

Docker初体验

2017-02-22 17:13:04

最近试用了一下比较火的docker容器技术，技术比较菜，踩了很多坑，简单总结一下使用心得。1、基本概念 镜像（Image），容器（Container），仓库（Repository）就我来理解，仓库是存储镜像的地方，我们从仓库获得镜像，有了镜像，我们可以根据相应的镜像启动容器了，容器与镜像可以看做类和对象的关系（或者操作系统和具体运行该操作系统的机器），容器是镜像的实例，它本质上是进程，容器启动起来后，我们就可以在容器上运行一些命令了。需要注意的是：如果容器内启动的命令不是那种持续运行的命令的话，容器启动后，命令执行结束之后，容器将自动关闭，我刚开始创建启动容器的时候就总犯错，因为容器启动之后，它就自己关闭了。。。2.安装 我使用的是centos操作系统，不得不说的是centos版本6安装docker全是坑，我为了填坑，把操作系统弄崩了，建议像我这样的小白还是使用centos 7以上的操作系统吧 1) 添加 yum 软件源 `sudo tee /etc/yum.repos.d/docker.repo <<-'EOF'`
[dockerrepo] name=Docker Repository baseurl=https://yum.dockerproject.org/repo/main/centos/7/ enabled=1
gpgcheck=1 gpgkey=https://yum.dockerproject.org/gpg EOF 2) 更新yum 软件源并安装docker `sudo yum update sudo yum install docker-engine` 3) 启动docker `sudo systemctl enable docker sudo systemctl start docker` 4) 添加docker组（不添加的话，访问不了docker） `sudo groupadd docker sudo /usr/sbin/usermod -aG docker $USER`（\$USER是变量，用户名，如root等）需要注意的是：如果你想要远程用户可以访问你的docker的话，需要修改docker的配置文件，设置端口 `sudo mkdir /etc/systemd/system/docker.service.d sudo vim /etc/systemd/system/docker.service.d/docker.conf` 然后在文件中输入（监听端口50000）：`ExecStart=/usr/bin/dockerd -H unix:///var/run/docker.sock -H tcp://0.0.0.0:50000` 3.简单样例 我使用的node镜像，下载最新的node镜像 `docker pull node:latest` 然后运行一个简单的容器 `docker run node:latest node -e 'console.log(2)'` 然后运行命令 `docker ps -a` 来查看刚建立的容器，可以发现容器已经退出了 4.集群 新版的docker内置了swarm集群，旧版的得下载swarm镜像才可以。1) 使用swarm镜像搭建swarm集群（使用命令的时候把ip1，ip2和ip3以及port替换成自己的ip地址和端口号，如113.22.32.12:50000，token://后面的字符串也替换成自己的）`node1: ip1:port node2: ip2:port manager: ip3:port` a)`docker run --rm swarm create` 获得token，之后使用该token创建(例token：`8c26457b1f36b454e9dc679bd791680b`) b)添加节点（在node1或者node2上面运行都可以，但是如果是在同一台机器上运行，如果该机器宕机，两个节点都无法访问了）`docker run -d swarm join --addr=ip1:port token://8c26457b1f36b454e9dc679bd791680b` `docker run -d swarm join --addr=ip2:port token://8c26457b1f36b454e9dc679bd791680b` 使用`docker run --rm swarm list token://8c26457b1f36b454e9dc679bd791680b`显示所有节点 c) 注意manager可以跟node起在同一台机器上，但是这样的话，要注意port不可以一样 `docker run -d -p port:2375 swarm manage token://8c26457b1f36b454e9dc679bd791680b` 在任一台机器上运行manager（在哪台机器上运行，ip就是哪台机器的ip地址） d)访问manager获得所有的容器 `docker -H ip3:port ps -a` 需要注意的是：如果一台机器宕机了，它上面的容器不会迁移，manager宕机的话，不影响node，只是没办法通过该manager进行访问了，docker启动的时候必须配置访问的端口号即tcp://0.0.0.0:50000设置端口为50000 2) 内置swarm 这个swarm比较简单，在manager节点上面运行`docker swarm init`，该节点自动成为manager 然后在该节点上运行 `docker swarm join-token manager`获得加入一个manager的命令 `docker swarm join-token worker`获得加入一个worker的命令 在目标机器上运行相应的命令即可将目标机器加入集群，作为worker或者manager 这样就获得了集群，之后可以在manager上面使用`docker node`管理节点，使用`docker service`管理服务 我们可以在集群上面创建我们的分布式服务了（`docker service create [OPTIONS] IMAGE [COMMAND] [ARG...]`），服务可以有多个task，分别运行在集群的各个node之上，也可以指定每个node都运行一个task。需要注意的是：在集群上面创建的服务自动负载均衡（在node上），而且node宕机，服务迁移（服务分多个task，每个task在一个container上面运行）；当有多个manager的时候，竞选一个作为leader，leader宕机的话，重新竞选，不过前提是满足剩余大多数manager才竞选，如果只有两个manager（包括leader），leader宕机，将不竞选leader。

