



链滴

搭建 docker 之路 - 初识 (1)

作者: [cuijianzhe](#)

原文链接: <https://ld246.com/article/1553592789196>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h2 id="一--安装docker环境">一、 安装 docker 环境</h2>

<h2 id="1-1-安装">1.1 安装</h2>

<p><code>[root@host ~]# yum install -y docker</code>

查看信息: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]# systemctl start docker
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
systemctl enable docker
</span></span><span class="highlight-line"><span class="highlight-cl">Created symlink f
om /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/
ocker.service.
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker version
</span></span><span class="highlight-line"><span class="highlight-cl">Client:
</span></span><span class="highlight-line"><span class="highlight-cl"> Version:      1.13
1
</span></span><span class="highlight-line"><span class="highlight-cl"> API version:   1.
6
</span></span><span class="highlight-line"><span class="highlight-cl"> Package version:
ocker-1.13.1-94.gitb2f74b2.el7.centos.x86_64
</span></span><span class="highlight-line"><span class="highlight-cl"> Go version:    g
1.10.3
</span></span><span class="highlight-line"><span class="highlight-cl"> Git commit:    b2
74b2/1.13.1
</span></span><span class="highlight-line"><span class="highlight-cl"> Built:        Tue
ar 12 10:27:24 2019
</span></span><span class="highlight-line"><span class="highlight-cl"> OS/Arch:      lin
x/amd64
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Server:
</span></span><span class="highlight-line"><span class="highlight-cl"> Version:      1.13
1
</span></span><span class="highlight-line"><span class="highlight-cl"> API version:   1.
6 (minimum version 1.12)
</span></span><span class="highlight-line"><span class="highlight-cl"> Package version:
ocker-1.13.1-94.gitb2f74b2.el7.centos.x86_64
</span></span><span class="highlight-line"><span class="highlight-cl"> Go version:    g
1.10.3
</span></span><span class="highlight-line"><span class="highlight-cl"> Git commit:    b2
74b2/1.13.1
</span></span><span class="highlight-line"><span class="highlight-cl"> Built:        Tue
ar 12 10:27:24 2019
</span></span><span class="highlight-line"><span class="highlight-cl"> OS/Arch:      lin
x/amd64
</span></span><span class="highlight-line"><span class="highlight-cl"> Experimental:  fa
se
</span></span></code></pre>
```

<h2 id="1-2-下载docker镜像">1.2 下载 docker 镜像</h2>

<p></p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]# docker search centos //从dockerHub中查找符合条件的镜像
```

	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
o/centos	The official build of CentOS.	5270	[OK]	
o/ansible/centos7-ansible	Ansible on Centos7	121		[OK]
o/jdeathe/centos-ssh	CentOS-6 6.10 x86_64 / CentOS-7 7.5.1804 x...	108		
o/consol/centos-xfce-vnc	Centos container with "headless" VNC sessi...	83		
o/imagine10255/centos6-lnmp-php56	centos6-lnmp-php56	52		
o/centos/mysql-57-centos7	MySQL 5.7 SQL database server	49		
o/tutum/centos	Simple CentOS docker image with SSH access	44		
o/gluster/gluster-centos	Official GlusterFS Image [CentOS-7 + Glu...	40		[
o/openshift/base-centos7	A Centos7 derived base image for Source-To...	40		
o/centos/postgresql-96-centos7	PostgreSQL is an advanced Object-Relationa...	37		
o/centos/python-35-centos7	Platform for building and running Python 3...	34		
o/kinogmt/centos-ssh	CentOS with SSH	26		[OK]
o/centos/httpd-24-centos7	Platform for running Apache httpd 2.4 or b...	22		
o/centos/php-56-centos7	Platform for building and running PHP 5.6 ...	20		
o/openshift/jenkins-2-centos7	A Centos7 based Jenkins v2.x image for use...	20		
o/pivotaldata/centos-gpdb-dev	CentOS image for GPDB development. Tag nam...	10		
o/openshift/wildfly-101-centos7	A Centos7 based WildFly v10.1 image for us...	6		
o/openshift/jenkins-1-centos7	DEPRECATED: A Centos7 based Jenkins v1.x i...	4		
o/darksheer/centos	Base Centos Image -- Updated hourly	3		[O

```

</span></span><span class="highlight-line"><span class="highlight-cl">docker.io  docker.
o/pivotaldata/centos          Base centos, freshened up a little with a ...  3
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io  docker.
o/pivotaldata/centos-mingw     Using the mingw toolchain to cross-compile...  2

</span></span><span class="highlight-line"><span class="highlight-cl">docker.io  docker.
o/blacklabelops/centos        CentOS Base Image! Built and Updates Daily!  1
[OK]
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io  docker.
o/openshift/wildfly-81-centos7  A Centos7 based WildFly v8.1 image for use...  1

</span></span><span class="highlight-line"><span class="highlight-cl">docker.io  docker.
o/pivotaldata/centos-gcc-toolchain  CentOS with a toolchain, but unaffiliated ...  1

</span></span><span class="highlight-line"><span class="highlight-cl">docker.io  docker.
o/smartentry/centos           centos with smartentry  0
</span></span></code></pre>
<p><strong>从公网 dockerhub 上拉取镜像 image</strong></p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[root@host ~]# docker pull docker.io/centos
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Using default tag:
atest
</span></span><span class="highlight-line"><span class="highlight-cl">Trying to pull rep
sitory docker.io/library/centos ...
</span></span><span class="highlight-line"><span class="highlight-cl">latest: Pulling fro
docker.io/library/centos
</span></span><span class="highlight-line"><span class="highlight-cl">8ba884070f61: Pul
complete
</span></span><span class="highlight-line"><span class="highlight-cl">Digest: sha256:8d
87d68857f5bc9595793279b33d082b03713341ddec91054382641d14db861
</span></span><span class="highlight-line"><span class="highlight-cl">Status: Download
d newer image for docker.io/centos:latest
</span></span></code></pre>
<p>开启 NAT 转发功能</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[root@host ~]# vim /etc/sysctl.conf
</span></span><span class="highlight-line"><span class="highlight-cl">net.ipv4.ip_forwar
= 1
</span></span><span class="highlight-line"><span class="highlight-cl">//执行sysctl -p生效
</span></span><span class="highlight-line"><span class="highlight-cl">[root@host ~]# sy
ctl -p
</span></span><span class="highlight-line"><span class="highlight-cl">net.ipv4.ip_forwar
= 1
</span></span><span class="highlight-line"><span class="highlight-cl">[root@host ~]# sy
temctl restart docker
</span></span><span class="highlight-line"><span class="highlight-cl">[root@host ~]# d
cker images
</span></span><span class="highlight-line"><span class="highlight-cl">REPOSITORY
TAG          IMAGE ID      CREATED      SIZE
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
latest       9f38484d220f  11 days ago  202 MB
</span></span></code></pre>
<h2 id="1-3实际应用">1.3 实际应用: </h2>

```

<h3 id="例1-运行一个--container--并加载镜像-centos-运行起来这个实例后-在实例中执行--bin-bash-命令">例 1: 运行一个 container 并加载镜像 centos, 运行起来这个实例后, 在实例中执行 /bin/bash 命令</h3>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[root@host ~]# docker run -it docker.io/centos:latest /bin/bash  ##启动一个实例
</span></span><span class="highlight-line"><span class="highlight-cl">[root@ff20a65c595 /]# ls
</span></span><span class="highlight-line"><span class="highlight-cl">anaconda-post.log
bin dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
</span></span><span class="highlight-line"><span class="highlight-cl">[root@ff20a65c595 /]# cat /etc/redhat-release
</span></span><span class="highlight-line"><span class="highlight-cl">CentOS Linux rele
se 7.6.1810 (Core)
</span></span><span class="highlight-line"><span class="highlight-cl">[root@ff20a65c595 /]# exit
</span></span><span class="highlight-line"><span class="highlight-cl">exit
</span></span><span class="highlight-line"><span class="highlight-cl">[root@host ~]#
</span></span></code></pre>
```

<p>docker 常用参数:

run 运行

-i 以交互模式运行容器, 通常不 -t 同时使用;

-t 为容器重新分配一个伪输入终端, 通常和 -i 同时使用; </p>

<h3 id="例2-在-container-中启动一个长久运行的进程-不断向-stdin-输出--hello-world---模拟一后台运行的服务">例 2: 在 container 中启动一个长久运行的进程, 不断向 stdin 输出 hello world 。模拟一个后台运行的服务</h3>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]# docker run -d docker.io/centos:latest /bin/sh -c "while true;do echo he
lo world; sleep 1; done"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">6246ea4f21efb67
18894fa78940210236e05d2e2953f942e5666488015b0f31
</span></span><span class="highlight-line"><span class="highlight-cl"> //容器的ID
</span></span></code></pre>
```

<h3 id="例3-从一个容器中取日志-查看输出的内容-可用于后期检查-docker-实例在标准输出中弹出错误信息或正常的信息-">例 3: 从一个容器中取日志, 查看输出的内容, 可用于后期检查 docker 实在标准输出中弹出的错误信息或正常的信息。</h3>

<p>语法: docker logs 容器实例的 Name/ID</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]# docker logs 6246ea4f21efb67018894fa78940210236e05d2e2953f942e
666488015b0f31
</span></span><span class="highlight-line"><span class="highlight-cl">hello world
</span></span><span class="highlight-line"><span class="highlight-cl">hello world
</span></span><span class="highlight-line"><span class="highlight-cl">hello world
</span></span><span class="highlight-line"><span class="highlight-cl">hello world
</span></span><span class="highlight-line"><span class="highlight-cl">hello world
</span></span><span class="highlight-line"><span class="highlight-cl">hello world
</span></span><span class="highlight-line"><span class="highlight-cl">hello world
</span></span></code></pre>
```

<p>查看正在运行的容器</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]# docker ps
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
```

```

IMAGE          COMMAND          CREATED          STATUS          PORTS
NAMES
</span></span><span class="highlight-line"><span class="highlight-cl">6246ea4f21ef
docker.io/centos:latest  "/bin/sh -c 'while..."  2 minutes ago    Up 2 minutes
compassionate_hugle
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker ps -a    #-a 列出所有容器（包含沉睡/退出状态的容器）；
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
IMAGE          COMMAND          CREATED          STATUS          PORTS
NAMES
</span></span><span class="highlight-line"><span class="highlight-cl">6246ea4f21ef
docker.io/centos:latest  "/bin/sh -c 'while..."  3 minutes ago    Up 3 minutes
compassionate_hugle
</span></span><span class="highlight-line"><span class="highlight-cl">ff20a65c5b95
docker.io/centos:latest  "/bin/bash"              3 hours ago      Exited (0) 3 hours ago
serene_yalow
</span></span></code></pre>
<p>杀死一个容器</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker kill 6246ea4f21ef
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">6246ea4f21ef
</span></span></code></pre>
<h3 id="例4-启动-停止-重启-container-容器实例">例 4：启动、停止、重启 container 容器实例<
h3>
<p>启动： run # 创建并运行 docker 实例</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker run -d docker.io/centos:latest /bin/sh -c "while true;do echo he
lo world; sleep 1; done"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">f639c4a00e22734
9c4faecd1480d01333c6049599d7e851342aa588e3e2926a
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker ps
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
IMAGE          COMMAND          CREATED          STATUS          PORTS
NAMES
</span></span><span class="highlight-line"><span class="highlight-cl">f639c4a00e22
docker.io/centos:latest  "/bin/sh -c 'while..."  7 seconds ago    Up 6 seconds
infallible_lewin
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker stop f639c4a00e22
</span></span><span class="highlight-line"><span class="highlight-cl">f639c4a00e22
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker ps
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
IMAGE          COMMAND          CREATED          STATUS          PORTS          NAME
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
</span></span></code></pre>
<h3 id="例5-删除指定-container---rm">例 5：删除指定 container ： rm</h3>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker rm f639c4a00e22

```



```

</span></span><span class="highlight-line"><span class="highlight-cl">f639c4a00e22
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker ps
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
IMAGE          COMMAND          CREATED          STATUS          PORTS
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<h2 id="二-docker镜像制作的方法-">二、docker 镜像制作的方法:</h2>
<p>Docker Image 的制作两种方法</p>
<p>方法 1: docker commit #保存 container 的当前状态到 image 后, 然后生成对应的 image<
r>
方法 2: docker build #使用 Dockerfile 文件自动化制作 image</p>
<h2 id="2-1-创建一个安装好-apache-web-服务器的容器镜像">2.1 创建一个安装好 apache web
服务器的容器镜像</h2>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker run -it docker.io/centos:latest /bin/bash
</span></span><span class="highlight-line"><span class="highlight-cl">[root@692713fffa
b /]# yum install -y httpd
</span></span><span class="highlight-line"><span class="highlight-cl"> #根据容器前状
做一个 image 镜像: 创建一个安装了 apache 工具的 centos 镜像语法: docker commit &lt;conta
ner 的 ID&gt;或&lt;image_name&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker ps -a
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
IMAGE          COMMAND          CREATED          STATUS          PORTS
NAMES
</span></span><span class="highlight-line"><span class="highlight-cl">692713ffabb
docker.io/centos:latest "/bin/bash" 14 minutes ago Exited (0) 2 minutes ago
priceless_kare
</span></span><span class="highlight-line"><span class="highlight-cl">6246ea4f21ef
docker.io/centos:latest "/bin/sh -c 'while..." 46 minutes ago Exited (137) 40 minutes ago
compassionate_hugle
</span></span><span class="highlight-line"><span class="highlight-cl">ff20a65c5b95
docker.io/centos:latest "/bin/bash" 3 hours ago Exited (0) 3 hours ago
serene_yalow
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker commit 692713ffabb docker.io/centos:apache
</span></span><span class="highlight-line"><span class="highlight-cl">sha256:d61468f3
896cf7f900f44b442c3e20960eba54bf877809f30a832e5c844ff15
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker images
</span></span><span class="highlight-line"><span class="highlight-cl">REPOSITORY
TAG          IMAGE ID          CREATED          SIZE
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
apache      d61468f39896      7 seconds ago    318 MB
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
latest      9f38484d220f      11 days ago      202 MB
</span></span><span class="highlight-line"><span class="highlight-cl">#使用新创建的 doc
er.io/centos: apache 镜像, 生成一台容器实例: 镜像, 生成一台容器实例:
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker run -it centos:apache /bin/bash
</span></span><span class="highlight-line"><span class="highlight-cl">[root@e105be293
2b /]# rpm -qa httpd
</span></span><span class="highlight-line"><span class="highlight-cl">httpd-2.4.6-88.el7

```

centos.x86_64

```
</span></span><span class="highlight-line"><span class="highlight-cl">[root@e105be2932b /]#
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">#查找安装包可以明基于docker的apacheweb服务器构建成功
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span></code></pre>
```

<p>以下操作要在 docker 物理机上操作：</p> <p>1、创建工作目录</p> ``` <pre><code class="highlight-chroma">[root@bogon ~]# mkdir /docker-build ``` ``` [root@bogon docker-build]# touch Dockerfile ``` ``` ``` ``` </code></pre> ``` <p>2、编辑 Dockerfile
 Dockerfile 用来创建一个自定义的 image,包含了用户指定的软件依赖等。</p> ``` <pre><code class="highlight-chroma">[root@bogon docker-build]# vim Dockerfile ``` ``` FROM docker.io/centos:latest ``` ``` MAINTAINER <www.cjzshilong.cn> ``` ``` RUN yum -y install httpd ``` ``` ADD start.sh /usr/local/bin/start.sh ``` ``` ADD index.html /var/www/html/index.html ``` ``` CMD /usr/local/bin/start.sh ``` ``` </code></pre> ``` <p>注释：
 FROM docker.io/centos:latest # FROM 基于哪个镜像 MAINTAINER <www.cjzshilong.cn> MAINTAINER 镜像创建者
 RUN yum -y install httpd #RUN 安装软件用
 ADD start.sh /usr/local/bin/start.sh
 ADD index.html /var/www/html/index.html CMD /usr/local/bin/start.sh
 ADD 将文件拷贝到新产生的镜像的文件系统对应的路径。所有拷贝到新镜像中的文件和文件夹权限 0755,uid 和 gid 为 0
 CMD echo hello world #container 启动时执行的命令或启动服务，但是一个 Dockerfile 中另能一条 CMD 命令，多条则另执行最后一条 CMD.</p> <p>3、创建 start.sh 脚本启动 httpd 服务和 apache 默认首页 index.html 文件</p> ``` <pre><code class="highlight-chroma">[root@bogon docker-build]# echo "/usr/sbin/httpd -DFOREGROUND" > start.sh ``` ``` [root@bogon docker-build]# chmod a+x start.sh ``` ``` [root@bogon doc ``` 原文链接：[搭建 docker 之路 - 初识 \(1\)](#)


```
er-build]# echo "docker images build test" &gt; index.html
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>注： /usr/sbin/httpd -DFOREGROUND 参数的含意： <br>
Systemd is designed to run processes "in the foreground", that is, they don't have to specifical
y run themselves as daemons. The processes don't exactly run in the foreground, they run un
er systemd and it captures their input and output, but from the process's perspective, it is the
same as running in the foreground. That is systemd's preferred method of operation, but it
does have compatibility with traditional daemons.</p>
<p>4、使用命令来创建新的 image<br>
语法： docker build -t 父镜像名： 镜像的 tag Dockerfile 文件所在路径<br>
-t :表示 tage, 镜像名<br>
例： 使用命令 docker build 来创建新的 image,并命名为 docker.io/centos:httpd</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon docker-build]# docker build -t docker.io/centos:httpd ./
</span></span><span class="highlight-line"><span class="highlight-cl">.....
</span></span><span class="highlight-line"><span class="highlight-cl">Complete!
</span></span><span class="highlight-line"><span class="highlight-cl"> ---&gt; 15fa43bf
4fd
</span></span><span class="highlight-line"><span class="highlight-cl">Removing interm
diate container 032876270cc9
</span></span><span class="highlight-line"><span class="highlight-cl">Step 4/6 : ADD sta
t.sh /usr/local/bin/start.sh
</span></span><span class="highlight-line"><span class="highlight-cl"> ---&gt; 7c5122e7
132
</span></span><span class="highlight-line"><span class="highlight-cl">Removing interm
diate container d4788388dfd4
</span></span><span class="highlight-line"><span class="highlight-cl">Step 5/6 : ADD in
ex.html /var/www/html/index.html
</span></span><span class="highlight-line"><span class="highlight-cl"> ---&gt; b02c0f7e
657
</span></span><span class="highlight-line"><span class="highlight-cl">Removing interm
diate container e8fd13da450e
</span></span><span class="highlight-line"><span class="highlight-cl">Step 6/6 : CMD /u
r/local/bin/start.sh
</span></span><span class="highlight-line"><span class="highlight-cl"> ---&gt; Running i
e28625d1a0e3
</span></span><span class="highlight-line"><span class="highlight-cl"> ---&gt; 7703e62f
080
</span></span><span class="highlight-line"><span class="highlight-cl">Removing interm
diate container e28625d1a0e3
</span></span><span class="highlight-line"><span class="highlight-cl">Successfully built
703e62fc080
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon doc
er-build]# docker images
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">REPOSITORY
TAG          IMAGE ID      CREATED       SIZE
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
httpd        7703e62fc080   About a minute ago   318 MB
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
apache       d61468f39896   27 minutes ago      318 MB
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
```

```

latest          9f38484d220f    11 days ago    202 MB
</span></span></code></pre>
<p>./表示当前目录，当前目录要含有 Dockerfile 文件</p>
<h2 id="三-Docker-images的发布">三、Docker images 的发布</h2>
<p>方法 1: Save Image To TarBall<br>
方法 2: Push Image To Docker Hub</p>
<h2 id="3-1-方法1-Save-Image-To-TarBall">3.1 方法 1: Save Image To TarBall</h2>
<p>保存 Image 到 tar 包<br>
语法: docker save -o 导出的镜像名.tar 本地镜像名: 镜像标签例: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
l -h
</span></span><span class="highlight-line"><span class="highlight-cl">总用量 311M
</span></span><span class="highlight-line"><span class="highlight-cl">-rw-----. 1 root
oot 311M 3月 26 18:38 docker.io-centos-httpd-image.tar
</span></span></code></pre>
<p>例: 使用导入本地镜像: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]# docker rmi docker.io/centos:httpd
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Untagged: docker.
o/centos:httpd
</span></span><span class="highlight-line"><span class="highlight-cl">Deleted: sha256:7
03e62fc0802dcb545ed4403dda4e343e0d51ae984db30efb2e32f0e045ef26
</span></span><span class="highlight-line"><span class="highlight-cl">Deleted: sha256:b
2c0f7e565757ec489dd759a45abf321273bca9959e3193dcf9a6d3de4adf2b
</span></span><span class="highlight-line"><span class="highlight-cl">Deleted: sha256:1
9184e36e9a941d7bd35ae854ac2677367f0f0882d2d8056cc1a9b025dad4c3
</span></span><span class="highlight-line"><span class="highlight-cl">Deleted: sha256:7
5122e7813298d44726f2c4df0f27be01cfe85ab2cf530d3729d3e84908dd29
</span></span><span class="highlight-line"><span class="highlight-cl">Deleted: sha256:ff
1787458e2102cfa4247fea79da7103d5017ef496ba8b9f83ad9296f1f60ea
</span></span><span class="highlight-line"><span class="highlight-cl">Deleted: sha256:1
fa43bfc4fd425002a5b528d061782f954643234fdd0beef31f380646b8471b
</span></span><span class="highlight-line"><span class="highlight-cl">Deleted: sha256:0
4e270a381efa029a8d0a26e377055ca8aa49d1d3fe595d2c05fb76824ff809
</span></span><span class="highlight-line"><span class="highlight-cl">Deleted: sha256:b
93b347b502d32efcbfe18746fda992b7675e36ead5fb727e237194c999cdaf
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker images
</span></span><span class="highlight-line"><span class="highlight-cl">REPOSITORY
TAG          IMAGE ID      CREATED      SIZE
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
apache      d61468f39896   34 minutes ago  318 MB
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
latest      9f38484d220f   11 days ago    202 MB
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker load -i docker.io-centos-httpd-image.tar
</span></span><span class="highlight-line"><span class="highlight-cl">f42df3e4c26b: Lo
ding layer [=====
116.6 MB/116.6 MB

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">b1e658f9a703: Lo
ding layer [=====&gt;]
3.584 kB/3.584 kB
</span></span><span class="highlight-line"><span class="highlight-cl">1ebe4270a160: Lo
ding layer [=====&gt;]
3.584 kB/3.584 kB
</span></span><span class="highlight-line"><span class="highlight-cl">Loaded image: do
ker.io/centos:httpd
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker images
</span></span><span class="highlight-line"><span class="highlight-cl">REPOSITORY
TAG          IMAGE ID      CREATED      SIZE
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
httpd        7703e62fc080  10 minutes ago  318 MB
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
apache       d61468f39896  35 minutes ago  318 MB
</span></span><span class="highlight-line"><span class="highlight-cl">docker.io/centos
latest       9f38484d220f  11 days ago    202 MB
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
</span></span></code></pre>
<h2 id="3-2-方法2-Push-Image-To-Docker-Hub-发布到外网">3.2 方法 2: Push Image To Doc
er Hub 发布到外网</h2>
<p>1、Signup on docker hub & create repo 注册一个帐号 <a href="https://ld246.com/
orward?goto=https%3A%2F%2Fhub.docker.com%2F" target="_blank" rel="nofollow ugc">ht
ps://hub.docker.com/</a> <br>
2、Login to docker hub<br>
// docker login -u userabc -p abc123 -e <a href="https://ld246.com/forward?goto=mailto%3
userab%40gmail.com" target="_blank" rel="nofollow ugc">userab@gmail.com</a> <br>
3、Push image to docker hub #上传镜像<br>
// docker push centos:httpd<br>
4、Pull image from docker hub #下载镜像<br>
// docker pull userabc/centos:httpd # 用户名/镜像名</p>
<h2 id="3-2-方法3-使用阿里云的私有仓库来发布-">3.2 方法 3: 使用阿里云的私有仓库来发布;</h
>
<p>待实验.....</p>
<h2 id="四-Container-容器端口映射">四、Container 容器端口映射</h2>
<h2 id="4-1-实战Container-端口映射">4.1 实战 Container 端口映射</h2>
<p>启动 Container</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker run -d -p 80:80 docker.io/centos:httpd
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">6a1fc818a972307
87b9130ad7460c098c2bd88d60eb9026d565c7c24b71fe33
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
</span></span></code></pre>
<p>注: -p 物理机的 80 端口:容器实例的 80 端口, 把容器中的 80 端口映射到物理机上的 80 端口
/p>
<p>在物理机上查看容器状态</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker ps
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
IMAGE          COMMAND          CREATED          STATUS          PORTS
NAMES
</span></span></code>

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">6a1fc818a972
docker.io/centos:httpd "/bin/sh -c /usr/l..." 43 seconds ago Up 42 seconds 0.0.0.0:80
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
netstat -antup | grep 80
</span></span><span class="highlight-line"><span class="highlight-cl">tcp 0 0 192
168.89.209:22 192.168.89.80:62820 ESTABLISHED 1397/sshd: root@pts
</span></span><span class="highlight-line"><span class="highlight-cl">tcp 0 0 192
168.89.209:22 192.168.89.80:62819 ESTABLISHED 1358/sshd: root@pts
</span></span><span class="highlight-line"><span class="highlight-cl">tcp6 0 0 ::
0 :::* LISTEN 6742/docker-proxy-c
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
</span></span></code></pre>
<p>测试 http<br>
</p>
<h2 id="4-2-访问正在运行的container容器实例">4.2 访问正在运行的 container 容器实例</h2>
<p>语 法： docker exec -it <container id | name> /bin/bash<br>
查看正在运行的容器 ID：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker ps
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
IMAGE COMMAND CREATED STATUS PORTS
NAMES
</span></span><span class="highlight-line"><span class="highlight-cl">6a1fc818a972
docker.io/centos:httpd "/bin/sh -c /usr/l..." 6 minutes ago Up 6 minutes 0.0.0.0:80-
gt;80/tcp unruffled_yonath
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
docker exec -it 6a1fc818a972 /bin/bash
</span></span><span class="highlight-line"><span class="highlight-cl">[root@6a1fc818a
72 /]# echo cuijianzhe &gt; /var/www/html/test.html
</span></span><span class="highlight-line"><span class="highlight-cl">[root@6a1fc818a
72 /]# exit
</span></span><span class="highlight-line"><span class="highlight-cl">exit
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
curl http://192.168.89.209/test.html
</span></span><span class="highlight-line"><span class="highlight-cl">cuijianzhe
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
</span></span></code></pre>
<p>查看物理机和容器的网络<br>
查看容器的 ip</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker exec -it 6a1fc818a972 /bin/bash
</span></span><span class="highlight-line"><span class="highlight-cl">[root@6a1fc818a
72 /]# yum install net-tools -y
</span></span><span class="highlight-line"><span class="highlight-cl">Loaded plugins: fa
testmirror, ovl
</span></span><span class="highlight-line"><span class="highlight-cl">Loading mirror sp
eds from cached hostfile
</span></span><span class="highlight-line"><span class="highlight-cl">* base: mirrors.zju
edu.cn
</span></span><span class="highlight-line"><span class="highlight-cl">* extras: mirrors.s

```

```

u.edu.cn
</span></span><span class="highlight-line"><span class="highlight-cl"> * updates: mirrors
shu.edu.cn
</span></span><span class="highlight-line"><span class="highlight-cl">Resolving Depend
ncies
</span></span><span class="highlight-line"><span class="highlight-cl">--&gt; Running tr
nsaction check
</span></span><span class="highlight-line"><span class="highlight-cl">---&gt; Package n
t-tools.x86_64 0:2.0-0.24.20131004git.el7 will be installed
</span></span><span class="highlight-line"><span class="highlight-cl">--&gt; Finished D
ependency Resolution
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Dependencies Re
olved
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">=====
=====
=====
=====
=====
</span></span><span class="highlight-line"><span class="highlight-cl"> Package
Arch Version Repository
y Size
</span></span><span class="highlight-line"><span class="highlight-cl">=====
=====
=====
=====
=====
</span></span><span class="highlight-line"><span class="highlight-cl">Installing:
</span></span><span class="highlight-line"><span class="highlight-cl"> net-tools
x86_64 2.0-0.24.20131004git.el7
ase 306 k
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Transaction Sum
ary
</span></span><span class="highlight-line"><span class="highlight-cl">=====
=====
=====
=====
=====
</span></span><span class="highlight-line"><span class="highlight-cl">Install 1 Package
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Total download si
e: 306 k
</span></span><span class="highlight-line"><span class="highlight-cl">Installed size: 918

</span></span><span class="highlight-line"><span class="highlight-cl">Downloading pac
ages:
</span></span><span class="highlight-line"><span class="highlight-cl">net-tools-2.0-0.24
20131004git.el7.x86_64.rpm
| 306 kB 00:00:00
</span></span><span class="highlight-line"><span class="highlight-cl">Running transacti
n check
</span></span><span class="highlight-line"><span class="highlight-cl">Running transacti

```



```

n test
</span></span><span class="highlight-line"><span class="highlight-cl">Transaction test s
ceeded
</span></span><span class="highlight-line"><span class="highlight-cl">Running transacti
n
</span></span><span class="highlight-line"><span class="highlight-cl"> Installing : net-t
ols-2.0-0.24.20131004git.el7.x86_64
1/1
</span></span><span class="highlight-line"><span class="highlight-cl"> Verifying : net-t
ols-2.0-0.24.20131004git.el7.x86_64
1/1
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Installed:
</span></span><span class="highlight-line"><span class="highlight-cl"> net-tools.x86_64
0:2.0-0.24.20131004git.el7

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Complete!
</span></span><span class="highlight-line"><span class="highlight-cl">[root@6a1fc818a
72 /]# ifconfig
</span></span><span class="highlight-line"><span class="highlight-cl">eth0: flags=4163&
t;UP,BROADCAST,RUNNING,MULTICAST&gt; mtu 1500
</span></span><span class="highlight-line"><span class="highlight-cl"> inet 172.17.0
2 netmask 255.255.0.0 broadcast 0.0.0.0
</span></span><span class="highlight-line"><span class="highlight-cl"> inet6 fe80::42
acff:fe11:2 prefixlen 64 scopeid 0x20&lt;link&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> ether 02:42:ac
11:00:02 txqueuelen 0 (Ethernet)
</span></span><span class="highlight-line"><span class="highlight-cl"> RX packets 1
4 bytes 326507 (318.8 KiB)
</span></span><span class="highlight-line"><span class="highlight-cl"> RX errors 0 d
ropped 0 overruns 0 frame 0
</span></span><span class="highlight-line"><span class="highlight-cl"> TX packets 1
2 bytes 11695 (11.4 KiB)
</span></span><span class="highlight-line"><span class="highlight-cl"> TX errors 0 d
ropped 0 overruns 0 carrier 0 collisions 0
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">lo: flags=73&lt;UP
LOOPBACK,RUNNING&gt; mtu 65536
</span></span><span class="highlight-line"><span class="highlight-cl"> inet 127.0.0.1
netmask 255.0.0.0
</span></span><span class="highlight-line"><span class="highlight-cl"> inet6 ::1 pref
ixlen 128 scopeid 0x10&lt;host&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> loop txqueue
len 1000 (Local Loopback)
</span></span><span class="highlight-line"><span class="highlight-cl"> RX packets 0
bytes 0 (0.0 B)
</span></span><span class="highlight-line"><span class="highlight-cl"> RX errors 0 d
ropped 0 overruns 0 frame 0
</span></span><span class="highlight-line"><span class="highlight-cl"> TX packets 0
bytes 0 (0.0 B)
</span></span><span class="highlight-line"><span class="highlight-cl"> TX errors 0 d
ropped 0 overruns 0 carrier 0 collisions 0
</span></span><span class="highlight-line"><span class="highlight-cl">

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">[root@6a1fc818a
72 /]#
</span></span></code></pre>
<p>物理机的 ip: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# ifconfig
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">docker0: flags=41
3<&lt;UP,BROADCAST,RUNNING,MULTICAST&gt; mtu 1500
</span></span><span class="highlight-line"><span class="highlight-cl">      inet 172.17.0
1 netmask 255.255.0.0 broadcast 0.0.0.0
</span></span><span class="highlight-line"><span class="highlight-cl">      inet6 fe80::42
a1ff:fee8:1077 prefixlen 64 scopeid 0x20&lt;link&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">      ether 02:42:a
:e8:10:77 txqueuelen 0 (Ethernet)
</span></span><span class="highlight-line"><span class="highlight-cl">      RX packets 3
531 bytes 2208751 (2.1 MiB)
</span></span><span class="highlight-line"><span class="highlight-cl">      RX errors 0 d
opped 0 overruns 0 frame 0
</span></span><span class="highlight-line"><span class="highlight-cl">      TX packets 3
270 bytes 74193226 (70.7 MiB)
</span></span><span class="highlight-line"><span class="highlight-cl">      TX errors 0 d
opped 0 overruns 0 carrier 0 collisions 0
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ens33: flags=416
&lt;UP,BROADCAST,RUNNING,MULTICAST&gt; mtu 1500
</span></span><span class="highlight-line"><span class="highlight-cl">      inet 192.168.
9.209 netmask 255.255.255.0 broadcast 192.168.89.255
</span></span><span class="highlight-line"><span class="highlight-cl">      inet6 fe80::e
9d:e9f2:8c11:9372 prefixlen 64 scopeid 0x20&lt;link&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">      ether 00:0c:2
:33:7b:83 txqueuelen 1000 (Ethernet)
</span></span><span class="highlight-line"><span class="highlight-cl">      RX packets 2
8009 bytes 207333025 (197.7 MiB)
</span></span><span class="highlight-line"><span class="highlight-cl">      RX errors 0 d
opped 1 overruns 0 frame 0
</span></span><span class="highlight-line"><span class="highlight-cl">      TX packets 9
075 bytes 8610553 (8.2 MiB)
</span></span><span class="highlight-line"><span class="highlight-cl">      TX errors 0 d
opped 0 overruns 0 carrier 0 collisions 0
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">lo: flags=73&lt;UP
LOOPBACK,RUNNING&gt; mtu 65536
</span></span><span class="highlight-line"><span class="highlight-cl">      inet 127.0.0.1
netmask 255.0.0.0
</span></span><span class="highlight-line"><span class="highlight-cl">      inet6 ::1 pref
xlen 128 scopeid 0x10&lt;host&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">      loop txqueue
en 1000 (Local Loopback)
</span></span><span class="highlight-line"><span class="highlight-cl">      RX packets 6
4 bytes 57000 (55.6 KiB)
</span></span><span class="highlight-line"><span class="highlight-cl">      RX errors 0 d
opped 0 overruns 0 frame 0
</span></span><span class="highlight-line"><span class="highlight-cl">      TX packets 6

```

```

4 bytes 57000 (55.6 KiB)
</span></span><span class="highlight-line"><span class="highlight-cl">      TX errors 0 d
opped 0 overruns 0 carrier 0 collisions 0
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">vethd559ccb: flag
=4163&lt;UP,BROADCAST,RUNNING,MULTICAST&gt; mtu 1500
</span></span><span class="highlight-line"><span class="highlight-cl">      inet6 fe80::8
ea:50ff:fe38:8696 prefixlen 64 scopeid 0x20&lt;link&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">      ether 8e:ea:5
:38:86:96 txqueuelen 0 (Ethernet)
</span></span><span class="highlight-line"><span class="highlight-cl">      RX packets 1
2 bytes 11695 (11.4 KiB)
</span></span><span class="highlight-line"><span class="highlight-cl">      RX errors 0 d
opped 0 overruns 0 frame 0
</span></span><span class="highlight-line"><span class="highlight-cl">      TX packets 1
4 bytes 326507 (318.8 KiB)
</span></span><span class="highlight-line"><span class="highlight-cl">      TX errors 0 d
opped 0 overruns 0 carrier 0 collisions 0
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">[root@bogon ~]#
</span></span></code></pre>
<p>目前相当于虚拟机的 NAT 模式<br>
</p>
<p>配置容器的 root 密码:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@bogon ~]# docker exec -it 6a1fc818a972 /bin/bash
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">[root@6a1fc818a
72 /]# echo 598941324 | passwd --stdin root
</span></span><span class="highlight-line"><span class="highlight-cl">Changing passwo
d for user root.
</span></span><span class="highlight-line"><span class="highlight-cl">passwd: all authen
ication tokens updated successfully.
</span></span><span class="highlight-line"><span class="highlight-cl">[root@6a1fc818a
72 /]#
</span></span></code></pre>
<p>在容器中安装 sshd 服务，然后把 22 端口映射出去，就可以远程登录容器了。</p>

```