

# 云计算部署与管理

**NSD CLOUD**

**DAY04**

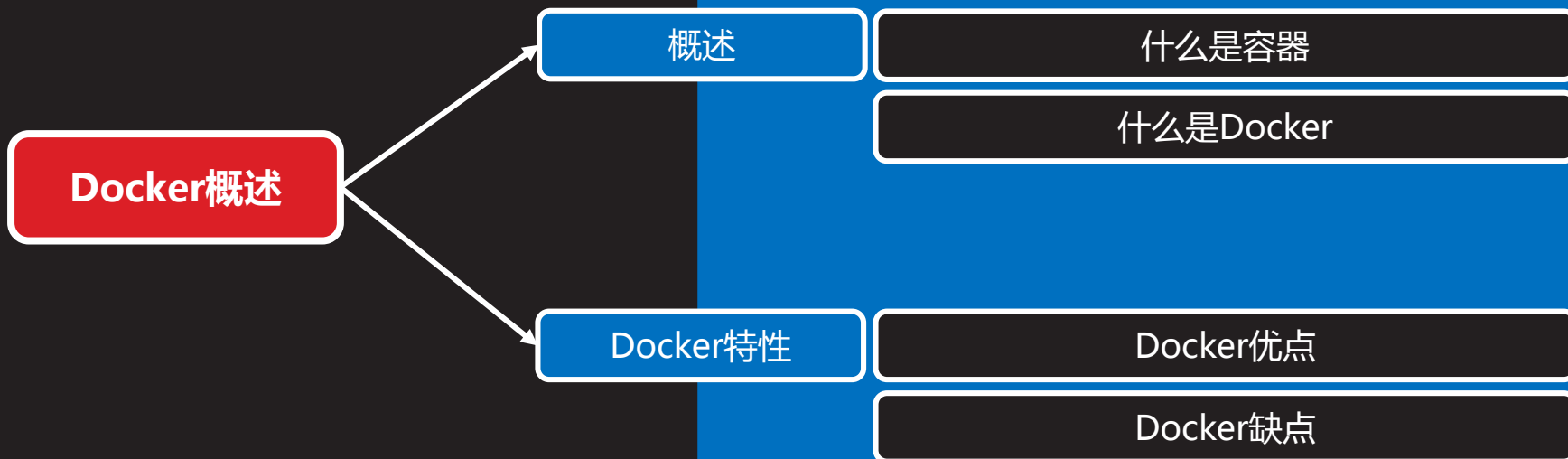
# 内容

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



# 什么是Docker

- Docker是完整的一套容器管理系统
- Docker提供了一组命令，让用户更加方便直接地使用容器技术，而不需要过多关心底层内核技术

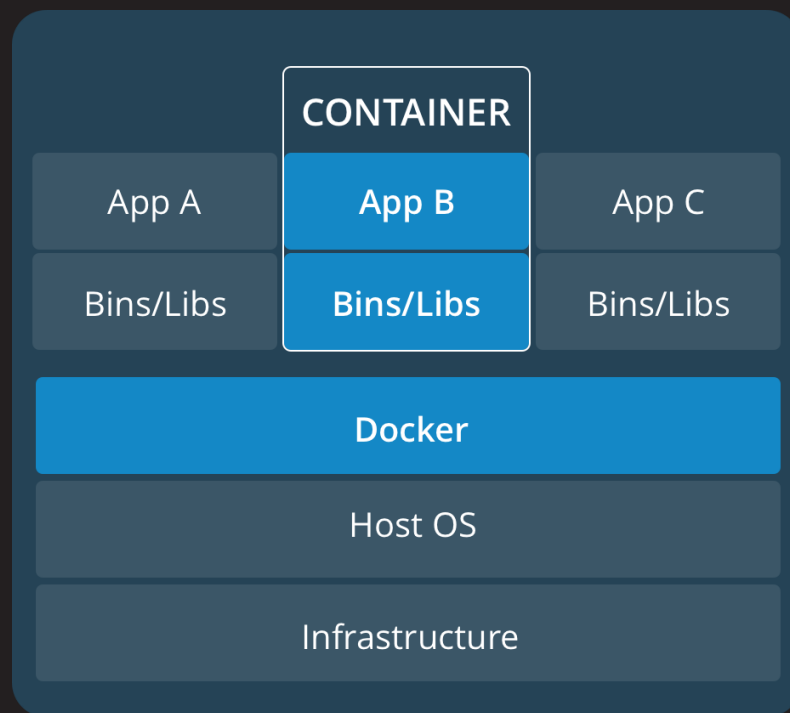
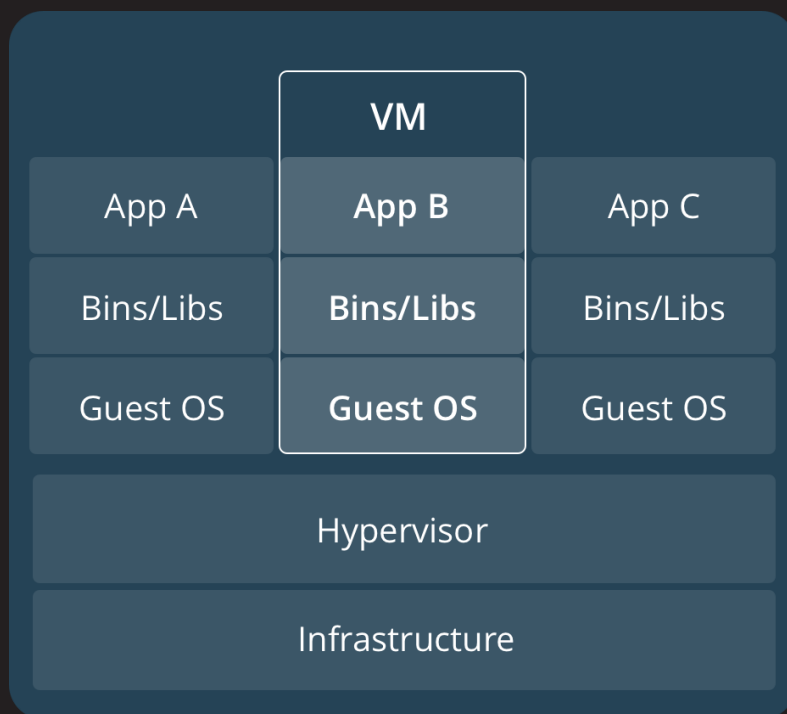


# Docker特性



# 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





# Docker缺点

- 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）



# 安装Docker

- 配置 yum 源
  - docker 软件包位于光盘 RHEL7-extras.iso 中
  - 在物理机的 ftp 目录下创建文件夹 extras

```
mkdir /var/ftp/extras
```
  - 把光盘挂到这个目录下

```
mount -t iso9660 -o ro,loop RHEL7OSP-extras.iso /var/ftp/rhelosp
```



# 安装Docker

- 在虚拟机中配置 yum 源，指向刚刚创建的目录：

```
[extras]
```

```
name=CentOS-$releasever - Extras
```

```
baseurl=ftp://192.168.1.254/extras
```

```
enabled=1
```

```
gpgcheck=0
```

- 软件包安装

```
[root@docker1 ~]# yum -y install docker
```

```
[root@docker1 ~]# systemctl restart docker
```

```
[root@docker1 ~]# systemctl enable docker
```

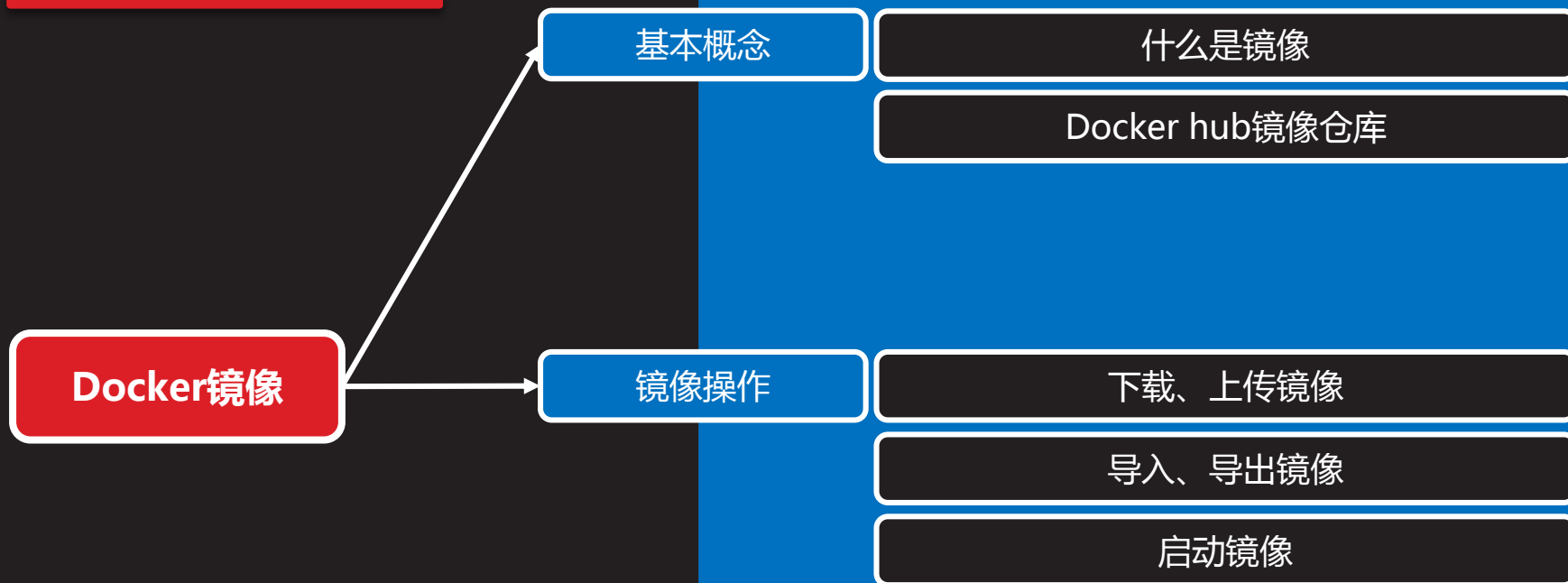


# 案例1：安装Docker

1. 准备两台虚拟机，IP 为192.168.1.31 和 192.168.1.32
2. 安装docker
3. 关闭防火墙



# Docker镜像



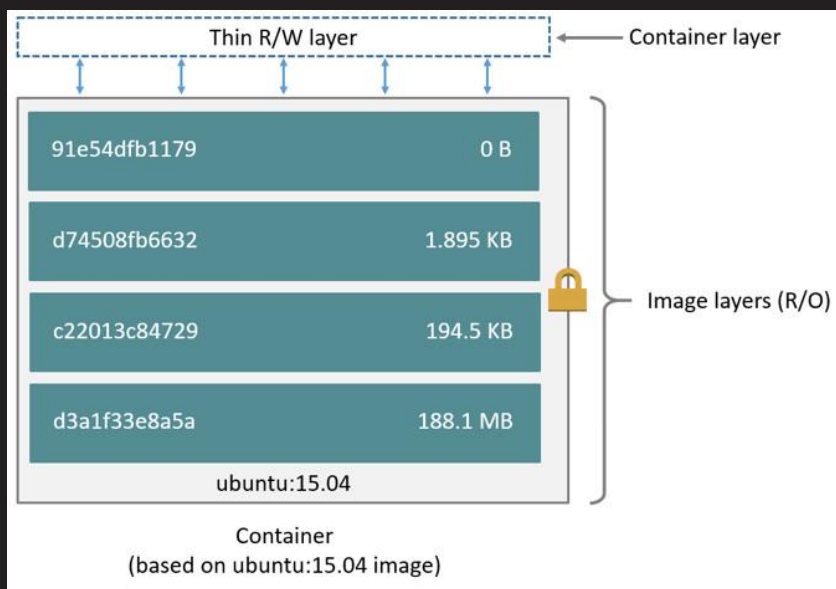


# 基本概念

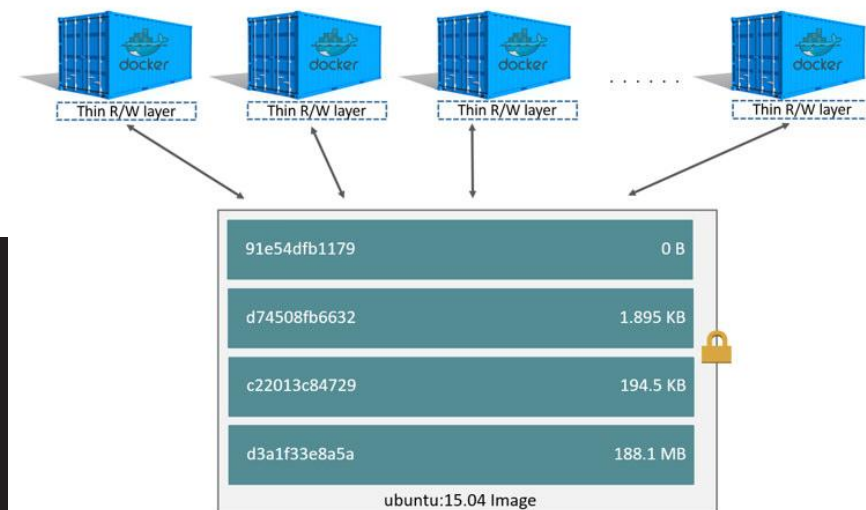
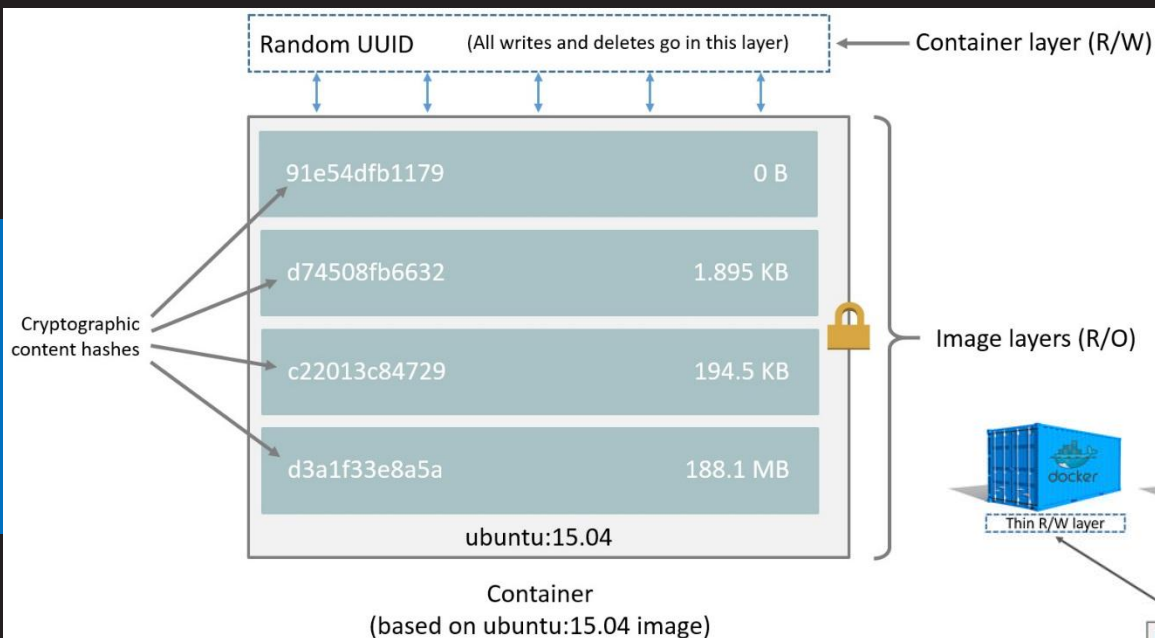
---

# 什么是镜像

- 在Docker中容器是基于镜像启动的
- 镜像是启动容器的核心
- 镜像采用分层设计
- 使用快照的COW技术，确保底层数据不丢失



# 什么是镜像（续1）



```
[root@docker1 ~]# docker images
```

# Docker hub镜像仓库

- <https://hub.docker.com>
- Docker官方提供公共镜像的仓库(Registry)

```
[root@docker1 ~]# docker search busybox  
[root@docker1 ~]# docker search centos  
[root@docker1 ~]# docker search nginx
```



# 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@docker1 ~]# docker help pull
docker pull [OPTIONS] NAME[:TAG|@DIGEST]
```

```
[root@docker1 ~]# docker pull docker.io/busybox
```

- 上传镜像（上传镜像到仓库）

```
[root@docker1 ~]# docdocker help push
Usage: docker push [OPTIONS] NAME[:TAG]
```

```
[root@docker1 ~]# docker push docker.io/busybox
```



# 导入、导出镜像

- 导出镜像（将本地镜像导出为tar文件）

```
[root@docker1 ~]# docker save docker.io/busybox:latest -o
busybox.tar
[root@docker1 ~]# ls
```

- 导入镜像（通过tar包文件导入镜像）

```
[root@docker1 ~]# scp nginx.tar root@192.168.1.31:/root
[root@docker2 ~]# ls
nginx.tar
[root@docker2 ~]# docker load -i nginx.tar
[root@docker2 ~]# docker images
```



# 启动镜像

- 启动centos镜像生成一个容器

```
[root@docker1 ~]# docker images
```

```
[root@docker1 ~]# docker run -it docker.io/centos:latest /bin/bash
```

- 开启另一个终端（查看容器信息）

```
[root@docker1 ~]# docker ps
```





## 案例2：镜像基本操作

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 top

docker rm

# 镜像常用命令

---

# 命令列表

- 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）

- 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

- 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@docker1 ~]# docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/busybox	latest	d8233ab899d4	10 days ago	1.199 MB



# docker history

- 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@docker1 ~]# docker history docker.io/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 inspect

- 查看镜像底层信息
  - 了解镜像环境变量、存储卷、标签等信息

```
[root@docker1 ~]# docker inspect docker.io/centos
```

```
... ..
{
  "Id": "sha256:980e0e4c79ec933406e467a296ce3b86685e6b42eed2f873745e6a91d718e37a",
  "RepoTags": [
    "docker.io/centos:latest"
  ],
  "RepoDigests": [],
  "Parent": "",
  "Comment": "",
  "Created": "2016-09-06T21:10:20.397787682Z",
  "Container": "37446a1771cbec3e85b76d9159fd6a5a92743655cb92a65661e8a174bad81c7e "
  ... ..
}
```





# docker rmi

- 删除本地镜像
  - 注意：启动容器时删除镜像会提示错误

```
[root@docker1 ~]# docker rmi docker.io/centos
Error response from daemon: conflict: unable to remove repository reference "
docker.io/centos" (must force) - container 3daba799135c is using its referenced
image 980e0e4c79ec
```



# 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

```
[root@docker1 ~]# docker save docker.io/busybox:latest -o busybox.tar  
[root@docker1 ~]# ls busybox.tar  
busybox.tar
```

- 使用tar包文件导入镜像

```
[root@docker1 ~]# docker load -i busybox.tar
```



# docker tag

- 重命名镜像名称（复制）

```
[root@docker1 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/centos	latest	42b4762643dc	4 weeks ago	109.2 M
	e1ddd7948a1c	4 weeks ago	1.163 MB	

```
[root@docker1 ~]# docker tag docker.io/centos:latest cen:v1
```

```
[root@docker1 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
cen	v1	76d6bc25b8a5	4 months ago	199.7 MB



# 容器常用命令

---

# 命令列表

- 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

```
[root@docker1 ~]# docker run -it docker.io/centos:latest bash  
[root@bbb51ac87b08 /]#exit
```

```
[root@docker1 ~]# docker run -itd docker.io/centos:latest bash  
b8f218f2341c12655e6092d7d2e7fd5229824fdefce84075fb3a9569ebf82079  
[root@docker1 ~]# 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

[root@docker1 ~]# docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b8f218f2341c	docker.io/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

```
[root@docker1 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
B8f218f2341c	docker.io/centos	"bash"	9 minutes ago	Up 8 minutes		jovial_Pasteur

```
[root@docker1 ~]# docker stop b8f
```





# docker attach|exec

## • 进入容器

– docker attach

进入容器，exit会导致容器关闭

– docker exec

进入容器，退出时不会关闭容器

```
[root@docker1 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
B8f218f2341c	docker.io/centos	"bash"	9 minutes ago	Up 8 minutes		jovial_Pasteur
7e4e51117c40	docker.io/centos	"bash"	5 seconds ago	Up 2 seconds		pedantic_bard

```
[root@docker1 ~]# docker attach b8f
```

```
[root@docker1 ~]# docker exec -it 7e4e bash
```



# docker inspect

- 查看容器底层信息

[root@docker1 ~]# docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
B8f218f2341c	docker.io/centos	"bash"	9 minutes ago	Up 8 minutes		jovial_Pasteur
7e4e51117c40	docker.io/centos	"bash"	5 seconds ago	Up 2 seconds		pedantic_bard

[root@docker1 ~]# docker inspect 7e4e

```
"Id": "0cc385eb6f63373397e678347890914347dc8efa3cefd809c1ed8c157651a261",
"Gateway": "172.17.0.1",
"IPAddress": "172.17.0.2",
"MacAddress": "02:42:ac:11:00:02"
```



# docker top

- 查看容器进程列表

```
[root@docker1 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b8f218f2341c	docker.io/centos	"bash"	9 minutes ago	Up 8 minutes		jovial_Pasteur
7e4e51117c40	docker.io/centos	"bash"	5 seconds ago	Up 2 seconds		pedantic_bard

```
[root@docker1 ~]# docker top 7e4e
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	3177	3163	0	21:25	pts/8	00:00:00	/bin/bash



# docker rm

- 删除容器
  - 注意，删除正在运行的容器时会提示错误

```
[root@docker1 ~]# docker rm 7e4e
```

```
[root@docker1 ~]# docker rm $(docker ps -aq)
```



# 案例3：镜像与容器常用指令

1. 镜像常用指令练习
2. 容器常用指令练习



# 总结和答疑

---

总结和答疑

删除容器

问题现象

故障分析及排除

# 删除容器

---

# 问题现象

- 删除容器，提示错误：
  - `docker rm 0cc`

Error response from daemon: You cannot remove a running container

0cc385eb6f63373397e678347890914347dc8efa3cef  
d809c1ed8c157651a261. Stop the container before  
attempting removal or use -f





# 故障分析及排除

- 原因分析
  - 容器没有关闭
- 解决办法
  - 先关闭容器，才能删除容器

