

计算机学院

**科学数据版权溯源原型系统**

**（安装指南）**

**北京理工大学**

**2019年12月**

目录

[1.Hyperledger Fabric 安装 3](#_Toc28637482)

[1.运行环境安装 3](#_Toc28637483)

[1.操作系统 3](#_Toc28637484)

[2.Go语言开发环境 3](#_Toc28637485)

[3.Curl工具 4](#_Toc28637486)

[4.Docker 4](#_Toc28637487)

[5.Docker-compose 5](#_Toc28637488)

[2.Fabric平台安装 6](#_Toc28637489)

[2 IPFS安装 8](#_Toc28637490)

[1.安装go-ipfs 8](#_Toc28637491)

[2.安装go-ipfs-api 11](#_Toc28637492)

[3 Beego框架安装 12](#_Toc28637493)

[4 Mysql安装 14](#_Toc28637494)

[5. 联盟链环境部署 15](#_Toc28637495)

[6 智能合约部署至peer节点 16](#_Toc28637496)

[7. 创建Mysql数据库 16](#_Toc28637497)

[8 启动IPFS文件系统的服务 17](#_Toc28637498)

[9 启动Web服务 18](#_Toc28637499)

[10 stop暂停服务 19](#_Toc28637500)

[11 restart重新启动服务 21](#_Toc28637501)

[12 停止服务 21](#_Toc28637502)

# 1.Hyperledger Fabric 安装

## 1.运行环境安装

### 1.操作系统

操作系统要求为 Ubuntu 16.04 Desktop

### 2.Go语言开发环境

Go语言版本要求不低于1.13版本。

1）下载地址 <https://studygolang.com/dl>

选择对应的Linux版本



下载后将压缩文件复制到Ubuntu的/usr/local目录下，执行解压命令：

|  |
| --- |
| sudo tar -zxvf go1.13.5.linux-amd64.tar.gz |

得到解压后的文件夹；

2）配置环境变量

打开控制台，输入

|  |
| --- |
| gedit ~/.bashrc |

编辑 ~/.bashrc文件，在文件末尾添加以下代码

|  |
| --- |
| export GOROOT=/usr/local/go  export GOPATH=/home/xxx/goworkspace #将xxx改为你的系统用户名  export GO111MODULE=auto  export GOPROXY=https://goproxy.io  export PATH=$PATH:$GOROOT/bin:$GOPATH/bin |

完成后关闭并保存，在命令行输入

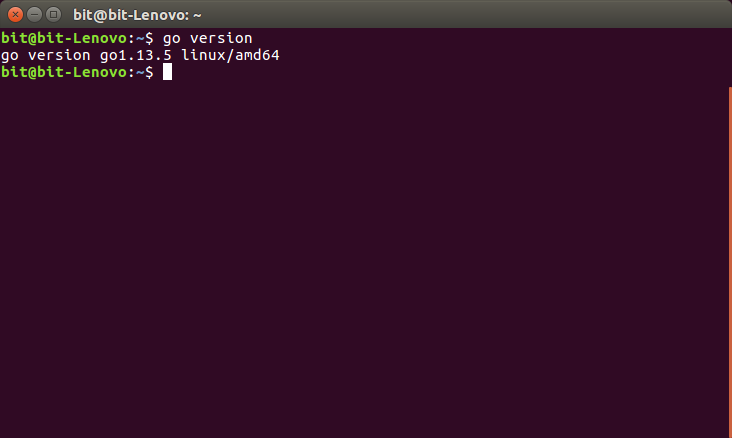
|  |
| --- |
| source ~/.bashrc |

使配置文件生效。

3）检查是否安装成功

在根目录下输入

|  |
| --- |
| go version |



出现以上字样 表示安装成功。

### 3.Curl工具

|  |
| --- |
| sudo apt install -y curl |

### 4.Docker

1）安装

**Docker版本要求不低于17.06.2**

首先安装相关包

|  |
| --- |
| $ sudo apt-get update  $ sudo apt-get install \  apt-transport-https \  ca-certificates \  curl \  software-properties-common |

添加软件仓库（使用阿里云仓库）

|  |
| --- |
| $ curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -  $ sudo add-apt-repository \  "deb [arch=amd64] https://mirrors.aliyun.com/docker-ce/linux/ubuntu \  $(lsb\_release -cs) \  stable" |

安装docker

|  |
| --- |
| $ sudo apt-get update  $ sudo apt-get install docker-ce # 安装最新版的docker |

添加访问权限

|  |
| --- |
| $ sudo gpasswd -a ${USER} docker |

重启docker

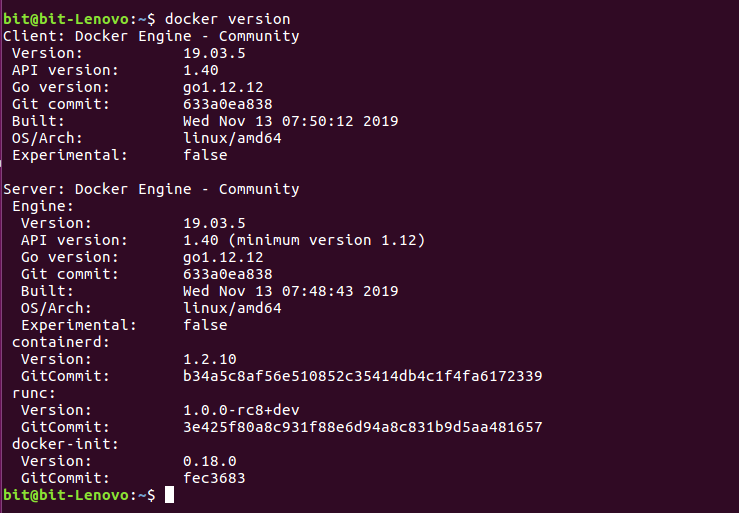
|  |
| --- |
| sudo service docker restart |

切换当前会话到新 group 或者重启 X 会话

|  |
| --- |
| newgrp – docker |

2）检查是否安装成功

|  |
| --- |
| docker version |



出现以上字样 表示安装成功。

### 5.Docker-compose

1）安装

**Docker-compose版本要求不低于 1.14.0**

下载 docker-compose-Linux-x86\_64，地址：

https://github.com/docker/compose/releases

把 docker-compose-Linux-x86\_64 放到 /usr/local/bin 下，并改名为 docker-compose

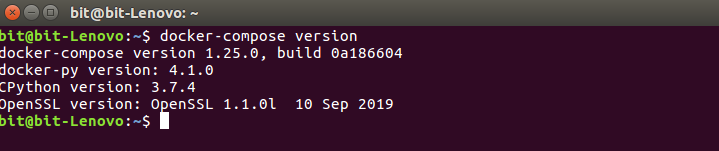
添加执行权限

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

2)检查是否安装成功

在根目录下输入

|  |
| --- |
| docker-compose version |



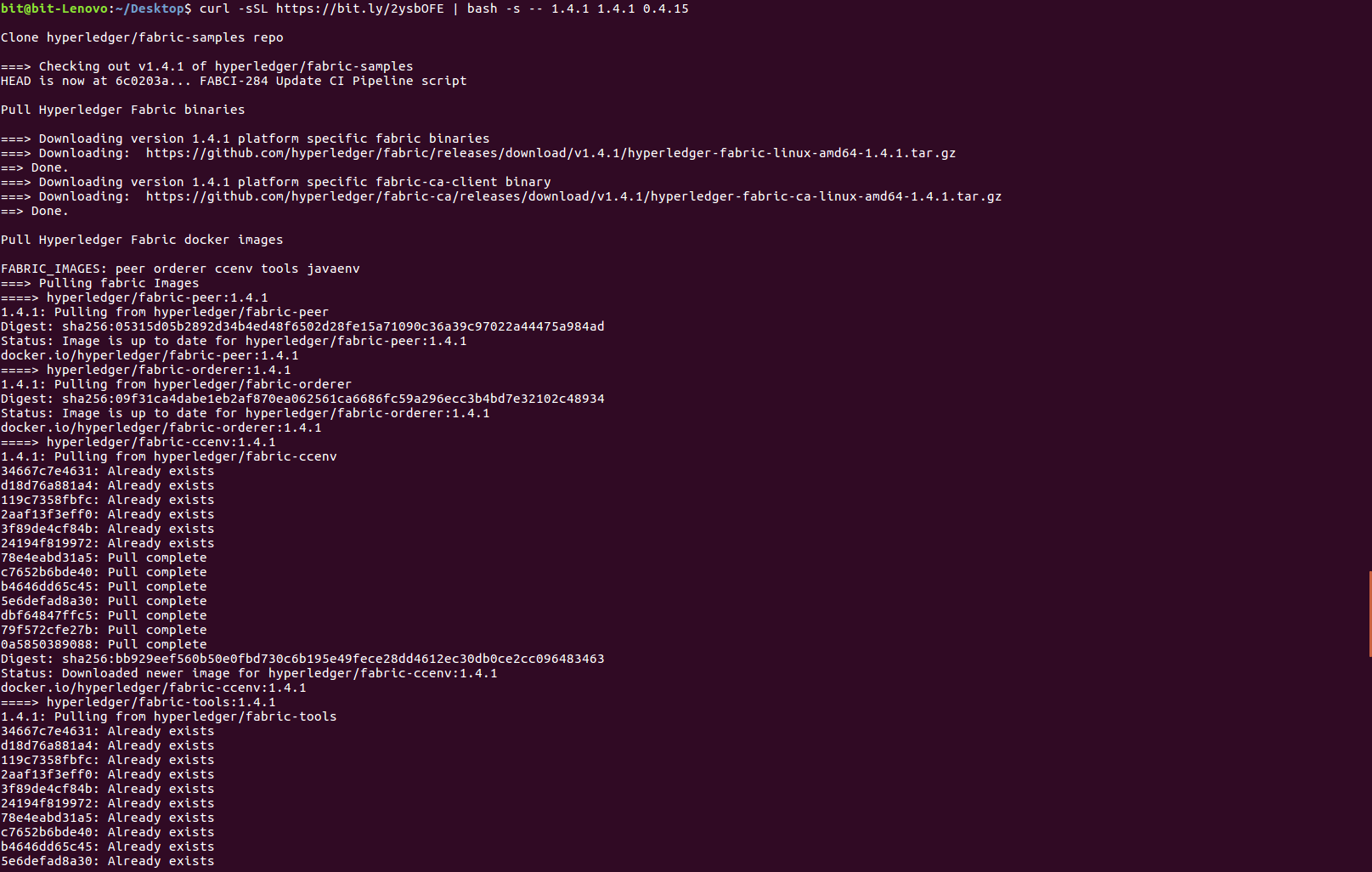
出现以上字样，表示安装成功。

### 6.Fabric平台安装

1）下载和安装

在/home/xxx/goworkspace （将xxx改为你的系统用户名）目录下新建src文件夹，在src目录下打开控制台，输入以下命令进行fabric的下载和安装

|  |
| --- |
| curl -sSL https://bit.ly/2ysbOFE | bash -s -- 1.4.1 1.4.1 0.4.15 |



等待安装完成后，src目录下会出现 fabric-samples文件夹。

|  |
| --- |
| **注**：由于某些网络原因，该脚本运行可能产生如下错误    如产生该错误，重复执行以上命令即可，直至正常运行。 |

2）配置环境变量

在控制台输入

|  |
| --- |
| gedit ~/.bashrc |

在文件末尾输入

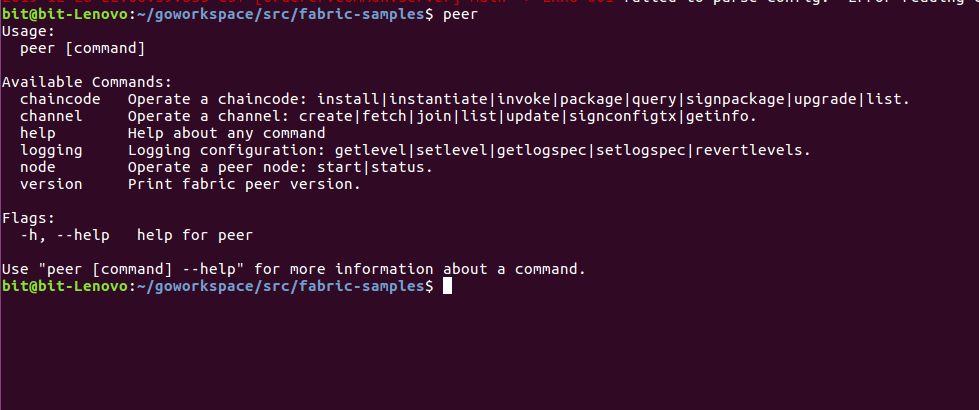
|  |
| --- |
| #fabricPath  export PATH=/home/xxx/goworkspace/src/fabric-samples/bin:$PATH  #将xxx改为你的系统用户名 |

完成后保存文件，并在控制台输入

|  |
| --- |
| source ~/.bashrc |

应用对环境变量的修改

在根目录下输入peer



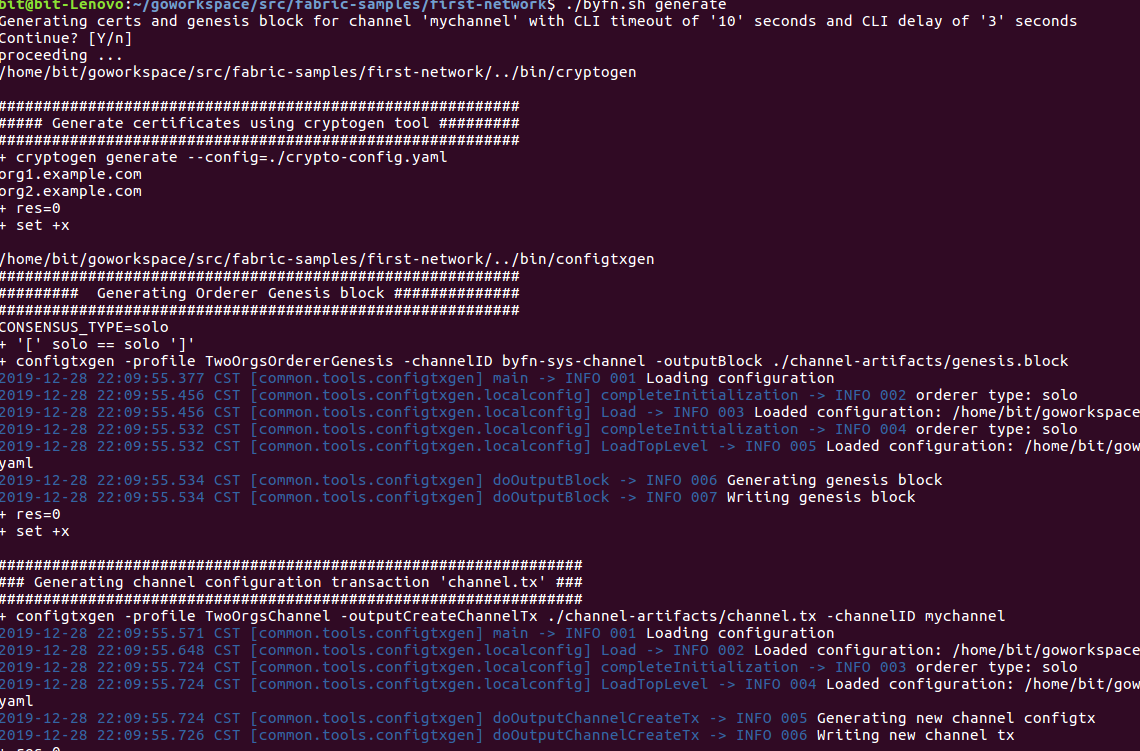
出现以上字样，表示环境变量设置成功。

3）检查是否安装成功

进入fabric-samples/first-network目录，

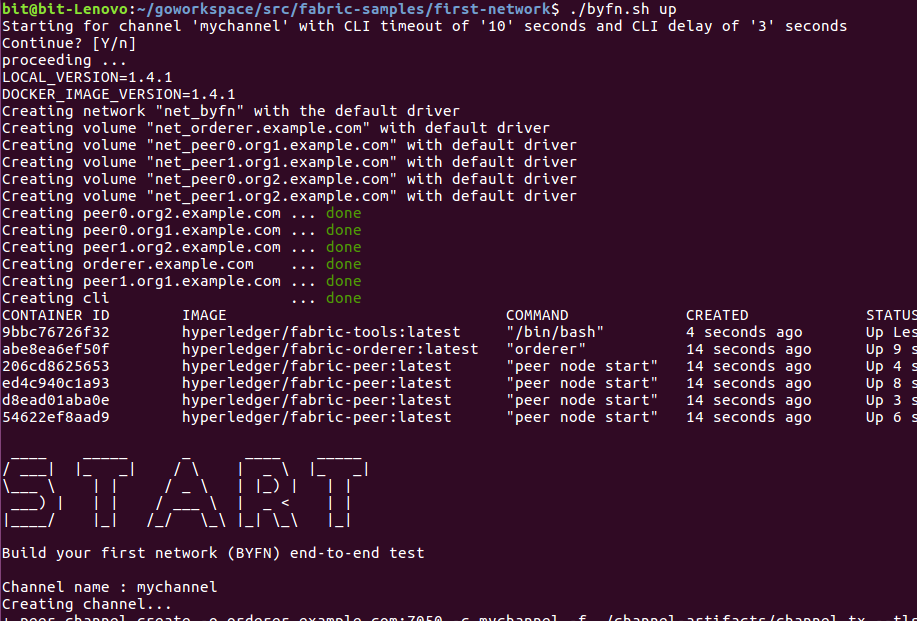
在控制台输入

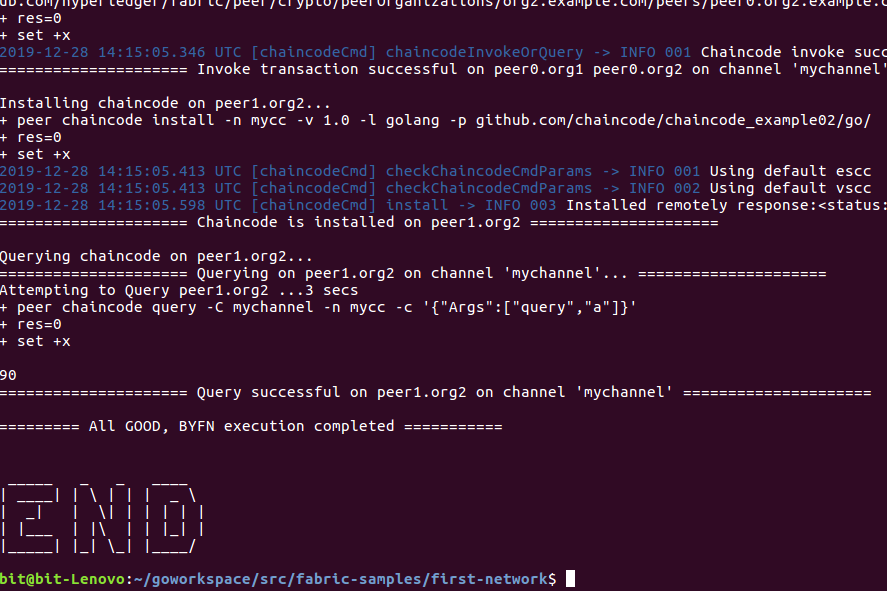
|  |
| --- |
| ./byfn.sh generate |



出现该字样后再输入

|  |
| --- |
| ./byfn.sh up |





最终出现以上字样后，表示fabric安装成功。

输入

|  |
| --- |
| ./byfn.sh down |

关闭服务并清理环境。（重要）

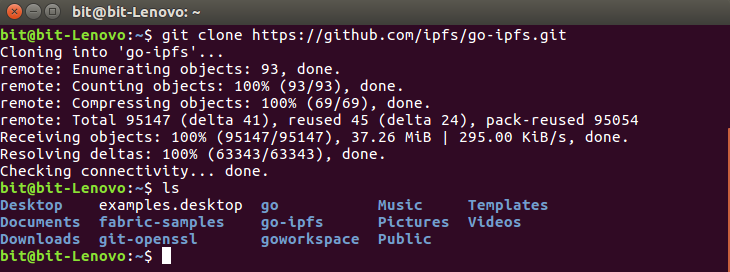
## 2 IPFS安装

### 1.安装go-ipfs

在家目录下输入

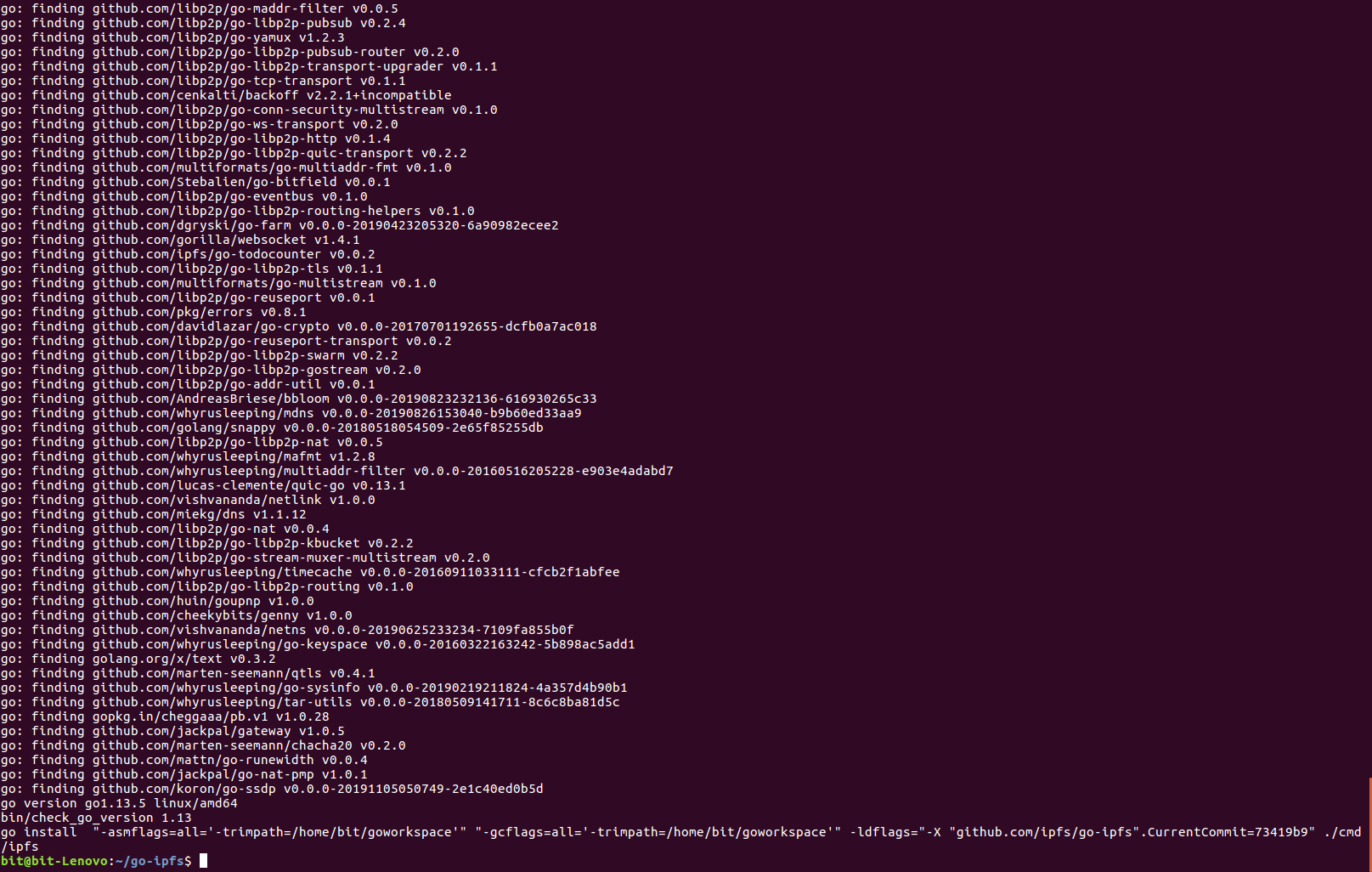
|  |
| --- |
| git clone https://github.com/ipfs/go-ipfs.git |

下载go-ipfs



进入go-ipfs目录，输入

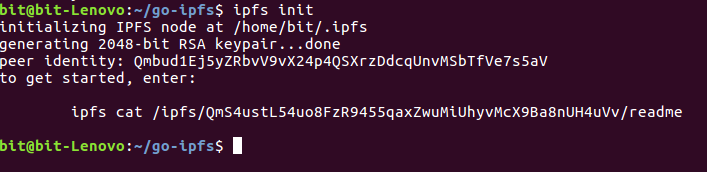
|  |
| --- |
| cd go-ipfs  make install |



tar -zxvf kubo\_v0.18.1\_linuv-amd64.tar.gz -C ~/

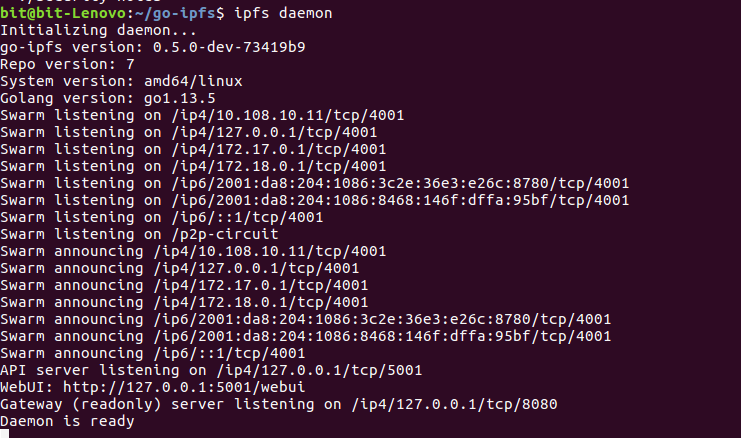
完成后输入

|  |
| --- |
| ipfs init |

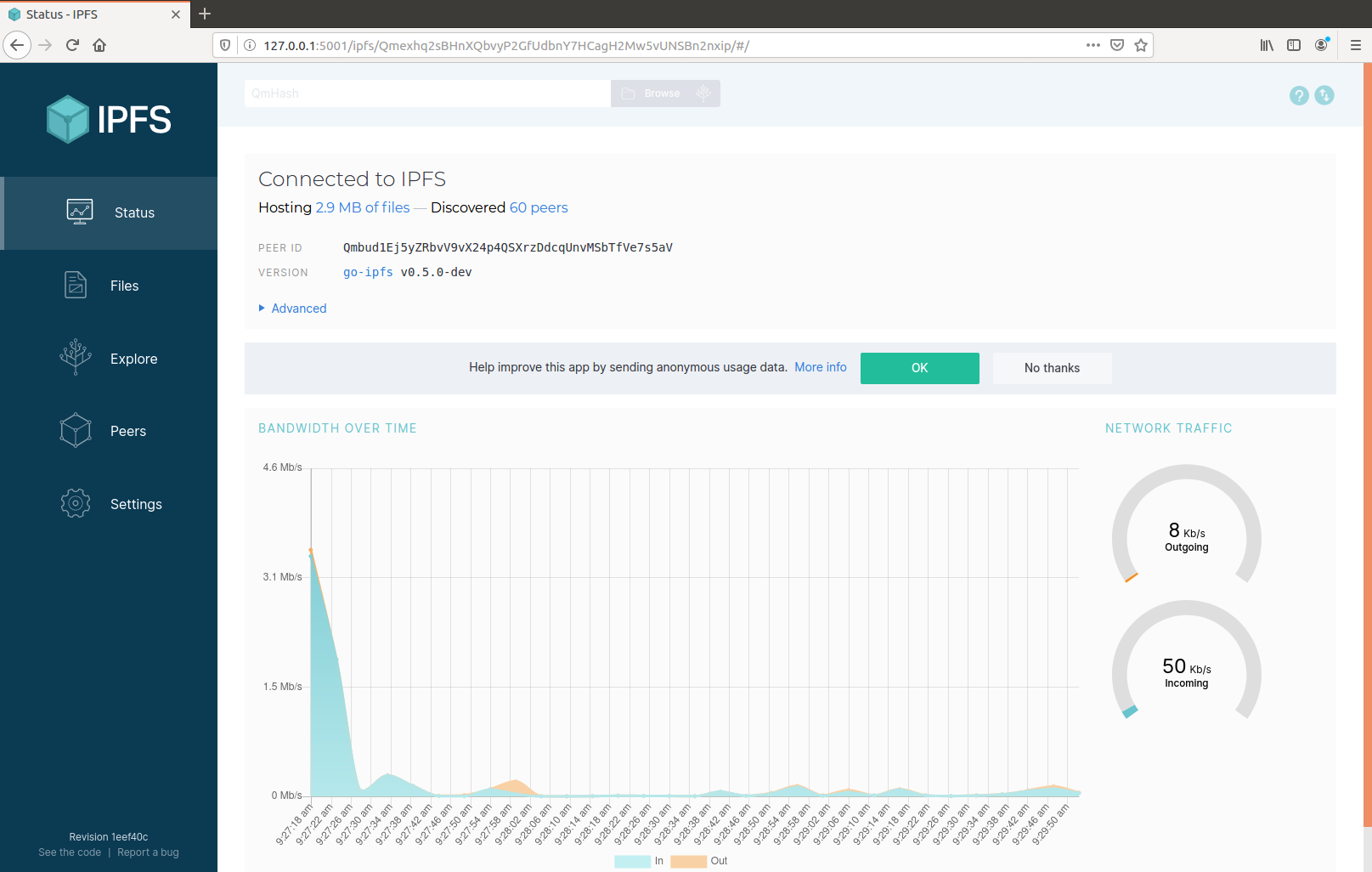


完成后再输入

|  |
| --- |
| ipfs daemon |



完成后打开浏览器，在地址栏输入127.0.0.1：5001/webui



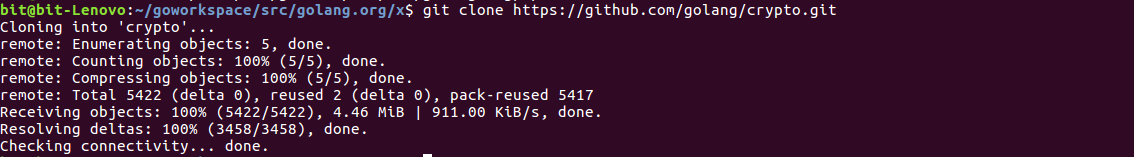
出现以上界面，表明go-ipfs安装成功

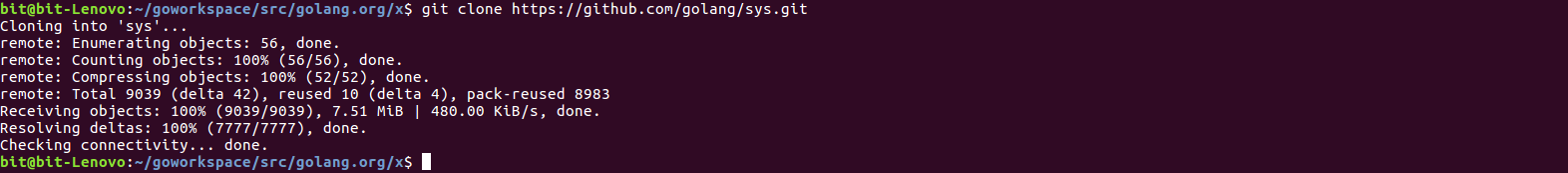
### 2.安装go-ipfs-api

1)安装

打开控制台，依次输入

|  |
| --- |
| mkdir -p $GOPATH/src/golang.org/x  cd $GOPATH/src/golang.org/x  git clone <https://github.com/golang/crypto.git>  git clone https://github.com/golang/sys.git |

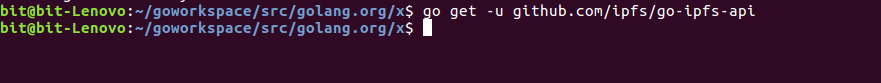




出现以上界面后，再输入

|  |
| --- |
| go get -u github.com/ipfs/go-ipfs-api |

等待正常完成后，如未报错，显示以下界面



2）检查是否安装成功

新建一个test.go文件

输入以下代码

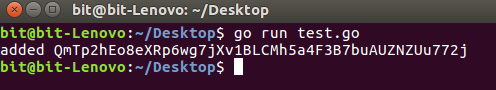
|  |
| --- |
| package main  import (  "fmt"  "strings"  "os"  shell "github.com/ipfs/go-ipfs-api"  )  func main() {  sh := shell.NewShell("localhost:5001")  cid, err := sh.Add(strings.NewReader("hello world!"))  if err != nil {  fmt.Fprintf(os.Stderr, "error: %s", err)  os.Exit(1)  }  fmt.Printf("added %s\n", cid)  } |

保存后，打开控制台，输入

|  |
| --- |
| ipfs init  ipfs daemon |

开启ipfs服务，完成后再打开一个新的控制台，进入test.go所在路径，输入

|  |
| --- |
| go run test.go |



出现以上界面，表示安装成功。



## 3 Beego框架安装

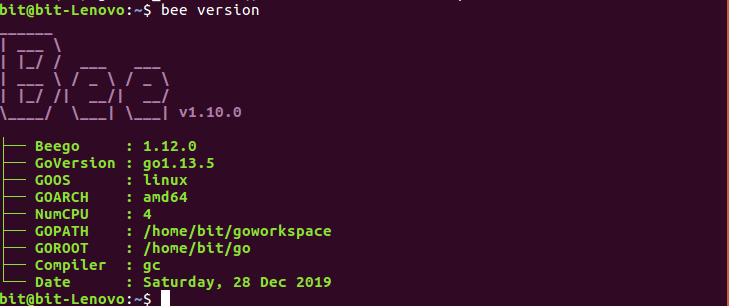
1）安装

在控制台输入以下两条命令

|  |
| --- |
| ~~go get github.com/astaxie/beego~~  ~~go get github.com/beego/bee~~ |
| go get github.com/beego/beego/v2  go get github.com/beego/bee/v2 |

安装完成后，在控制台输入

|  |
| --- |
| bee version |

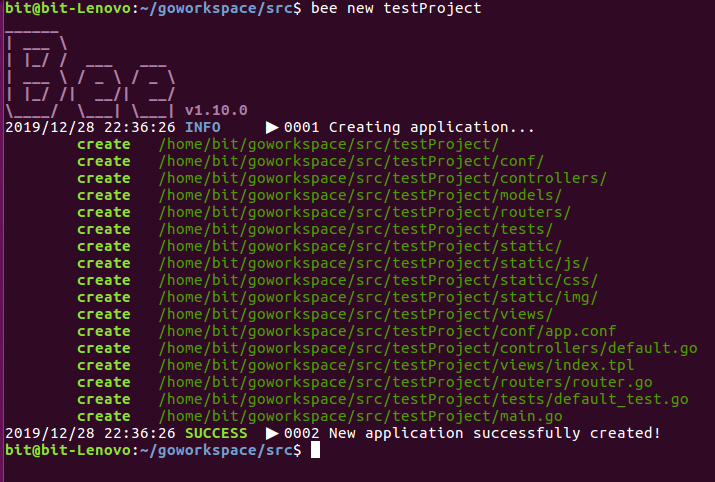


如果出现以上字样，表明安装成功。

2）检查是否安装成功

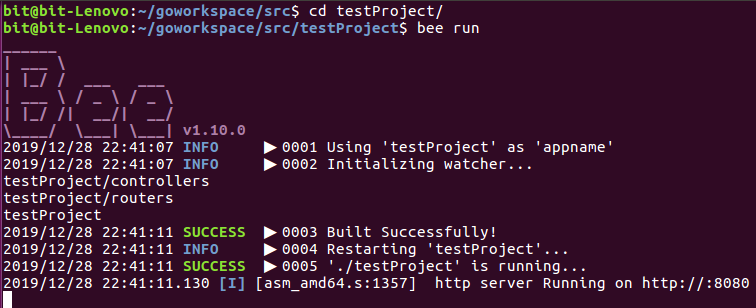
在 /home/xxx/goworkspace/src（将xxx替换为你的系统用户名） 目录下，在控制台输入te

|  |
| --- |
| bee new testProject |

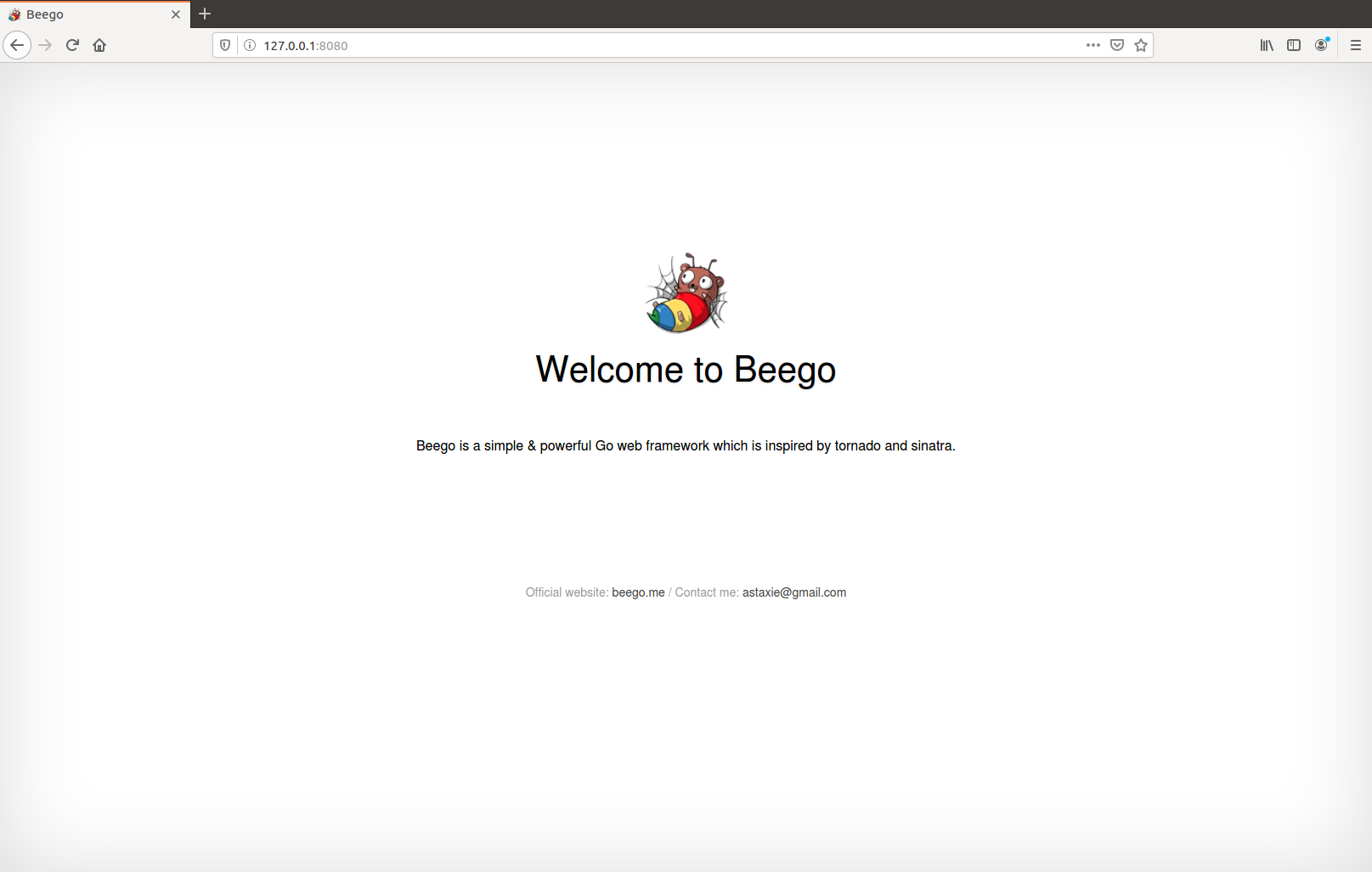


进入testProject目录之后输入

|  |
| --- |
| bee run |



打开浏览器访问127.0.0.1:8080



出现以上界面，表示beego安装成功。

## 4 Mysql安装

完全卸载

sudo rm /var/lib/mysql/ -R

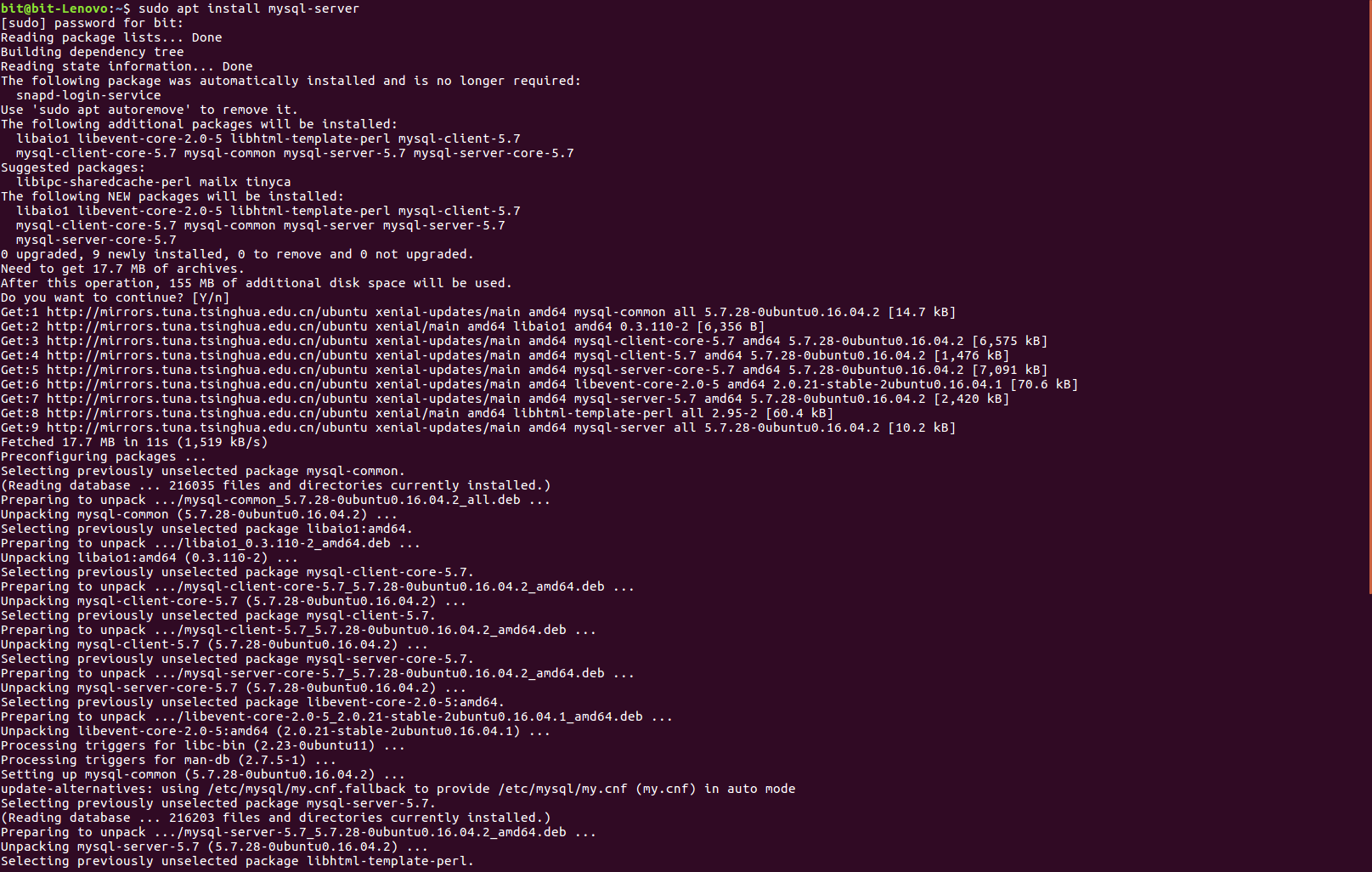
sudo rm /etc/mysql/ -R

sudo apt-get autoremove mysql\* --purge

sudo apt-get remove apparmor

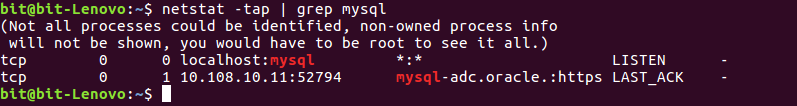
打开控制台，输入以下命令（安装过程中root用户密码设置为123456）

|  |
| --- |
| sudo apt install mysql-server |



安装完成后，在控制台输入

|  |
| --- |
| netstat -tap | grep mysql |



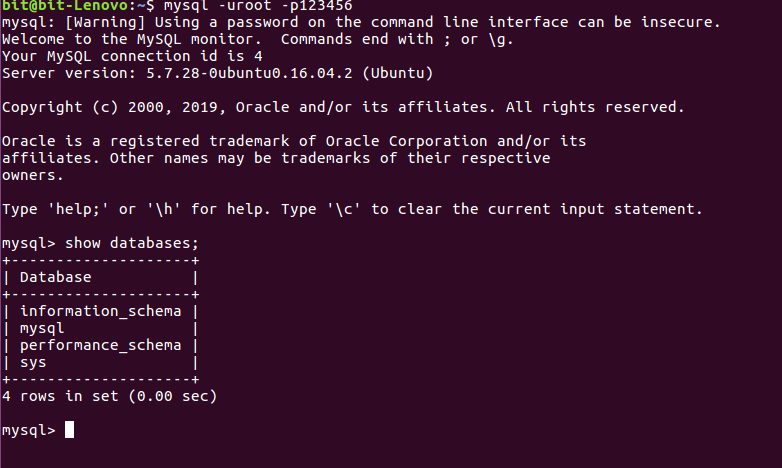
显示mysql的端口处于监听状态表示安装成功。

输入

|  |
| --- |
| sudo mysql -u用户名 -p密码 |

进入mysql，输入

|  |
| --- |
| show databases; |



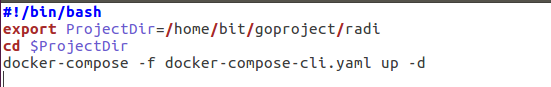
出现以上界面，表示mysql安装成功。

## 5. 联盟链环境部署

将radi文件夹放置在非$GOPATH的路径下（路径中不要出现中文）,使用pwd命令查看自己电脑上radi文件夹的绝对路径如下图。



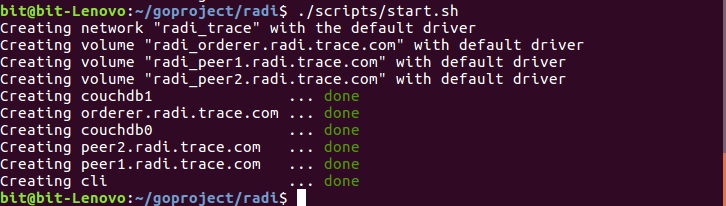
radi/scripts文件夹下有一些.sh脚本,需要对其进行修改。将start.sh和radiInit.sh脚本的第二行export ProjectDir=/home/bit/goproject/radi替换为自己电脑上的radi文件夹的绝对路径如下图；对radiProvenance文件夹下的scripts/webInit.sh脚本进行同样的路径替换。



返回radi目录下，输入

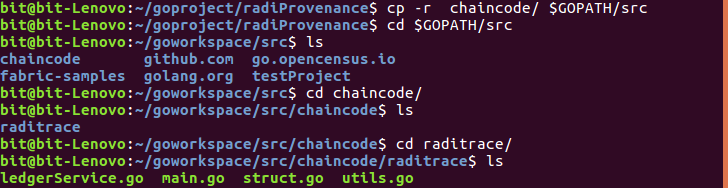
|  |
| --- |
| ./scripts/start.sh |

运行start.sh脚本，搭建联盟链测试环境（一组织、两peer节点、节点启用couchDB作为状态账本、一orderer节点），成功截图如下



## 6 智能合约部署至peer节点

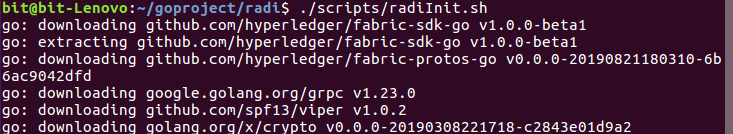
从radiProvenance文件夹下将chaincode文件夹拷贝至$GOPATH/src目录下



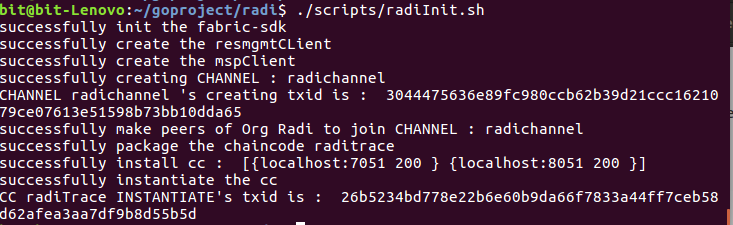
返回radi文件夹下，在radi文件夹下输入

|  |
| --- |
| ./scripts/radiInit.sh |

运行radiInit.sh脚本，如下图所示（此处首次运行需要连网，运行脚本时会下载一些依赖库）。



智能合约部署成功效果如图所示：

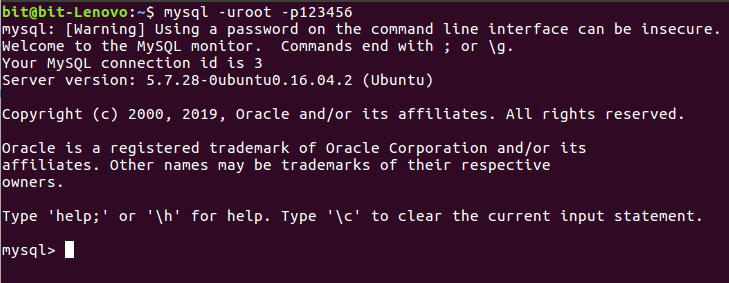


## 7. 创建Mysql数据库

在任意位置打开一个shell，输入

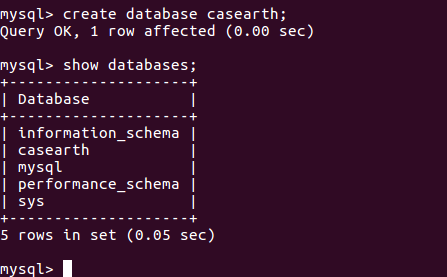
|  |
| --- |
| mysql -uroot -p123456 |

进入mysql数据库，如下图所示。

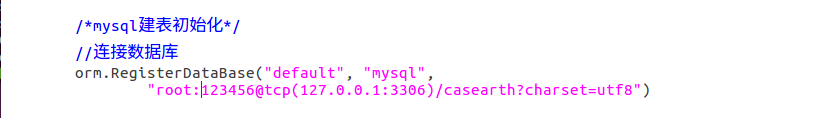


创建数据库casearth，如下图所示:

create database casearth;



|  |
| --- |
| [注]：这里需要mysql数据库用户名为root，密码为123456，创建数据库名为casearth。因为Web服务会连接此数据库，若自己设置的用户名密码或数据库名和如上所述不一致，则需要去radiProvenance/models/model.go文件夹中修改对应位置的代码使用自己的用户名、密码和数据库名，需要修改的地方如图所示 |
|  |

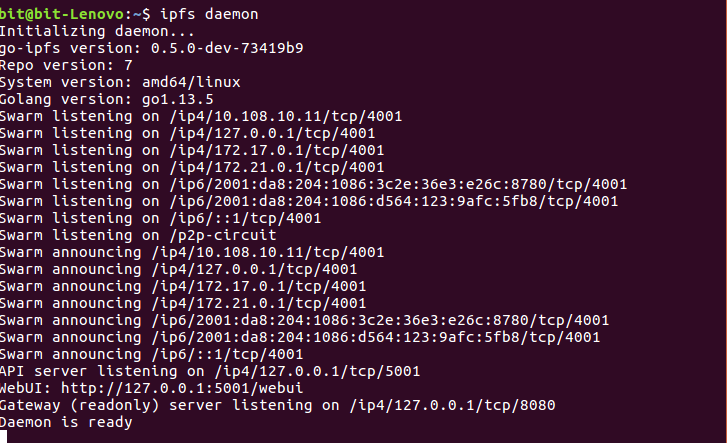
 

## 8 启动IPFS文件系统的服务

打开任意的终端，输入

|  |
| --- |
| ipfs init  ipfs daemon |

启动ipfs文件系统后如图所示（命令可能需要加sudo）

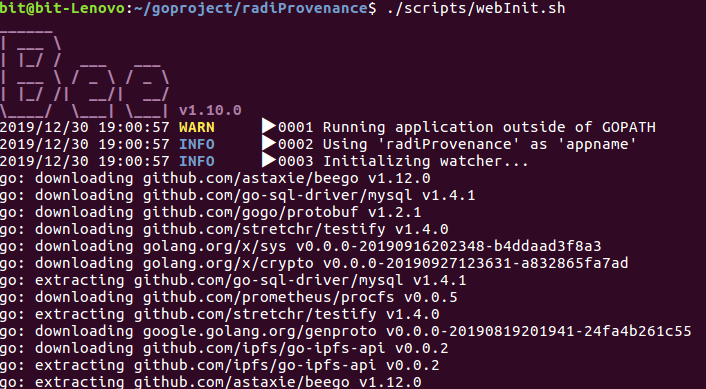


## 9 启动Web服务

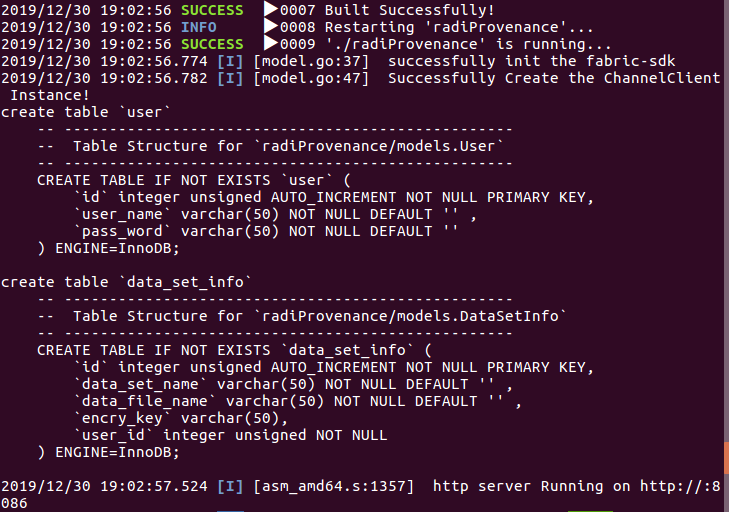
Shell cd进入radiProvenance文件夹，在此文件夹路径下输入

|  |
| --- |
| ./scripts/webInit.sh |

启动web服务（此处首次运行需要连网，下载一部分依赖库，如下图所示）

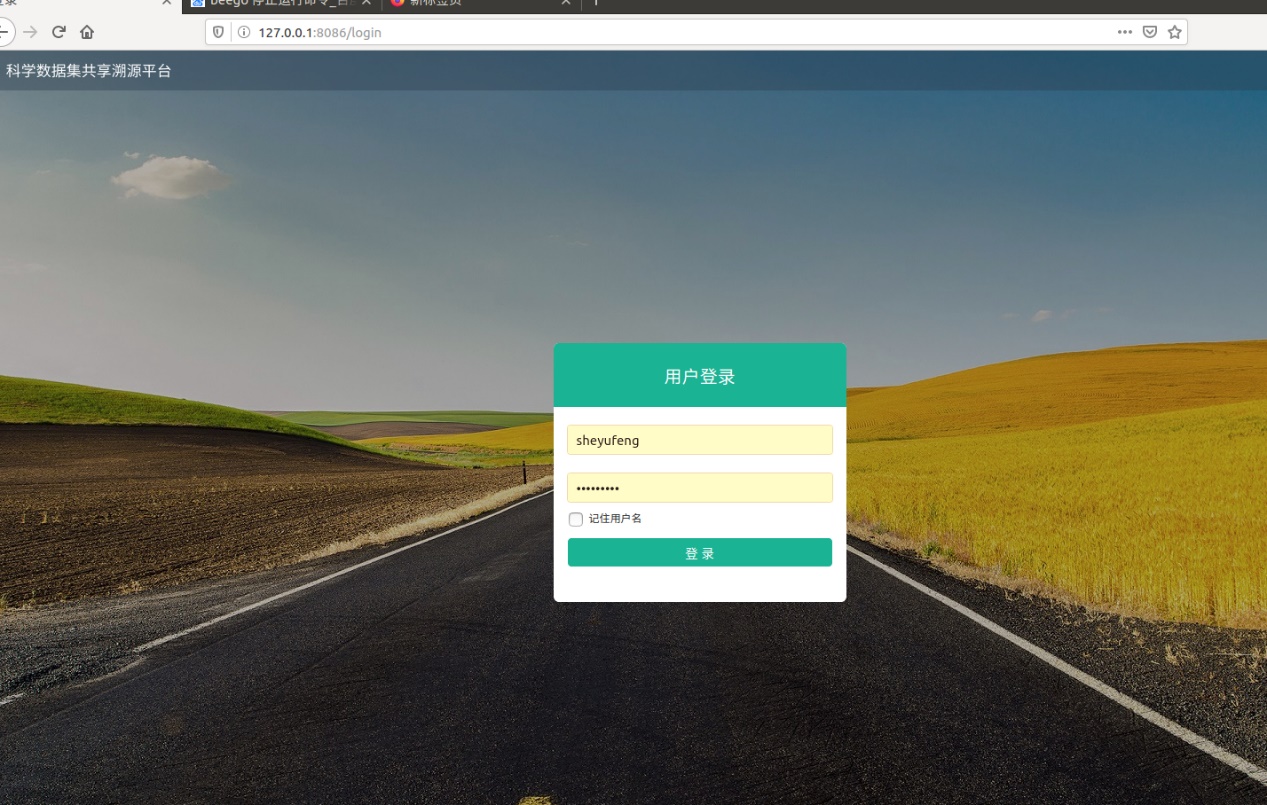


Web服务成功启动效果如下图所示。



此时，Web服务运行在本机的8086端口,在浏览器访问127.0.0.1:8086/register

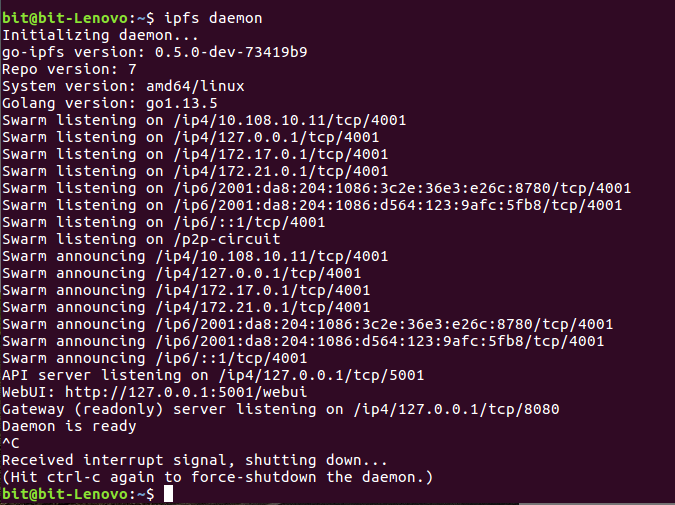
进入用户注册界面如下图所示，此时系统部署成功！（此时shell后台可以看web应用的日志信息）



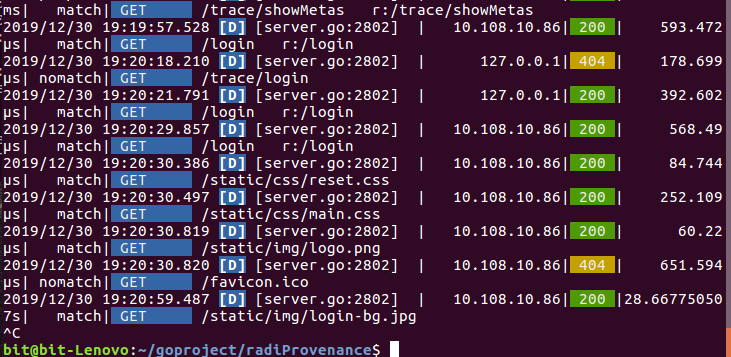
## 10 stop暂停服务

不清除IPFS文件系统、联盟链账本、mysql的数据，仅暂时停止服务。

a. ctrl+c关掉ipfs文件系统的运行进程如下图所示。



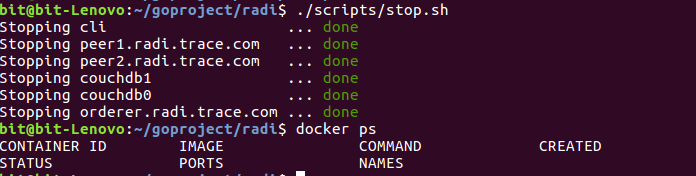
b. ctrl+c关掉Web服务进程如下图所示。



c. 暂停docker容器的服务。Shell cd 进入radi文件夹下，输入

|  |
| --- |
| ./scripts/stop.sh |

运行stop.sh脚本，停止docker容器的服务，如下图所示。



此时暂停服务成功

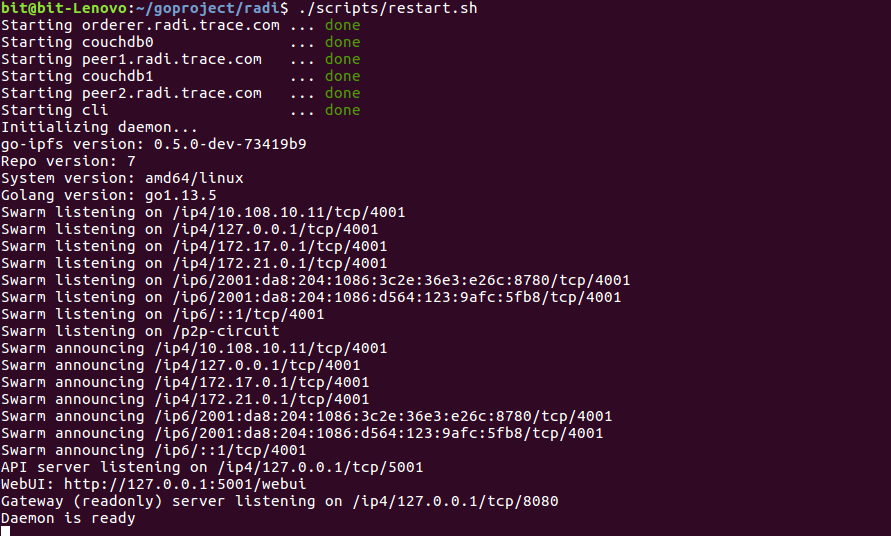
## 11 restart重新启动服务

在运行stop.sh暂停服务后，restart重新启动系统。

a. shell cd进入radi文件夹下，输入

|  |
| --- |
| ./scripts/restart.sh |

重新启动ipfs和docker容器的服务如下图所示。



b. 重新启动web服务，shell cd进入radiProvenance文件夹，运行./scripts/webInit.sh，如下图所示。

此时系统重新启动



## 12 停止服务

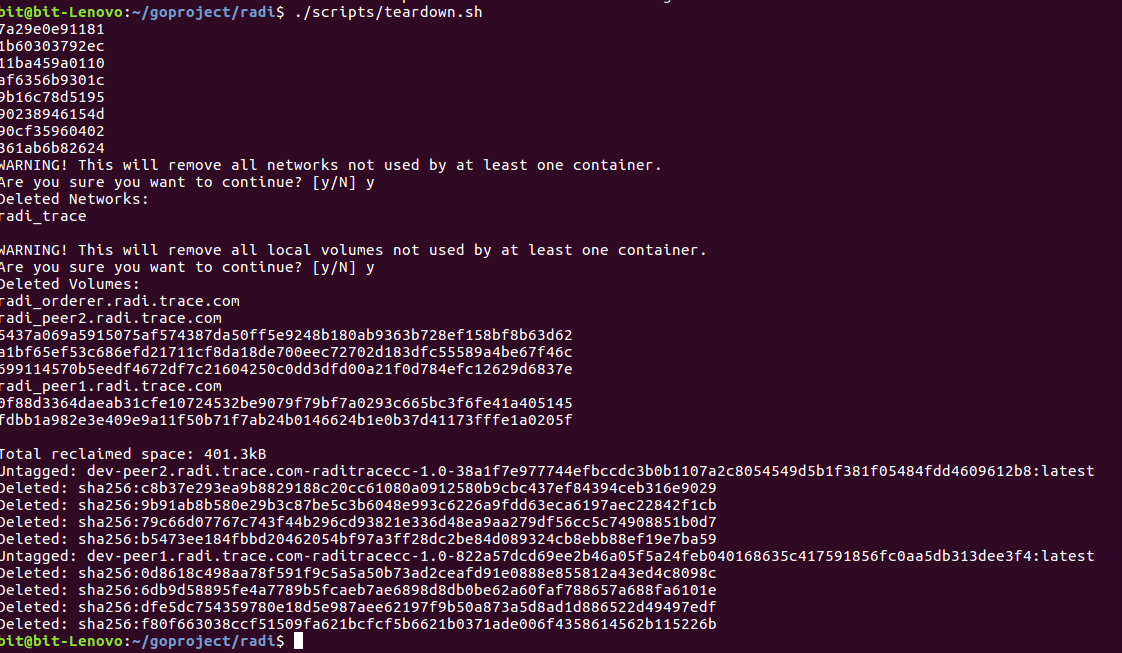
停止服务，并删除所有数据

a. 同stop暂停服务一样，ctrl+c停止web服务和ipfs服务。

b. 删除docker容器及其挂载卷，以及ipfs文件系统中的文件。Shell cd进入radi文件夹，输入

|  |
| --- |
| ./scripts/teardown.sh |

删除成功如下图所示。



c. 删除mysql数据表

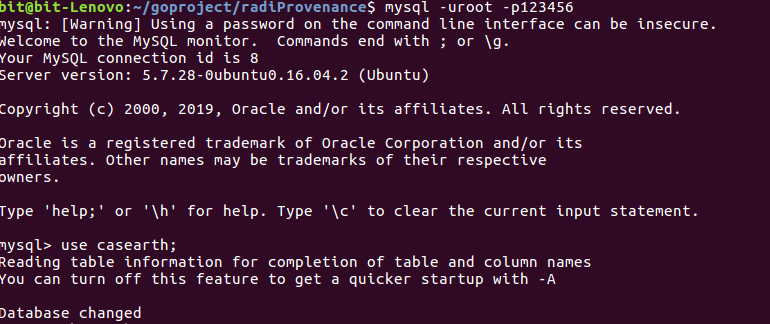
输入

|  |
| --- |
| mysql -uroot -p123456 |

进入mysql数据库交互界面，输入

|  |
| --- |
| use casearth; |

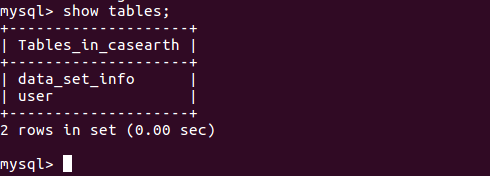
如下图所示。



输入

|  |
| --- |
| show tables; |

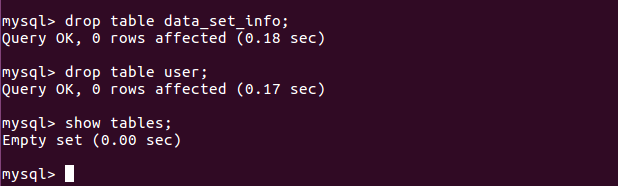
查看数据库casearth下的数据表。

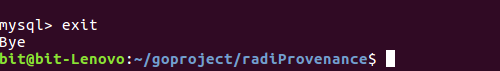


输入

|  |
| --- |
| drop table data\_set\_info; |
| drop table user; |

删除两张数据表，此时数据库casearth中没有数据表,如下图所示。

输入exit，退出mysql数据库，如下图所示。



此时，完全删除之前的IPFS、联盟链账本、mysql中的数据，在teardown之后，再次开启服务请参考5-9的操作（已经创建和修改过的文件和数据库无需再次修改和创建）。