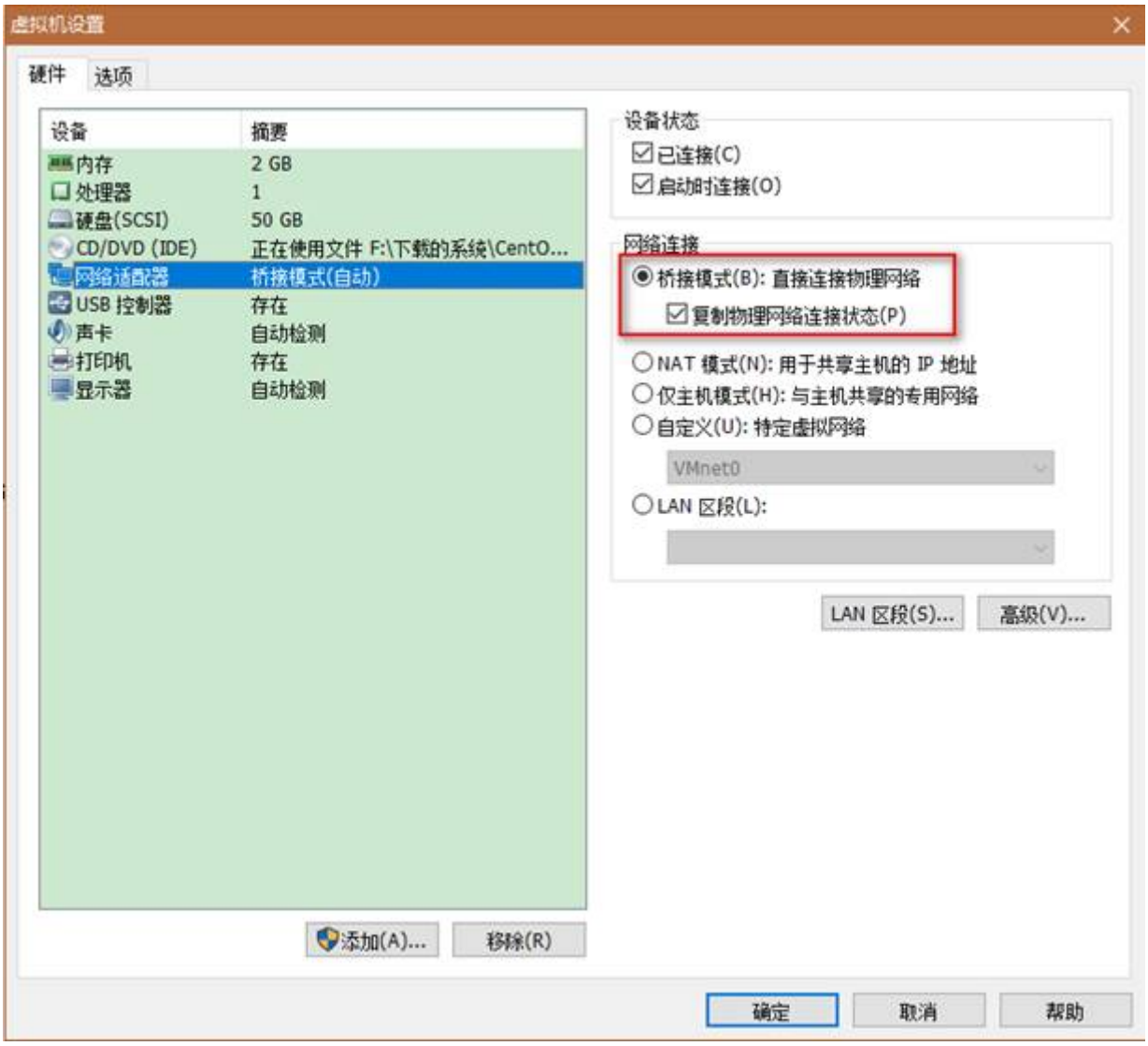
docker的安装

原创 Mr大表哥 2017-02-27 21:40:49 评论(0) 254人阅读

CentOS7的docker安装：
方法一：
Prerequisites（先决条件）
Docker requires a 64-bit installation regardless of yourCentOS version. Also, your kernel must be 3.10 at minimum, which CentOS 7 runs.
To check your current kernel version, open a terminaland use uname -r to display your kernel version（翻译：docker要求64位安装，不管你的CentOS版本。另外，你的内核必须至少是3.10，而CentOS 7运行。要查看当前的内核版本，打开一个终端使用uname -r显示您的内核版本）。



```
[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens33
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=0367ef04-9f53-43d3-98aa-ff0ddbc7bc15
DEVICE=ens33
ONBOOT=yes
[root@localhost ~]# systemctl restart network
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:23:c0:9c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global dynamic ens33
       valid_lft 86397sec preferred_lft 86397sec
    inet6 fe80::be0:9b99:ed88:9bd8/64 scope link
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN qlen 1000
    link/ether 52:54:00:ac:2c:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN qlen 1000
    link/ether 52:54:00:ac:2c:0f brd ff:ff:ff:ff:ff:ff
[root@localhost ~]#
```

正式开始做Centos7的docker安装：

1）先确定内核版本是不是3.8以上

```
[root@localhost ~]# uname -r
3.10.0-514.el7.x86_64
[root@localhost ~]#
```

2）Docker 软件包已经包括在默认的 CentOS-Extras 软件源里。因此想要安装docker，只需要运行下面的 yum 命令

```
[root@localhost ~]# yum -y install docker
```

```
[root@localhost ~]# yum -y install docker
已加载插件：fastestmirror, langpacks
/var/run/yum.pid 已被锁定，PID 为 15269 的另一个程序正在运行。
Another app is currently holding the yum lock; waiting for it to exit...
  另一个应用程序是：PackageKit
    内存：146 M RSS （1.0 GB VSZ）
    已启动： Fri Jan 13 14:57:39 2017 - 04:30之前
    状态  ：睡眠中，进程ID：15269
Another app is currently holding the yum lock; waiting for it to exit...
  另一个应用程序是：PackageKit
    内存：146 M RSS （1.0 GB VSZ）
    已启动： Fri Jan 13 14:57:39 2017 - 04:32之前
    状态  ：睡眠中，进程ID：15269
^C

Exiting on user cancel.
[root@localhost ~]# rm -rf /var/run/yum.pid
[root@localhost ~]# yum -y install docker
```

Yum安装docker,这时会报一个错误,这时按ctrl+c将其强制停止。报错的原因是因为PID15269被占用，删除/var/run.yum.pid，然后重新执行yum安装docker即可。

3）启动 Docker 服务：

安装完成后，使用下面的命令来启动docker服务，并将其设置为开机启动

```
[root@localhost ~]# systemctl start docker.service
[root@localhost ~]# systemctl enable docker.service
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[root@localhost ~]#
```

4）查看docker版本 （docker version和docker info都是查看版本的命令，不同之处在于，docker version 显示 Docker 版本信息，而docker info 显示 Docker 系统信息，包括镜像和容器数）

```
[root@localhost ~]# docker version
Client:
 Version:         1.10.3
 API version:     1.22
 Package version: docker-common-1.10.3-59.el7.centos.x86_64
 Go version:      go1.6.3
 Git commit:      3999ccb-unsupported
 Built:           Thu Dec 15 17:24:43 2016
 OS/Arch:         linux/amd64

Server:
 Version:         1.10.3
 API version:     1.22
 Package version: docker-common-1.10.3-59.el7.centos.x86_64
 Go version:      go1.6.3
 Git commit:      3999ccb-unsupported
 Built:           Thu Dec 15 17:24:43 2016
 OS/Arch:         linux/amd64
[root@localhost ~]#
```

```
[root@localhost ~]# docker info
Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 0
Server Version: 1.10.3
Storage Driver: devicemapper
 Pool Name: docker-253:0-100669301-pool
 Pool Blocksize: 65.54 kB
 Base Device Size: 10.74 GB
 Backing Filesystem: xfs
 Data file: /dev/loop0
 Metadata file: /dev/loop1
 Data Space Used: 11.8 MB
 Data Space Total: 107.4 GB
 Data Space Available: 46.06 GB
 Metadata Space Used: 581.6 kB
 Metadata Space Total: 2.147 GB
 Metadata Space Available: 2.147 GB
 Udev Sync Supported: true
 Deferred Removal Enabled: false
 Deferred Deletion Enabled: false
 Deferred Deleted Device Count: 0
 Data loop file: /var/lib/docker/devicemapper/devicemapper/data
```

5）验证docker安装是否正确

```
[root@localhost ~]# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
Trying to pull repository docker.io/library/hello-world ...
latest: Pulling from docker.io/library/hello-world

c04b14da8d14: Pull complete
Digest: sha256:0256e8a36e2070f7bf2d0b0763dbabdd67798512411de4cdcf9431a1feb60fd9
Status: Downloaded newer image for docker.io/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.
 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 Hub account:
 https://hub.docker.com

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

[root@localhost ~]#
```

看见到这一行就说明docker安装成功了。

方法二：

1）配置yum文件

可以用老方法去配置文件，也可以用脚本的方式，脚本的方式如下：

```
# sudo tee /etc/yum.repos.d/docker.repo<<-'EOF'
[dockerrepo]
name=DockerRepository
baseurl=https://yum.dockerproject.org/repo/main/centos/7/
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
EOF
```

（注：如果我们既想把输出保存到文件中，又想在屏幕上看到输出内容，就可以使用tee命令了。tee命令读取标准输入，把这些内容同时输出到标准输出和（多个）文件中）

我下面用老方法配置yum文件

```
[root@localhost ~]# vim /etc/yum.repos.d/docker.repo
```

```
[dockerrepo]
name=DockerRepository
baseurl=https://yum.dockerproject.org/repo/main/centos/7/
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
```

2）安装docker

```
[root@localhost ~]# sudo yum -y install docker-engine
```

3）开启docker服务

```
[root@localhost ~]# sudo service docker start
Redirecting to /bin/systemctl start  docker.service
[root@localhost ~]#
```

4）验证docker安装是否正确

```
[root@localhost ~]# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world

c04b14da8d14: Pull complete
Digest: sha256:0256e8a36e2070f7bf2d0b0763dbabdd67798512411de4cdcf9431a1feb60fd9
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.
 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 Hub account:
 https://hub.docker.com

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

docker默认使用的是unix socket：

```
[root@localhost ~]# ss -ax | grep docker
u_str  LISTEN     0      128    /var/run/docker.sock 78450                 * 0
u_str  LISTEN     0      128    /var/run/docker/libcontainerd/docker-containerd.sock 78471                 *
0
u_str  LISTEN     0      128    /run/docker/libnetwork/a8f67075470fd4ab64f8d05e15ae59ea0ec9d3d1eb92145e60a260
8f1fc86dbe.sock 78738                 * 0
u_str  ESTAB      0      0      /var/run/docker/libcontainerd/docker-containerd.sock 78475                 *
78474
[root@localhost ~]#
```

Centos6.5的docker安装：

在RedHat/CentOS环境下安装Docker。官方文档要求Linux kernel至少3.8以上，且docker只能运行在64位的系统中。由于RHEL6和CentOS6的内核版本为2.6，因此必须要先升级内核。

1）先确定系统版本和内核版本

```
[root@localhost ~]# uname -r
2.6.32-431.el6.x86_64
[root@localhost ~]# cat /etc/issue
CentOS release 6.5 (Final)
Kernel \r on an \m

[root@localhost ~]#
```

2）安装elrepo yum源

```
[root@localhost ~]# rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
[root@localhost ~]# rpm -Uvh http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noarch.rpm
Retrieving http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noarch.rpm
Preparing...                ########################################### [100%]
   1:elrepo-release         ########################################### [100%]
[root@localhost ~]#
```

安装elrepo yum源之前先要导入elrepo的key；
第二步是安装elrepo yum源。

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

安装lt内核 yum --enablerepo=elrepo-kernel -y installkernel-lt

由于是生产环境我们安装稳定的lt内核

```
[root@localhost ~]# yum --enablerepo=elrepo-kernel -y install kernel-lt
Loaded plugins: fastestmirror, refresh-packagekit, security
Determining fastest mirrors
 * base: mirrors.tuna.tsinghua.edu.cn
 * elrepo: repos.lax-noc.com
 * elrepo-kernel: repos.lax-noc.com
 * extras: mirror.lzu.edu.cn
 * updates: mirrors.neusoft.edu.cn
elrepo                                                   | 2.9 kB     00:00
elrepo/primary_db                                        | 727 kB     00:01
elrepo-kernel                                            | 2.9 kB     00:00
elrepo-kernel/primary_db                                 |  19 kB     00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package kernel-lt.x86_64 0:3.10.104-1.el6.elrepo will be installed
```

它自动安装的是3.10的版本。

4）修改grub.conf文件vim /etc/grub.conf

default=0

```
[root@localhost ~]# vim /etc/grub.conf
```

```
 9 #boot=/dev/sda
10 default=0
11 timeout=5
12 splashimage=(hd0,0)/grub/splash.xpm.gz
13 hiddenmenu
14 title CentOS (3.10.104-1.el6.elrepo.x86_64)
15         root (hd0,0)
```

表示第一个title下的内容为默认启动的kernel（一般新安装的内核在第一个位置），也就是说让3.10的内核设为默认启动内核。

5）修改完了就重启系统让其生效

```
[root@localhost ~]# reboot
```

6）再次查看内核版本

```
[root@localhost ~]# uname -r
3.10.104-1.el6.elrepo.x86_64
[root@localhost ~]#
```

其实升级内核还一种办法命令如下所示：

（但还是建议用第一种方法）

1、yum 安装带 aufs 模块的 3.10 内核

```
#cd /etc/yum.repos.d
#wget http://www.hop5.in/yum/el6/hop5.repo
#yum install kernel-ml-aufs kernel-ml-aufs-devel
```

2、修改 grub 的主配置文件/etc/grub.conf，设置 default=0，表示第一个 title 下的内容为默认启动的 kernel（一般新安装的内核在第一个位置），重启系统，这时候你的内核就成功升级了。

```
[root@localhost ~]# uname -r
3.10.5-3.el6.x86_64
```

查看内核是否支持 aufs：

```
[root@localhost ~]# grep aufs /proc/filesystems
nodev   aufs
```

7）对于 CentOS6，可以使用EPEL库安装 Docker

（科普知识：EPEL（Extra Packages for EnterpriseLinux，企业版Linux的额外软件包）是Fedora小组维护的一个软件仓库项目，为RHEL/CentOS提供他们默认不提供的软件包。这个源兼容RHEL及像CentOS和ScientificLinux这样的衍生版本。我们可以很容易地通过yum命令从EPEL源上获取上万个在CentOS自带源上没有的软件。EPEL提供的软件包大多基于其对应的Fedora软件包，不会与企业版Linux发行版本的软件发生冲突或替换其文件。）

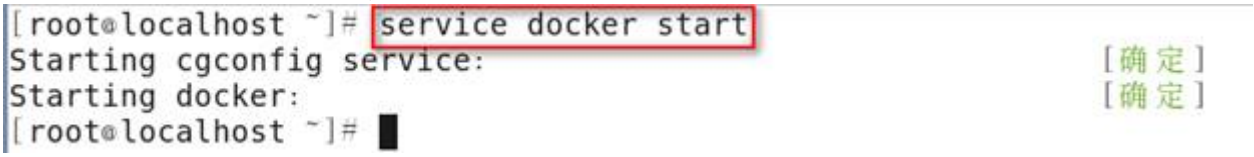
更多关于EPEL 项目的细节可以到以下网站获取：https://fedoraproject.org/wiki/EPEL

```
[root@localhost ~]# yum install http://mirrors.yun-idc.com/epel/6/i386/epel-release-6-8.noarch.rpm
```

可以在yum和install中间加个"-y"，这样的话就不用手动输入y了。

```
Total size: 22 k
Installed size: 22 k
Is this ok [y/N]: y
```

8）启动docker服务



至此6.5上安装docker就完成了！