想理解、会用Docker，这篇文章就够了！	http://baijiahao.baidu.com/s?id=1586633082072582334&wfr=spider&for=pc
CentOS Docker 安装	https://www.runoob.com/docker/centos-docker-install.html
Docker 官方使用文档	https://docs.docker.com/engine/reference/builder/

一、环境准备：

1.1、查看linux内核版本和系统

> Docker目前要求Linux系统内核版本不低于3.10，并且系统是64位，对应CentOS 7及以后的版本

```
uname -ir
```

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# uname -ir
3.10.0-693.2.2.el7.x86_64 x86_64
```

如果CentOS系统过低，可以通过 sudo yum update命令升级

> 说明：生产环境对软件版本和内核版本要求非常精确，别没事有事随便的进行yum update操作
> yum update：升级所有包同时也升级软件和系统内核
> yum upgrade：只升级所有包，不升级软件和系统内核

```
sudo yum update
```

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# sudo yum update
Loaded plugins: fastestmirror
base                                                     | 3.6 kB  00:00:00
epel                                                     | 5.3 kB  00:00:00
extras                                                   | 2.9 kB  00:00:00
updates                                                  | 2.9 kB  00:00:00
(1/7): epel/x86_64/group_gz                              |  90 kB  00:00:00
(2/7): base/7/x86_64/group_gz                            | 165 kB  00:00:00
(3/7): epel/x86_64/updateinfo                            | 1.0 MB  00:00:00
(4/7): extras/7/x86_64/primary_db                        | 153 kB  00:00:00
(5/7): base/7/x86_64/primary_db                          | 6.0 MB  00:00:00
(6/7): updates/7/x86_64/primary_db                       | 4.2 MB  00:00:00
(7/7): epel/x86_64/primary_db                            | 6.9 MB  00:00:00
Determining fastest mirrors
Resolving Dependencies
--> Running transaction check
---> Package GeoIP.x86_64 0:1.5.0-11.el7 will be updated
```

```
  Cleanup    : bash-4.2.46-29.el7_4.x86_64                                 467/472
  Cleanup    : nspr-4.13.1-1.0.el7_3.x86_64                                468/472
  Cleanup    : nss-softokn-freebl-3.28.3-8.el7_4.x86_64                    469/472
  Cleanup    : glibc-2.17-196.el7.x86_64                                   470/472
  Cleanup    : tzdata-2017b-1.el7.noarch                                   471/472
  Cleanup    : libgcc-4.8.5-16.el7.x86_64                                  472/472
```

1.2、查看是否已经安装过docker

docker命令

```
docker
```

或 查看docker安装目录

```
which docker
```

或 运行hello-world镜像

```
sudo docker run hello-world
```

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# which docker
/usr/bin/which: no docker in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin)
[root@iZbp16e5pr6sbudyuvo3ruZ ~]#
```

1.3、如果需要卸载旧版本的docker

```
sudo yum remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-engine
```

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# uname -ir
3.10.0-693.2.2.el7.x86_64 x86_64
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# sudo yum remove docker \
>                   docker-client \
>                   docker-client-latest \
>                   docker-common \
>                   docker-latest \
>                   docker-latest-logrotate \
>                   docker-logrotate \
>                   docker-engine
Loaded plugins: fastestmirror
No Match for argument: docker
No Match for argument: docker-client
No Match for argument: docker-client-latest
No Match for argument: docker-common

No Match for argument: docker-latest
No Match for argument: docker-latest-logrotate
No Match for argument: docker-logrotate
No Match for argument: docker-engine
No Packages marked for removal
```

1.4、为了方便添加软件源支持 devicemapper存储类型，需要安装yum-utils、device-mapper-persistent-data、lvm2

> yum-utils 提供了 yum-config-manager
> device mapper 存储驱动程序需要 device-mapper-persistent-data 和 lvm2
> device-mapper-persistent-data
> lvm2

换行命令

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

```
3  lvm2
```

不换行命令

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

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# sudo yum install -y yum-utils \
>     device-mapper-persistent-data \
>     lvm2
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
---> Package device-mapper-persistent-data.x86_64 0:0.8.5-1.el7 will be installed
--> Processing Dependency: libaio.so.1(LIBAIO_0.4)(64bit) for package: device-mapper-persistent-data-0.8.
5-1.el7.x86_64
--> Processing Dependency: libaio.so.1(LIBAIO_0.1)(64bit) for package: device-mapper-persistent-data-0.8.
5-1.el7.x86_64
--> Processing Dependency: libaio.so.1()(64bit) for package: device-mapper-persistent-data-0.8.5-1.el7.x8
6_64
---> Package lvm2.x86_64 7:2.02.185-2.el7_7.2 will be installed
--> Processing Dependency: lvm2-libs = 7:2.02.185-2.el7_7.2 for package: 7:lvm2-2.02.185-2.el7_7.2.x86_64
--> Processing Dependency: liblvm2app.so.2.2(Base)(64bit) for package: 7:lvm2-2.02.185-2.el7_7.2.x86_64
--> Processing Dependency: libdevmapper-event.so.1.02(Base)(64bit) for package: 7:lvm2-2.02.185-2.el7_7.2
.x86_64
--> Processing Dependency: liblvm2app.so.2.2()(64bit) for package: 7:lvm2-2.02.185-2.el7_7.2.x86_64
--> Processing Dependency: libdevmapper-event.so.1.02()(64bit) for package: 7:lvm2-2.02.185-2.el7_7.2.x86
_64
---> Package yum-utils.noarch 0:1.1.31-52.el7 will be installed
--> Processing Dependency: python-kitchen for package: yum-utils-1.1.31-52.el7.noarch
--> Processing Dependency: libxml2-python for package: yum-utils-1.1.31-52.el7.noarch
```

```
--------------------------------------------------------------------------------------------------------------
Total                                                       12 MB/s | 4.0 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 7:device-mapper-event-libs-1.02.158-2.el7_7.2.x86_64                                      1/10
  Installing : libaio-0.3.109-13.el7.x86_64                                                              2/10
  Installing : device-mapper-persistent-data-0.8.5-1.el7.x86_64                                          3/10
  Installing : 7:device-mapper-event-1.02.158-2.el7_7.2.x86_64                                           4/10
  Installing : 7:lvm2-libs-2.02.185-2.el7_7.2.x86_64                                                     5/10
  Installing : libxml2-python-2.9.1-6.el7_2.3.x86_64                                                     6/10
  Installing : python-chardet-2.2.1-3.el7.noarch                                                         7/10
  Installing : python-kitchen-1.1.1-5.el7.noarch                                                         8/10
  Installing : yum-utils-1.1.31-52.el7.noarch                                                            9/10
  Installing : 7:lvm2-2.02.185-2.el7_7.2.x86_64                                                         10/10
  Verifying  : python-chardet-2.2.1-3.el7.noarch                                                         1/10
  Verifying  : yum-utils-1.1.31-52.el7.noarch                                                            2/10
  Verifying  : device-mapper-persistent-data-0.8.5-1.el7.x86_64                                          3/10
  Verifying  : 7:lvm2-2.02.185-2.el7_7.2.x86_64                                                          4/10
  Verifying  : libxml2-python-2.9.1-6.el7_2.3.x86_64                                                     5/10
  Verifying  : 7:lvm2-libs-2.02.185-2.el7_7.2.x86_64                                                     6/10
  Verifying  : python-kitchen-1.1.1-5.el7.noarch                                                         7/10
  Verifying  : 7:device-mapper-event-1.02.158-2.el7_7.2.x86_64                                           8/10
  Verifying  : libaio-0.3.109-13.el7.x86_64                                                              9/10
  Verifying  : 7:device-mapper-event-libs-1.02.158-2.el7_7.2.x86_64                                     10/10

Installed:
  device-mapper-persistent-data.x86_64 0:0.8.5-1.el7          lvm2.x86_64 7:2.02.185-2.el7_7.2
  yum-utils.noarch 0:1.1.31-52.el7

Dependency Installed:
  device-mapper-event.x86_64 7:1.02.158-2.el7_7.2  device-mapper-event-libs.x86_64 7:1.02.158-2.el7_7.2
  libaio.x86_64 0:0.3.109-13.el7                   libxml2-python.x86_64 0:2.9.1-6.el7_2.3
  lvm2-libs.x86_64 7:2.02.185-2.el7_7.2            python-chardet.noarch 0:2.2.1-3.el7
  python-kitchen.noarch 0:1.1.1-5.el7

Complete!
[root@iZbp16e5pr6sbudyuvo3ruZ ~]#
```

二、通过yum安装Docker

2.1、添加Docker稳定版本的yum软件源：

```
sudo yum-config-manager \
 --add-repo \
 https://download.docker.com/linux/centos/docker-ce.repo
```

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# sudo yum-config-manager \
>     --add-repo \
>     https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
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
```

2.2、安装 Docker Engine-Community

> 安装最新版本的 Docker Engine-Community 和 containerd，或者转到2.3中安装特定版本

```
sudo yum install docker-ce docker-ce-cli containerd.io
```

如果提示您接受 GPG 密钥，请选是

或者使用命令

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

安装过程：

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# sudo yum install -y docker-ce
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
---> Package docker-ce.x86_64 3:19.03.5-3.el7 will be installed
--> Processing Dependency: container-selinux >= 2:2.74 for package
--> Processing Dependency: containerd.io >= 1.2.2-3 for package: 3
--> Processing Dependency: docker-ce-cli for package: 3:docker-ce-
--> Processing Dependency: libcgroup for package: 3:docker-ce-19.0
--> Running transaction check
---> Package container-selinux.noarch 2:2.107-3.el7 will be instal
--> Processing Dependency: policycoreutils-python for package: 2:c
---> Package containerd.io.x86_64 0:1.2.10-3.2.el7 will be install
---> Package docker-ce-cli.x86_64 1:19.03.5-3.el7 will be installe
---> Package libcgroup.x86_64 0:0.41-21.el7 will be installed
--> Running transaction check
```

安装结果会展示安装版本

```
Installed:
  docker-ce.x86_64 3:19.03.5-3.el7

Dependency Installed:
  audit-libs-python.x86_64 0:2.8.5-4.el7           checkpolicy.x86_64 0:2.5-8.el7
  container-selinux.noarch 2:2.107-3.el7           containerd.io.x86_64 0:1.2.10-3.2.el7
  docker-ce-cli.x86_64 1:19.03.5-3.el7             libcgroup.x86_64 0:0.41-21.el7
  libsemanage-python.x86_64 0:2.5-14.el7           policycoreutils-python.x86_64 0:2.5-33.el7
  python-IPy.noarch 0:0.75-6.el7                   setools-libs.x86_64 0:3.3.8-4.el7

Complete!
[root@iZbp16e5pr6sbudyuvo3ruZ ~]#
```

2.3、列出并排序存储库中可用的版本。按版本号（从高到低）对结果进行排序

```
yum list docker-ce --showduplicates | sort -r
```

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# yum list docker-ce --showduplicates | sort -r
Loading mirror speeds from cached hostfile
Loaded plugins: fastestmirror
docker-ce.x86_64            3:19.03.5-3.el7                     docker-ce-stable
docker-ce.x86_64            3:19.03.4-3.el7                     docker-ce-stable
docker-ce.x86_64            3:19.03.3-3.el7                     docker-ce-stable
docker-ce.x86_64            3:19.03.2-3.el7                     docker-ce-stable
docker-ce.x86_64            3:19.03.1-3.el7                     docker-ce-stable
docker-ce.x86_64            3:19.03.0-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.9-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.8-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.7-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.6-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.5-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.4-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.3-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.2-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.1-3.el7                     docker-ce-stable
docker-ce.x86_64            3:18.09.0-3.el7                     docker-ce-stable
docker-ce.x86_64            18.06.3.ce-3.el7                    docker-ce-stable
docker-ce.x86_64            18.06.2.ce-3.el7                    docker-ce-stable
docker-ce.x86_64            18.06.1.ce-3.el7                    docker-ce-stable
docker-ce.x86_64            18.06.0.ce-3.el7                    docker-ce-stable
docker-ce.x86_64            18.03.1.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            18.03.0.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.12.1.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.12.0.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.09.1.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.09.0.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.06.2.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.06.1.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.06.0.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.03.3.ce-1.el7                    docker-ce-stable
docker-ce.x86_64            17.03.2.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.03.1.ce-1.el7.centos             docker-ce-stable
docker-ce.x86_64            17.03.0.ce-1.el7.centos             docker-ce-stable
Available Packages
[root@iZbp16e5pr6sbudyuvo3ruZ ~]#
```

2.3.2、通过其完整的软件包名称安装特定版本

> 该软件包名称是软件包名称（docker-ce）加上版本字符串（第二列），从第一个冒号（:）一直到第一个连字符，并用连字符（-）分隔。
> 例如：3:19.03.5-3.e17 对应 docker-ce-19.03.5

```
sudo yum install docker-ce-19.03.5 docker-ce-cli--19.03.5 containerd.io
```

2.4、启动Docker服务

```
sudo systemctl start docker
```

没有提示才是正确的

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# sudo systemctl start docker
[root@iZbp16e5pr6sbudyuvo3ruZ ~]#
```

通过运行 hello-world 映像来验证是否正确安装了 Docker Engine-Community 。

```
sudo docker run hello-world
```

表示运行本地hello-world镜像，本地是没有下载的，然后docker会尝试先从默认镜像仓库下载该镜像，随后启动镜像。

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:4df8ca8a7e309c256d60d7971ea14c27672fc0d10c5f303856d7bc48f8cc17ff
Status: Downloaded newer image for hello-world:latest

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://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

2.4、查看本地已经安装的镜像信息：

```
1 docker images
```

```
[root@iZbp16e5pr6sbudyuvo3ruZ ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world         latest              fce289e99eb9        11 months ago       1.84kB
[root@iZbp16e5pr6sbudyuvo3ruZ ~]#
```