1. 安装Docker
2. 安装CentOS7
3. 安装系统

从网上下载CentOS7的镜像，创建虚拟机，安装CentOS7系统。检查系统的网卡开启情况，使用ip addr命令，也可以安装net-tools来使用ifconfig命令。

1. 配置yum的源

之后，配置yum的配置文件，将阿里云的配置文件整个替换掉/etc/yum.repos.d目录中的CentOS-Base.repo配置文件，再通过以下命令添加阿里云的Docker yum源：

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

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

1. 安装指定版本的Docker
2. 查看服务器上的Docker

首先查看阿里云服务器上的Docker版本：

yum list docker-ce –showduplicates

1. 安装指定Docker

然后安装指定的docker版本和docker-selinux版本，如17.03.0：

# Install docker

# on a new system with yum repo defined, forcing older version and ignoring obsoletes introduced by 17.06.0

yum install -y --setopt=obsoletes=0 \

docker-ce-17.03.2.ce-1.el7.centos.x86\_64 \

docker-ce-selinux-17.03.2.ce-1.el7.centos.noarch

1. 启动Docker服务

systemctl enable docker

systemctl start docker

通过hello-world验证Docker的运行情况：

docker run hello-world

1. 设置Docker国内加速地址

申请阿里云的私人Docker镜像仓库地址，然后在/etc/docker目录下创建daemon.json文件，输入以下字符，包括你的Docker 仓库地址：

{

"registry-mirrors": ["<your accelerate address>"]

}

然后重启deamon服务：

systemctl daemon-reload

systemctl restart dockerr

通过查看docker信息，就可以看到docker的镜像仓库地址的变化：

docker info

1. Docker命令
2. Docker指令的格式

docker 命令关键字 一系列参数

如：

docker run --name xx –link db:mysql -p 8080:80 -d xxx

1. 命令关键字

docker info : 作为守护进程的系统资源设置

docker search: 查询Docker仓库中的镜像

docker pull : 从Docker仓库中拉取镜像

docker images: 查询当前本地的镜像

docker rmi : 删除Docker镜像，可以通过镜像的名称删除，也可以通过镜像的ID号前四位来简便地删除，要注意，镜像只有在没有容器使用该镜像时可以删除

docker ps : 查询当前的运行着的容器，加上-a参数，能够查询出关闭的容器,加上--no-trunc参数，可以查看完整的128位容器ID

docker run ; 容器的创建启动，可通过添加--restart=always设置为与docker同时启动，不会因为docker重启而停止；

添加 -h x.xx.xxx 参数，设置容器的主机名称

添加 --dns xx.xx.xx.xx 设置容器使用的DNS服务器

添加 --dns-search 设置DNS搜素

添加 --add-host hostname:IP 注入主机名与IP的解析关系

添加 --rm 设置在容器停止时自动删除容器

docker start/stop : 容器启动停止，可通过容器ID或容器别名操作

docker rm : 删除容器，注意只有在容器停止时可以删除

docker inspect : 查看详细的容器基本信息

docker logs : 查看容器的日志

docker stats : 查看容器所占用的系统资源，如CPU，硬盘，IO等

docker exec 容器名 容器内执行的命令： 可以向容器中发送命令

docker exec -it 容器名 /bin/bash : 进入容器内部的bash中，执行命令

docker命令除了可以单条地使用外，还可以赋值，解析变量，嵌套使用，如将一条命令的结果传递给另一条命令：

docker rm -f $(docker ps -a -q)

1. Docker-Compose
2. 单一容器的管理

每个容器创建后，都会分配一个CONTAINER ID作为容器的唯一表示，后续对容器的启动，停止，修改，删除等操作，都会通过CONTAINER ID来完成。

1. 多容器管理

Docker提倡一个容器一个进程，假设一个服务需要多个进程所组成，就需要多个容器组成一个系统，相互分工和配合对外提供完整服务。Docker-Compose就是这样一个容器编排工具，能够让用户通过在一个模板文件（YAML格式）中定义一组相关联的容器，编排不同容器之间的启动优先级等等。

1. Docker-Compose的安装

通过以下命令，高速安装Docker-Compose：

curl -L https://get.daocloud.io/docker/compose/releases/download/1.25.5/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose

chmod +x /usr/local/bin/docker-compose

通过docker-compose --version 查看安装是否成功。

1. Docker-Compose的用法

通过在 docker-compose 命令后添加以下的参数来使用：

1. -f

指定使用的yaml文件的位置

1. ps

显示Docker-Compose编排的所有容器信息 。

1. restart

重新启动容器。

1. logs

查看日志信息。

1. config -q

验证yaml配置文件是否正确。

1. stop/start

停止，启动容器。

1. up -d

启动容器项目

1. pause/unpause

暂停容器，恢复容器。

1. rm

删除容器