

云原生应用2: docker在线/离线安装与基础用法

原创

闪耀的瞬间  于 2019-08-27 22:35:31 发布  250  收藏 1

分类专栏: [云原生](#) 文章标签: [docker](#) [kubernetes](#) [容器](#) [云原生](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/zhuyu19911016520/article/details/100099464>

版权



[云原生 专栏收录该内容](#)

14 篇文章 2 订阅

订阅专栏

本篇博客能让你了解到docker的安装与基础语法

什么是 Docker?

Docker 是一个开源的应用容器引擎, 是一种资源虚拟化技术, 让开发者可以打包他们的应用以及依赖包到一个可移植的容器中, 然后发布到任何流行的 Linux 机器上。虚拟化技术演进路径可分为三个时代:

- a.物理机时代, 多个应用程序可能跑在一台物理机器上
- b.虚拟机时代, 一台物理机器启动多个虚拟机实例, 一个虚拟机跑多个应用程序
- c.容器化时代, 一台物理机上启动多个容器实例, 一个容器跑多个应用程序

Docker 的构成

Docker 由镜像、镜像仓库、容器三个部分组成

镜像: 跨平台、可移植的程序+环境包

镜像仓库: 镜像的存储位置, 有云端仓库和本地仓库之分, 官方镜像仓库地址 (<https://hub.docker.com/>)

容器: 进行了资源隔离的镜像运行时环境

1.docker安装与基础用法

依赖环境: Linux kernel 3.10+

下面使用 Centos7.6演示: 查看内核与系统版本: `uname -a`, `cat /etc/centos-release`

```
[root@centos1 ~]# uname -a
Linux centos1 3.10.0-957.21.3.el7.x86_64 #1 SMP Tue Jun 18 16:35:19 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
[root@centos1 ~]# cat /etc/centos-release
CentOS Linux release 7.6.1810 (Core)
[root@centos1 ~]#
```

2.安装docker

- 1.在阿里云开源镜像仓库找到docker-ce库， <https://mirrors.aliyun.com/docker-ce/linux/centos/>， 找到 docker-ce.repo ， 并右键复制链接地址， 在centos上 cd /etc/yum.repos.d/ ， wget <https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo>
- 2.执行yum repolist， 再安装docker-ce， yum install docker-ce， 中间步骤输入 y 同意
如果出现 container-selinux >= 2.9问题无法安装， 请先下载 container-selinux
下载： wget http://mirror.centos.org/centos/7/extras/x86_64/Packages/container-selinux-2.107-3.el7.noarch.rpm
安装： rpm -ivh container-selinux-2.107-3.el7.noarch.rpm
- 3.设置docker镜像加速器， dockerhub在国外， 因此docker镜像的下载很慢， 需要设置加速地址， 有很多第三方镜像加速地址， 如： 阿里云镜像加速器、163、docker-cn等， 设置加速器： 1.mkdir -p /etc/docker， 2.vi /etc/docker/daemon.json， 3.输入 {"registry-mirrors": ["https://xxx.mirror.aliyuncs.com", "https://registry.docker-cn.com"]} 里面是数组， 可以输入多个
- 4.启动docker， systemctl start docker.service， 开机启动， systemctl enable docker.service
- 5.查看docker版本， docker version， docker info， 可看到运行中的容器， docker 版本， 以及刚刚设置的镜像加速器地址， 可看到机器上有一个已停止的容器

```
[root@centos1 yum.repos.d]# docker info
Client:
  Debug Mode: false

Server:
  Containers: 1
  Running: 0
  Paused: 0
  Stopped: 1
  Images: 1
  Server Version: 19.03.1
  Storage Driver: overlay2
    Backing Filesystem: xfs
    Supports d_type: true
    Native Overlay Diff: true
  Logging Driver: json-file
  Cgroup Driver: cgroupfs
  Plugins:
    volume: local
    Network: bridge host ipvlan macvlan null overlay
    Log: awslogs fluentd gcplogs gelf journald json-file local logentries
  Swarm: inactive
  Runtimes: runc
  Default Runtime: runc
  Init Binary: docker-init
  containerd version: 894b81a4b802e4eb2a91d1ce216b8817763c29fb
  runc version: 425e105d5a03fabd737a126ad93d62a9eeede87f
  init version: fec3683
  Security Options:
    seccomp
    Profile: default
  Kernel Version: 3.10.0-957.21.3.el7.x86_64
  Operating System: CentOS Linux 7 (Core)
  OSType: linux
  Architecture: x86_64
  CPUs: 1
  Total Memory: 1.777GiB
  Name: centos1
  ID: 74NC:BK5I:LTAK:XIA3:6JCS:XMVK:6JX6:E36T:LV7I:YJBO:VTDD:QKF7
  Docker Root Dir: /var/lib/docker
  Debug Mode: false
  Registry: https://index.docker.io/v1/
  Labels:
  Experimental: false
  Insecure Registries:
    127.0.0.0/8
  Registry Mirrors:
    https://xxx.mirror.aliyuncs.com/
  Live Restore Enabled: false
```

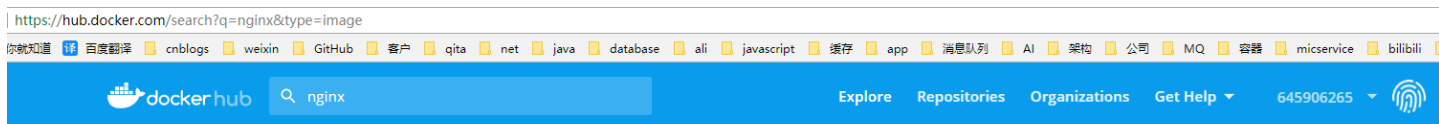
3.docker基础用法

帮助文档可让你快速了解它， 输入docker， 或 docker image --help 查看帮助文档
安装一个nginx来快速了解docker的使用， 以 nginx 为例进行说明：

- 1.查询镜像，有2种方式，一是使用 `docker search nginx` 命令查询，二是在<https://hub.docker.com/> 页面上输入nginx查询
- 2.下载镜像：`docker pull nginx:latest`，或指定标签版本 tag
- 3.查看镜像列表：`docker image ls`
- 4.启动容器：`docker run -it -d --name nginx1 -p 8080:80 nginx:latest`，运行一个有交互窗口的容器，名称叫nginx1，且容器的80端口映射宿主机的8080端口，启动后可通过 `http://宿主机IP:8080` 访问nginx
- 5.查看容器：`docker ps`，`docker ps -a`（可查看已停止的容器）
- 6.停止容器：`docker stop nginx1`
- 7.删除容器：`docker rm nginx1`，再通过`docker ps -a`就看不到停止的容器了
- 8.查看容器详细信息：`docker inspect nginx1`
- 9.查看容器中日志信息：`docker container logs nginx1`

上面命令截图

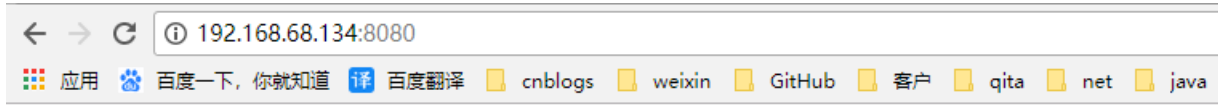
```
[root@centos1 yum.repos.d]# docker search nginx
NAME                DESCRIPTION                STARS     OFFICIAL   AUTOMATED
nginx                Official build of Nginx.   11871     [OK]
jwilder/nginx-proxy Automated Nginx reverse proxy for docker con... 1644
richarvey/nginx-php-fpm Container running Nginx + PHP-FPM capable of... 738
linuxserver/nginx   An Nginx container, brought to you by LinuxS... 73
bitnami/nginx        Bitnami nginx Docker Image 70
tiangolo/nginx-rtmp Docker image with Nginx using the nginx-rtmp... 51
nginxdemos/hello     NGINX webserver that serves a simple page co... 24
jc21/nginx-proxy-manager Docker container for managing Nginx proxy ho... 23
nginx/nginx-ingress  NGINX Ingress Controller for kubernetes     21
jlesage/nginx-proxy-manager Docker container for Nginx Proxy Manager     20
schmunk42/nginx-redirect A very simple container to redirect HTTP tra... 17
blacklabelops/nginx Dockerized Nginx Reverse Proxy Server.       12
centos/nginx-18-centos7 Platform for running nginx 1.8 or building n... 11
centos/nginx-112-centos7 Platform for running nginx 1.12 or building ... 10
nginxinc/nginx-unprivileged Unprivileged NGINX Dockerfiles              9
webdevops/nginx      Nginx container                             8
nginx/nginx-prometheus-exporter NGINX Prometheus Exporter                   5
sophos/nginx-vts-exporter Simple server that scrapes Nginx vts stats a... 5
lscience/nginx        Nginx Docker images that include Consul Temp... 5
mailu/nginx           Mailu nginx frontend                         3
pebbletech/nginx-proxy nginx-proxy sets up a container running ngin... 2
travix/nginx          Nginx reverse proxy                          2
centos/nginx-110-centos7 Platform for running nginx 1.10 or building ... 0
wodby/nginx           Generic nginx                                 0
ansibleplaybookbundle/nginx-apb An APB to deploy NGINX                      0
```



```
[root@centos1 yum.repos.d]# docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest   5a3221f0137b  11 days ago   126MB
[root@centos1 yum.repos.d]# docker restart nginx1
nginx1
[root@centos1 yum.repos.d]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
0ed369474576  nginx:latest "nginx -g 'daemon of..." 13 hours ago  Up 3 seconds  0.0.0.0:8080->80/tcp  nginx1
[root@centos1 yum.repos.d]# docker stop nginx1
nginx1
[root@centos1 yum.repos.d]# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
0ed369474576  nginx:latest "nginx -g 'daemon of..." 13 hours ago  Exited (0) 7 seconds ago  nginx1
[root@centos1 yum.repos.d]# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
```

```
[root@centos1 yum.repos.d]# docker run -it -d --name nginx1 -p 8080:80 nginx:latest
26921c81544c739e02af3e90e2a3c6725cf04bec4918629b2680da4a3f437a27
[root@centos1 yum.repos.d]# docker container logs nginx1
192.168.68.1 - - [27/Aug/2019:14:32:25 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (windows NT 10.0; win64; x64) AppleWebKit/537.36 (KHTML, li
ke Gecko) Chrome/64.0.3282.186 Safari/537.36" "-"
```

访问刚刚运行的 nginx 容器



Welcome to nginx!

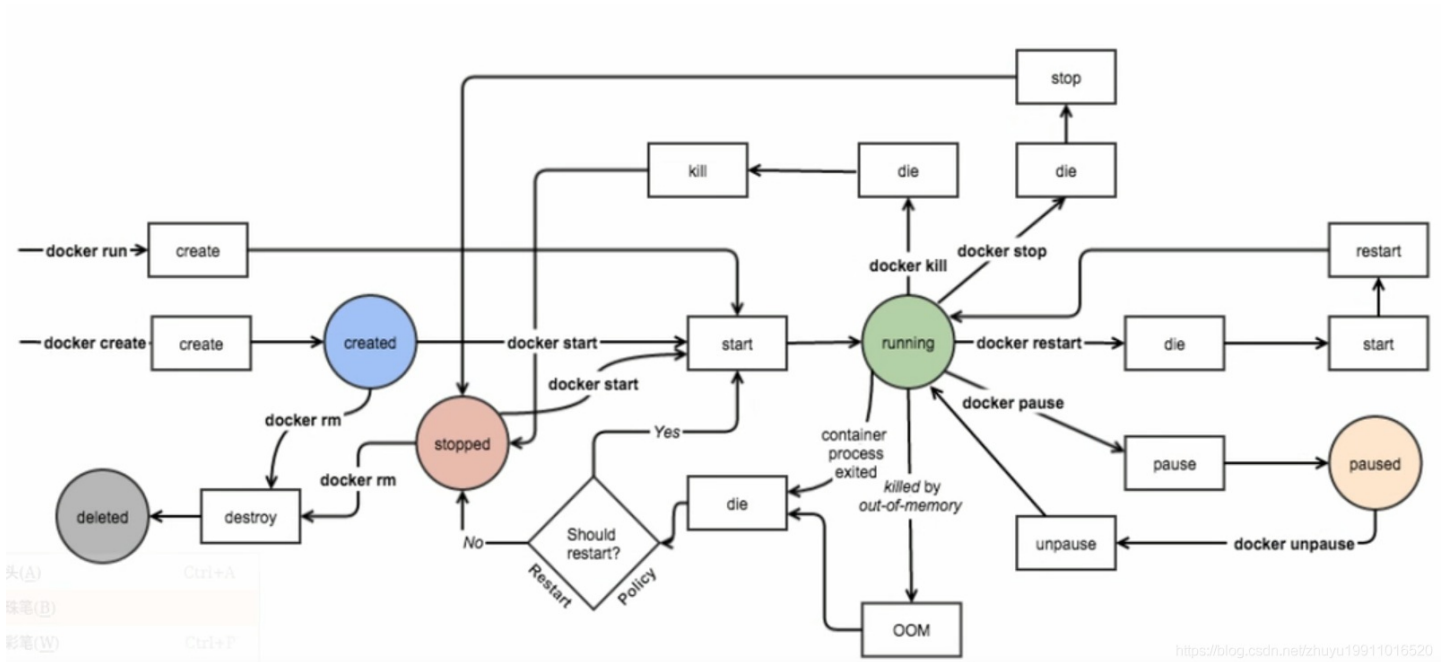
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

<https://blog.csdn.net/zhuyu19911016520>

docker运行状态图:



<https://blog.csdn.net/zhuyu19911016520>

阿里云镜像加速器，需要先注册，然后在 容器镜像服务-镜像加速器中查看，如下图：

容器镜像服务 | 镜像加速器

默认实例 | 加速器

使用加速器可以提升获取Docker官方镜像的速度

加速器地址

https://[redacted].mirror.aliyuncs.com [复制](#)

操作文档

Ubuntu **CentOS** Mac Windows

1. 安装 / 升级Docker客户端

推荐安装 1.10.0 以上版本的Docker客户端，参考文档 [docker-ce](#)

2. 配置镜像加速器

针对Docker客户端版本大于 1.10.0 的用户

您可以通过修改daemon配置文件 `/etc/docker/daemon.json` 来使用加速器

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<'EOF'
{
  "registry-mirrors": ["https://[redacted].mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

<https://blog.csdn.net/zhuyu19911016520>

下一篇：云原生应用3：docker镜像管理

配置一个 jenkins 容器：vi start_jenkins.sh，内容如下：

```
docker run -d --name jenkins -u root -it --restart=always
-p 7080:8080 -p 50000:50000
-p 8190:8190
-p 8191:8191
--privileged=true
-v /var/jenkins_home:/var/jenkins_home
-v /home:/home
-v /usr/bin/docker:/usr/bin/docker
-v /var/run/docker.sock:/var/run/docker.sock
-v /usr/lib64/libltdl.so.7:/usr/lib/x86_64-linux-gnu/libltdl.so.7
-v /etc/localtime:/etc/localtime -v /etc/timezone:/etc/timezone
jenkinsci/blueocean
```

-v /etc/localtime:/etc/localtime -v /etc/timezone:/etc/timezone 这行特别重要，表示容器的时间使用宿主机的时间，否则你应用内容的时间会少8个小时

docker也可以离线安装

离线包下载地址tg: https://download.docker.com/linux/static/stable/x86_64/

离线包下载地址rpm: https://download.docker.com/linux/centos/7/x86_64/stable/Packages/

上传文件，解压

```
[root@dev1 opt]# tar -xvf docker-18.06.1-ce.tgz
docker/
docker/docker-containerd
docker/docker-proxy
docker/docker
docker/docker-runc
docker/dockerd
docker/docker-containerd-ctr
docker/docker-containerd-shim
docker/docker-init
```

复制 [root@dev1 opt] cp docker/* /usr/bin/

设置开机自启动，vim /etc/systemd/system/docker.service，内容如下：

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

添加文件权限并启动docker: `chmod 777 -R /etc/systemd/system/docker.service`

重载unit配置文件: `systemctl daemon-reload`

启动Docker(2种方式): `systemctl start docker / dockerd &`

设置开机自启: `systemctl enable docker.service`

关闭selinux: `vi /etc/selinux/config`, 修改 `SELINUX=disabled` 重新启动系统即可
修改iptables

```
vi /etc/sysctl.conf
```

添加以下内容

```
net.bridge.bridge-nf-call-ip6tables = 1
```

```
net.bridge.bridge-nf-call-iptables = 1
```

刷新配置: `sysctl -p`

测试

```
[root@centos75-1 ~]# docker info
```