1 安装Docker

（1）yum 包更新到最新

```
sudo yum update
```

（2）安装需要的软件包， yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动依赖的

```
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

（3）设置yum源为阿里云

```
sudo yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

（4）安装docker

```
sudo yum install docker-ce
```

（5）安装后查看docker版本

```
docker -v
```

2 设置ustc的镜像

ustc是老牌的linux镜像服务提供者了，还在遥远的ubuntu 5.04版本的时候就在用。ustc的docker镜像加速器速度很快。ustc docker mirror的优势之一就是不需要注册，是真正的公共服务。

https://lug.ustc.edu.cn/wiki/mirrors/help/docker

编辑该文件：

```
vi /etc/docker/daemon.json
```

在该文件中输入如下内容：

```
{
"registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]
}
```

3 Docker的启动与停止

systemctl命令是系统服务管理器指令

启动docker：

```
systemctl start docker
```

停止docker：

```
systemctl stop docker
```

重启docker：

```
systemctl restart docker
```

查看docker状态：

```
systemctl status docker
```

开机启动：

```
systemctl enable docker
```

查看docker概要信息

```
docker info
```

查看docker帮助文档

```
docker --help
```

验证docker是否安装成功安装，如果返回如下信息，表示docker安装成功

```
[root@docker ~]# docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://cloud.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/
```


启动并加入开机启动

```
$ sudo systemctl start docker
$ sudo systemctl enable docker
```

验证安装是否成功(有client和service两部分表示docker安装启动都成功了)

```
$ docker version
```

```
[root@localhost local]# docker version
Client:
 Version:       17.12.0-ce
 API version:   1.35
 Go version:    go1.9.2
 Git commit:    c97c6d6
 Built: Wed Dec 27 20:10:14 2017
 OS/Arch:       linux/amd64

Server:
 Engine:
  Version:      17.12.0-ce
  API version:  1.35 (minimum version 1.12)
  Go version:   go1.9.2
  Git commit:   c97c6d6
  Built:        Wed Dec 27 20:12:46 2017
  OS/Arch:      linux/amd64
  Experimental: false
```

在docker容器安装httpd服务，并把宿主机的80端口映射给httpd的80端口

```
[root@docker ~]# docker run -d -p 80:80 httpd
Unable to find image 'httpd:latest' locally
```

```
latest: Pulling from library/httpd
f49cf87b52c1: Pull complete
02ca099fb6cd: Pull complete
de7acb18da57: Pull complete
770c8edb393d: Pull complete
0e252730aeae: Pull complete
6288e83d58fa: Pull complete
a91ad03b2178: Pull complete
Digest: sha256:643ca2ed9f6caf1f392184aee05a8f2cd478bdacbd350ea6b4dbc8f5b8e400a8
Status: Downloaded newer image for httpd:latest
0747f02bb2304ef964ed153e6e81f3f083965f95f12ad9a70c95b7f329243a2c
docker: Error response from daemon: driver failed programming external connectivity on endpoint mystifying_jennings (8644011e3dab6725c314f6e634444ca6b6cfe3792d6f504a12af2e065d4c448f): Error starting userland proxy: listen tcp 0.0.0.0:80: bind: address already in use.
```

安装后在web页面输入宿主机地址访问http容器