

Docker读书笔记

镜像

1.从仓库下载镜像到本地

```
#不指定TAG时获取latest标签的镜像,registry默认为
registry.hub.docker.com
docker pull [registry/]NAME[:TAG]
#删除某个仓库中的某个镜像
docker rmi [REPOSITORY:TAG]
#创建一个容器，并在其中运行指定应用
docker run -t -i [REPOSITORY:TAG] [COMMAND]
#查看镜像的完整信息
docker inspect [IMAGE_ID]
```

2.查看本地已有镜像

```
docker images
```

3.搜索远端仓库的镜像

```
docker search [key-word]
```

4.创建镜像

```
#-a镜像作者，-m提交信息，-p提交时暂停镜像运行
docker commit [-a ,-m,-p] CONTAINER_ID [REPOSITORY[:TAG]]
#导出镜像到本地Tar文件
docker save -o file_name.tar [REPOSITORY[:TAG]]
#导入本地文件到镜像 docker load --input file_name.tar
```

5.向Docker Hub推送自己的镜像

```
docker push [REPOSITORY[:TAG]]
```

容器

1.创建并且启动一个容器

```
#基于IMAGE创建容器,-i标准输入输出打开,-t指定一个终端,-d守护运行态的容器
docker create -it [REPOSITORY[:TAG]]
#启动容器 docker start [container]
#基于IMAGE创建容器,并且启动
docker run -it [REPOSITORY[:TAG]]
#容器日志
docker logs [container]
#终止容器,发送SIGTERM,等待若干秒后发送SIGKILL
docker stop [container]
```

2.获取容器的进程PID

```
PID=$(docker inspect --format "{{.State.Pid}}" )
```

3.导入和导出容器

```
#导出运行中的容器到一个文件,创建容器的快照
docker export [CONTAINER] > file_name.tar
#导入容器的快照文件,成为一个镜像
cat file_name.tar | docker import - [REPOSITORY[:TAG]]
```

4.删除容器

```
docker rm [container id]
```

仓库

```
####1.注册服务器 -- 存放仓库的具体服务器
```

2.下载远程镜像

```
docker pull [registry]/[REPOSITORY[:TAG]]
```

3.搭建本地私有仓库

```
#自动下载一个名为registry的容器
docker run -d -p 5000: 5000 registry
#kolla建立的私有仓库
docker run -d -v [local_path]:[container_path] -p 4000:5000 --
restart=always --name registry registry:2
```

数据管理

1.数据卷

```
#在docker run命令中使用-v在容器内创建一个数据卷
docker run -d -P --name [container_name] -v [data_volume_path]
[REPOSITORY[:TAG]] [COMMAND]
#可以指定挂载一个本地目录到容器中作为数据卷
docker run -d -P --name [container_name] -v
[local_path:data_volume_path:mode] [REPOSITORY[:TAG]] [COMMAND]
#使用-v命令删除容器挂载的数据卷，否者对应的数据卷不会删除
docker rm -v [container_id]
```

2.数据卷容器

```
#创建数据卷容器，本质上就只是一个容器而已，但是该容器中有一个数据卷，其
他容器可以通过--volumes-from来挂载该数据卷中，达到若干容器共享这个数据
卷
docker run -it [data_volume_path] --name [container_name_1]
[REPOSITORY[:TAG]]
docker run -it --volumes-from [container_name_1] --name
[container_name_2] [REPOSITORY[:TAG]]
```

3.数据卷的相关操作

#备份,把数据卷打包到本地目录

```
docker run -it --volume-from [container_name_1] --name worker -v  
${pwd}:/backup [REPOSITORY[:TAG]] tar cvf /backup/backuo.tar  
[data_volume_path]
```

#恢复,类似备份命令

```
docker run -it --volume-from [container_name_to_restore] --name  
worker -v ${pwd}:/backup [REPOSITORY[:TAG]] tar xvf  
/backup/backup.tar
```

配置修改

对Centos来说,配置文件为/etc/sysconfig/docker, docker守护进程启动时会读取该文件。

```
1 # docker启动脚本 /usr/lib/systemd/system/docker.service  
2 [Unit]  
3 Description=Docker Application Container Engine  
4 Documentation=http://docs.docker.com  
5 After=network.target  
6 Wants=docker-storage-setup.service  
7 Requires=docker-cleanup.timer  
8  
9 [Service]  
10 Type=notify  
11 NotifyAccess=all  
12 EnvironmentFile=-/run/containers/registries.conf  
13 EnvironmentFile=-/etc/sysconfig/docker  
14 EnvironmentFile=-/etc/sysconfig/docker-storage  
15 EnvironmentFile=-/etc/sysconfig/docker-network  
16 Environment=GOTRACEBACK=crash  
17 Environment=DOCKER_HTTP_HOST_COMPAT=1  
18 Environment=PATH=/usr/libexec/docker:/usr/bin:/usr/sbin  
19 ExecStart=/usr/bin/dockerd-current \  
20     --add-runtime docker-runc=/usr/libexec/docker/docker-runc-current \  
21     --default-runtime=docker-runc \  
22     --exec-opt native.cgroupdriver=systemd \  
23     --userland-proxy-path=/usr/libexec/docker/docker-proxy-current \  
24     $OPTIONS \
```

```

25     $DOCKER_STORAGE_OPTIONS \
26     $DOCKER_NETWORK_OPTIONS \
27     $ADD_REGISTRY \
28     $BLOCK_REGISTRY \
29     $INSECURE_REGISTRY\
30     $REGISTRIES
31 ExecReload=/bin/kill -s HUP $MAINPID
32 LimitNOFILE=1048576
33 LimitNPROC=1048576
34 LimitCORE=infinity
35 TimeoutStartSec=0
36 Restart=on-abnormal
37 MountFlags=slave
38 KillMode=process
39
40 [Install]
41 WantedBy=multi-user.target
42

```

注意ExecStart中定义了一些变量，这些变量可以在/etc/sysconfig/docker中定义

```

1 # /etc/sysconfig/docker
2
3 # Modify these options if you want to change the way the docker daemon runs
4 OPTIONS='--selinux-enabled --log-driver=journald --signature-verification=false'
5 if [ -z "${DOCKER_CERT_PATH}" ]; then
6     DOCKER_CERT_PATH=/etc/docker
7 fi
8
9 # Do not add registries in this file anymore. Use /etc/containers/registries.conf
10 # from the atomic-registries package.
11 #
12
13 # docker-latest daemon can be used by starting the docker-latest unitfile.
14 # To use docker-latest client, uncomment below lines
15 #DOCKERBINARY=/usr/bin/docker-latest
16 #DOCKERDBINARY=/usr/bin/dockerd-latest
17 #DOCKER_CONTAINERD_BINARY=/usr/bin/docker-containerd-latest
18 #DOCKER_CONTAINERD_SHIM_BINARY=/usr/bin/docker-containerd-shim-latest

```

```
19 INSECURE_REGISTRY = '--insecure-registry 172.24.9.11:4000' # 配置不安全镜像仓库
20 OPTIONS='--selinux-enabled' #开启selinux
21 DOCKER_CERT_PATH=/etc/docker #证书目录
22
```

配置/etc/docker/daemon.json 可以配置registry-mirrors , 在www.daocloud.io上可以配置镜像加速