|  |
| --- |
| **Docker安装、私有库配置** |
|  |
| **版本：V1.0** |

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制人: | 王先健 | 日期: | 2017-6-26 |
| 审核人: |  | 日期: |  |
| 批准人: |  | 日期: |  |

**南京国通智能科技有限公司**

南京国通智能科技有限公司对本文件资料享有著作权及其它专属权利，未经书面许可，不得将该等文件资料（其全部或任何部分）披露予任何第三方，或进行修改后使用。

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **作者** | **修订日期** | **修订内容** | **批准人** |
| 1.0 | 王先健 | 2017-6-26 | 初稿形成 | - |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目 录**

目录

[一、 Docker简介 4](#_Toc486262151)

[二、 docker的安装 5](#_Toc486262152)

[三、 docker的基本操作命令 6](#_Toc486262153)

[3.1、搜索镜像 6](#_Toc486262154)

[3.2、下载镜像 6](#_Toc486262155)

[3.3、查看镜像 6](#_Toc486262156)

[3.4、使用镜像创建容器 7](#_Toc486262157)

[3.5、查看容器 7](#_Toc486262158)

[3.6、再次启动容器 7](#_Toc486262159)

[3.7、删除容器 8](#_Toc486262160)

[四、 docker建立私有仓库 8](#_Toc486262161)

[4.1、下载仓库并启动容器 8](#_Toc486262162)

[4.2、访问私有仓库 9](#_Toc486262163)

[4.3、创建镜像链接为基础镜像打个标签 9](#_Toc486262164)

[4.4、修改docker配置文件 9](#_Toc486262165)

[4.5、上传镜像到私有仓库 9](#_Toc486262166)

[4.6、查看镜像是否提交 10](#_Toc486262167)

# Docker简介

Docker 提供了一个可以运行你的应用程序的封套(envelope)，或者说容器。 Docker 扩展了 Linux 容器（Linux Containers），或着说 LXC，通过一个高层次的 API 为进程单独提供了一个轻量级的虚拟环境。Docker 利用了 LXC， cgroups 和 Linux 自己的内核。和传统的虚拟机不同的是，一个 Docker 容器并不包含一个单独的操作系统，而是基于已有的基础设施中操作系统提供的功能来运行的。

 Docker类似虚拟机的概念，但是与虚拟化技术的不同点在于下面几点：  
    1.虚拟化技术依赖物理CPU和内存，是硬件级别的；而docker构建在操作系统上，利用操作系统的containerization技术，所以docker甚至可以在虚拟机上运行。  
    2.虚拟化系统一般都是指操作系统镜像，比较复杂，称为“系统”；而docker开源而且轻量，称为“容器”，单个容器适合部署少量应用，比如部署一个redis、一个memcached。  
    3.传统的虚拟化技术使用快照来保存状态；而docker在保存状态上不仅更为轻便和低成本，而且引入了类似源代码管理机制，将容器的快照历史版本一一记录，切换成本很低。  
    4.传统的虚拟化技术在构建系统的时候较为复杂，需要大量的人力；而docker可以通过Dockfile来构建整个容器，重启和构建速度很快。更重要的是Dockfile可以手动编写，这样应用程序开发人员可以通过发布Dockfile来指导系统环境和依赖，这样对于持续交付十分有利。  
    5.Dockerfile可以基于已经构建好的容器镜像，创建新容器。Dockerfile可以通过社区分享和下载，有利于该技术的推广。  
       Docker 由下面这些组成：  
    1. Docker 服务器守护程序（server daemon），用于管理所有的容器。  
    2. Docker 命令行客户端，用于控制服务器守护程序。  
    3. Docker 镜像：查找和浏览 docker 容器镜像。

# docker的安装

[root@localhost ~]# uname –a

#查看内核版本，必须3.10以上的内核方可安装docker

Linux localhost.localdomain 3.10.0-514.el7.x86\_64 #1 SMP Tue Nov 22 16:42:41 UTC 2016 x86\_64 x86\_64 x86\_64 GNU/Linux

[root@localhost ~]# rpm -qa | grep docker

#查看服务器是否安装过docker

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

[root@localhost ~]# systemctl start docker #启动docker

[root@localhost ~]# systemctl enable docker #开机启动docker

[root@localhost ~]# docker info #查看docker的信息

Containers: 9

Running: 1

Paused: 0

Stopped: 8

Images: 9

Server Version: 1.12.6

………….

[root@localhost ~]# curl -sSL https://get.daocloud.io/daotools/set\_mirror.sh | sh -s <http://7a5bf838.m.daocloud.io> #添加国内加速器

docker version >= 1.12

{"registry-mirrors": ["http://7a5bf838.m.daocloud.io"]}

Success.

You need to restart docker to take effect: sudo systemctl restart docker

[root@localhost ~]# systemctl restart docker

# docker的基本操作命令

## 3.1、搜索镜像

命令：docker search <image>：在docker index中搜索image

[root@localhost ~]# docker search centos #centos后面什么都不带，代表着tag为latest

## 3.2、下载镜像

命令：docker pull <image> ：从docker registry server 中下拉image

[root@localhost ~]# docker pull ubuntu

## 3.3、查看镜像

docker images： 列出images

docker images -a ：列出所有的images（包含历史）

docker rmi  <image ID>： 删除一个或多个image

## 3.4、使用镜像创建容器

[root@localhost ~]# docker run docker.io/centos /bin/echo hello world

hello world

交换式运行

[root@localhost ~]# docker run -ti --name hehe docker.io/centos /bin/bash

[root@787f21f93414 /]#

## 3.5、查看容器

docker ps ：列出当前所有正在运行的container  
docker ps -l ：列出最近一次启动的container  
docker ps -a ：列出所有的container（包含历史，即运行过的container）  
docker ps -q ：列出最近一次运行的container ID

## 3.6、再次启动容器

docker start/stop/restart <container> ：开启/停止/重启container  
 docker start [container\_id] ：再次运行某个container （包括历史container）  
docker attach [container\_id] ：连接一个正在运行的container实例（即实例必须为start状态，可以多个窗口同时attach 一个container实例）  
docker start -i <container> ：启动一个container并进入交互模式（相当于先start，在attach）

docker run -i -t <image> /bin/bash ：使用image创建container并进入交互模式, login shell是/bin/bash  
 docker run -i -t -p <host\_port:contain\_port> ：映射 HOST 端口到容器，方便外部访问容器内服务，host\_port 可以省略，省略表示把 container\_port 映射到一个动态端口。  
 注：使用start是启动已经创建过得container，使用run则通过image开启一个新的container。

## 3.7、删除容器

 docker rm <container...> ：删除一个或多个container  
 docker rm `docker ps -a -q` ：删除所有的container  
 docker ps -a -q | xargs docker rm ：同上, 删除所有的container

# docker建立私有仓库

## 4.1、下载仓库并启动容器

[root@localhost ~]# docker pull docker.io/registry

[root@localhost ~]# mkdir -p /mnt/registry

[root@localhost ~]# docker run -d -p 5000:5000 --privileged=true -v /mnt/registry:/tmp/registry docker.io/registry

eac7b9190f68af0d24cab08b5b60933789d5647c9a33853d2f1676459acf9d22

默认情况下，会将仓库存放于容器的/tmp/registry目录下，这样如果容器被删除，则存放于容器中的镜像也会丢失，所以我们一般情况下会指定本地一个目录挂载到容器的/tmp/registry下， 两个目录下都有！registry的默认存储路径是/tmp/registry，只是个临时目录，一段时间之后就会消失·所以使用-v参数，指定个本地持久的路径，

## 4.2、访问私有仓库

[root@localhost ~]# curl -X GET http://192.168.221.132:5000/v2/\_catalog

{"repositories":[]}

## 4.3、创建镜像链接为基础镜像打个标签

[root@localhost ~]# docker tag docker.io/ubuntu 192.168.221.132:5000/ubuntu

## 4.4、修改docker配置文件

vim /etc/sysconfig/docker

添加以下内容

ADD\_REGISTRY='--add-registry 192.168.221.132:5000'

INSECURE\_REGISTRY='--insecure-registry 192.168.221.132:5000'

## 4.5、上传镜像到私有仓库

[root@localhost ~]# docker push 192.168.221.132:5000/ubuntu

报错：

The push refers to a repository [192.168.221.132:5000/ubuntu]

Get https://192.168.221.132:5000/v1/\_ping: http: server gave HTTP response to HTTPS client

因为： Docker从1.3.x之后，与docker registry交互默认使用的是https，但是此处搭建私有仓库却只提供http服务，所以当和私有仓库交互时报上述错误。因此需要在启动docker server时增加启动参数为默认使用http访问。

解决方案：

[root@localhost ~]# vi /etc/docker/daemon.json 添加如下内容

{"insecure-registries": ["192.168.221.132:5000"]}

[root@localhost docker]# systemctl restart docker

[root@localhost docker]# docker start eac7b9190f68

[root@localhost docker]# docker push 192.168.221.132:5000/ubuntu

The push refers to a repository [192.168.221.132:5000/ubuntu]

0566c118947e: Pushed

6f9cf951edf5: Pushed

182d2a55830d: Pushed

5a4c2c9a24fc: Pushed

cb11ba605400: Pushed

latest: digest: sha256:a0ee7647e24c8494f1cf6b94f1a3cd127f423268293c25d924fbe18fd82db5a4 size: 1357

## 4.6、查看镜像是否提交

[root@localhost ~]# curl -XGET 192.168.221.132:5000/v2/\_catalog

{"repositories":["ubuntu"]}