

1 docker虚拟化容器技术

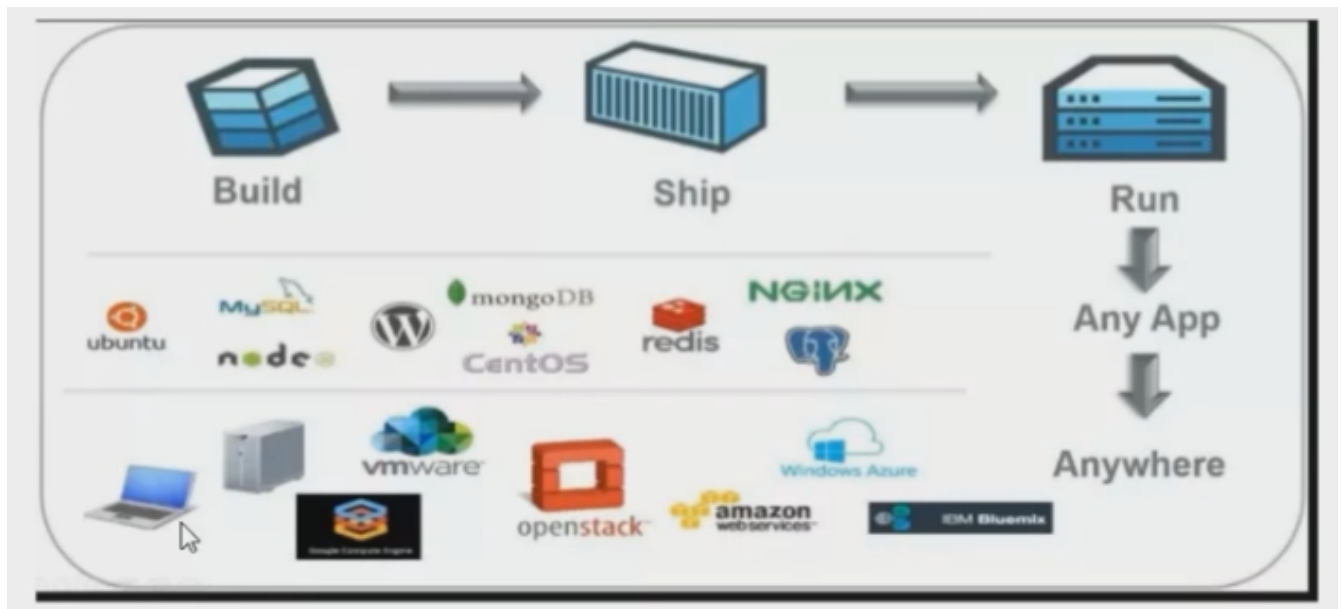
1.1 传统的交付模式

只交付软件，不交付环境

1.2 docker是什么？

一种容器技术 docker基于go语言实现 目标： build , ship and run any app ,anywhere

一次封装，到处运行



Docker是一个开源的引擎，可以轻松的为任何应用创建一个轻量级的、可移植的、自给自足的容器。

开发者在笔记本上编译测试通过的容器可以批量地在生产环境中部署，包括VMs（虚拟机）、bare metal、OpenStack 集群和其他的基础应用平台。

1.3 docker为什么会出现？

开发与运维的鸿沟

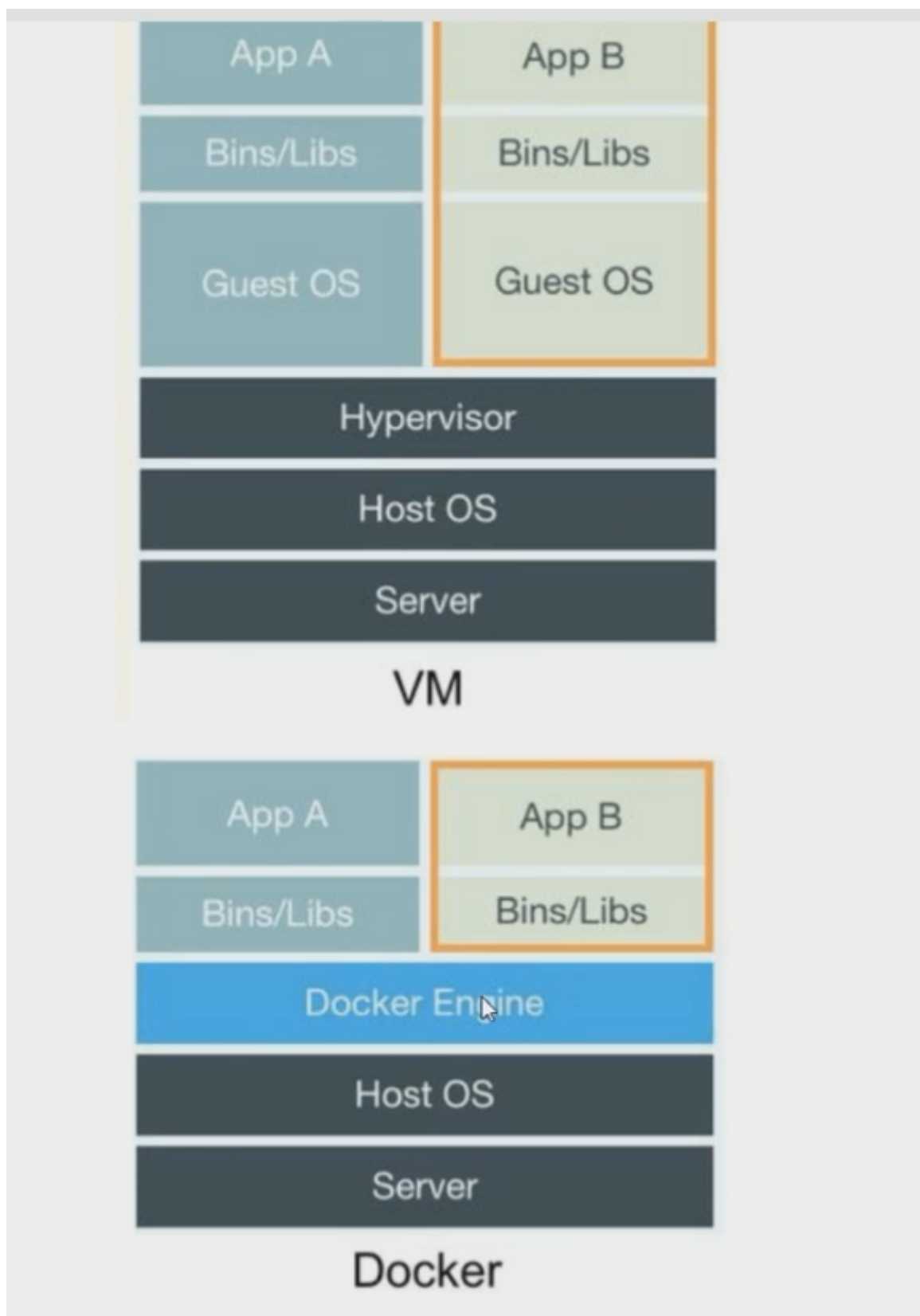
减少运维工作量（100台机器要安装同一个软件）

1.4 docker解决了什么问题？

解决了运行环境和配置问题，方便做持续集成，有助于整体发布

1.5 docker能干什么？

docker与传统虚拟化技术(例如：vmware)的区别？



虚拟化技术就是虚拟了整套环境

缺点：虚拟化技术资源占用多，启动慢，而docker启动秒级的

1.6 docker版本：dockerCE（社区版） DockerEE（企业版）

docker安装：

1. Install required packages.

```
$ sudo yum install -y yum-utils \
  device-mapper-persistent-data \
  lvm2
```

2. Use the following command to set up the stable repository.

```
$ sudo yum-config-manager \
  --add-repo \
  https://download.docker.com/linux/centos/docker-ce.repo
```

3. Install the *latest version* of Docker Engine - Community and containerd

```
$ sudo yum install docker-ce docker-ce-cli containerd.io
```

4. Start Docker.

```
$ sudo systemctl start docker
```

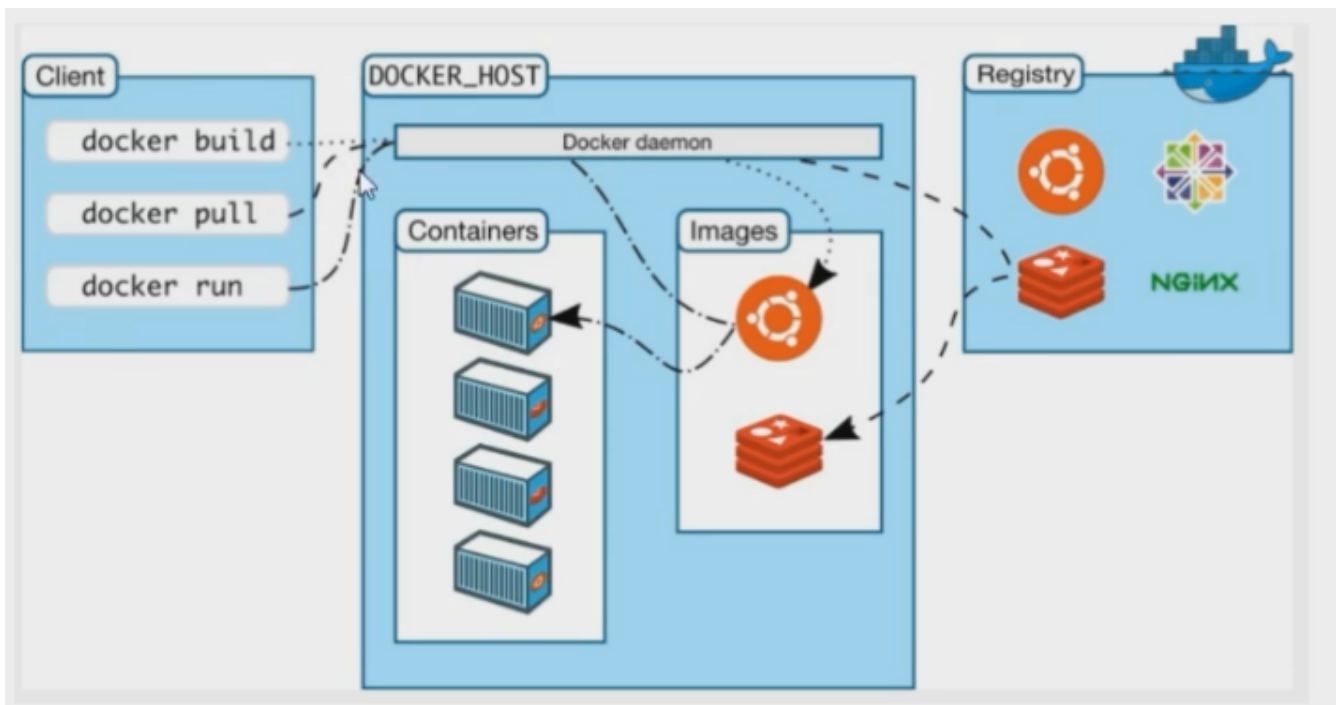
5. docker version

```
$ sudo docker run hello-world
```

6. Verify that Docker Engine - Community is installed correctly by running the hello-world image.

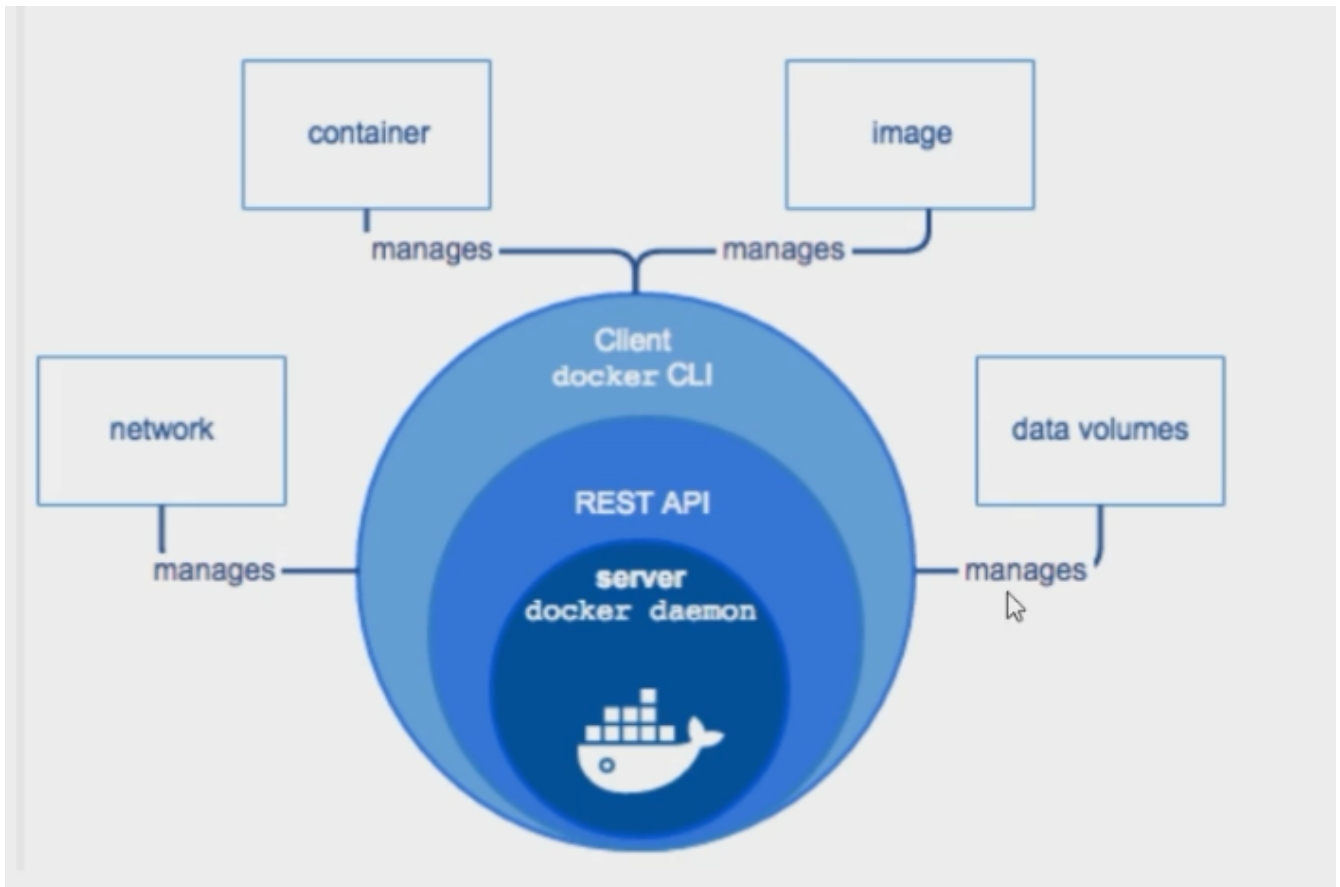
```
$ sudo docker run hello-world
```

1.7 docker架构介绍



docker客户端（控制台）-->docker pull 通过socket通信 调用docker后台进程，先去本地看是否存在redis镜像，若没有就从远程仓库下载下来，再通过 run 命令将镜像做成一个容器（放到集装箱中）并且运行出来

1.8 docker是怎么工作的？



2. vmware 安装 centos7

- VMware安装
- centos7 安装 网路 nat模式
 - ping 127.0.0.1
 - ping www.baidu.com 不通
 - 查看ifconfig
 - cd /etc/sysconfig/network-scripts/
vim ifcfg-ens33

```

BOOTPROTO=static #静态ip
....
ONBOOT=yes #启动时开启网卡
IPV6_PRIVACY=no
IPADDR=192.168.25.101
NETMASK=255.255.255.0
GATEWAY=192.168.25.2
PREFIX=24
DNS1=114.114.114.114
DNS2=8.8.8.8

```

- service network restart 重启网卡
 - xshell 连接不上虚拟机
 - windows - 网络和共享中心 - 更改适配器设置 - vmnet8 - 属性 - IPv4 - 使用指定ip地址与虚拟机ip地址设在同一网段
 - centos7 快照使用
- linux定位
 - 操作系统
 - 适合服务器系统，不适合办公系统
- linux相对windows优点
 - 稳定
 - 安全性高
 - 免费
- linux系统目录介绍
 - / 表示根目录
 - ~ 表示 /root
 - /etc 存放系统配置目录
 - /home 除了root以外，所有用户都会默认在home下新建一个以用户名作为名称的文件夹
用户 xxx 对/home/xxx 具有完全操作权限
 - /root 用户root 单独文件夹
 - /usr 所有用户安装的软件都放入到这个文件夹中
在/usr/local下新建一个tmp。所有的压缩包都上传到tmp中
- linux常用命令
 - pwd
 - cd 进入文件夹
 - cd .. 向上跳一级文件夹
 - cd 路径 进入到指定文件夹
 - mkdir 创建空文件夹
 - ls 平铺 || 详细列表
 - vi 和 vim(高级文本)

- 若文件不存在，带有创建文件功能
 - vi 普通编辑 vim高级编辑（带有颜色）
 - :wq 保存退出 :q 不保存退出 :q! 强制退出
- touch 创建空文件
- cat 查看文件全部内容
- head [-n] 文件名 查看文件 前 n 行，默认前10行
- tail [-n] 文件名 查看文件 后 n 行，默认前10行
 - tailf 动态显示文件后n行内容,常用在显示tomcat日志文件功能
- echo '内容' >> 文件名 向文件中添加一些内容
- ifconfig 打印网卡信息
- reboot 重启
- tar zxvf 文件名 解压缩
- cp [-r] 原文件 新文件路径 复制文件
 - -r 表示复制文件夹
- ctrl + c 中断
- mv 原文件 新文件 剪切 且具备重命名功能
- rm [-r] [-f] 文件名 删除
 - -r 删除目录
 - -f 强制删除，不需要确认
- clear 清屏
- 不同用户登陆
 - 普通用户 \$


```
[tomcat@server-661fed9e-dae2-423f-87f8-7649d7d4f7da ~]$
```
 - root 用户 #


```
[root@xingfei local]#
```
- linux所有需要写路径的地方支持两种写法：
 - 全路径：以 / 开头
 - 相对路径：以当前资源一步一步寻找其他资源过程

3. 客户端工具 XSHELL

centos 默认只开启22端口 其他端口都被防火墙拦截

4.filezilla 工具使用

- FTP协议
 - File transfer Protocal
 - 互联网中,计算机与计算机之间传输文件需要遵守的协议
- filezilla
 - 基于FTP协议的客户端工具

- 使用filezilla可以快速完成文件传输功能

5. JDK安装

- windows下jdk安装
 - 配置 JAVA_HOME
 - 配置 PATH --》指向bin
 - 配置 classpath dt.jar tools.jar (jdk1.7后不用配) --》为了使用javac java等命令
- linux下jdk安装
 - 压缩包上传至服务器/usr/local/tmp
 - 解压压缩包
 - tar zxvf jdk的tar包
 - 将解压后的文件复制到/usr/local/jdk8
 - cp -r 文件名 /usr/local/jdk8
 - 配置环境变量，执行命令
 - vim /etc/profile
 - 修改文件内容
 - linux分割使用冒号
 - \$ 变量名表示引用
 - export 表示（不存在则）添加或（存在则）修改变量
 - #export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL
 - export JAVA_HOME=/usr/local/jdk8
 - export PATH=\$JAVA_HOME/bin:\$PATH
 - export CLASSPATH=.:\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar
 - 解析文件或重启系统
 - source /etc/profile
 - 验证配置是否成功
 - java -version
 - javac Hello.java 编译
 - java Hello 运行Hello.class

6.TOMCAT 配置

- tomcat压缩包上传至服务器/usr/local/tmp
 - tar zxvf apache-tomcat-8.0.43.tar.gz
 - cp -r 文件名 /usr/local/tomcat
- 配置环境变量 TOMCAT_HOME 和 CATALINA_HOME 两个路径一样
 - catalina是tomcat的实例服务也叫主服务
 - vim /etc/profile
 - export TOMCAT_HOME=/usr/local/tomcat
 - export CATALINA_HOME=/usr/local/tomcat
- 解析文件或重启系统

- source /etc/profile
- cd /usr/local/tomcat/bin
 - startup.bat bat 是windows可执行文件的扩展名
 - startup.sh sh是 linux系统 可执行文件的扩展名 绿的都是可执行文件
 - 怎么启动 ./启动 ./startup.sh ./shutdown.sh
- 访问 虚拟机 ip: 8080 端口 失败
 - 防火墙默认只放开22端口, 解决: 1关闭防火墙 2 修改防火墙, 放行8080端口
 - vim /etc/sysconfig/iptables
 - iptables 防火墙配置文件名称
 - sysconfig 目录下的都是系统服务
 - 放行8080端口
 - -A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT

8080-9090 从8080到9090全放行
 - 重启服务

固定语法 service 服务名 restart

 - service iptables restart 重启
 - service iptables start 启动
 - service iptables stop 停止
 - 启动tomcat 进入tomcat/bin
 - ./startup.sh 直接启动
 - ./startup.sh & tailf /usr/local/tomcat/logs/catalina.out 启动并动态打印启动信息
- 在CentOS 7或RHEL 7或Fedora中防火墙由firewalld来管理
 - 要添加范围例外端口 如 1000-2000

语法: 启用区域端口和协议组合

端口可以是一个单独的端口 或者是一个端口范围 -

协议可以是 tcp 或 udp

 - firewall-cmd --zone=public --add-port=80/tcp --permanent (--permanent永久生效, 没有 此参数重启后失效)
 - firewall-cmd --zone=public --add-port=1000-2000/tcp --permanent
 - 重新载入 firewall-cmd --reload
 - 查看 firewall-cmd --zone=public --query-port=80/tcp
 - 删除 firewall-cmd --zone=public --remove-port=80/tcp --permanent
- 还原传统的管理方式
 - 执行一下命令:
 - systemctl stop firewalld systemctl mask firewalld
 - 安装iptables-services
 - yum install iptables-services

- 设置开机启动 `systemctl enable iptables`

`systemctl stop iptables` `systemctl start iptables` `systemctl restart iptables` `systemctl reload iptables`

- 保存设置

- `service iptables save`