

Docker环境离线安装（二进制包部署模式）

1、下载docker的安装文件

https://download.docker.com/linux/static/stable/x86_64/

← → ↺ 🔒 download.docker.com/linux/static/stable/x86_64/

Index of linux/static/stable/x86_64/

../	
docker-17.03.0-ce.tgz	2019-10-18 22:27:42 26.5 MiB
docker-17.03.1-ce.tgz	2019-10-18 22:27:42 26.5 MiB
docker-17.03.2-ce.tgz	2019-10-18 22:27:43 26.5 MiB
docker-17.06.0-ce.tgz	2019-10-18 22:27:43 28.3 MiB
docker-17.06.1-ce.tgz	2019-10-18 22:27:43 28.3 MiB
docker-17.06.2-ce.tgz	2019-10-18 22:27:43 28.3 MiB
docker-17.09.0-ce.tgz	2019-10-18 22:27:43 29.0 MiB
docker-17.09.1-ce.tgz	2019-10-18 22:27:43 29.0 MiB
docker-17.12.0-ce.tgz	2019-10-18 22:27:43 32.7 MiB
docker-17.12.1-ce.tgz	2019-10-18 22:27:43 32.7 MiB
docker-18.03.0-ce.tgz	2019-10-18 22:27:43 37.0 MiB
docker-18.03.1-ce.tgz	2019-10-18 22:27:44 37.0 MiB
docker-18.06.0-ce.tgz	2019-10-18 22:27:44 41.8 MiB
docker-18.06.1-ce.tgz	2019-10-18 22:27:44 41.8 MiB
docker-18.06.2-ce.tgz	2019-10-18 22:27:44 41.8 MiB
docker-18.06.3-ce.tgz	2019-10-18 22:27:44 41.8 MiB
docker-18.09.0.tgz	2019-10-18 22:27:45 44.9 MiB
docker-18.09.1.tgz	2019-10-18 22:27:45 45.8 MiB
docker-18.09.2.tgz	2019-10-18 22:27:46 45.8 MiB
docker-18.09.3.tgz	2019-10-18 22:27:46 45.8 MiB

2、将docker-18.06.3-ce.tgz文件上传到centos7-linux系统上，用ftp工具上传即可

3、解压

```
tar -zxvf docker-18.06.3-ce.tgz
```

4、将解压出来的docker文件复制到 /usr/bin/ 目录下

```
cp docker/* /usr/bin/
```

5、进入/etc/systemd/system/目录,并创建docker.service文件

```
cd /etc/systemd/system/  
touch docker.service
```

6、打开docker.service文件,将以下内容复制

```
vim docker.service
```

```
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
After=network-online.target firewall.service
Wants=network-online.target

[Service]
Type=notify
# the default is not to use systemd for cgroups because the delegate issues
# still
# exists and systemd currently does not support the cgroup feature set required
# for containers run by docker
ExecStart=/usr/bin/dockerd --selinux-enabled=false --insecure-
registry=192.168.200.128
ExecReload=/bin/kill -s HUP $MAINPID
# Having non-zero Limit*s causes performance problems due to accounting overhead
# in the kernel. We recommend using cgroups to do container-local accounting.
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
# Uncomment TasksMax if your systemd version supports it.
# Only systemd 226 and above support this version.
#TasksMax=infinity
TimeoutStartSec=0
# set delegate yes so that systemd does not reset the cgroups of docker
# containers
Delegate=yes
# kill only the docker process, not all processes in the cgroup
KillMode=process
# restart the docker process if it exits prematurely
Restart=on-failure
StartLimitBurst=3
StartLimitInterval=60s

[Install]
WantedBy=multi-user.target
```

注意：--insecure-registry=192.168.200.128 此处改为你自己服务器ip

7、给docker.service文件添加执行权限

```
chmod 777 /etc/systemd/system/docker.service
```

8、重新加载配置文件（每次有修改docker.service文件时都要重新加载下）

```
systemctl daemon-reload
```

9、启动

```
systemctl start docker
```

10、设置开机启动

```
systemctl enable docker.service
```

11、查看docker状态

```
systemctl enable docker.service
```

出现下面这个界面就代表docker安装成功。

```
[root@localhost system]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/etc/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://docs.docker.com
[root@localhost system]# systemctl start docker
[root@localhost system]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/etc/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since 四 2020-05-14 16:50:32 CST; 2s ago
     Docs: https://docs.docker.com
  Main PID: 2249 (dockerd)
    Tasks: 17
   Memory: 21.1M
   CGroup: /system.slice/docker.service
           └─2249 /usr/bin/dockerd --selinux-enabled=false --insecure-registry=192.168.200.128
             └─2255 docker-containerd --config /var/run/docker/containerd/containerd.toml

5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.492379906+08:00" level=info msg="pickfi
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.492951532+08:00" level=info msg="pickfi
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.492975666+08:00" level=info msg="Loading
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.678916697+08:00" level=info msg="Default
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.798966587+08:00" level=info msg="Loading
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.830836003+08:00" level=info msg="Docker
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.831139821+08:00" level=info msg="Daemon
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.855238884+08:00" level=warning msg="Cou
5月 14 16:50:32 localhost.localdomain dockerd[2249]: time="2020-05-14T16:50:32.863758382+08:00" level=info msg="API lis
5月 14 16:50:32 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
```

12、配置镜像加速器,默认是到国外拉取镜像速度慢,可以配置国内的镜像如: 阿里、网易等等。下面配置一下网易的镜像加速器。打开docker的配置文件: /etc/docker/daemon.json文件:

```
vim /etc/docker/daemon.json
```

配置如下:

```
{"registry-mirrors": ["http://hub-mirror.c.163.com"]}
```

配置完后:wq保存配置并重启docker 一定要重启不然加速是不会生效的!!!

```
service docker restart
```

Docker版本离线升级（二进制包部署模式）

1.目标

通过离线软件包方式，将 docker 19.03.05 升级为 20.10.12版本，升级完成后保证原有镜像不丢失，原有容器重启后正常。

操作系统：Redhat 7.6

2.备份数据卷，容器，镜像

简单的说就是挂载路径，以及/var/lib/docker/路径下的所有东西都备份。/var/lib/docker/路径下，容器，镜像，网络配置等等一系列的东西都在这下面。

```
cp -r source_path target_path
```

3.下载最新软件包

https://download.docker.com/linux/static/stable/x86_64/

```
docker-20.10.12.tgz
```

4.升级

4.1停止docker服务

```
systemctl stop docker
```

4.1卸载Docker服务

查看当前版本

```
rpm -qa | grep docker
```

卸载软件包

```
yum erase docker \
docker-client \
docker-client-latest \
docker-common \
docker-latest \
docker-latest-logrotate \
docker-logrotate \
docker-selinux \
docker-engine-selinux \
docker-engine \
docker-ce
```

删除相关配置文件

```
find /etc/systemd -name '*docker*' -exec rm -f {} \;
find /etc/systemd -name '*docker*' -exec rm -f {} \;
find /lib/systemd -name '*docker*' -exec rm -f {} \;
```

4.3升级

按《Docker环境安装：centos7 离线安装docker》章节进行部署即可，docker.service文件若做过个性化改造，可保留

5.启动服务，验证版本，查看镜像，启动容器，删除备份

```
systemctl start docker
```

验证版本，查看镜像，启动容器，删除备份省略。

Docker拉取arm镜像- 拉取指定平台镜像

1.1 方法一

如上manifest实验功能开启后，可通过如下命令拉取其他CPU平台的镜像。

```
#x86平台docker拉取arm镜像
docker pull --platform=arm64 镜像名:版本

#示例
docker pull --platform=arm64 nginx:latest
```

--platform：该参数是用于拉取指定平台的镜像，也是实验性功能，在开启manifest功能后就会出现。通过该参数可以手动指定需要的CPU平台镜像，而不用自动去识别。

补充：docker manifest简介

使用镜像创建一个容器，该镜像必须与 Docker 宿主机系统架构一致，例如x86_64 架构的系统中只能使用x86_64的镜像创建容器。

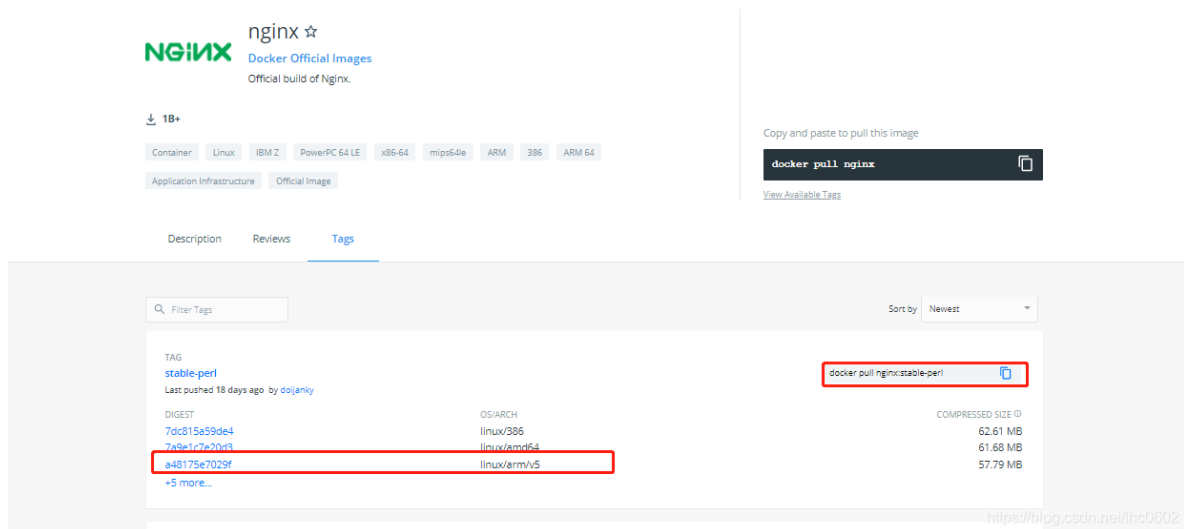
docker manifest特性可支持用户在不同系统架构的机器上分别运行不同的架构的镜像。这一点基本不需要用户做任何适配，非常的方便。

manifest list是一个镜像清单列表，用于存放多个不同os/arch的镜像信息；主要用到manifest的目的，其实还是多用于存放不同的os/arch信息，也就是方便我们在不同的CPU架构（arm或者x86）或者操作系统中，通过一个镜像名称拉取对应架构或者操作系统的镜像，（这个尤其是在K8S中，对于异构CPU的服务器中的镜像显得尤为有效。）

注：manifest文件仅仅是针对于已经在仓库中的镜像！！！换句话说，就是这个镜像是从仓库中pull下来的！如果这个镜像是自己build的，需要先push到仓库中，否则，这个镜像是没有manifest文件的！！同样的，如果你pull了一个镜像，tag了一下，再去看这个manifest文件，也是没有的，因为tag后的镜像不在镜像仓库中。

3.2 方法二

通过镜像的 sha256值拉取（大概率会失败）



#通过sha256值拉取镜像

docker pull nginx:stable-

perl@sha256:a48175e7029f0ae21b8b4e2526d6c3dd7278a8479be0e666d729b6234108f4e1

3.2 方法三

购买aws云主机完成镜像拉取、同步：

#step1: 购买aws的arm架构ECS主机

#step2: 在服务器上拉取arm版本镜像

#step3: 导出镜像，推送至dockerhub私人仓库

#step4: x86个人虚拟机下载镜像，导出文件