

云计算部署与管理

NSD CLOUD

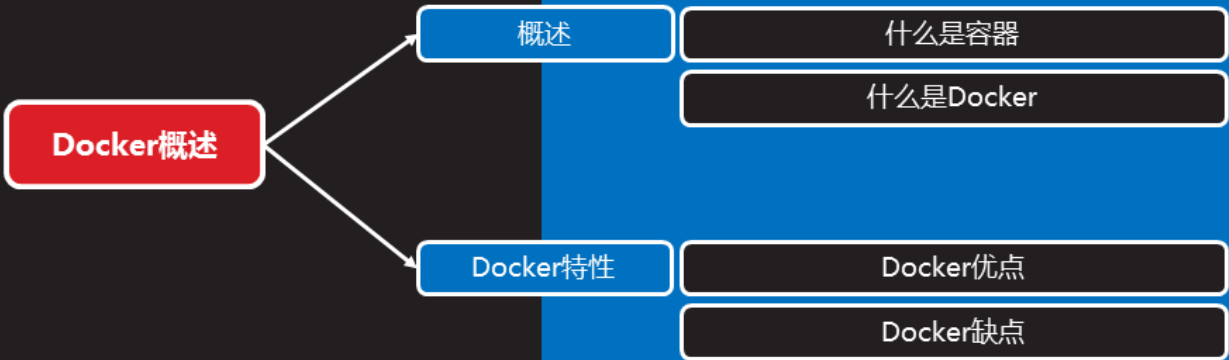
DAY05

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	Docker概述
	10:30 ~ 11:20	部署Docker
	11:30 ~ 12:00	Docker镜像
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	Docker基本命令
	16:10 ~ 17:10	
	17:20 ~ 18:00	总结和答疑



Docker概述



概述

什么是容器

- 容器技术已经成为应用程序封装和交付的核心技术
- 容器技术的核心有以下几个内核技术组成：
 - Cgroups (Control Groups) -资源管理
 - NameSpace-进程隔离
 - SELinux安全
- 由于是在物理机上实施隔离，启动一个容器，可以像启动一个进程一样快速

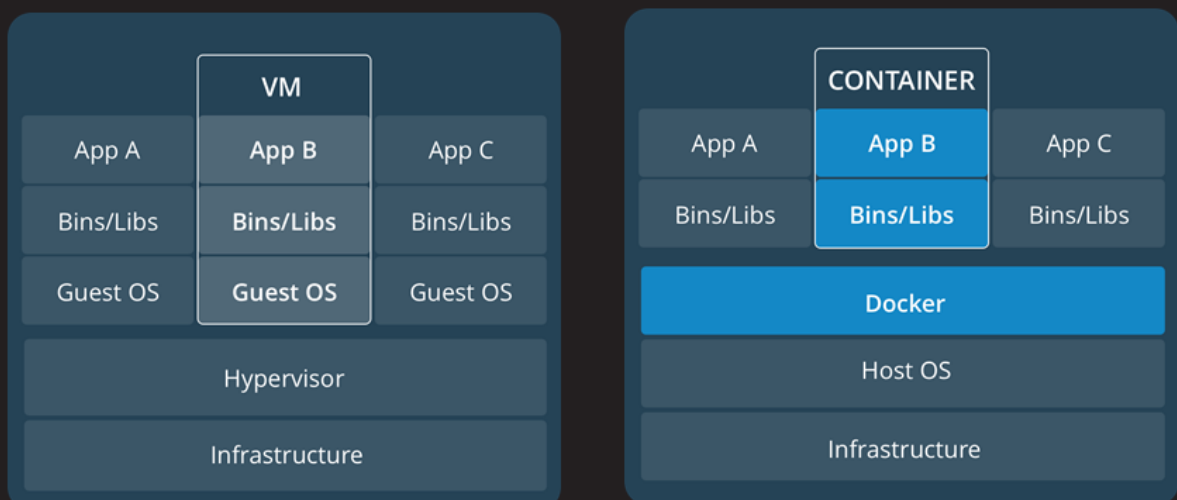


Docker特性

Docker优点

- 相比于传统的虚拟化技术，容器更加简洁高效
- 传统虚拟机需要给每个VM安装操作系统
- 容器使用的共享公共库和程序

知识讲解



Docker缺点

知识讲解

- 容器的隔离性没有虚拟化强
- 共用Linux内核，安全性有先天缺陷
- SELinux难以驾驭
- 监控容器和容器排错是挑战



部署Docker

部署Docker

安装软件

安装前准备

安装Docker

安装软件



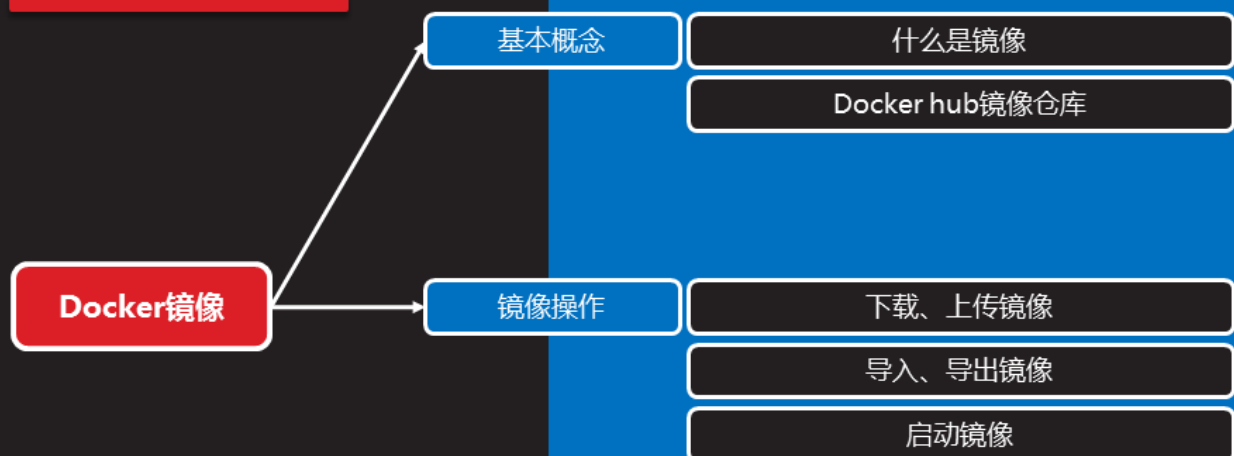
安装前准备

- 需要64位操作系统
- 至少RHEL6.5以上的版本，强烈推荐RHEL7
- 关闭防火墙（不是必须）

知识讲解



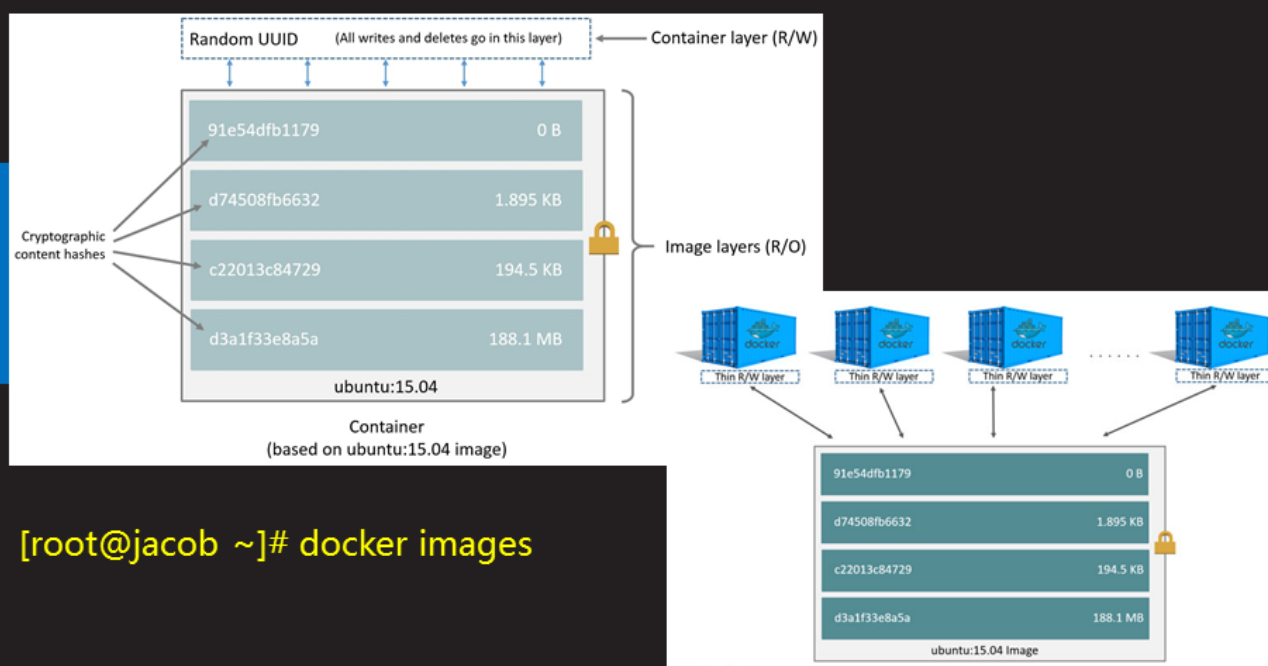
Docker镜像



基本概念

什么是镜像 (续1)

知识讲解



```
[root@jacob ~]# docker images
```



镜像操作

下载、上传镜像

- 下载镜像（从镜像仓库中下载镜像）

```
[root@server0 ~]# docker help pull
docker pull [OPTIONS] NAME[:TAG]@DIGEST
```

```
[root@server0 ~]# docker pull rhel7
```

- 上传镜像（上传镜像到仓库）

```
[root@server0 ~]# docdocker help push
Usage: docker push [OPTIONS] NAME[:TAG]
```

```
[root@server0 ~]# docker push rhel7
```



导入、导出镜像

知识讲解

- 导入镜像（通过tar包文件导入镜像）

```
[root@server0 ~]# docker load < xx.tar  
[root@server0 ~]# docker images
```

- 导出镜像（将本地镜像导出为tar文件）

```
[root@server0 ~]# docker images  
[root@server0 ~]# docker save image_name > xx.tar
```



启动镜像

知识讲解

- 启动centos镜像生成一个容器

```
[root@server0 ~]# docker images  
[root@server0 ~]# docker run -it centos bash
```

- 开启另一个终端（查看容器信息）

```
[root@server0 ~]# docker ps
```



案例2：镜像基本操作

课堂练习

1. 导入镜像
2. 导出镜像
3. 启动镜像



Docker基本命令

Docker基本命令

镜像常用命令

命令列表

docker images

docker history

docker inspect

docker rmi

docker save|load

docker tag

命令列表

docker run

docker ps

docker stop|start|restart

docker attach|exec

docker inspect

docker top

docker rm

容器常用命令

命令列表

知识讲解

- 命令列表
 - **docker images** //查看镜像列表
 - **docker history** //查看镜像制作历史
 - **docker inspect** //查看镜像底层信息
 - **docker pull** //下载镜像
 - **docker push** //上传镜像



命令列表（续1）

知识讲解

- 命令列表
 - **docker rmi** //删除本地镜像
 - **docker save** //镜像另存为tar包
 - **docker load** //使用tar包导入镜像
 - **docker search** //搜索镜像
 - **docker tag** //修改镜像名称和标签



docker images

知识讲解

- 查看镜像列表
 - 镜像仓库名称
 - 镜像标签
 - 镜像ID
 - 创建时间
 - 大小

```
[root@jacob ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
centos	latest	980e0e4c79ec	8 months ago	196.7 MB



docker history

知识讲解

- 查看镜像历史
 - 了解镜像制作过程
 - 详细参考后面的dockerfile内容

```
[root@jacob ~]# docker history centos
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
980e0e4c79ec	8 months ago	/bin/sh -c #(nop) CMD ["/bin/bash"]	0 B	
<missing>	8 months ago	/bin/sh -c #(nop) LABEL name=CentOS Base Ima	0 B	
<missing>	8 months ago	/bin/sh -c #(nop) ADD file:e336b45186086f7d9d	196.7 MB	
<missing>	8 months ago	/bin/sh -c #(nop) MAINTAINER https://github.	0 B	



docker save|load

- 保存本地镜像另存为tar文件
 - 方便其他人使用tar包导入镜像

知识讲解

```
[root@jacob ~]# docker save centos > centos.tar  
[root@jacob ~]# ls centos.tar  
centos.tar
```

- 使用tar包文件导入镜像

```
[root@jacob ~]# docker load < centos.tar
```



容器常用命令

命令列表

知识讲解

- 命令列表
 - **docker run** //运行容器
 - **docker ps** //查看容器列表
 - **docker stop** //关闭容器
 - **docker start** //启动容器
 - **docker restart** //重启容器
 - **docker attach|exec** //进入容器
 - **docker inspect** //查看容器底层信息
 - **docker top** //查看容器进程列表
 - **docker rm** //删除容器



docker run

- 使用镜像启动容器

知识讲解

```
[root@jacob ~]# docker run -it centos bash
[root@bbb51ac87b08 /]#exit

[root@jacob ~]# docker run -itd centos bash
b8f218f2341c12655e6092d7d2e7fd5229824fdefce84075fb3a9569ebf82079
[root@jacob ~]# docker ps
```



docker ps

• 列出容器列表

- docker ps 查看正在运行的容器
- docker ps -a 查看所有容器列表
- docker ps -aq 仅显示容器id

知识讲解

```
[root@server0 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b8f218f2341c	centos	"bash"	9 minutes ago	Up 8 minutes		jovial_pasteur



docker stop|start|restart

• 管理容器

- docker stop 关闭容器
- docker start 开启容器
- docker restart 重启容器

知识讲解

```
[root@server0 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b8f218f2341c	centos	"bash"	9 minutes ago	Up 8 minutes		jovial_Pasteur

```
[root@server0 ~]# docker stop b8f
```



