**《课程综合实践II》实验报告五**

**Geth多节点部署及基于Ganache和Truffle的智能合约实践**

**姓名\_\_\_ 沈云昊 专业 电子信息工程**

**学号 3220104126 联系方式 2766636730@qq.com**

## 一、实验目的

1. 在上次实验的基础上，完成基于Geth的多节点私有链部署。

2. 在Ubuntu系统上安装必需的nodeJS和npm环境。

3. 安装Ganache和Truffle，并在此基础上开发简单的智能合约。

## 二、实验内容及要求

**准备工作：为了方便后续的学习和实践，建议同学们自带电脑，在自己电脑上进行实验内容，并完成实验报告。**

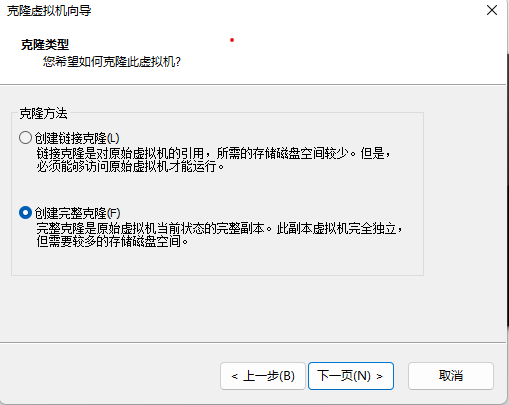
**实验步骤（仔细阅读，按照步骤完成实验）**

**（本次实验全程在ubuntu上进行操作）**

### 基于Geth的多节点部署(请截图保存在实验记录中)

上次实验我们成功的在一台主机上搭建了Geth私有链，并在该节点上新建了多个账户进行挖矿和转账交易操作，本次我们将搭建基于两个虚拟机的双节点私有链，具体配置如下：

1. 为保证环境的一致性，首先将之前已经配置好环境的虚拟机克隆一份，在后续的步骤中选择下一页—虚拟机中的当前状态—创建完整克隆—完成



1. 开启两个虚拟机，将genesis.json文件中的chainID修改为666或者其他数字，修改完成后分别查看两台虚拟机的IP地址并记录，在两台虚拟机的桌面上分别建立工作目录multiNodes，命令如下：

mkdir ~/Desktop/multiNodes

1. 在两台虚拟机上分别进行私有链的初始化。

cd ~/Desktop

geth --datadir ./multiNodes init genesis.json

1. 在两台虚拟机上分别启动节点，命令如下：

geth --port 8100 --networkid 666 --datadir ./multiNodes --maxpeers=3 --http --http.port 8545 --http.addr 虚拟机IP --http.corsdomain "\*" --http.api="eth,web3,personal,admin,txpool" console --allow-insecure-unlock --nodiscover

在第二台虚拟机上启动节点时，**需要改变命令行参数--port和--http.port的值，并填写第二台虚拟机的IP**

1. 启动完成后，我们在两个节点上分别添加一个账户。

personal.newAccount("123456")

1. 在第二台虚拟机上获取enode信息，并在另一台虚拟机上执行添加节点命令。

获取enode信息：

admin.nodeInfo.enode

将获取到的enode信息改造为如下形式，其中虚拟机IP地址和port参数均为第二台主机的：

"enode://cb740d086effb5313fb87ea71f27adcd1e57fd60d73938b0bd068bdebb8a4de27a96b826df0773d00424096cbda1c3f7731dabd80a7fa165d12e0e053b30af58@虚拟机IP地址:启动时命令行参数port的值?discport=0"

在第一台主机添加节点：

admin.addPeer("enode://cb740d086effb5313fb87ea71f27adcd1e57fd60d73938b0bd068bdebb8a4de27a96b826df0773d00424096cbda1c3f7731dabd80a7fa165d12e0e053b30af58@虚拟机IP地址:启动时命令行参数port的值?discport=0")

验证是否添加成功

admin.peers

成功应该会显示添加的节点信息。

1. 选择任一节点开始挖矿并隔一段时间停止，输入命令

miner.start()

miner.stop()

此时观察另一节点命令行内的输出（如果出现连接断开是正常现象，等一段时间后，按上述步骤进行重连即可），等节点同步后，输入命令查看两个节点的区块高度，

eth.blockNumber

如果一致说明同步成功。

### Ganache客户端安装及操作

Ganache是一个可以快速启动个人以太坊区块链，并可以使用它来运行测试，执行命令、检查状态，同时控制链条的运行方式的工具，由于其基于node环境运行，所以我们先需要安装相应的环境。

### 2.1 Node和nvm安装

a. 更新软件源列表

sudo apt update

b. 安装npm包管理工具

sudo apt install npm

c. 查看node和npm版本

npm version

node --version

**如果npm版本低于8.1.3、node版本低于16.13.0，请继续下面步骤**

d. 清除npm缓存，安装node.js版本管理工具，用n将node.js升级至最新稳定版

npm cache clean --force

sudo npm install -g n --force

sudo n stable

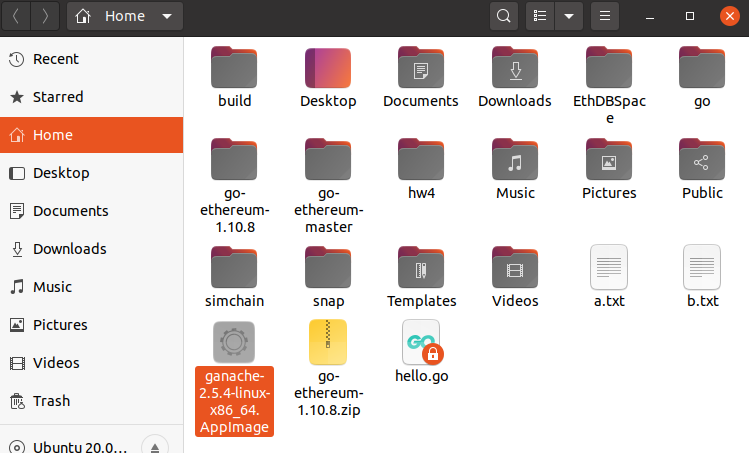
e. 关闭终端后，重新打开终端，将npm更新到8.1.3后再次查看npm和node的版本

sudo apt install npm

sudo npm install -g [npm@8.1.3](mailto:npm@8.1.3)

### 2.2 安装运行Ganache

将学在浙大上的Ganache安装包解压后拖入虚拟机的工作目录内，



双击打开即可，打开后选择quickStart即可生成拥有10个账户的私有链，每个账户有100以太币。

### Truffle安装及操作

### 3.1 truffle安装

在控制台中，利用npm包管理工具，输入以下命令安装truffle并验证是否安装成功

sudo npm install -g truffle

truffle --version

### 3.2 创建项目

a. truffle的webpack工具需要在github上下载，因此我们需要先修改host文件，输入以下命令：

sudo gedit /etc/hosts

在hosts文件内加以下两行：

0.0.0.0:443 github.com

199.232.68.133 raw.githubusercontent.com

b. 在桌面上新创建一个文件夹作为工作目录，在工作目录内用unbox工具新建一个truffle项目，输入以下命令。

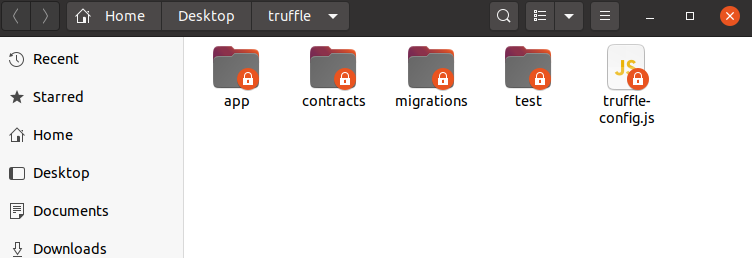
cd ~/Desktop

sudo mkdir mytruffle

cd mytruffle

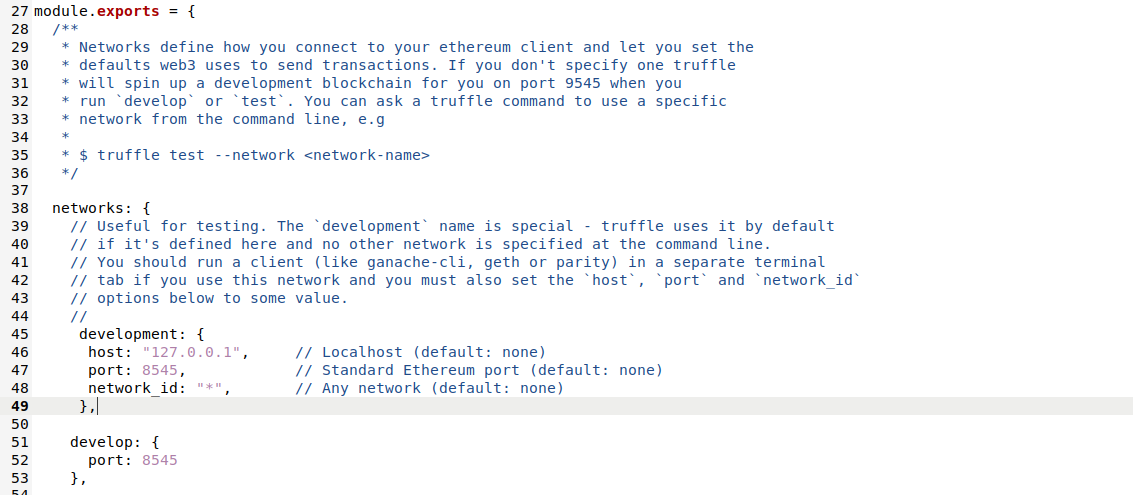
sudo truffle unbox webpack

在文件夹内出现如下目录结构说明构建成功。

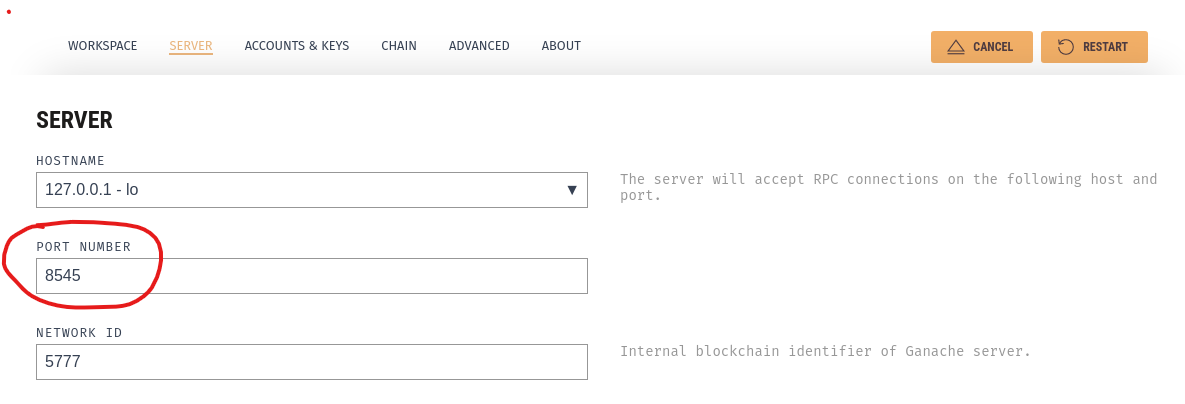


c. 在保证打开图形化ganache的前提下，修改项目配置，进行项目编译运行并查看效果。

修改truffle-config.js文件，将development前的注释移除



其中，**port**必须与图形化ganache工具中的port**保持一致。**



如不一致，请自行修改端口后，保存重启ganache工具

项目配置修改完成后，输入以下命令，运行初始化自带的合约。

sudo truffle compile

sudo truffle migrate

回到ganache图形化界面，应可以看到block栏里多了几个区块。

### 3.3 简单的智能合约实践

a. 在contracts文件中新建文件mathControl.sol，并在其中填入内容

pragma solidity >=0.4.22 <0.9.0;

contract MathContract {

function add(uint a,uint b) public pure returns(uint){

return a+b;

}

function sub(uint a,uint b) public pure returns(uint) {

return a-b;

}

}

可以看到这是一个比较简单的加减合约

b. 在migrations文件夹新建一个将合约部署到私有链上的文件3\_initial\_migration.js，在其中填入内容如下

var Math=artifacts.require("./MathContract.sol");

module.exports = function (deployer) {

deployer.deploy(Math);

};

c. 在test文件夹新建一个合约调用文件mathcontract.js，在其中填入内容如下。

var MathContract=artifacts.require("MathContract")

contract('MathContract',function(accounts){

it("first",function(){

return MathContract.deployed().then(function(instance){

console.log(instance.address)

instance.add(1,1).then(function(result) {

console.log(result)

assert.equal(result,2,"add prompt");

})

instance.sub(5,1).then(function(result) {

console.log(result)

assert.equal(result,4,"sub prompt");

})

})

})

it("second",function(){

return MathContract.deployed().then(function(instance){

console.log(instance.address)

instance.add(1,1).then(function(result) {

console.log(result.toNumber())

assert.equal(result,2,"add prompt");

})

instance.sub(5,1).then(function(result) {

console.log(result.toNumber())

assert.equal(result,4,"sub prompt");

})

})

})

})

d. 部署并测试合约，输入命令如下：

sudo truffle compile

sudo truffle migrate

truffle test ~/Desktop/mytruffle/test/mathcontract.js

查看输出结果和ganache内区块变化

**作业上交内容与事项：**

1．实验报告文档：按照要求完成实验并将关键步骤实验结果进行截图记录。注意文档工整；

2．程序代码：如果有程序代码或其他相关材料，可汇总压缩所有文件并上传压缩包。注意文件命名；

**本期实验作业上交期限：**

请在实验设置的截止日期内提交实验报告，若逾期提交，成绩会被适当打折。

**本次作业上交内容：**

* 实验报告文档(.docx)

## 三、实验感受及记录

* 1. **实验感受（本次实验遇到的问题、主要收获等内容）**

我通过Geth搭建了多个节点的区块链网络，理解了以太坊节点之间的通信和共识机制，进一步认识到区块链去中心化的本质及其对网络安全的保障作用。在智能合约开发过程中，Ganache提供的本地区块链环境使测试变得更加便捷，而Truffle则简化了智能合约的编译、部署和测试，极大地提升了开发效率。

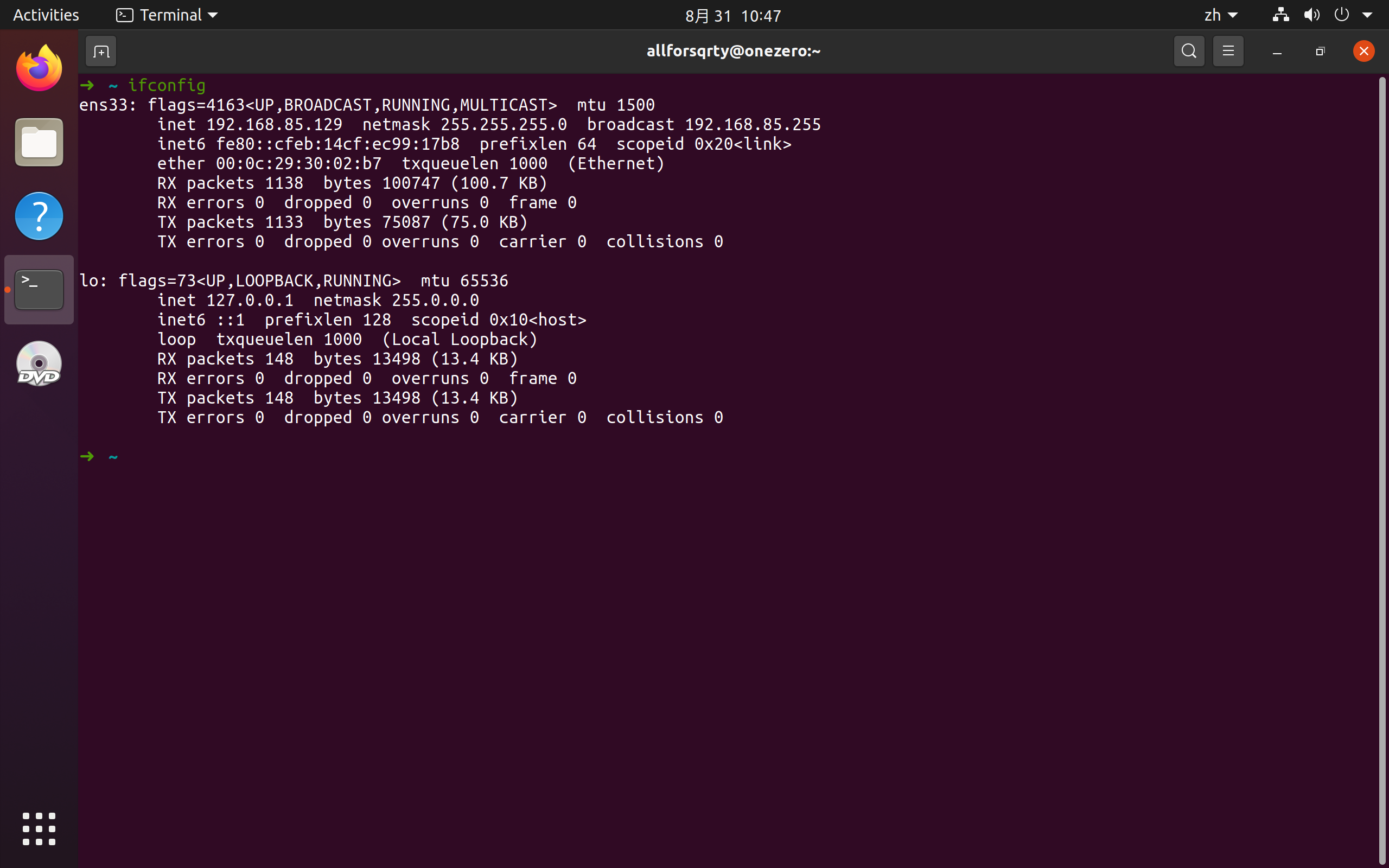
同时，在智能合约开发环节中，我也遇到了一些问题。虽然Truffle简化了流程，但由于对其命令行工具不够熟悉，我在部署时出现了一些错误，尤其是在处理复杂的合约结构和依赖关系时。通过反复调试和查阅文档，我逐渐掌握了工具的使用技巧。

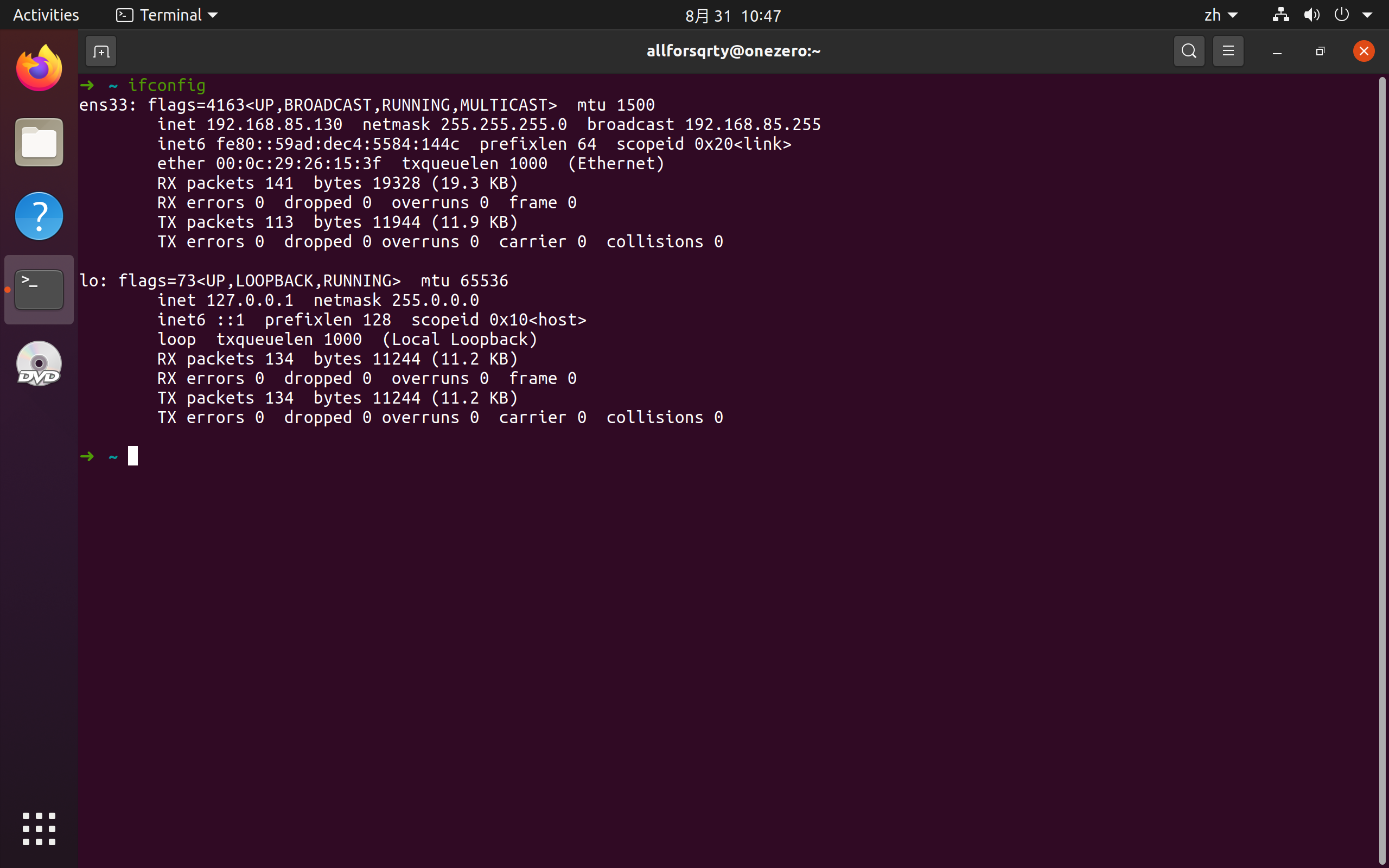
本次实验让我在实际操作中加深了对区块链技术的理解，掌握了节点部署和智能合约开发的基本技能，让我对区块链技术的应用前景有了更深刻的认识。

* 1. **实验记录（实验过程中关键步骤截图记录及文字描述）**

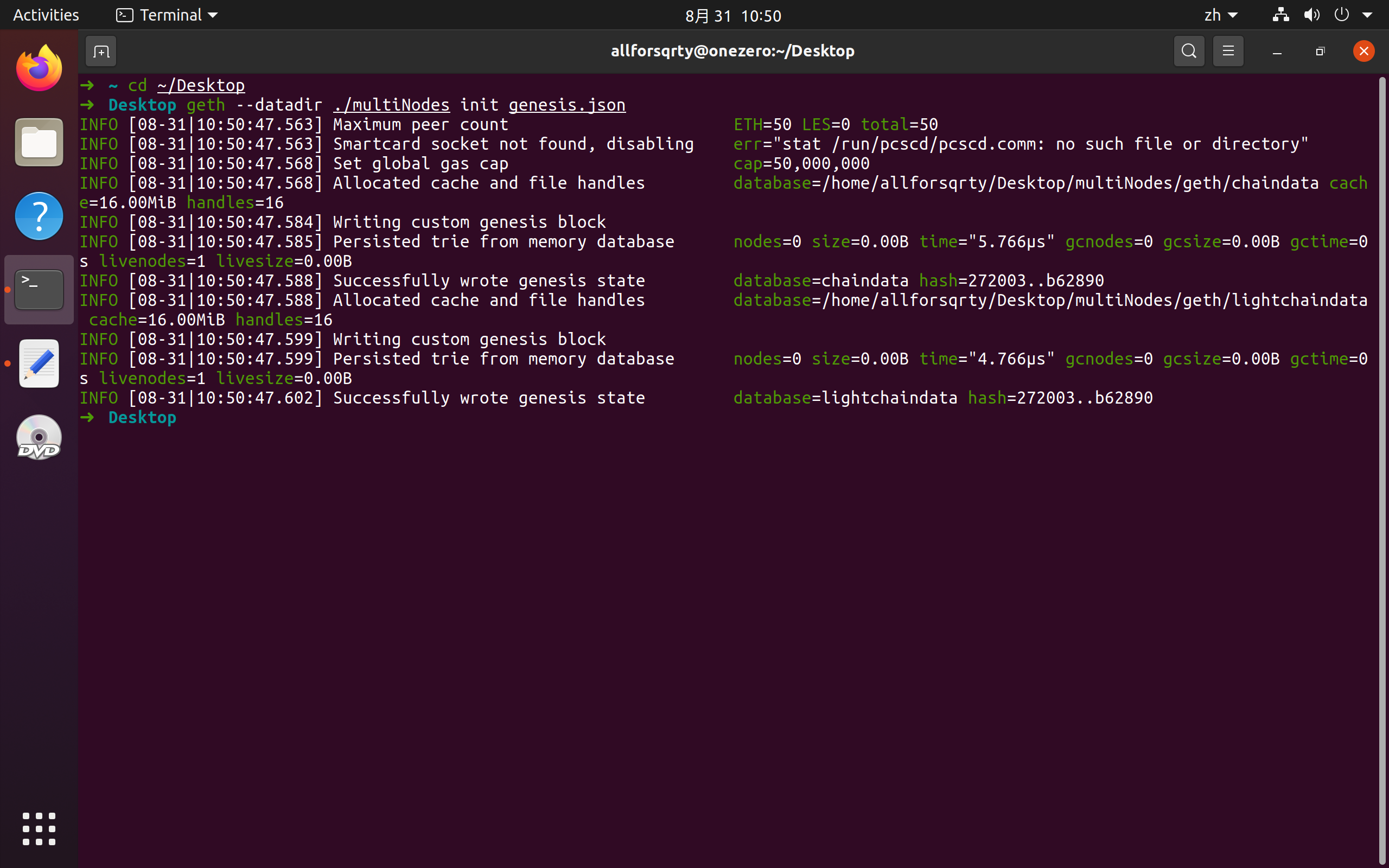
**1 基于Geth的多节点部署**

1. 首先克隆一台虚拟机，分别并命为ubuntu\_64与ubuntu\_64\_clone，开启两台虚拟机。
2. 将genesis.json文件中的chainID修改为666，并查看两台虚拟机的ip地址，结果如下。并在两台虚拟机的桌面上分别建立工作目录multiNodes。

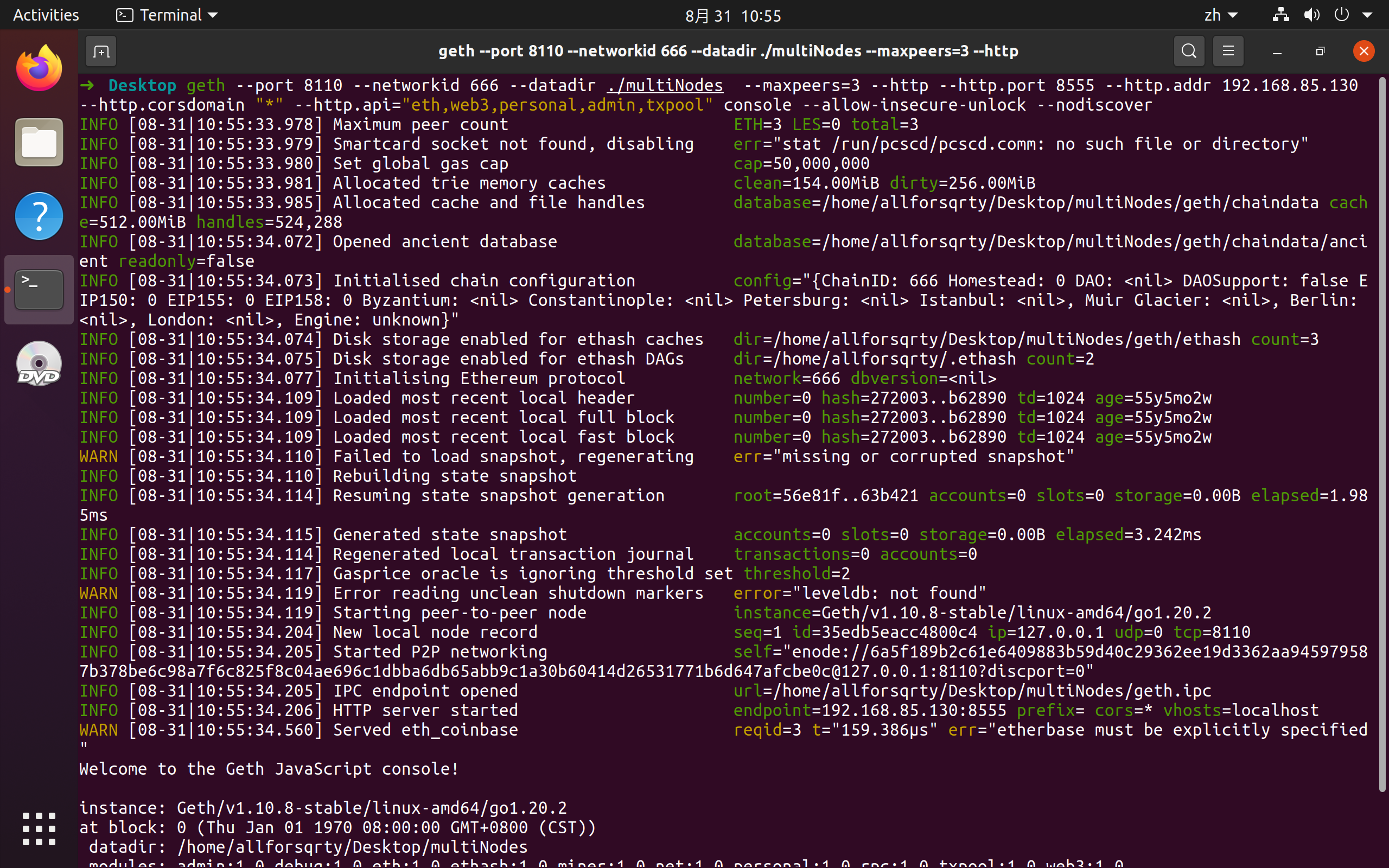




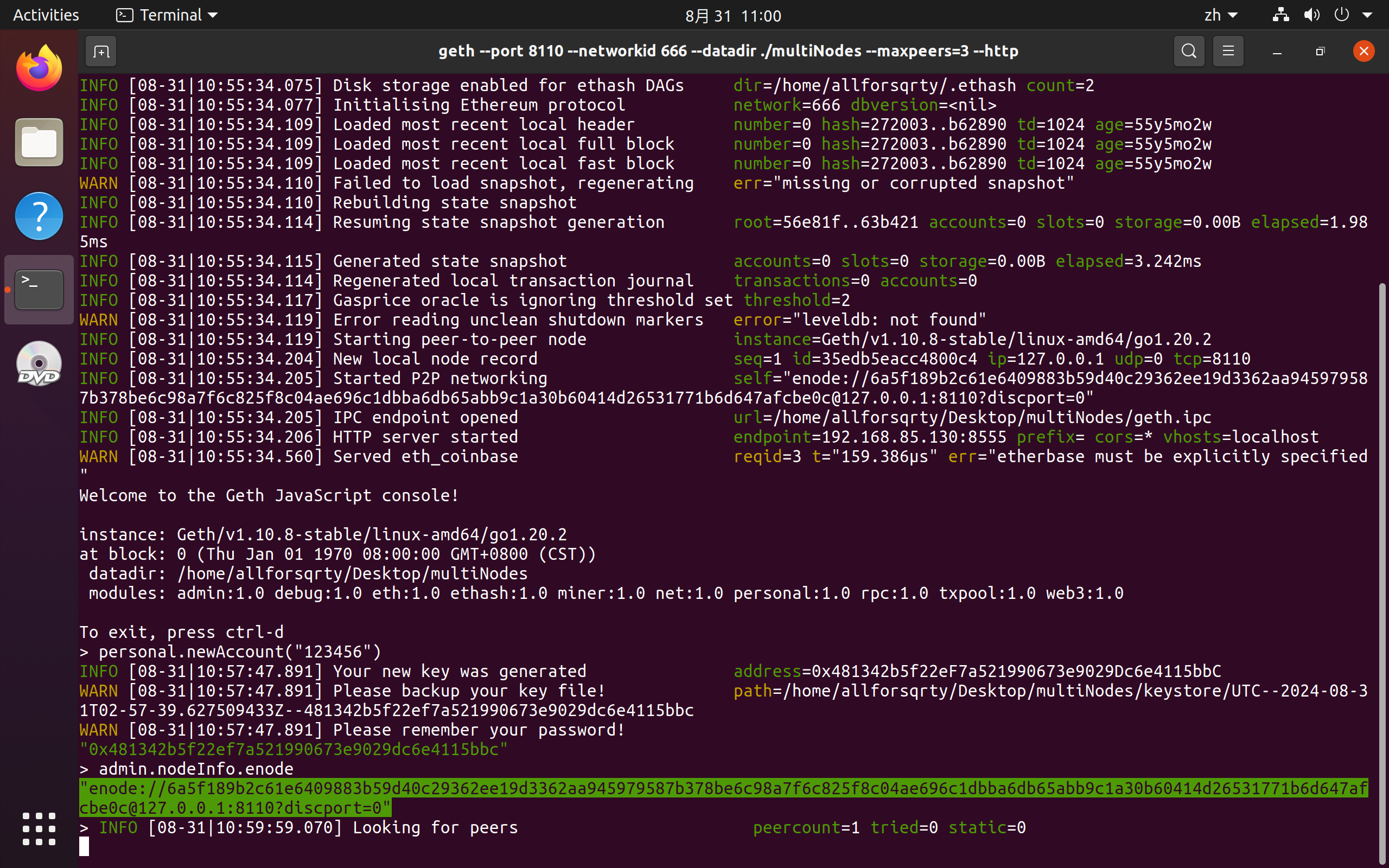
1. 在两台虚拟机中，完成私有链的初始化



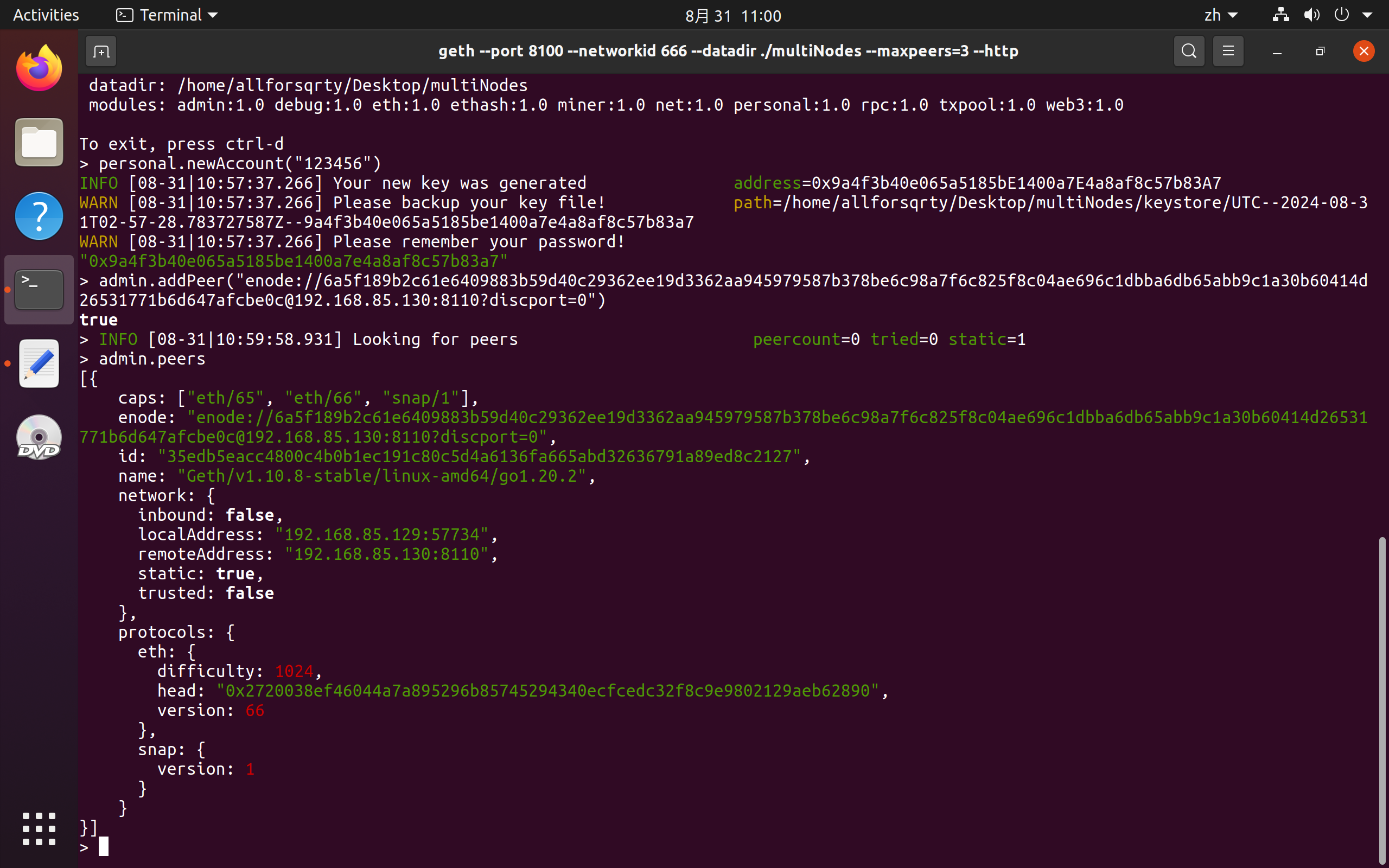
1. 在两台虚拟机上启动节点，第二台虚拟机的端口号需发生变化，并分别添加账户。



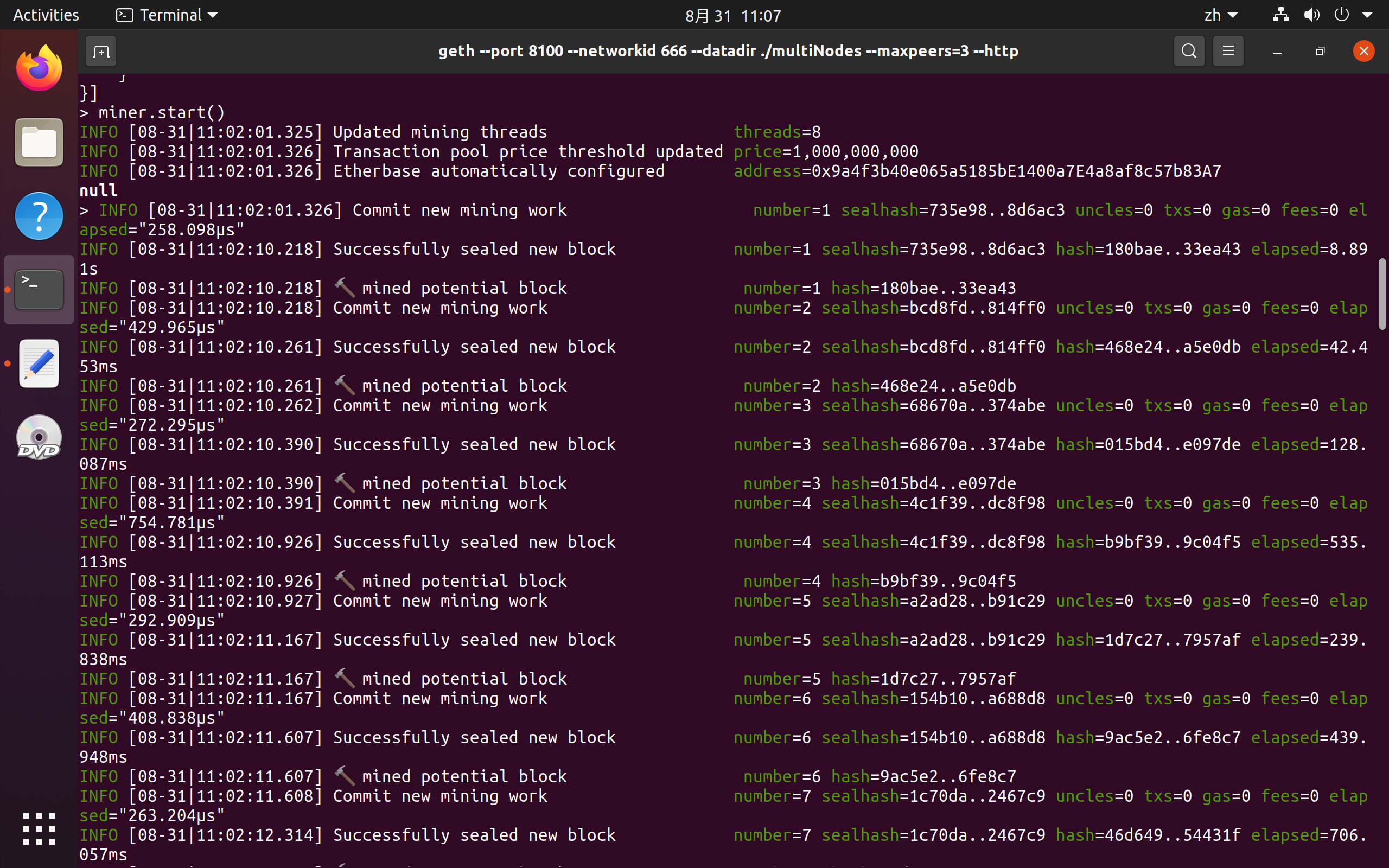
1. 在第二台虚拟机上获取enode信息

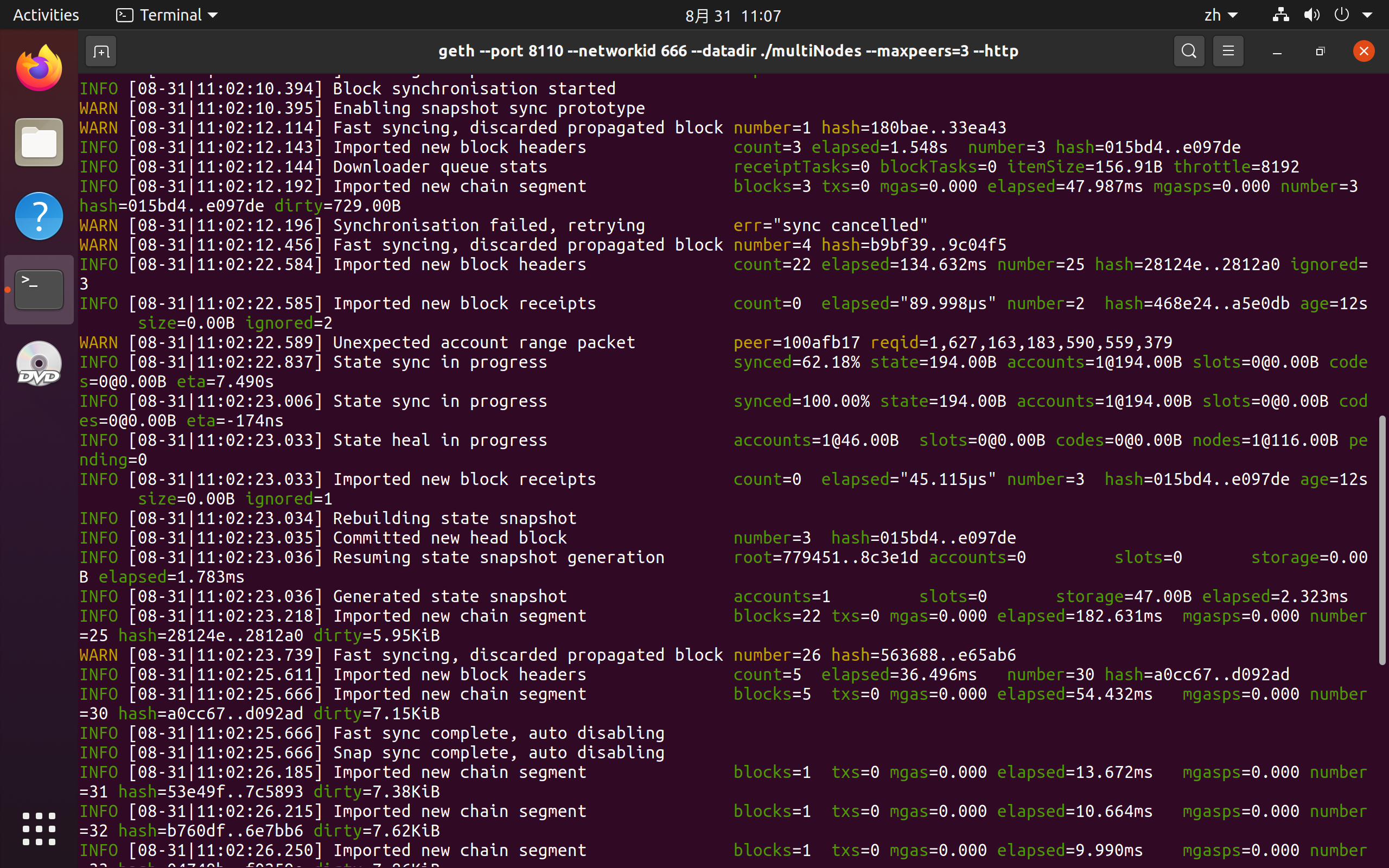


1. 改造enode信息后，在第一台虚拟机上添加节点，执行admin.peers观察发现，节点已经成功添加。



1. 在第一台虚拟机上开始挖矿，第二台虚拟机上同步显示信息。





1. Miner.stop()后查看两个节点的区块高度，发现相同，均为51。

