[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLOUD/DAY04/CASE/01/index.html" \l "page_top_case)

# NSD CLOUD DAY04

1. [案例1：安装Docker](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLOUD/DAY04/CASE/01/index.html" \l "case1)
2. [案例2：镜像基本操作](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLOUD/DAY04/CASE/01/index.html" \l "case2)
3. [案例3：镜像与容器常用指令](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLOUD/DAY04/CASE/01/index.html" \l "case3)

## 1 案例1：安装Docker

### 1.1 问题

本案例要求配置yum源并安装Docker：

* 准备两台虚拟机，IP为192.168.1.31和192.168.1.32
* 安装docker
* 关闭防火墙

### 1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置yum源

1）配置IP（虚拟机配置静态ip）docker1和docker2主机同样操作

1. [root@localhost ~]# echo docker1 > /etc/hostname
2. [root@localhost ~]# hostname docker1
3. [root@localhost ~]# echo docker2 > /etc/hostname
4. [root@localhost ~]# hostname docker2
5. [root@docker1 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
6. # Generated by dracut initrd
7. DEVICE="eth0"
8. ONBOOT="yes"
9. IPV6INIT="no"
10. IPV4\_FAILURE\_FATAL="no"
11. NM\_CONTROLLED="no"
12. TYPE="Ethernet"
13. BOOTPROTO="static"
14. IPADDR="192.168.1.31"
15. PREFIX=24
16. GATEWAY=192.168.1.254
17. [root@docker1 ~]# systemctl restart network
18. [root@docker2 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
19. # Generated by dracut initrd
20. DEVICE="eth0"
21. ONBOOT="yes"
22. IPV6INIT="no"
23. IPV4\_FAILURE\_FATAL="no"
24. NM\_CONTROLLED="no"
25. TYPE="Ethernet"
26. BOOTPROTO="static"
27. IPADDR="192.168.1.32"
28. PREFIX=24
29. GATEWAY=192.168.1.254
30. [root@docker1 ~]# systemctl restart network

2）配置yum客户端，配置之前openstack的扩展源(RHEL7-extras.iso)即可（docker1和docker2主机同样操作）

1. [root@docker1 ]# vim /etc/yum.repos.d/rhel.repo
2. ...
3. [local\_extras]
4. name=CentOS-$releasever - Extras
5. baseurl="ftp://192.168.1.254/extras"
6. enabled=1
7. gpgcheck=0
8. [root@docker2 ]# vim /etc/yum.repos.d/rhel.repo
9. ...
10. [local\_extras]
11. name=CentOS-$releasever - Extras
12. baseurl="ftp://192.168.1.254/extras"
13. enabled=1
14. gpgcheck=0

4）安装docker（docker1和docker2主机同样操作）

1. [root@docker1 ~]# yum -y install docker
2. [root@docker1 ~]# systemctl restart docker
3. [root@docker1 ~]# systemctl enable docker
4. [root@docker1 ~]# ifconfig     //有docker0说明环境部署完成
5. docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
6. inet 172.17.0.1 netmask 255.255.0.0 broadcast 0.0.0.0
7. ether 02:42:3e:e7:3f:6e txqueuelen 0 (Ethernet)
8. RX packets 0 bytes 0 (0.0 B)
9. RX errors 0 dropped 0 overruns 0 frame 0
10. TX packets 0 bytes 0 (0.0 B)
11. TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
12. [root@docker1 ~]# docker version        //查看版本
13. [root@docker2 ~]# yum -y install docker
14. [root@docker2 ~]# systemctl restart docker
15. [root@docker2 ~]# systemctl enable docker
16. [root@docker2 ~]# ifconfig     //有docker0说明环境部署完成
17. docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
18. inet 172.17.0.1 netmask 255.255.0.0 broadcast 0.0.0.0
19. ether 02:42:53:82:b9:d4 txqueuelen 0 (Ethernet)
20. RX packets 0 bytes 0 (0.0 B)
21. RX errors 0 dropped 0 overruns 0 frame 0
22. TX packets 0 bytes 0 (0.0 B)
23. TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
24. [root@docker2 ~]# docker version        //查看版本

## 2 案例2：镜像基本操作

### 2.1 问题

本案例要求熟悉镜像的基本操作：

* 导入镜像
* 导出镜像
* 启动镜像

### 2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：docker镜像

1）下载镜像

1. [root@docker1 ~]# docker pull docker.io/busybox
2. [root@docker1 bulid]# docker pull docker.io/busybox
3. Using default tag: latest
4. Trying to pull repository docker.io/library/busybox ...
5. latest: Pulling from docker.io/library/busybox
6. Digest: sha256:061ca9704a714ee3e8b80523ec720c64f6209ad3f97c0ff7cb9ec7d19f15149f

2）上传镜像

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

3）查看镜像

1. [root@docker1 ~]# docker images
2. REPOSITORY TAG IMAGE ID CREATED SIZE
3. docker.io/busybox latest d8233ab899d4 10 days ago 1.199 MB

4）查找busybox镜像

1. [root@docker1 ~]# docker search docker.io/busybox

5）导出busybox镜像为busybox.tar

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

6）导入镜像

1. [root@docker1 ~]# scp nginx.tar root@192.168.1.32:/root
2. [root@docker2 ~]# ls
3. nginx.tar
4. [root@docker2 ~]# docker load -i nginx.tar
5. 3c816b4ead84: Loading layer 58.47 MB/58.47 MB
6. 787822cf1b17: Loading layer 54.44 MB/54.44 MB
7. 89decbdf7fb7: Loading layer 3.584 kB/3.584 kB
8. Loaded image: docker.io/nginx:latest/3.584 kB
9. [root@docker2 ~]# docker images
10. REPOSITORY TAG IMAGE ID CREATED SIZE
11. docker.io/nginx latest 42b4762643dc 4 weeks ago 109.2 M e1ddd7948a1c 4 weeks ago 1.163 MB

7）删除镜像

1. [root@docker1 docker]# docker rmi docker.io/nginx
2. Untagged: docker.io/nginx:latest
3. Deleted: sha256:42b4762643dcc9bf492b08064b55fef64942f055f0da91289a8abf93c6d6b43c
4. Deleted: sha256:e0e55dd2303b3e3ec852acae267d1f8a3eea27a22c64a5829304ecee4d3f559c
5. Deleted: sha256:4062cf272cdd99e83b1c21f712e5e1359c91ecf92925e56c62133c3324b84e45
6. Deleted: sha256:3c816b4ead84066ec2cadec2b943993aaacc3fe35fcd77ada3d09dc4f3937313

步骤二：一次性导入多个镜像

1. [student@room9pc01 04]$ cd /linux-soft/04/
2. [student@room9pc01 04]$ scp -r docker/ root@192.168.1.31:/root/
3. [root@docker1 ~]# cd docker
4. [root@docker1 docker]# ls
5. centos.tar nginx.tar redis.tar registry.tar ubuntu.tar
6. [root@docker1 docker]# docker images
7. REPOSITORY TAG IMAGE ID CREATED SIZE
8. [root@docker1 docker]# for i in \* ; do docker load -i $i; done
9. bcc97fbfc9e1: Loading layer 208.2 MB/208.2 MB
10. Loaded image: docker.io/centos:latest208.2 MB
11. Loaded image: docker.io/nginx:latest
12. a098f8909d23: Loading layer 338.4 kB/338.4 kB
13. c3bc3362d4ff: Loading layer 3.034 MB/3.034 MB
14. bf10d8534af6: Loading layer 36.4 MB/36.4 MB
15. 0949ce28adcc: Loading layer 1.536 kB/1.536 kB
16. b1f287cac208: Loading layer 3.584 kB/3.584 kB
17. Loaded image: docker.io/redis:latest/3.584 kB
18. 7bff100f35cb: Loading layer 4.672 MB/4.672 MB
19. 6b6e0aba7201: Loading layer 1.587 MB/1.587 MB
20. 18429e86e6ad: Loading layer 20.08 MB/20.08 MB
21. 7b33b7de3d52: Loading layer 3.584 kB/3.584 kB
22. 726e86b708de: Loading layer 2.048 kB/2.048 kB
23. Loaded image: docker.io/registry:latest048 kB
24. adcb570ae9ac: Loading layer 89.95 MB/89.95 MB
25. 7604c8714555: Loading layer 15.87 kB/15.87 kB
26. 9e9d3c3a7458: Loading layer 11.26 kB/11.26 kB
27. 27a216ffe825: Loading layer 3.072 kB/3.072 kB
28. Loaded image: docker.io/ubuntu:latest3.072 kB
29. [root@docker1 docker]# docker images

导入多个镜像如图-1所示：

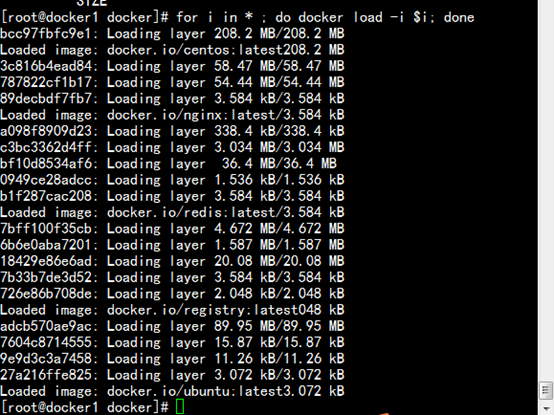


图-1

步骤三：启动镜像

1）启动centos镜像生成一个容器

启动镜像时若不知道后面的命令加什么:

1、可以猜（如：/bin/bash、/bin/sh）

2、可以不加后面的命令，默认启动

1. [root@docker1 docker\_images]# docker run -it docker.io/centos /bin/bash
2. [root@7a652fc72a9f /]# ls /
3. anaconda-post.log bin dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
4. [root@7a652fc72a9f /]# cd /etc/yum.repos.d/
5. [root@7a652fc72a9f yum.repos.d]# ls
6. CentOS-Base.repo CentOS-Debuginfo.repo CentOS-Sources.repo CentOS-fasttrack.repo
7. CentOS-CR.repo CentOS-Media.repo CentOS-Vault.repo
8. [root@7a652fc72a9f yum.repos.d]# rm -rf C\*
9. [root@7a652fc72a9f yum.repos.d]# ls
10. [root@7a652fc72a9f yum.repos.d]#vi dvd.repo //在容器里面配置一个yum源
11. [local]
12. name=local
13. baseurl=ftp://192.168.1.254/system
14. enable=1
15. gpgcheck=0
16. [root@7a652fc72a9f yum.repos.d]# yum -y install net-tools //安装软件
17. [root@7a652fc72a9f yum.repos.d]# exit
18. exit

## 3 案例3：镜像与容器常用指令

### 3.1 问题

本案例要求掌握镜像与容器的常用命令：

* 镜像常用指令练习
* 容器常用指令练习

### 3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：镜像常用命令

1）查看后台运行的容器

1. [root@docker1 docker]# docker run -d docker.io/nginx //启动nginx的镜像
2. [root@docker1 docker]# docker ps    //查看后台运行的容器
3. CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4. 81458156f6e8 docker.io/nginx "nginx -g 'daemon off" 9 seconds ago Up 8 seconds 80/tcp pedantic\_goldberg

2）只显示容器ID

1. [root@docker1 docker]# docker ps -q
2. 81458156f6e8

3）显示所有的容器,包括没有启动的

1. [root@docker1 docker]# docker ps -a

4）显示所有的容器ID

1. [root@docker1 docker]# docker ps -qa
2. 81458156f6e8
3. 3656f1978967

5）查看centos镜像历史（制作过程），如图-2所示：

1. [root@docker1 docker\_images]# docker history docker.io/centos

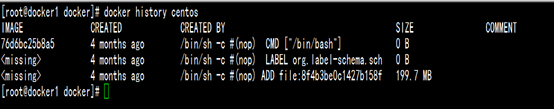


图-2

7）删除镜像，启动容器时删除镜像会失败，先删除容器,再删除镜像

格式：docker rmi 镜像名

1. [root@docker1 docker]# docker rmi nginx //nginx为镜像名
2. Error response from daemon: conflict: unable to remove repository reference "nginx" (must force) - container 81458156f6e8 is using its referenced image 42b4762643dc
3. //删除时报错
5. [root@docker1 docker]# docker stop 81
6. 81
7. [root@docker1 docker]# docker rm 81
8. 81
9. [root@docker2 ~]# docker rmi docker.io/nginx //删除nginx镜像
10. Untagged: docker.io/nginx:latest
11. Deleted: sha256:42b4762643dcc9bf492b08064b55fef64942f055f0da91289a8abf93c6d6b43c
12. Deleted: sha256:e0e55dd2303b3e3ec852acae267d1f8a3eea27a22c64a5829304ecee4d3f559c
13. Deleted: sha256:4062cf272cdd99e83b1c21f712e5e1359c91ecf92925e56c62133c3324b84e45
14. Deleted: sha256:3c816b4ead84066ec2cadec2b943993aaacc3fe35fcd77ada3d09dc4f3937313

8）修改镜像的名称和标签,默认标签为latest

1. [root@docker1 docker\_images]# docker tag docker.io/centos:latest docker.io/cen:v1
2. [root@docker2 ~]# docker images
3. REPOSITORY TAG IMAGE ID CREATED SIZE
4. docker.io/centos latest 42b4762643dc 4 weeks ago 109.2 MB
5. docker.io/cen v1 42b4762643dc 4 weeks ago 109.2 MB

9）查看镜像的底层信息，如图-3所示：

1. [root@docker1 docker\_images]# docker inspect docker.io/centos

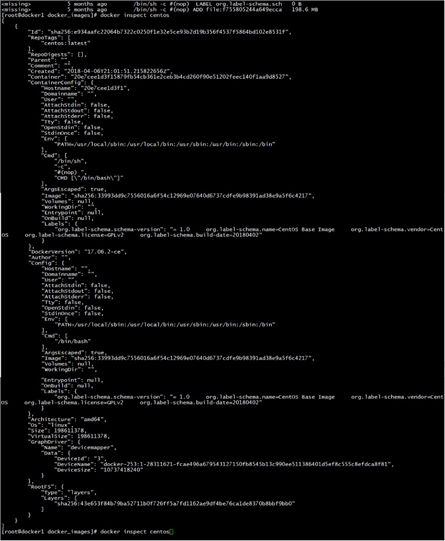


图-3

10）修改镜像的标签

1. [root@docker1 docker\_images]# docker tag docker.io/centos:latest docker.io/cen:v1
2. [root@docker1 docker\_images]# docker images
3. REPOSITORY TAG IMAGE ID CREATED SIZE
4. docker.io/cen v1 e934aafc2206 5 months ago 198.6 MB
5. [root@docker1 docker\_images]# docker rmi docker.io/centos //删除centos
6. [root@localhost ~]# docker run -it centos
7. //启动的时候，因为是用标签标签启动的，所以会重新通过ID下载
8. [root@docker1 docker]# docker run -it docker.io/centos
9. Unable to find image 'centos:latest' locally
10. Trying to pull repository registry.access.redhat.com/centos ...
11. Trying to pull repository docker.io/library/centos ...
12. latest: Pulling from docker.io/library/centos
13. a02a4930cb5d: Pull complete
14. Digest: sha256:184e5f35598e333bfa7de10d8fb1cebb5ee4df5bc0f970bf2b1e7c7345136426
15. [root@b6c5f16461f5 /]#
16. [root@localhost ~]# docker run -it docker.io/cen:v1
17. //通过新建的标签启动docker.io/cen:v1

步骤二：容器命令

1）关闭容器

命令：docker stop 容器ID

1. [root@docker1 docker]# docker stop 0f //0f为容器ID
2. 0f

2）启动容器

1. [root@docker1 docker]# docker start 0f
2. 0f

3）重启容器

1. [root@docker1 docker]# docker restart 0f
2. 0f

4）删除容器

运行中删除不掉，先关闭容器

1. [root@docker1 docker]# docker rm 0f //删除失败
2. Error response from daemon: You cannot remove a running container 0f63706692e15134a8f07655a992771b312b8eb01554fc37e1a39b03b28dd05c. Stop the container before attempting removal or use -f
3. [root@docker1 docker]# docker stop 0f //关闭容器
4. 0f
5. [root@docker1 docker]# docker rm 0f         //删除成功
6. 0f
7. [root@docker1 docker]#

5）连接容器attach|exec

1. [root@docker1 docker]# docker attach 0f
2. [root@docker1 docker]# docker ps //容器关闭
3. CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4. [root@docker1 docker]# docker exec -it 0f /bin/bash
5. [root@docker1 docker]# docker ps        //容器不会关闭
6. CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7. 0b3c50284a1c docker.io/centos "/bin/bash" 15 minutes ago Up 15 minutes tiny\_lamarr
8. [root@docker1 docker]# docker top f7        //查看容器进程列表
9. [root@docker1 docker]# docker run -itd docker.io/centos:latest
10. 76e7577cff5d8a3ec877d3ea564fed2fb502bd4a0447705bec269a646d414d07
11. [root@docker1 docker]# ps
12. PID TTY TIME CMD
13. 5552 pts/0 00:00:00 bash
14. 6739 pts/0 00:00:00 ps
15. [root@docker1 docker]# docker exec -it 76 /bin/bash
16. [root@76e7577cff5d /]# sleep 50 &
17. [1] 30
18. [root@76e7577cff5d /]# exit
19. exit
20. [root@docker1 docker]# docker top 76
21. UID PID PPID C STIME TTY TIME CMD
22. root 6708 6691 0 17:00 pts/1 00:00:00 /bin/bash
23. UID PID PPID C STIME     TTY         TIME      CMD
24. root 2744 2729 0 18:01 pts/4 00:00:00 /bin/bash

6）过滤查看mac和ip地址

1. [root@docker1 docker]# docker inspect -f '{{.NetworkSettings.MacAddress}}' 76
2. 02:42:ac:11:00:04
3. [root@docker1 docker]# docker inspect -f '{{.NetworkSettings.IPAddress}}' 76
4. 172.17.0.4

7）修改nginx的显示内容

1. [root@docker1 docker\_images]# docker run -it docker.io/nginx:latest



1. [root@docker1 docker\_images]# docker exec -it 56 /bin/bash
2. root@56ec8154f8e0:/# nginx -T /usr/share/nginx/html/
3. nginx: invalid option: "/usr/share/nginx/html/" //查找并显示结果
4. root@56ec8154f8e0:/# echo aaa > /usr/share/nginx/html/index.html
5. //修改主页显示的内容
6. root@56ec8154f8e0:/# nginx -T
7. root@56ec8154f8e0:/# cat /usr/share/nginx/html/index.html
8. aaa

8）过滤查看nginx的ip地址

1. [root@docker1 docker]# docker inspect -f '{{.NetworkSettings.IPAddress}}' a6
2. 172.17.0.3
3. [root@docker1 docker]# curl 172.17.0.3
4. aaa