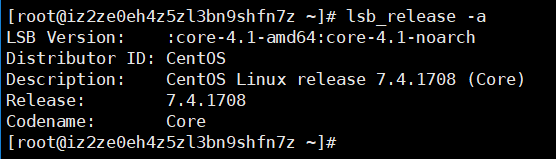
**Centos上搭建Hyperledger Fabric 1.2**

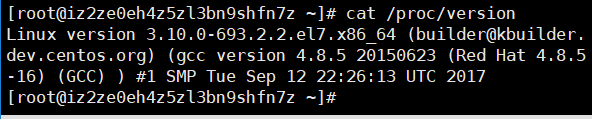
本文是在阿里云服务器上进行Hyperledger Fabric 1.2的环境搭建。

**第一部分：服务器环境准备**

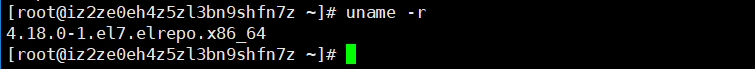
1、使用 lsb\_release -a命令查看服务器系统信息如下：



2、使用cat /proc/version命令查看内核版本



3、为了防止后期运行节点时因Linux内核版本过低导致运行失败，建议将Linux内核版本升级至4.x以上（内核版本升级参考另外一篇文章）



**第二部分：安装Docker和Docker-Compose**

Fabric的节点通过Docker容器来运行，启动Fabric网络中的节点需要预先安装Docker、Docker-Compose，然后在网上拉取相关的Docker镜像，再通过配置Compose文件来启动各节点。

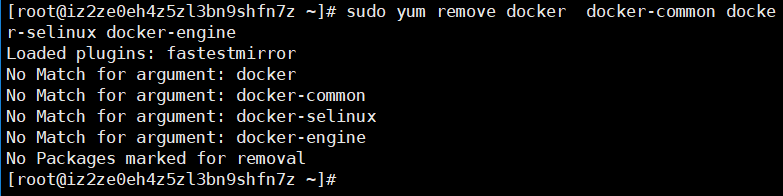
1. 安装Docker

更新本地依赖（本操作是为了将所有的内置库更新到最新版，因为Docker最新版本的安装需要对应的依赖都是相对较新的版本，为了避免安装依赖的麻烦，此操作很重要。）：

# sudo yum update

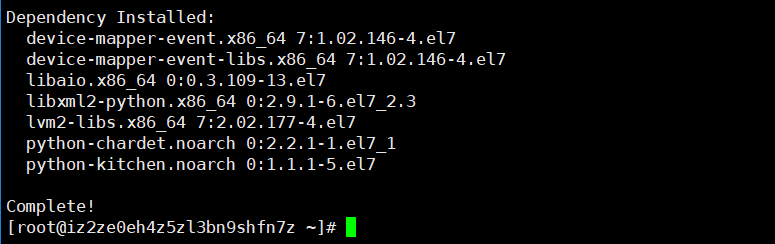
如果安装过旧版本的话需要先卸载旧版本：

# sudo yum remove docker docker-common docker-selinux docker-engine



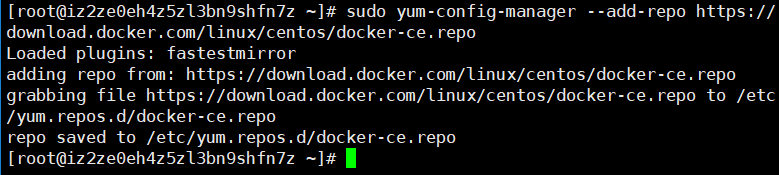
安装需要的软件包（yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动依赖的）：

# sudo yum install -y yum-utils device-mapper-persistent-data lvm2



设置yum源：

# sudo yum-config-manager –add-repo https://download.docker.com/linux/centos/docker-ce.repo

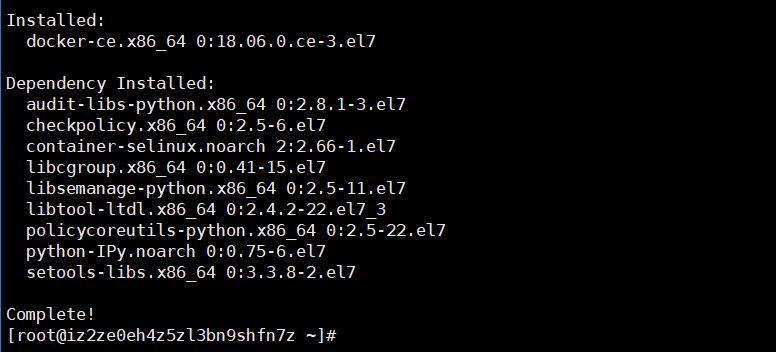


可以查看仓库中所有docker版本然后选择特定版本进行安装：

# yum list docker-ce --showduplicates | sort –r

本文安装最新的稳定版：

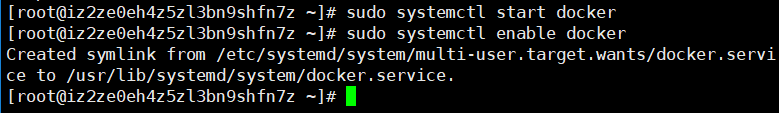
# sudo yum install docker-ce



启动并加入开机启动：

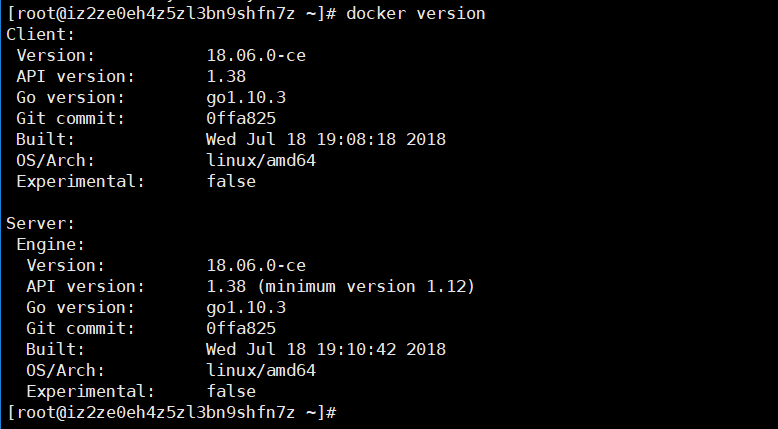
# sudo systemctl start docker

# sudo systemctl enable docker



查看docker版本验证是否安装成功，有client和service两部分表示docker安装启动都成功了：

# docker version



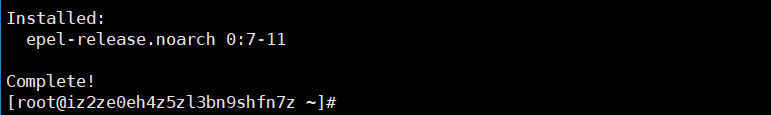
备注：

[安装Docker](https://docs.docker.com/install/linux/docker-ce/centos/)

1. 安装Docker-Compose

安装epel依赖：

# yum -y install epel-release



安装pip工具：

# yum install python-pip

更新pip：

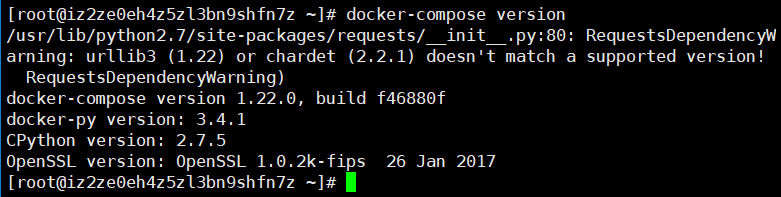
# pip install --upgrade pip

安装docker-compose：

# pip install docker-compose

查看docker-compose版本：

# docker-compose version



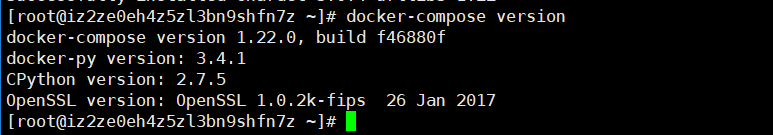
可以看到图中有一个waring，原因是：python库中urllib3 (1.22) or chardet (2.2.1) 的版本不兼容，可通过执行以下下命令可解决：

# pip uninstall urllib3

# pip uninstall chardet

# pip install requests

再次查看docker-compose版本，warning消除：



备注：

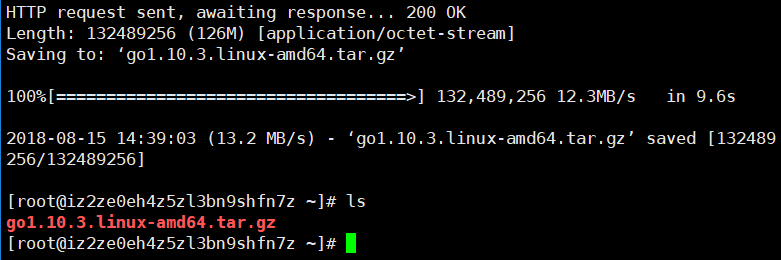
[安装Dokcer-Compose](https://docs.docker.com/compose/install/#install-compose)

**第三部分：安装Go语言环境和Nodejs环境**

1、安装Go语言环境

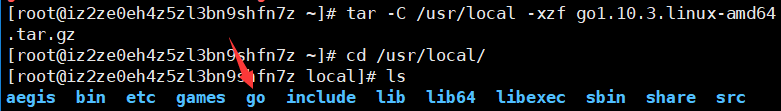
下载安装包：

# wget https://storage.googleapis.com/golang/go1.10.3.linux-amd64.tar.gz



将安装包解压到/usr/local/：

# tar -C /usr/local -xzf go1.10.3.linux-amd64.tar.gz



在~/.bahsrc文件中配置环境变量：

# vim ~/.bashrc

添加以下内容：

export GOROOT=/usr/local/go

export GOPATH=$HOME/go

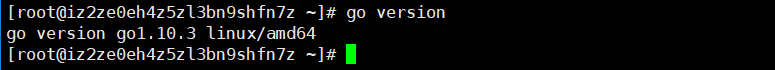
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin

保存文件退出，并执行以下命令使环境变量设置立即生效：

# source ~/.bashrc

查看go的版本信息，确定是否成功安装：

# go version



备注：

[安装Go](https://golang.org/doc/install)

1. 安装nodejs环境

安装npm：

# yum install npm

安装nvm：

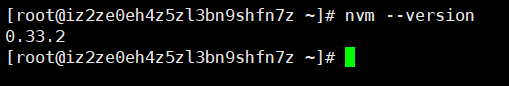
wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.33.2/install.sh | bash

再执行以下命令设置环境变量：

# export NVM\_DIR="$HOME/.nvm"

# [ -s "$NVM\_DIR/nvm.sh" ] && . "$NVM\_DIR/nvm.sh"

输入nvm --version查看安装：

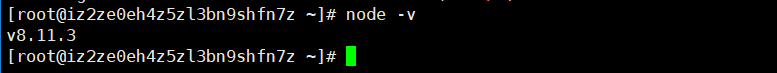


安装nodejs：

# nvm install v8.11.3

查看node版本：

# node –v



备注：如果之前安装过其他版本的node，需要先进行删除卸载。

[Nodejs安装参考链接](https://www.tiny-calf.com/2017/07/20/Node.js%E5%BF%AB%E9%80%9F%E5%AE%89%E8%A3%85/)

**第四部分 安装git并拉取fabric源码**

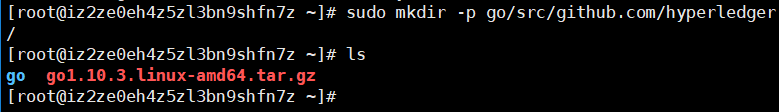
1. 安装git 拉取fabric

安装git：

# yum install git

新建文件夹放置代码：

# sudo mkdir -p go/src/github.com/hyperledger/



进入文件夹目录：

# cd go/src/github.com/hyperledger/



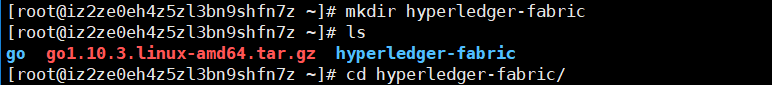
拉取fabric的代码：

# git clone <https://github.com/hyperledger/fabric.git>

1. 下载fabric-samples项目

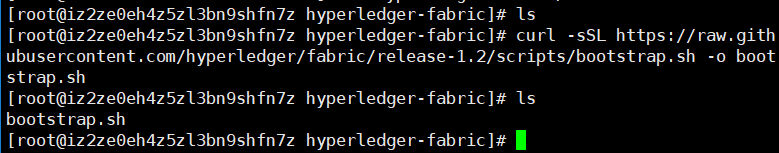
创建放置fabric-samples项目的目录：

# mkdir hyperledger-fabric



然后进入该目录，用curl下载引导脚本bootstrap.sh：

# curl -sSL https://raw.githubusercontent.com/hyperledger/fabric/release-1.2/scripts/bootstrap.sh -o bootstrap.sh



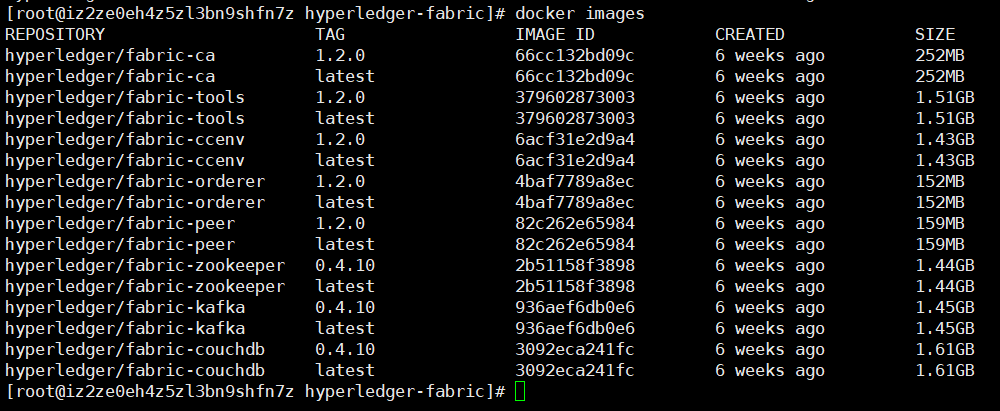
执行引导脚本（这一步通过运行脚本去安装fabric，并且会下载fabric-sample项目源码，而且还会摘取运行节点所需的docker镜像。下载过程需要耗费比较长的时间。）：

# sudo bash ./bootstrap.sh 1.2.0 1.2.0 0.4.10

备注：由于网络原因，命令执行可能会失败，多执行几次。

下载完，可以查看已经下载的docker相关镜像：

# docker images



添加环境变量：

# vi ~/.bashrc

增加以下内容：

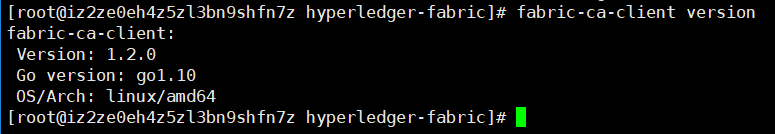
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin:/root/hyperledger-fabric/fabric-samples/bin

使环境变量立即生效：

# source ~/.bashrc

检验环境变量是否设置成功：

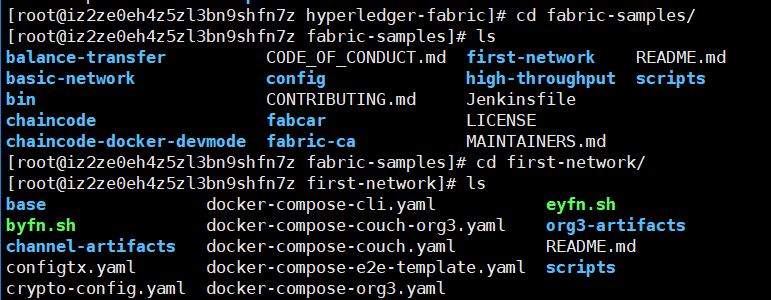
# fabric-ca-client version



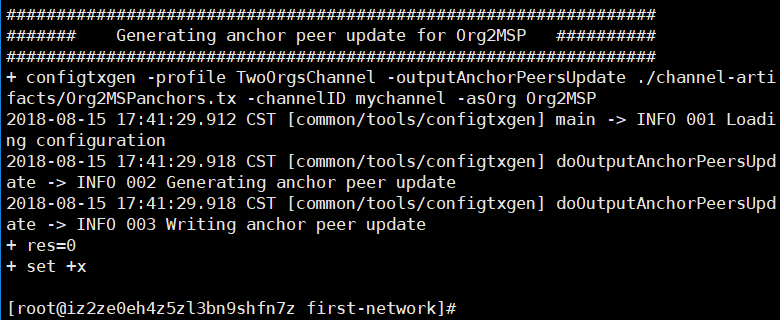
**第五部分：运行测试网络**

1. 运行first-network测试网络

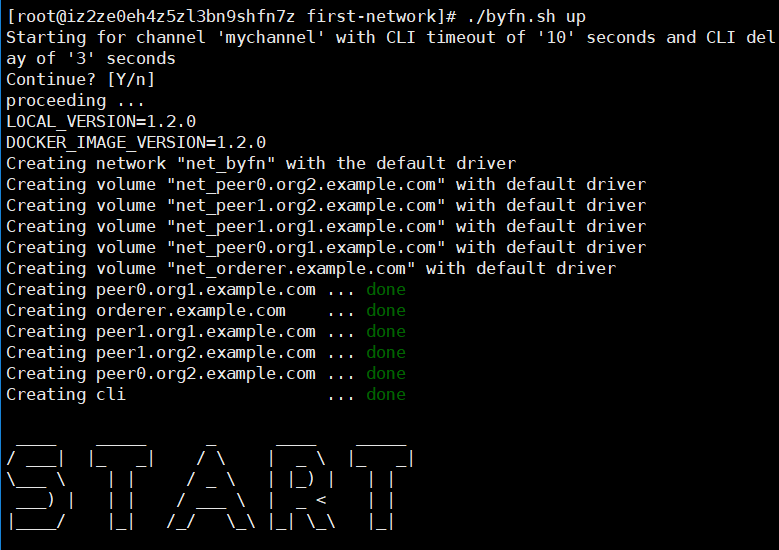
进入fabric-samples/first-network/目录下：



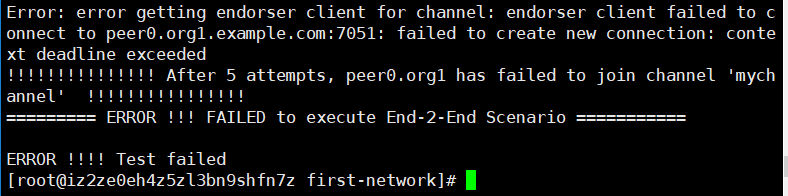
执行./byfn.sh generate生成相应的配置文件：



执行./byfn.sh up 启动网络：



运行出现以下错误：

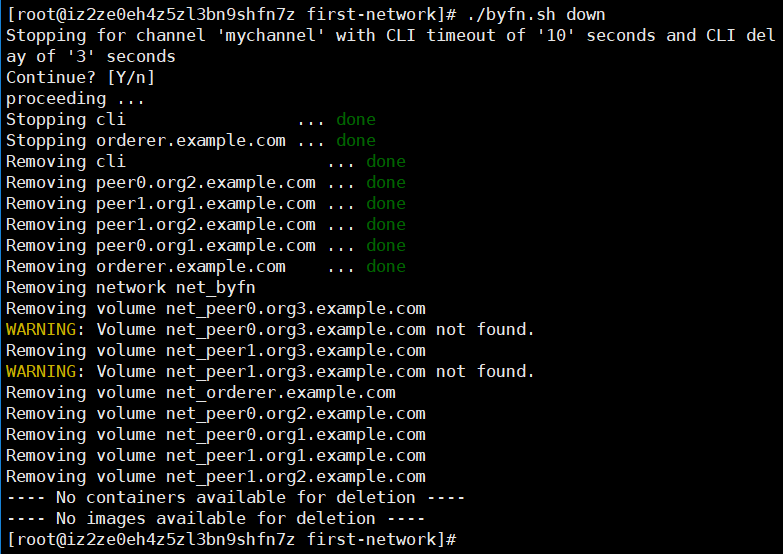


查了很久都找不到原因，最后发现是因为阿里云服务器自身的问题导致运行失败的，参考链接如下：

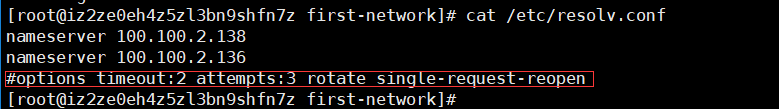
[阿里云服务器运行Hyperledger fabric测试网络异常分析1](https://yq.aliyun.com/articles/238940)

[阿里云服务器运行Hyperledger fabric测试网络异常分析2](https://blog.csdn.net/weishiym/article/details/79629297)

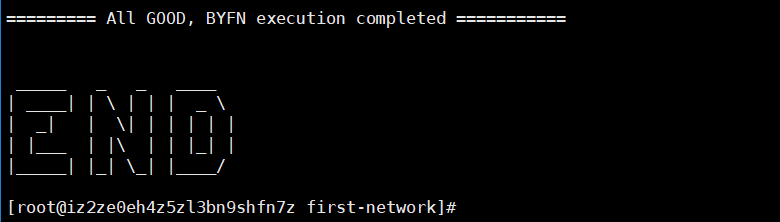
先执行./byfn.sh down关闭网络：



由于阿里云服务器自身的问题，需要修改 /etc/resolv.conf 配置，将 options timeout:2 attempts:3 rotate single-request-reopen 这一行内容注释掉：



再次执行./byfn.sh up 启动网络，运行成功：



可以用docker命令查看镜像运行情况：

# docker ps



2、运行e2e\_cli测试网络

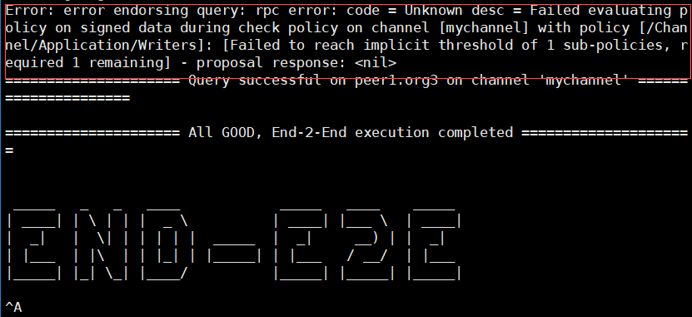
进入e2e\_cli目录：

# cd /root/go/src/github.com/hyperledger/fabric/examples/e2e\_cli

执行./network\_setup.sh up：



虽然执行成功，但是日志中有一个错误：

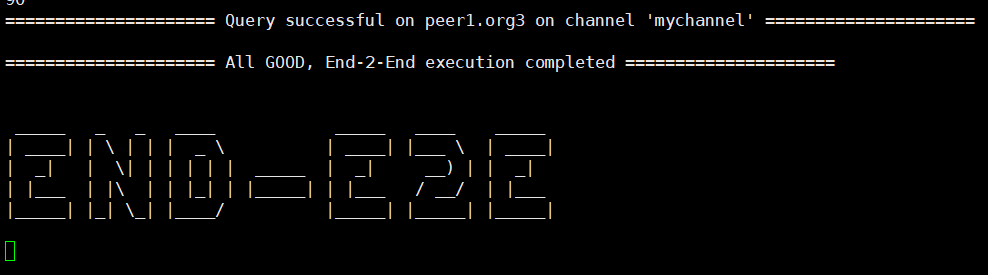


虽然运行成功了，但是这个错误是一个bug，官方jira已经提交了这个bug，但是不知道为什么最新代码还没有解决这个问题，经过查找资料，找到一个博主提供 的规避方案解决了该问题（[参考链接](https://zhulg.github.io/2018/08/02/fabric1-2%E4%B8%8Be2e-cli%E9%97%AE%E9%A2%98%E8%AE%B0%E5%BD%95/)）：

修改 e2e\_cli下的configtx.yaml，在Org3处进行添加Org3MSP.member，如下：



执行./network\_setup.sh down清空镜像，再次执行./network\_setup.sh up启动网络，错误消除：



**第六部分 其他**

1、参考链接

[Hyperledger Fabric 官网搭建教程](https://hyperledger-fabric.readthedocs.io/en/latest/getting_started.html)

[参考链接1](https://hyperledgercn.github.io/hyperledgerDocs/build_network_zh/)

[参考链接2](http://www.lijiaocn.com/%E9%A1%B9%E7%9B%AE/2018/07/19/hyperledger-fabric-1-2-0.html)

[参考链接3](https://github.com/aberic/fabric-net-server)

[参考链接4](https://hub.docker.com/u/hyperledger/)

1. docker常用命令

启动：

# service docker start

停止：

# service docker stop

重启：

# servcie docker restart

查看启动的容器列表：

# docker ps

查看创建的所有容器：

# docker ps –a

启动某个容器：

# docker start docker\_name

停止某个容器：

# docker stop docker\_name

重启某个容器：

# docker restart docker\_name

查看指定容器的日志记录：

# docker logs -f docker\_name

终止陈旧容器：

# docker rm -f $(docker ps -aq)

清理缓存：

# docker network prune

杀死所有正在运行的容器：

# docker kill $(docker ps -a -q)

删除所有已经停止的容器：

# docker rm $(docker ps -a -q)

删除所有镜像：

# docker rmi $(docker images -q)

强制删除所有镜像：

# docker rmi -f $(docker images -q)