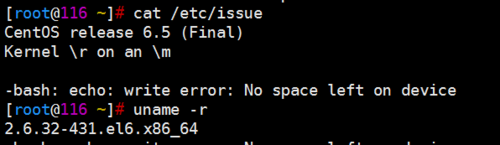
一、Centos6升级内核

1、操作系统为centos6.5，内核为2.6



2、先更新nss

yum update nss

728b46102889.png

3、安装elrepo的yum源，升级内核需要使用elrepo的yum源，在安装yum源之前还需要我们导入elrepo的key

[root@localhost ~]# rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org

[root@localhost ~]# rpm -Uvh http://elrepo.org/elrepo-release-6-8.el6.elrepo.noarch.rpm



4、升级内核

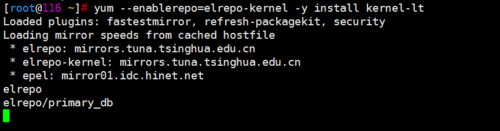
在yum的elrepo源中有ml和lt两种内核，其中ml(mainline)为最新版本的内核，lt为长期支持的内核。

安装ml内核使用如下命令：

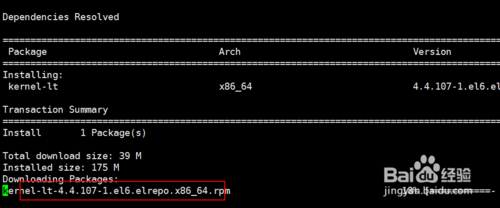
yum --enablerepo=elrepo-kernel -y install kernel-ml

安装lt内核使用如下命令：

yum --enablerepo=elrepo-kernel -y install kernel-lt



5、我们这里选择安装lt内核

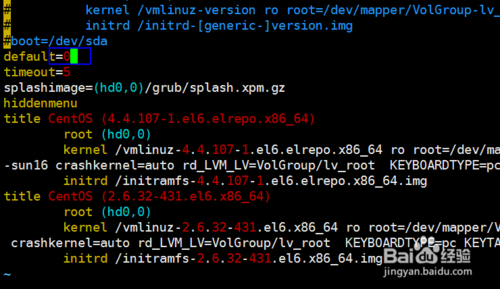


6、修改grub.conf文件

内核升级完毕后修改内核的启动顺序

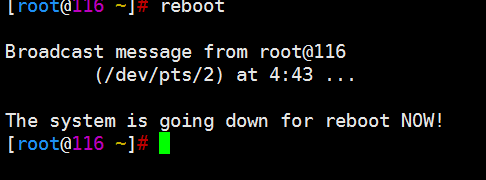
vim /etc/grub.conf

default=0



7、重启操作系统

reboot



8、操作系统还是6.5，内核升级成功

[root@116 ~]# uname -r

4.4.107-1.el6.elrepo.x86\_64

二、安装 Docker

以下 命令都是在 root 权限下运行的。

Docker是Enterprise Linux（EPEL）的额外包的一部分，EPEL是用于RHEL发行版的非标准包的社区库。首先，我们将安装EPEL仓库：

# rpm -iUvh http://dl.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm

更新 yum 源：

# yum update -y

现在让我们通过安装 docker-io 软件包来安装 Docker：

# yum -y install docker-io

安装完成后，我们需要启动 Docker 守护进程：

# service docker start

最后，可选地，我们让 Docker 在服务器启动时启动：

# chkconfig docker on

经过以上步骤，Docker 就已经安装成功了！来跑跑试试：

# docker info #查看 docker 基本信息

启动运行nginx

# docker run -d --name mynginx nginx #启动nginx镜像，没有会自动pull

# -p 指定端口映射，将80映射为host的91

进入容器方法（推荐）：

[root@localhost docker] docker inspect --format "{{.State.Pid}}" mynginx #获取容器pid 19769
[root@localhost docker] nsenter --target 19769 --mount --uts --ipc --net --pid #进入容器(推荐方法)

就这样！你现在在一个 centos Dockers 容器里面使用一个 bash shell。

要断开或分离 shell 而不退出，使用 Ctrl-p + Ctrl-q。（注：就是先按 Ctrl + p，再按 Ctrl + q）。

以上就是在 CentOS6.5 中安装 Docker 的过程了。

**docker命令：**

存储镜像：

# docker save -o ubuntu\_14.04.tar ubuntu:14.04

载入镜像：

# docker load < ubuntu\_14.04.tar 或者使用
# cat ubuntu.tar | docker import - test/ubuntu:v1.0

停止ubuntu容器

# docker stop myUbuntu

移除本地镜像：

# docker rmi training/sinatra #使用docker rmi <IMAGE ID>删除，但是必须先要docker rm myUbuntu才能删除镜像

清理所有未打过标签的本地镜像：

# docker rmi $(docker images -q -f "dangling=true")

其中 -q 和 -f 是 quiet，–filter 的缩写, 完整的命令其实可以写着下面这样，是不是更容易理解一点？

# docker rmi $(docker images --quiet --filter "dangling=true")

注： 容器是否会长久运行，是和docker run指定的命令有关，和 -d 参数无关。

要获取容器的输出信息，可以通过 docker logs 命令。

# docker logs [container ID or NAMES]

删除容器：

# docker rm 默认并不会删除运行中的容器

有关容器和镜像的底层信息:

# docker inspect container/image

可以查看：容器实例的IP地址、端口绑定列表、特定端口映射的搜索、收集配置的详细信息

从容器内复制文件到指定的路径上：

# docker cp container:path hostpath

使用Dockerfile来构建镜像：

# docker build [options] PATH | URL

–rm=true表示构建成功后，移除所有中间容器

–no-cache=false表示在构建过程中不使用缓存

启动bash控制台并修改index.html内容

# docker exec -it webserver bash

查看容器修改内容

# docker diff webserver