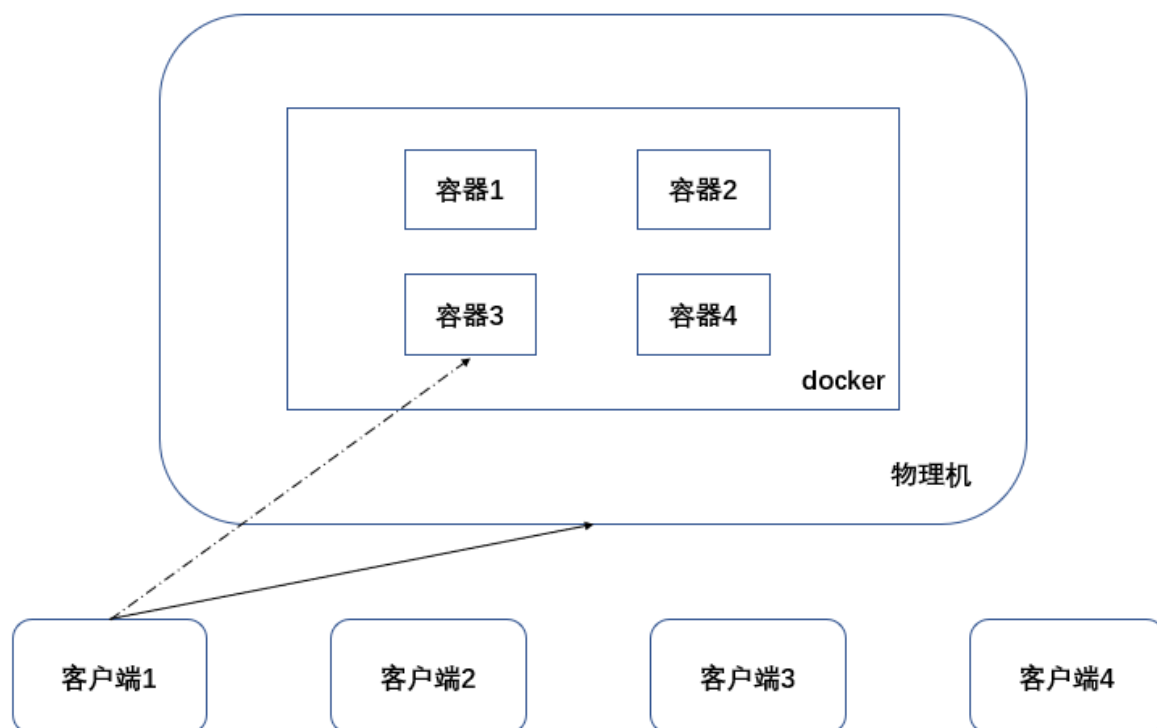


Docker

介绍

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中，然后发布到任何流行的 Linux 或 Windows 操作系统的机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。



安装

官方网址: <https://www.docker.com/>

安装依赖包

```
yum install -y yum-utils \
```

配置仓库

```
yum-config-manager \
--add-repo \
https://download.docker.com/linux/centos/docker-ce.repo    #默认是国外的地址
```

安装Docker引擎

```
yum install docker-ce
```

启动Docker

```
systemctl start docker
```

测试Docker是否安装成功

```
docker version
```

运行hello world

```
docker run hello-world
```

配置开机启动

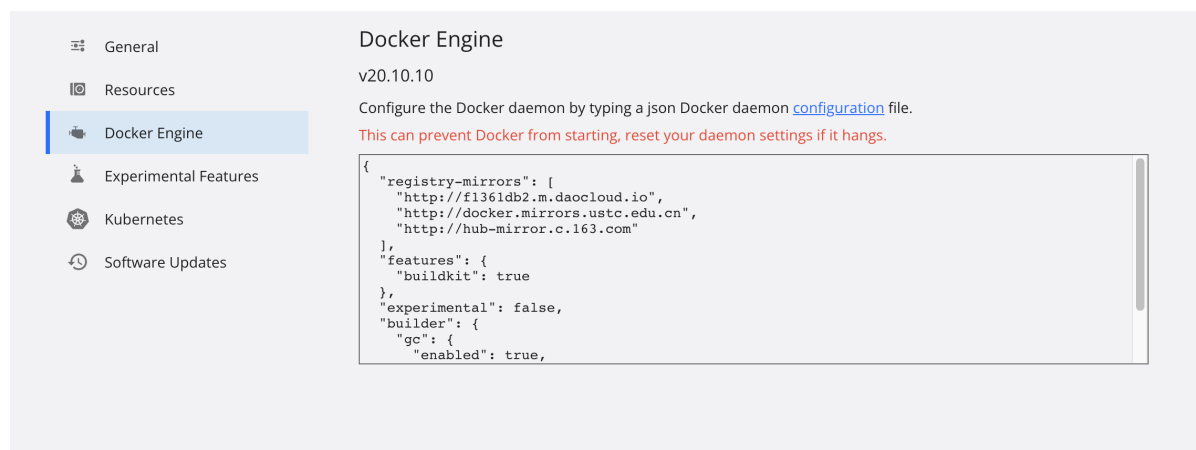
```
systemctl enable docker
```

更改源

国内的镜像源有

- docker官方中国区 <https://registry.docker-cn.com>
- 网易 <http://hub-mirror.c.163.com>
- ustc <http://docker.mirrors.ustc.edu.cn>
- 阿里云 <http://<你的ID>.mirror.aliyuncs.com>
- DaoCloud <http://f1361db2.m.daocloud.io>

1、在Dashboard中，点击设置->Docker Engine 中修改



```
{
  "registry-mirrors": [
    "http://f1361db2.m.daocloud.io",
    "http://docker.mirrors.ustc.edu.cn",
    "http://hub-mirror.c.163.com"
  ],
  "features": {
    "buildkit": true
  },
  "experimental": false,
  "builder": {
```

```
"gc": {  
  "enabled": true,  
  "defaultKeepStorage": "20GB"  
}  
}  
}
```

指定 `docker info` 查看是否配置成功

```
Registry Mirrors:  
http://f1361db2.m.daocloud.io/  
http://docker.mirrors.ustc.edu.cn/  
http://hub-mirror.c.163.com/
```

[Mac安装Docker](#)

[Xshell国内下载地址](#)

基本命令

Docker

```
docker version  
docker info  
docker run hello-world  
docker image ls  
docker container ls --all
```

List Docker CLI commands

```
docker 命令 --help
```

镜像命令

<code>docker images</code>	列出所有镜像
<code>docker search \${image:v}</code>	搜索镜像
<code>docker pull \${image:v}</code>	下载镜像
<code>docker rmi \${image:v}</code>	删除镜像

容器命令

<code>docker ps</code>	列出运行中的容器
<code>docker ps -a</code>	列出所有状态的容器
<code>docker ps -q</code>	列出运行中的容器id

<code>docker run \${image:v}</code>	通过镜像启动容器
<code>-d</code>	后台方式运行
<code>-it</code>	交互方式运行，能够进入容器查看内容
<code>-p</code>	端口映射

<code>exit</code>	退出容器
<code>ctrl+Q+P</code>	退出但不停止容器
<code>docker inspect \${containerID}</code>	查看容器信息
<code>docker rm \${containerID}</code>	删除容器
<code>docker stop \${containerID}</code>	停止正在运行的容器
<code>docker kill \${containerID}</code>	强制停止正在运行的容器
<code>docker start \${containerID}</code>	启动容器
<code>docker restart \${containerID}</code>	重启容器
<code>docker exec -it \${containerID} /bin/bash</code>	进入一个正在运行中的容器

扩展命令

<code>docker cp sourcePath \${containerID}:destinationPath</code>	宿主机拷贝到容器
<code>docker cp \${containerID}:destinationPath sourcePath</code>	容器拷贝到宿主机
<code>docker commit \${containerID} image:v</code>	保存一个container到image
<code>docker save -o destinationPath imageName</code>	保存一个image
<code>docker load -i sourcePath</code>	加载一个文件到image

启动一个demo (演示环境)



No containers running

Try running a container: Copy and paste this command into your terminal and then come back

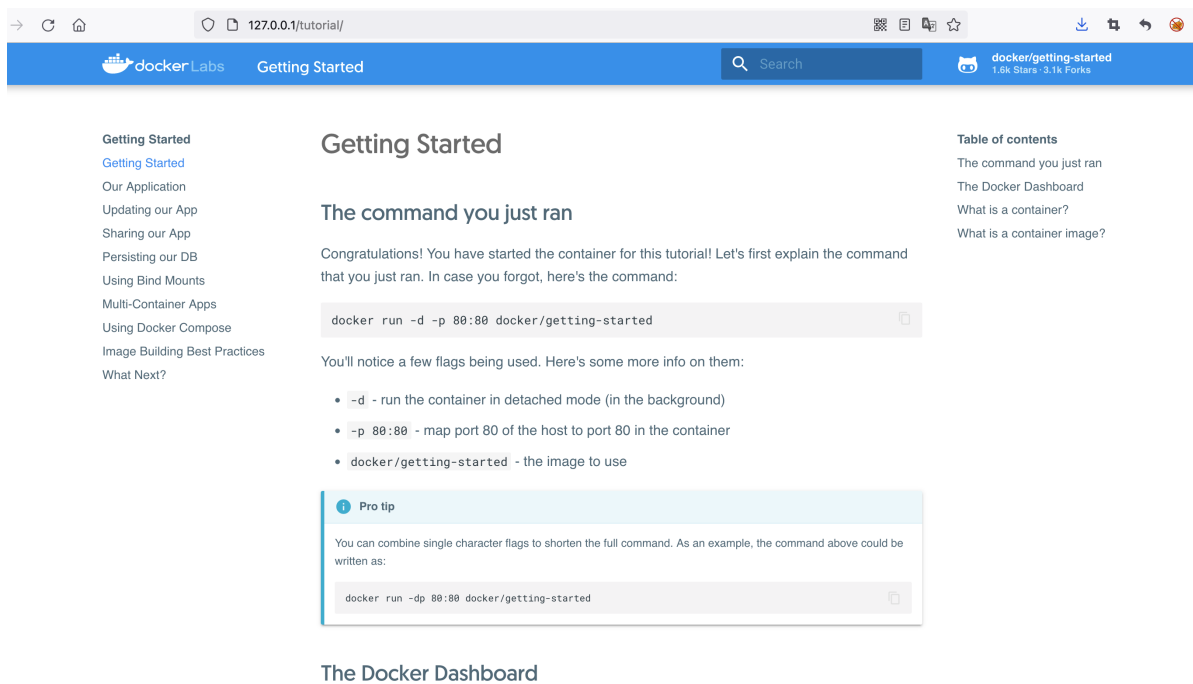
```
docker run -d -p 80:80 docker/getting-started
```



[Explore more in the Docker Docs](#)

```
docker run -d -p 80(主机端口):80(容器端口) docker/getting-started
```

浏览器访问 127.0.0.1



docker hub & 常用环境

官方镜像 <https://hub.docker.com/explore/>

安装配置Tomcat

参考链接: <https://hub.docker.com/tomcat/>

Docker Pull Command

```
$ docker pull tomcat
```

启动

```
$ docker run -it --rm -p 8888:8080 tomcat
```

启动带挂载目录

```
$ docker run -v /home/bage/mnt-tomcat:/mnt -it --rm -p 8888:8080 tomcat
```

安装配置MySQL

<https://hub.docker.com/mysql>

下载

```
docker pull mysql
```

Start a mysql server instance

```
docker run --name bage-mysql -v /home/bage/data/mysql:/var/lib/mysql -e  
MYSQL_ROOT_PASSWORD=bage -p 3306:3306 -d bage-mysql
```

其中

```
--name 起名 bage-mysql  
-v 挂载目录 /home/bage/data/mysql:/var/lib/mysql  
-e 设置root密码 MYSQL_ROOT_PASSWORD=bage  
-p 映射端口 3306:3306  
-d bage-mysql
```

进入mysql容器

```
docker exec -it bage-mysql /bin/bash
```

在容器内部进行登录

```
mysql -u root -p
```

创建数据库

```
Create database mydb;
```

创建用户

```
CREATE USER 'bage'@'%' IDENTIFIED BY 'bage';
```

授权

```
grant all privileges on mydb.* to 'bage'@'%';
```

远程无法登陆问题 <https://blog.csdn.net/gf0515/article/details/80466213>

修改密码规则

```
ALTER user 'bage'@'%' IDENTIFIED BY 'bage' PASSWORD EXPIRE NEVER;
```

更新密码

```
ALTER user 'bage'@'%' IDENTIFIED WITH mysql_native_password BY 'bage';
```

刷新

```
FLUSH PRIVILEGES;
```

安装配置Postgres

参考链接: <https://hub.docker.com/postgres/>

Docker Pull Command

```
docker pull postgres
```

start a postgres instance 方式1

```
docker run -it -p 5432:5432 --name bage-postgres -e POSTGRES_PASSWORD=postgres -d postgres
```

start a postgres instance 方式2

```
docker run --name bage-postgres -e POSTGRES_PASSWORD=postgres -d postgres
```

connect to it from an application

```
docker run -it --rm --link bage-postgres:postgres postgres psql -h postgres -U postgres
```

创建数据库:

```
CREATE TABLE weather (  
    cityvarchar(80),  
    temp_lo int,    -- low temperature  
    temp_hi int,    -- high temperature  
    prcpreal,    -- precipitation  
    datedate  
);
```

安装配置Nginx

下载安装:

```
docker pull nginx
```

启动:

```
docker run -p 80:80 --name tmp-nginx-container -d nginx
```

进入容器:

```
docker exec -it tmp-nginx-container /bin/bash
```

默认配置文件位置:

```
/etc/nginx/nginx.conf  
/etc/nginx/conf.d/default.conf
```

新建一个临时编辑目录:

```
mkdir -p /home/bage/workspace/docker/docker/
```

拷贝配置文件出来进行编辑:

```
docker cp tmp-nginx-container:/etc/nginx/conf.d/default.conf  
/home/bage/workspace/docker/docker/
```

编辑修改后进行返回:

```
docker cp /home/bage/workspace/docker/docker/ tmp-nginx-  
container:/etc/nginx/nginx.conf
```