同类型命令在相同层创建



# 后期工作

- 模拟HA部署场景
  - 网络配置与Native/VM不同
  - NAT映射漂移问题
- 测试环境
  - 功能验证
  - 统一部署与管理

# Thank you

[www.huawei.com](http://www.huawei.com)