

一、完成centos上安装Docker

1、安装 Docker Engine-Community

```
[root@localhost ~]# sudo yum install -y yum-utils \
> device-mapper-persistent-data \
> lvm2
已加载插件：fastestmirror, langpacks
base | 3.6 kB 00:00
epel/x86_64/metalink | 8.8 kB 00:00
epel | 4.7 kB 00:00
https://hkg.mirror.rackspace.com/epel/7/x86_64/repodata/repomd.xml: [Errno -1]
epomd.xml does not match metalink for epel
正在尝试其它镜像。
epel
extras | 2.9 kB 00:00
updates | 2.9 kB 00:00
(1/5): extras/7/x86_64/primary_db | 194 kB 00:00
(2/5): epel/x86_64/group_gz | 95 kB 00:01
(3/5): updates/7/x86_64/primary_db | 1.3 MB 00:01
(4/5): epel/x86_64/updateinfo | 1.0 MB 00:01
(5/5): epel/x86_64/primary_db | 6.8 MB 00:11
```

2、设置仓库（这里设置的官网）

```
[root@localhost ~]# sudo yum-config-manager \
> --add-repo \
> https://download.docker.com/linux/centos/docker-ce.repo
已加载插件：fastestmirror, langpacks
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum
repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
```

3、安装 Docker 最新版本

```
[root@localhost ~]# sudo yum install docker-ce docker-ce-cli containerd.io
已加载插件：fastestmirror, langpacks
docker-ce-stable | 3.5 kB 00:00:00
(1/2): docker-ce-stable/x86_64/updateinfo | 55 B 00:00:00
(2/2): docker-ce-stable/x86_64/primary_db | 43 kB 00:00:00
Loading mirror speeds from cached hostfile
* base: mirrors.aliyun.com
* epel: sg.fedora.ipserverone.com
* extras: mirrors.aliyun.com
* updates: mirrors.aliyun.com
正在解决依赖关系
--> 正在检查事务
--> 软件包 containerd.io.x86_64.0.1.2.13-3.2.el7 将被 安装
--> 正在处理依赖关系 container-selinux >= 2:2.74, 它被软件包 containerd.io-1.2.13-3.2.el7.x86
_64 需要
--> 软件包 docker-ce.x86_64.3.19.03.10-3.el7 将被 安装
--> 软件包 docker-ce-cli.x86_64.1.19.03.10-3.el7 将被 安装
--> 正在检查事务
```

4、安装完成并启动Docker

```
作为依赖被安装:
container-selinux.noarch 2:2.119.1-1.c57a6f9.el7

作为依赖被升级:
libselinux.x86_64 0:2.5-15.el7          libselinux-python.x86_64 0:2.5-15.el7
libsemanage-python.x86_64 0:2.5-14.el7  libsepol.x86_64 0:2.5-10.el7
selinux-policy.noarch 0:3.13.1-266.el7  selinux-policy-targeted.noarch 0:3.13.1-266.

完毕!
[root@localhost ~]# sudo systemctl start docker
```

5、配置阿里云的镜像加速并重启服务

```
[root@localhost /]# sudo mkdir -p /etc/docker
[root@localhost /]# sudo tee /etc/docker/daemon.json <<- 'EOF'
> {
>   "registry-mirrors": ["https://5tnugegb.mirror.aliyuncs.com"]
> }
> EOF
{
  "registry-mirrors": ["https://5tnugegb.mirror.aliyuncs.com"]
}
[root@localhost /]# sudo systemctl daemon-reload
[root@localhost /]# sudo systemctl restart docker
[root@localhost /]#
```

二、在docker中安装mysql

1、安装mysql镜像

```
[root@localhost /]# docker pull mysql:5.7
5.7: Pulling from library/mysql
afb6ec6fdc1c: Downloading 2.172MB/27.1MB
0bdc5971ba40: Download complete
97ae94a2c729: Downloading 1.975MB/4.178MB
f777521d340e: Downloading 199.7kB/1.419MB
1393ff7fc871: Waiting
a499b89994d9: Waiting
7ebe8eefbaf8: Waiting
4eec965ae405: Waiting
a531a782d709: Waiting
270aeddb45e3: Waiting
b25569b61008: Waiting
```

2、查看docker中的镜像

```
[root@localhost /]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql	5.7	a4fd462add	9 days ago	448MB

```
[root@localhost /]#
```

3、创建相应的文件目录-并使用docker命令启动mysql

```

[root@localhost /]# mkdir /mydata
[root@localhost /]# ls
bin  dev  home  lib64  mnt  opt  root  sbin  sys  usr  下载
boot  etc  lib  media  mydata  proc  run  srv  tmp  var
[root@localhost /]# cd mydata
[root@localhost mydata]# ls
[root@localhost mydata]# mkdir mysql
[root@localhost mydata]# docker run -p 3306:3306 --name mysql \
> -v /mydata/mysql/log:/var/log/mysql \
> -v /mydata/mysql/data:/var/lib/mysql \
> -v /mydata/mysql/conf:/etc/mysql \
> -e MYSQL_ROOT_PASSWORD=root \
> -d mysql:5.7
b61a3c50164c3d9d02c63f6380b3930379f54f71fe46ab838208977a410f9937
[root@localhost mydata]#

```

4、idea连接虚拟机中的mysql--并创建数据库--进入Docker与查看

5、创建数据库和表

```

mysql> use test;
Database changed
mysql> create table tabletest(id int(11));
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| tabletest      |
+-----+
1 row in set (0.00 sec)

mysql>

```


6、退出mysql-并退出mysql容器--关闭这个容器

```
mysql> exit
Bye
root@b61a3c50164c:/# docker stop b61a3c50164c
bash: docker: command not found
root@b61a3c50164c:/# exit
exit
[root@localhost data]# docker stop b61a3c50164c
b61a3c50164c
[root@localhost data]#
```

三、在docker中安装Nginx

1、下载Nginx1.10的docker镜像

```
[root@localhost data]# docker pull nginx:1.10
1.10: Pulling from library/nginx
6d827a3ef358: Pulling fs layer
1e3e18a64ea9: Pulling fs layer
556c62bb43ac: Download complete
```

2、从容器中拷贝nginx配置

```
[root@localhost data]# docker run -p 80:80 --name nginx \
> -v /mydata/nginx/html:/usr/share/nginx/html \
> -v /mydata/nginx/logs:/var/log/nginx \
> -d nginx:1.10
ea00e91d2a362b469ac348a1e3943c2b0c6e5b20de6c82c996662b9f2472b716
[root@localhost data]# 1 2 3 4
```

3、将容器中的配置文件拷到指定目录mydata下

```
[root@localhost data]# docker container cp nginx:/etc/nginx /mydata/nginx/
[root@localhost data]# ls
auto.cnf      client-cert.pem  ibdata1        mysql          public_key.pem  sys
ca-key.pem    client-key.pem   ib_logfile0     performance_schema  server-cert.pem  test
ca.pem        ib_buffer_pool  ib_logfile1     private_key.pem   server-key.pem
[root@localhost data]# cd ..
[root@localhost mysql]# ls
conf  data  log
[root@localhost mysql]# cd ..
[root@localhost mydata]# ls
mysql  nginx
[root@localhost mydata]#
```

4、修改文件名称

```
[root@localhost nginx]# mv nginx conf
[root@localhost nginx]# ls
conf  html  logs
[root@localhost nginx]#
```

5、终止并删除容器

```
[root@localhost nginx]# docker stop nginx
nginx
[root@localhost nginx]# docker rm nginx
nginx
```

6、启动Nginx并查看进程

```
[root@localhost nginx]# docker run -p 80:80 --name nginx \
> -v /mydata/nginx/html:/usr/share/nginx/html \
> -v /mydata/nginx/logs:/var/log/nginx \
> -v /mydata/nginx/conf:/etc/nginx \
> -d nginx:1.10
372f26a69b0960f45882ba54e2db40b016021b4d764a966f74fcd88a2338f8ee
[root@localhost nginx]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
372f26a69b09	nginx:1.10	"nginx -g 'daemon of..."	42 seconds ago	Up		

```
41 seconds
0.0.0.0:80->80/tcp, 443/tcp  nginx
[root@localhost nginx]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
372f26a69b09	nginx:1.10	"nginx -g 'daemon of..."	49 seconds ago	Up 48 seconds	0.0.0.0:80->80/tcp, 443/tcp	nginx