1. 安装步骤
2. 切换至root用户
3. 安装ssh服务（教程02有详细介绍）
4. 安装git、curl、pip

yum install git -y

yum install curl -y

yum install epel-release

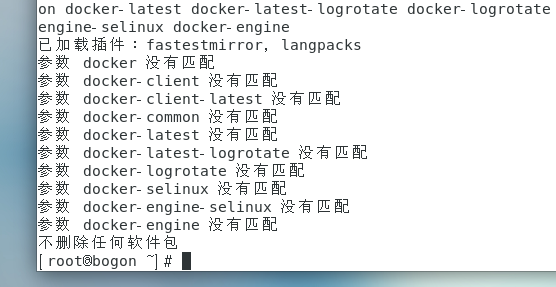
yum install python-pip -y

pip install --upgrade pip

1. 安装docker
2. 如果安装过docker，请使用下列命令卸载：

yum remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-selinux docker-engine-selinux docker-engine

因为我没有安装docker，使用命令后如图：



1. 安装yum-utils提供的yum-config-manager，device-mapper-persistent-data和lvm2，命令如下：yum install -y yum-utils device-mapper-persistent-data lvm2

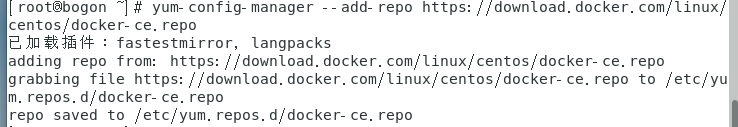
如图：



1. 设置稳定的存储库

yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

如图：



1. 可以选择性的启用edge和测试存储库，这些存储库包含在Docker中，默认是禁用的，命令如下：

yum-config-manager --enable docker-ce-edge

 yum-config-manager --enable docker-ce-test

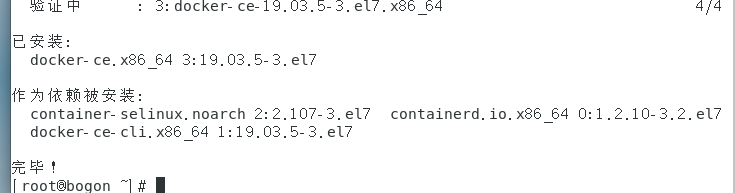
1. 以下命令为禁用edgecunc存储库：

yum-config-manager --disable docker-ce-edge

1. 安装最新版的docker-ce

yum install docker-ce –y

最后complete！如图



1. 查看docker版本（docker –version）如图

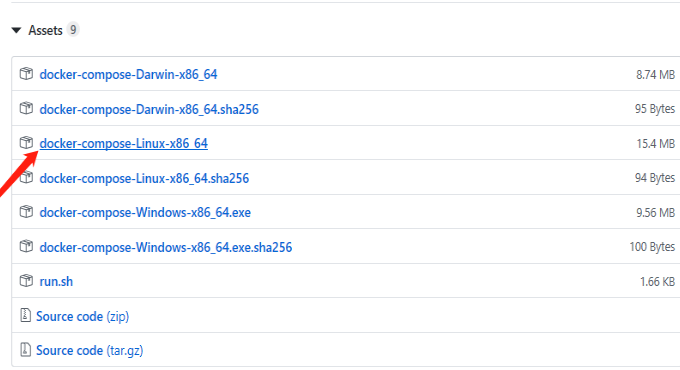


1. 启动docker并设置为开机自启

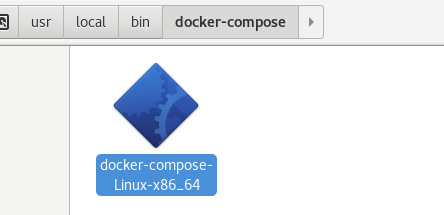
systemctl start docker

chkconfig docker on

1. 安装docker-compose（离线安装）：
2. 下载最新版的docker-compose-Linux-x86\_64，下载地址为：<https://github.com/docker/compose/releases>。下载完后上传至/tmp/docker/docker-compose文件夹下，若无此文件夹，创建即可。（已下载好）

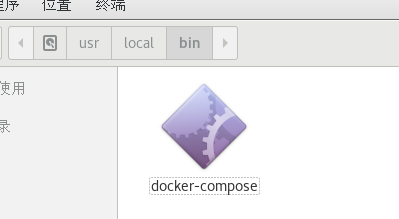


1. 下载好之后将其移动至/usr/local/bin，文件夹下



1. 使用以下命令进行改名：

rename docker-compose-Linux-x86\_64 docker-compose docker-compose-Linux-x86\_64



1. 赋予可执行权限：

chmod +x /usr/local/bin/docker-compose

1. 查看版本信息：docker-compose --version
2. 安装golang
3. 安装必要的依赖：

yum install libtool libltdl-dev

1. 下载go语言包：<https://studygolang.com/dl>（已下载好）
2. 将下载的语言包go1.12.9.linux-amd64.tar.gz上传至文件夹/tmp/docker
3. 解压至/usr/local目录下：# tar -C /usr/local -zxvf go1.12.9.linux-amd64.tar.gz
4. 配置go语言环境变量，参照配置jdk环境变量，在/etc/profile文件后加以下内容（别忘了使文件生效）：

vim /etc/profile

添加

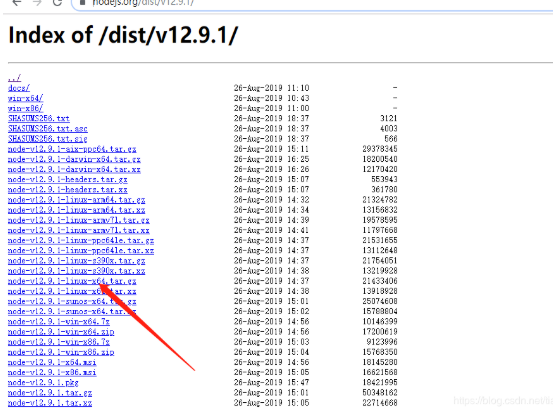
#go环境变量

export PATH=$PATH:/usr/local/go/bin  
export GOPATH=/opt/gopath

esc后:wq，保存

source /etc/profile（使文件生效）

1. 查看go版本：go version
2. 安装node.js和npm
3. 下载nodejs最新版：<https://nodejs.org/dist/v12.9.1/>（已下载好）如图：



1. mkdir /tmp/docker/nodejs
2. cd /tmp/docker/nodejs将node-v12.9.1-linux-x64.tar.gz上传到/tmp/docker/nodejs目录并解压。

解压缩命令：tar -xzvf node-v12.9.1-linux-x64.tar.gz

1. 配置环境变量：vim /etc/profile，添加以下内容

# nodejs environment  
export NODE\_HOME=/tmp/docker/nodejs/node-v12.9.1-linux-x64  
export PATH=$PATH:${NODE\_HOME}/bin

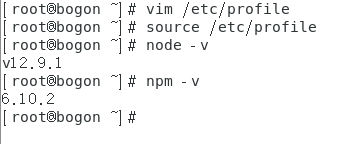
Esc后:wq，保存退出

source /etc/profile

1. 查看版本（如图）

node –v

npm –v



1. 升级gcc
2. 查看gcc版本：gcc –v



1. 执行以下命令进行升级

yum install centos-release-scl

yum install devtoolset-7-gcc\*

scl enable devtoolset-7 bash

which gcc

gcc –v



1. 各版本信息如下

go version

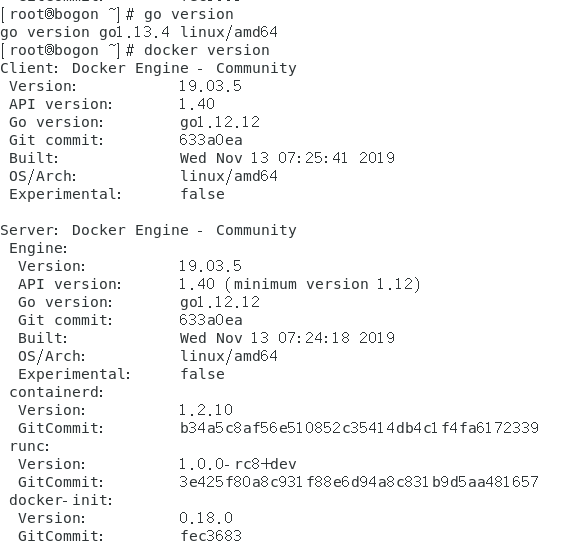
docker version

docker-compose version

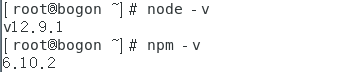
gcc –v

node –v

npm –v







1. 下载fabric源码：

mkdir -p /usr/local/go/src/github.com/hyperleder

cd /usr/local/go/src/github.com/hyperleder

git clone <https://github.com/hyperledger/fabric.git>（已下载好，可直接移动到文件夹中然后执行 [tar –xvf fabric.tar]进行解压）

cd fabric

git branch –a

git checkout release-1.4

1. 下载镜像

# docker pull hyperledger/fabric-peer:latest

# docker pull hyperledger/fabric-orderer:latest

# docker pull hyperledger/fabric-tools:latest

# docker pull hyperledger/fabric-ccenv:latest

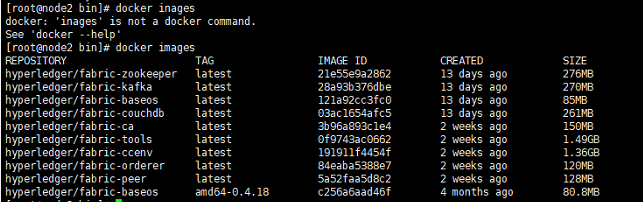
# docker pull hyperledger/fabric-baseos:latest

# docker pull hyperledger/fabric-kafka:latest

# docker pull hyperledger/fabric-zookeeper:latest

# docker pull hyperledger/fabric-couchdb:latest

下载时间很长，完毕如图：



1. 运行first-network

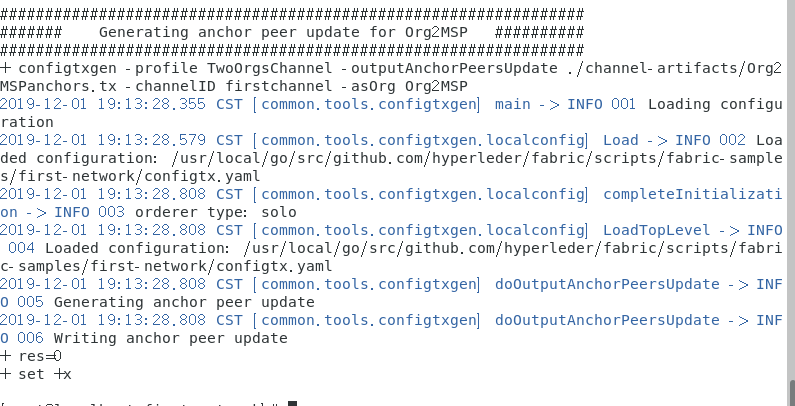
#cd fabric/

编译源码：

#make release

#cd fabric-samples/first-network

创建第一个channel（firstchannel为通道名称，可以自己定义）：# ./byfn.sh -m generate -c firstchannel



启动网络：./byfn.sh -m up -c firstchannel

成功结果如图：



1. 测试fabric是否成功：

另启动一终端；

输入docker exec –it cli bash进入cli客户端；

输入peer chaincode query -C firstchannel -n mycc -c '{"Args":["query","a"]}'命令查看用户a的余额为90，如图，表示搭建成功。

