实验六 使用Truffle框架搭建宠物商店Dapp

【实验介绍】

本实验开发一个去中心化应用（Dapp），让大家领养宠物。truffle box已提供pet-shop网站应用的框架代码，我们仅需编写智能合约及其交互部分。



官网地址：https://www.trufflesuite.com/tutorial

【实验要求】

1. 创建Truffle项目
2. 编写智能合约
3. 编译和部署智能合约到区块链
4. 如何通过Web3和智能合约交互
5. 学习Ganache和MetaMask的使用

【实验过程】

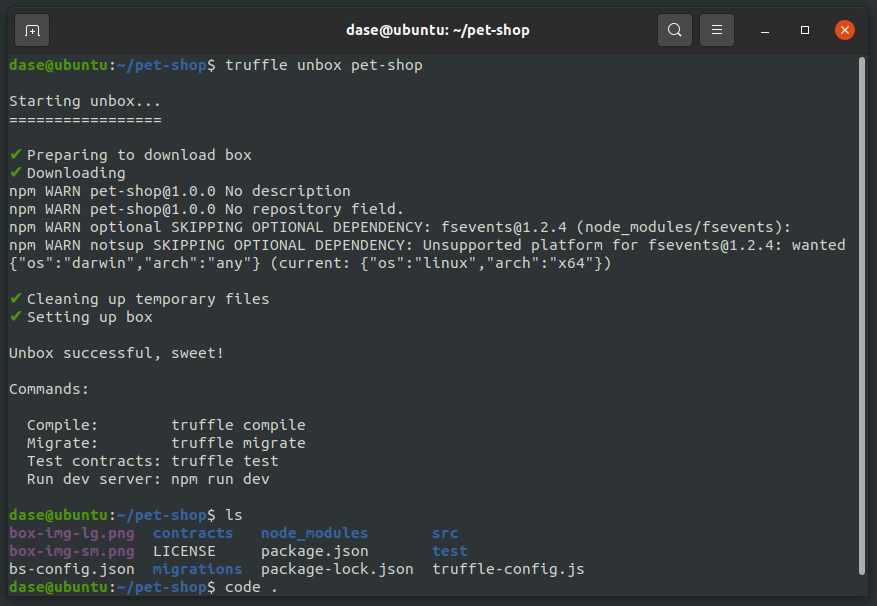
1. 创建项目
2. 创建项目目录并进入

|  |
| --- |
| **mkdir** pet-shop  **cd** pet-shop |

1. 初始化项目

使用truffle unbox拉取github仓库的box框架代码，等待下载构建完成。

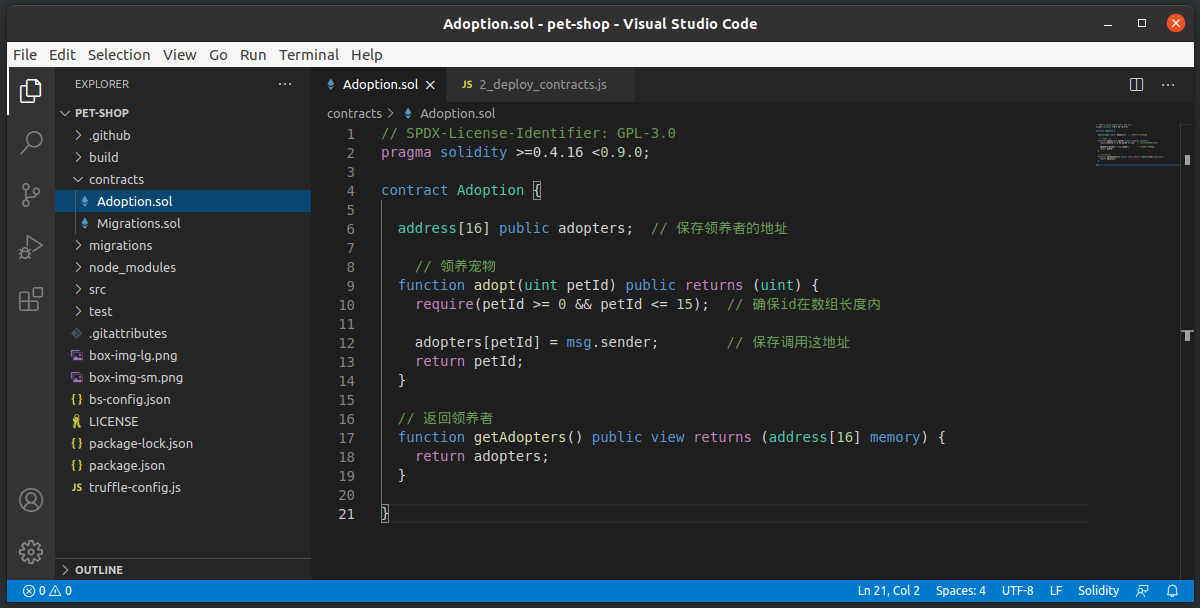
|  |
| --- |
| truffle unbox pet-shop |



contracts/智能合约的文件夹，所有的智能合约文件都放置在这里，里面包含一个重要的合约Migrations.sol。migrations/用来处理部署（迁移）智能合约，迁移是一个额外特别的合约用来保存合约的变化。test/智能合约测试用例文件夹。truffle.js/配置文件。

1. 编写智能合约

在contracts/目录下添加智能合约文件Adoption.sol:

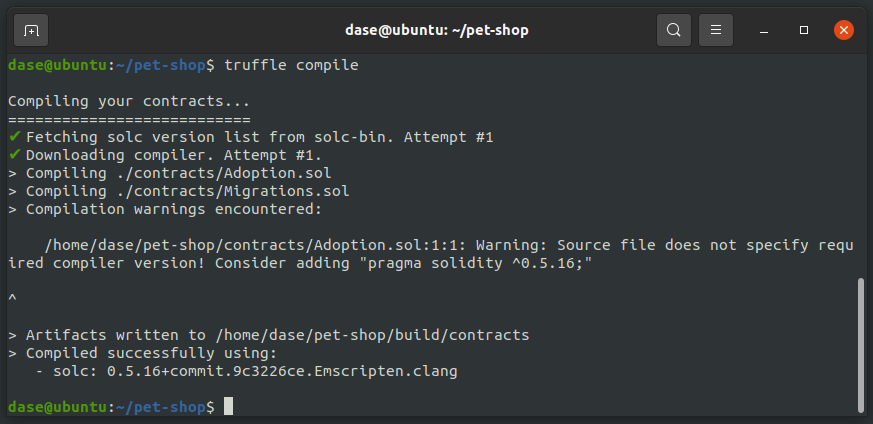


1. 编译部署智能合约

Truffle集成了一个开发者控制台，可用来生成一个开发链用来测试和部署智能合约。

1. 编译

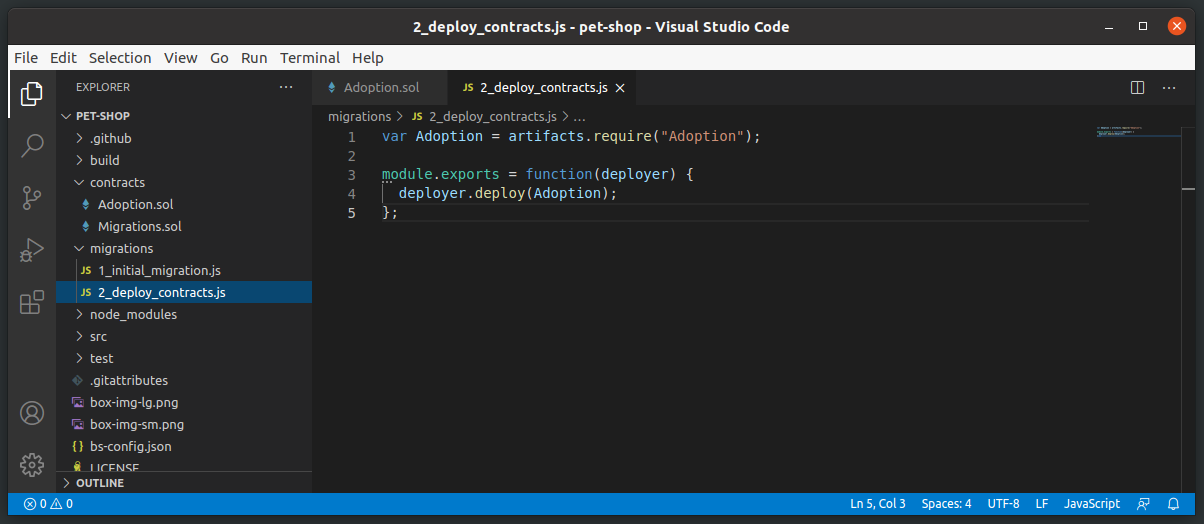
在项目目录下输入truffle compile命令进行编译：



1. 编写部署脚本

编译后就可以将其部署到区块链上。在migrations/文件夹下已经有一个1\_initial\_migration.js部署脚本，用来部署Migrations.sol合约。Migrations.sol用来确保不会部署相同的合约。

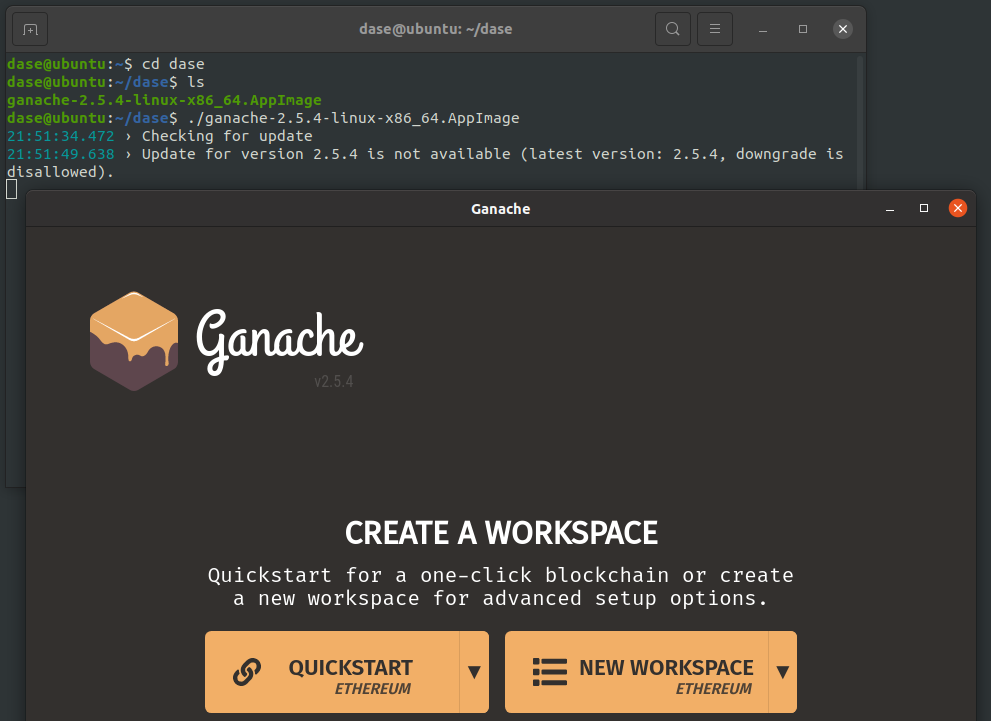
创建一个自己的部署脚本2\_deploy\_contracts.js：



1. 开启区块链

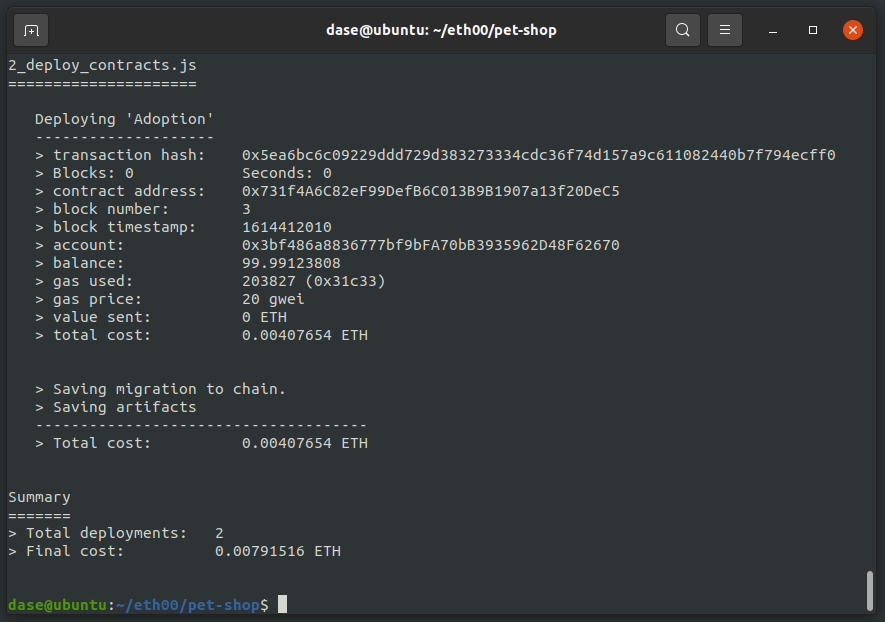
部署需要区块链，使用[Ganache](http://truffleframework.com/ganache/)可快速开启一个私链来进行开发测试，默认会在7545端口上运行一个开发链。

新开一个终端，在dase/目录下输入./ganache-2.5.4-linux-x86\_64.AppImage运行Ganache，点击QUICKSTART快速启动一个以太坊区块链。

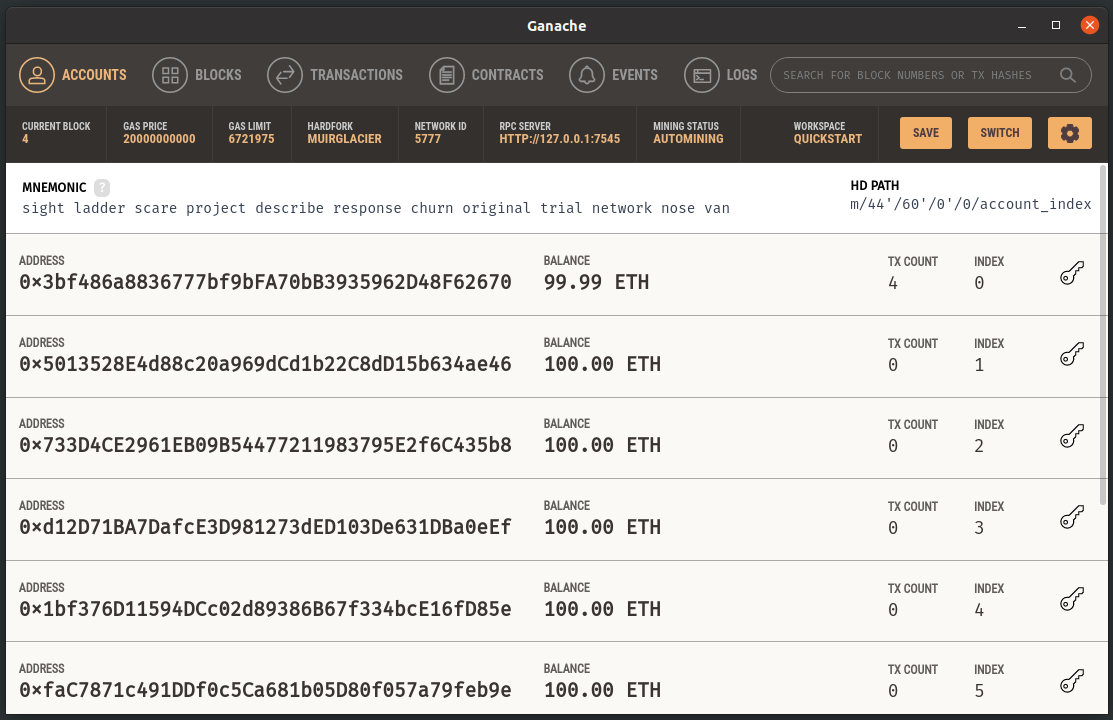


1. 部署

执行部署命令truffle migrate，编译情况如下，可以看到部署合约消耗的比特币。

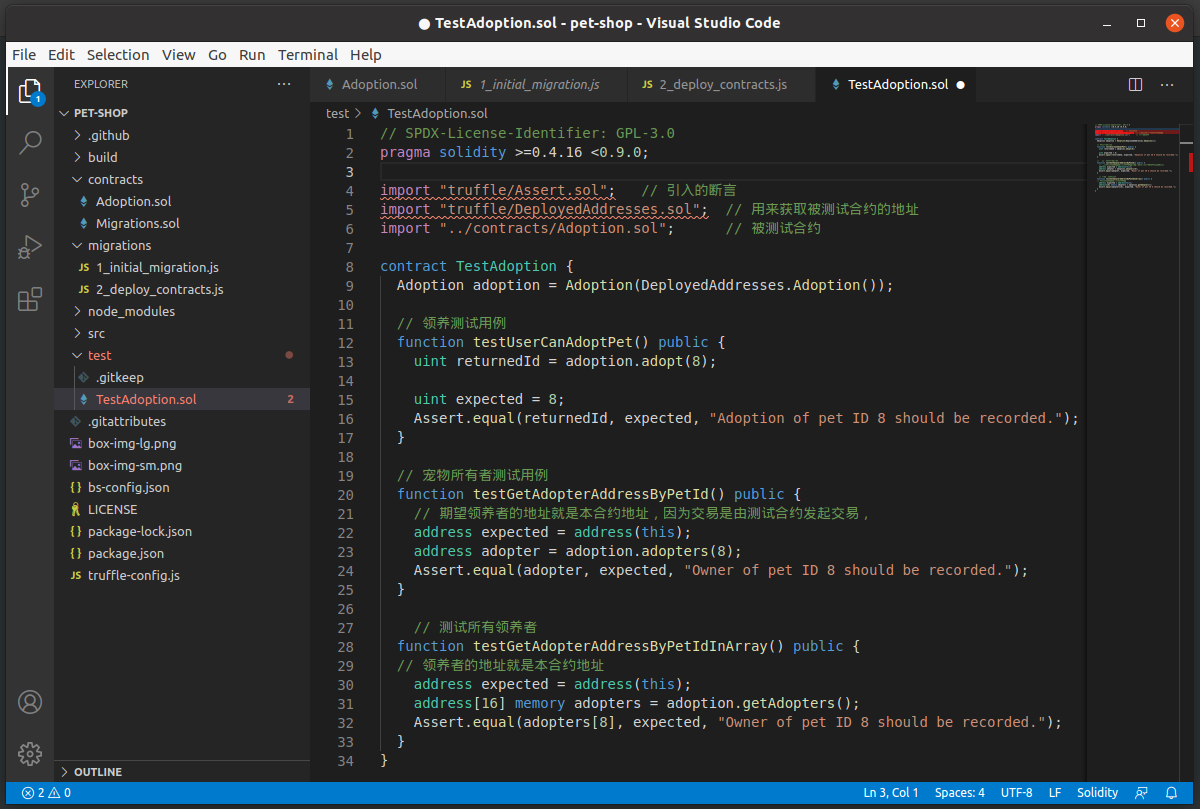


Ganache账户页面可见消耗的以太币，区块页面可见产生了四个区块，此时智能合约已部署完毕。

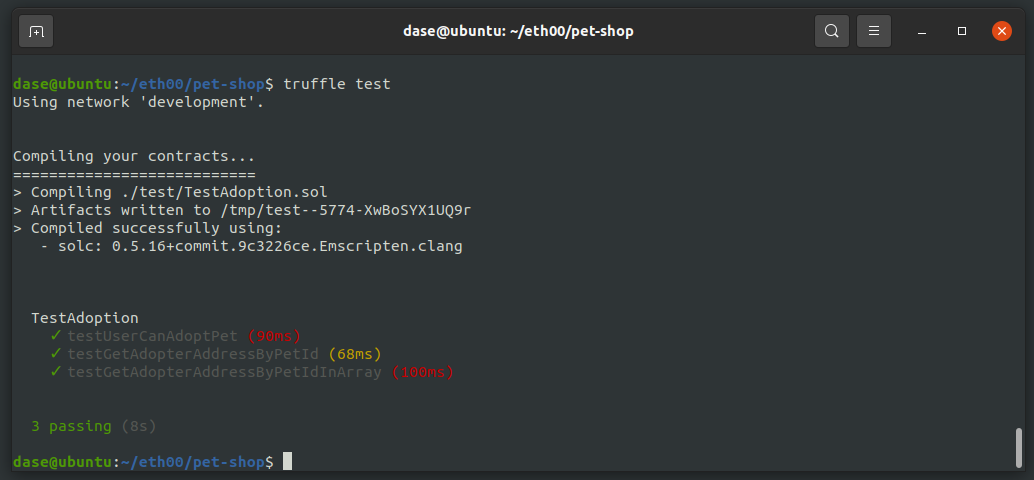


1. 测试

在test/目录下新建一个TestAdoption.sol，编写测试合约，其中Assert.sol及DeployedAddresses.sol由truffle框架提供：



在终端执行truffle test进行测试：



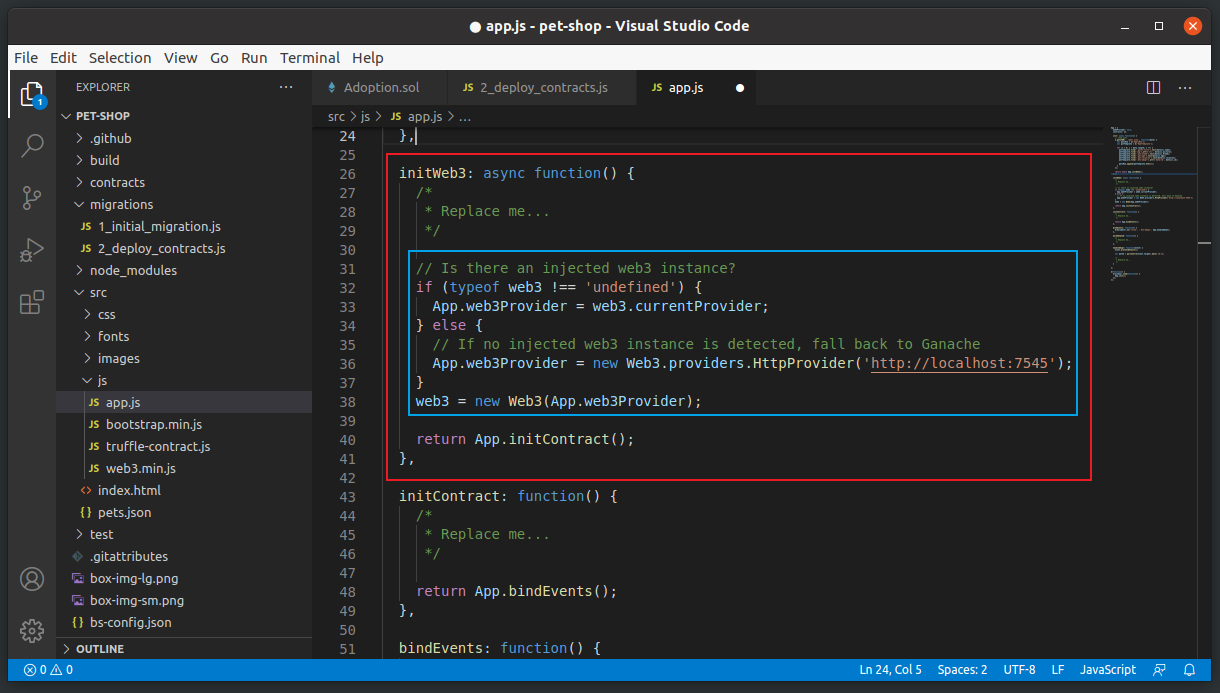
上图表示测试通过，可通过Gnanache观察区块链的变化。

1. 创建用户接口和智能合约交互

Truffle Box pet-shop里，已经包含了应用的前端代码，代码在src/文件夹下。

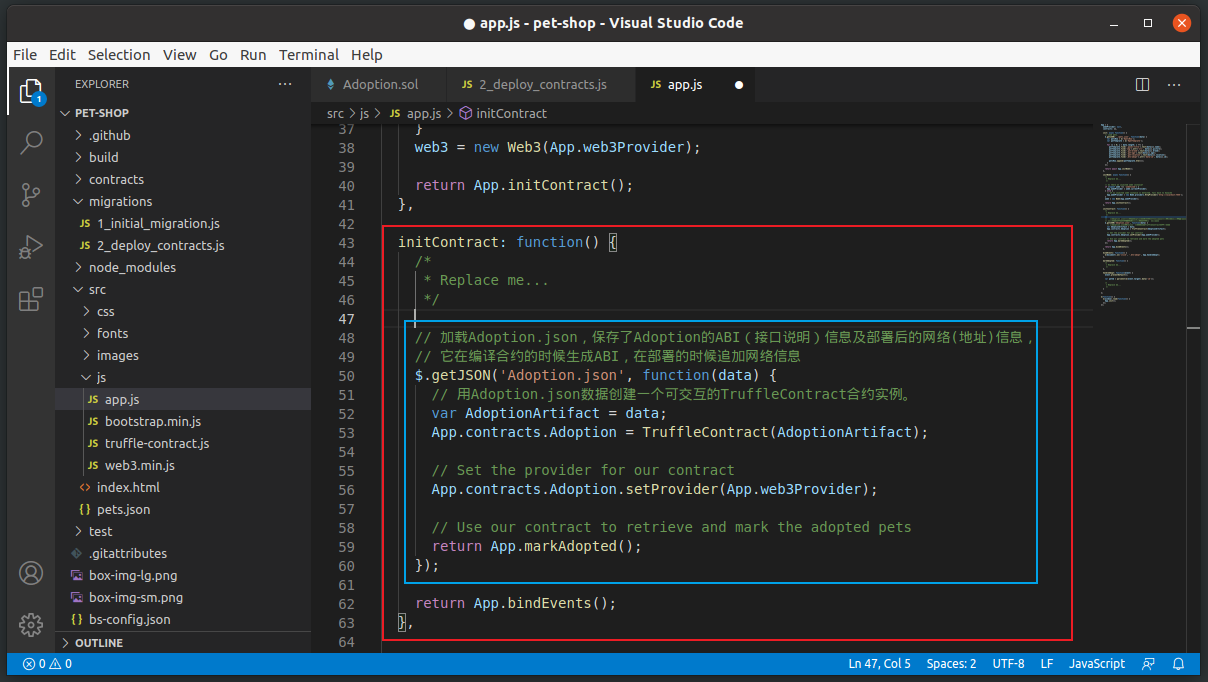
1. 初始化web3

修改app.js中的initWeb3: **function**()，添加蓝框内容：



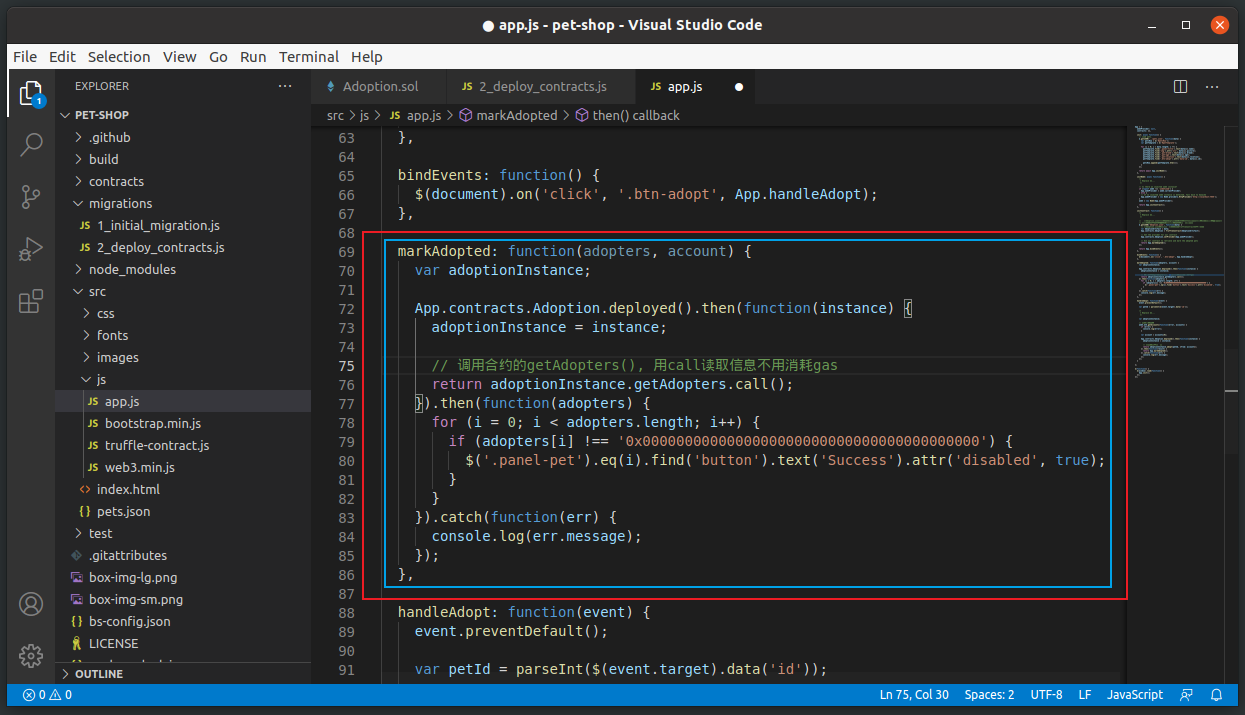
1. 实例化合约

使用truffle-contract保存合约部署的信息，自动修改合约地址，添加代码如下：

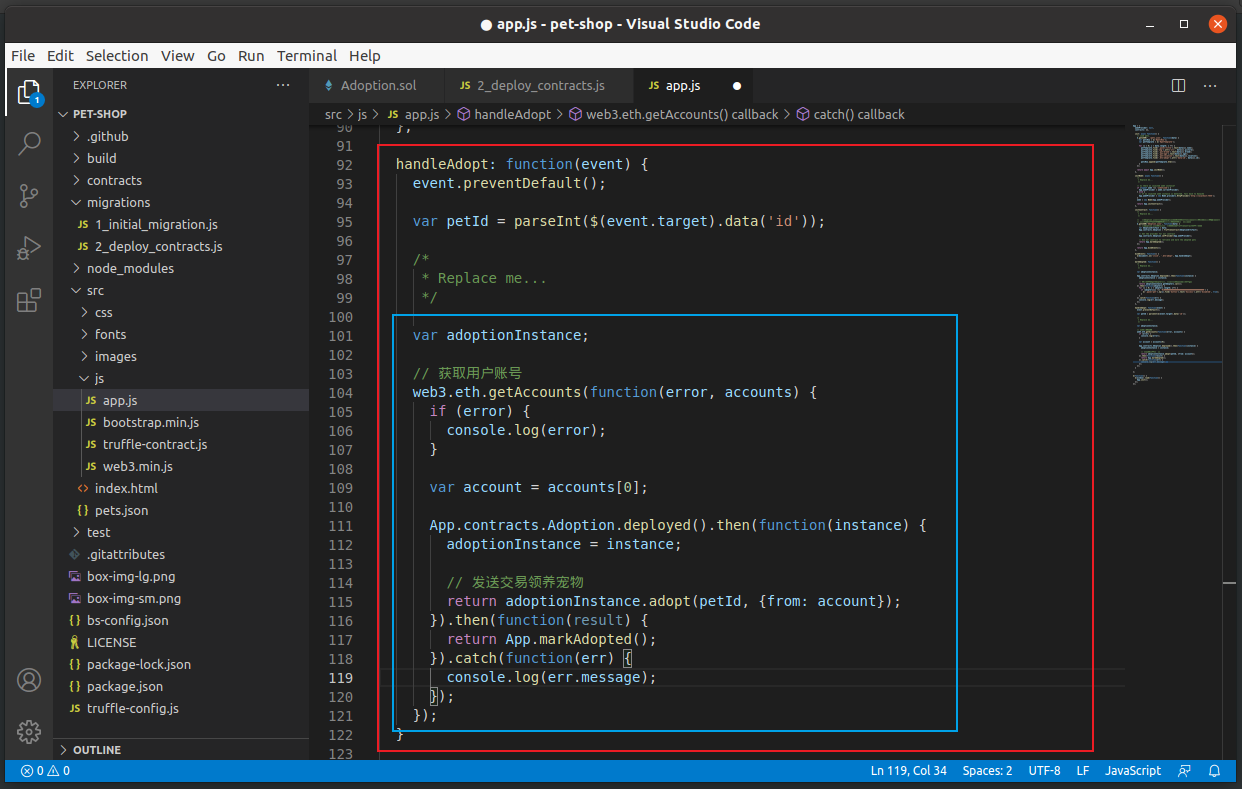


1. 处理领养

修改markAdopted: **function**(adopters, account)如下：

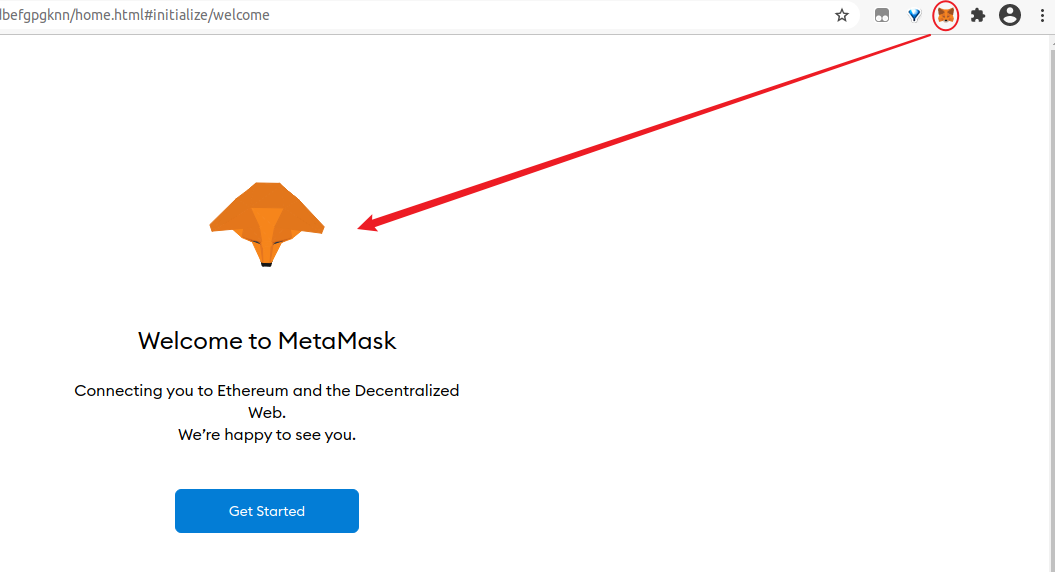


修改handleAdopt: **function**(event)如下：

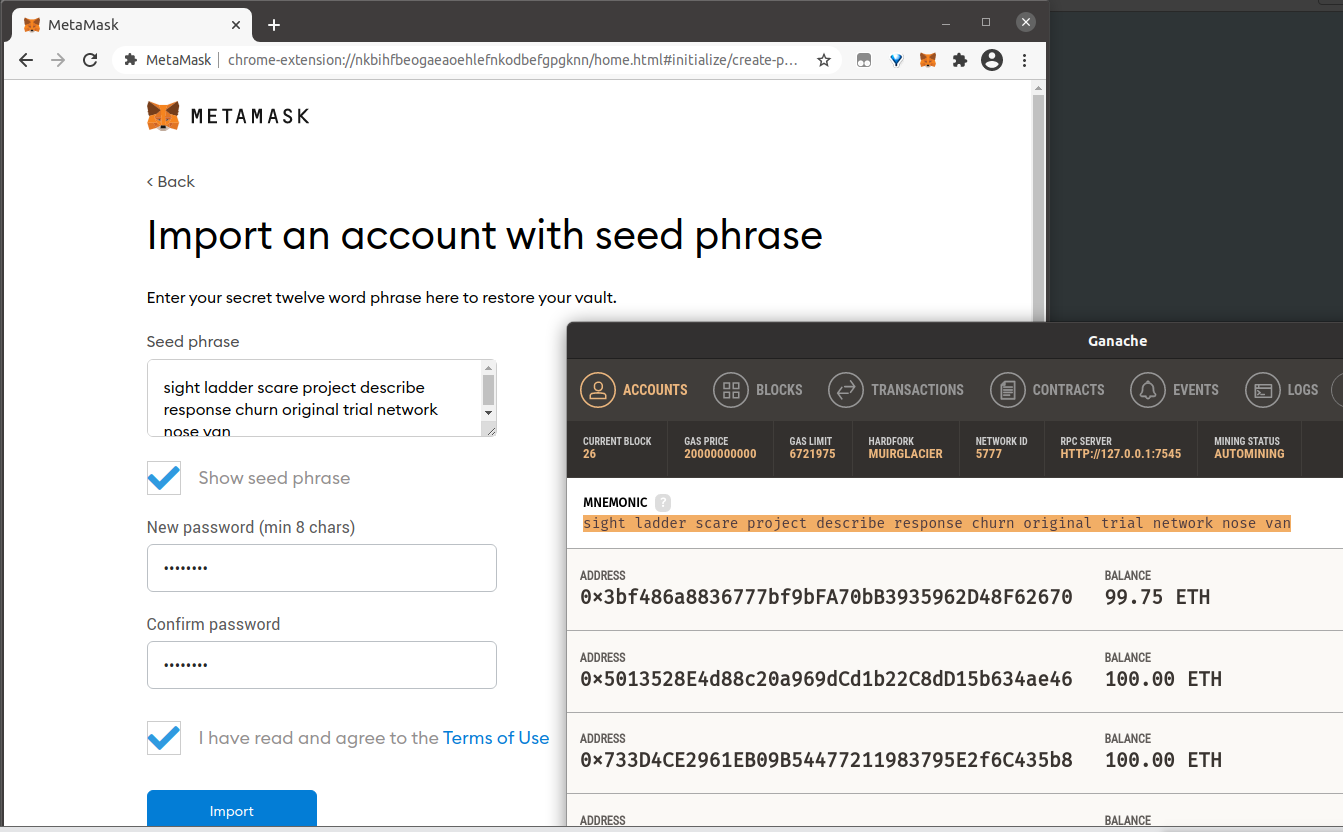


1. 在浏览器中运行

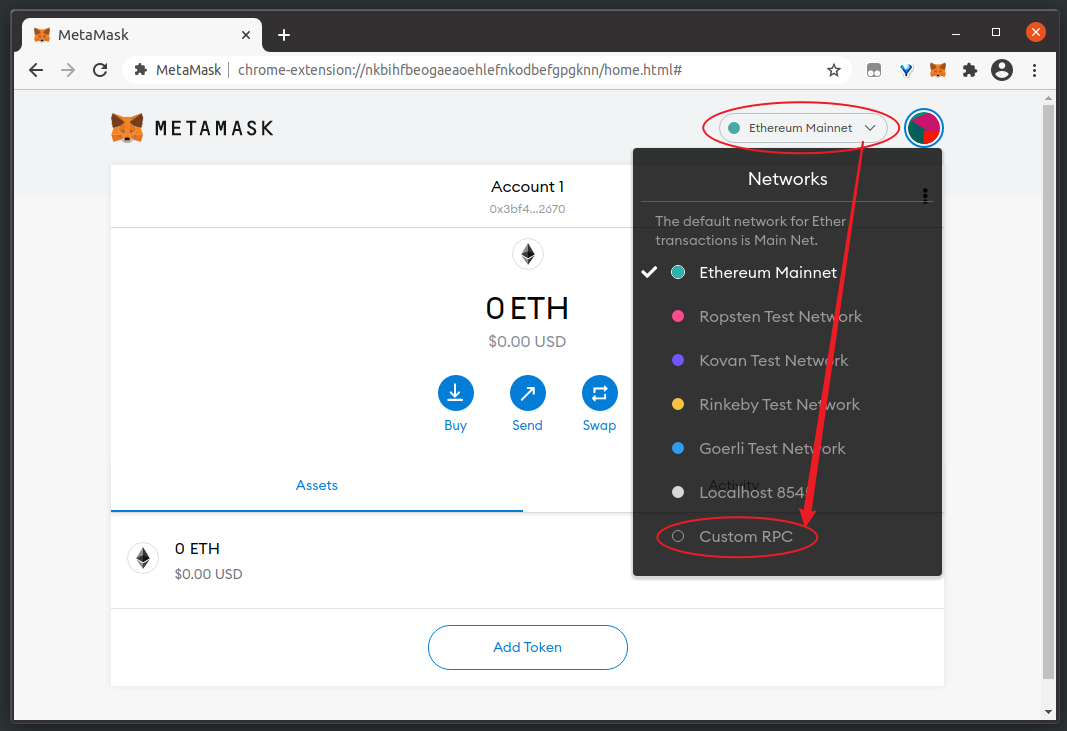
打开chrome浏览器，点击右上方MetaMask插件，进入MetaMask：

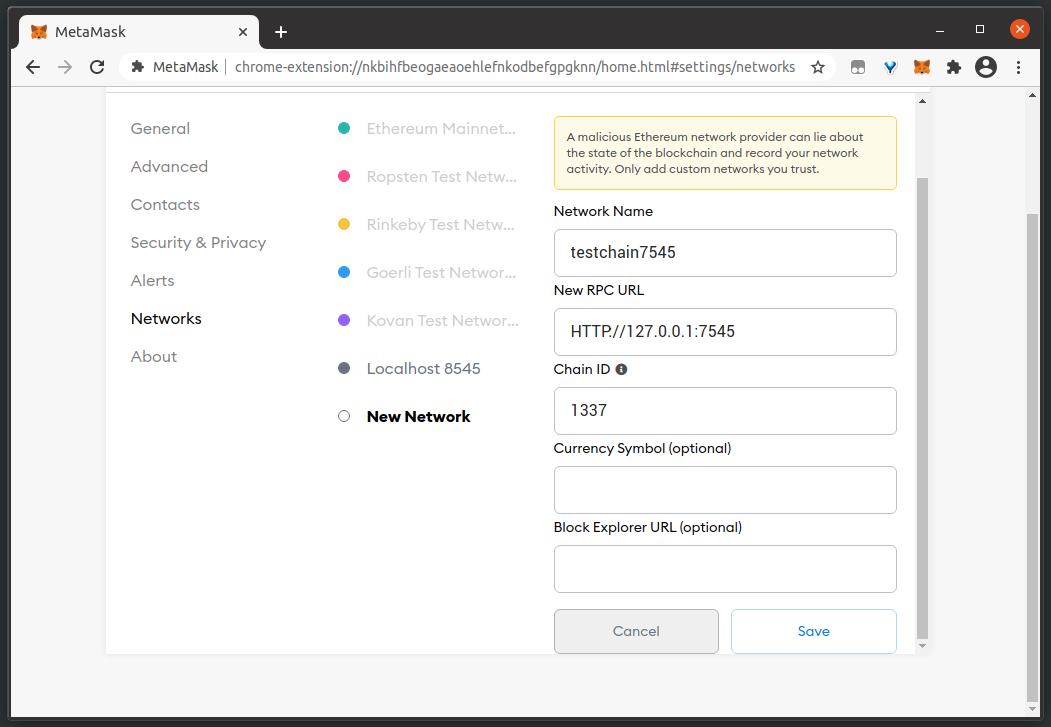


点击Get Start，选择左边的Import Wallet，进入输入界面，Seed phrase为Ganache显示的助记符（每次随机生成），密码为dasedase(也可自行设定)。

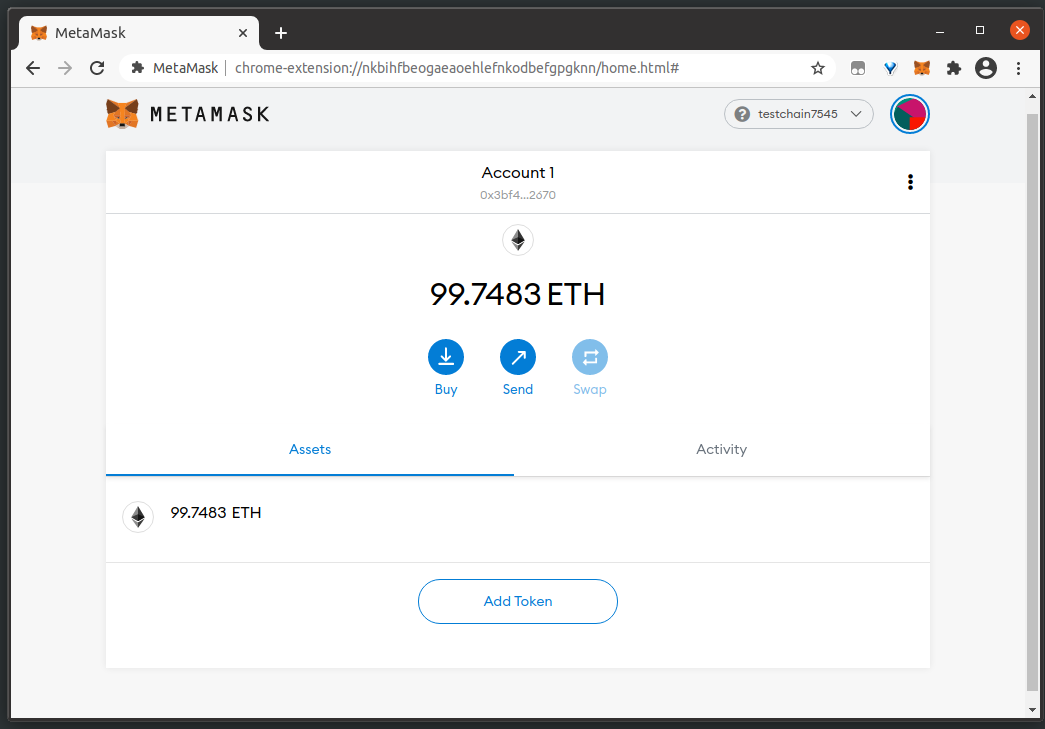


通过下图步骤连上本次实验的测试开发链：



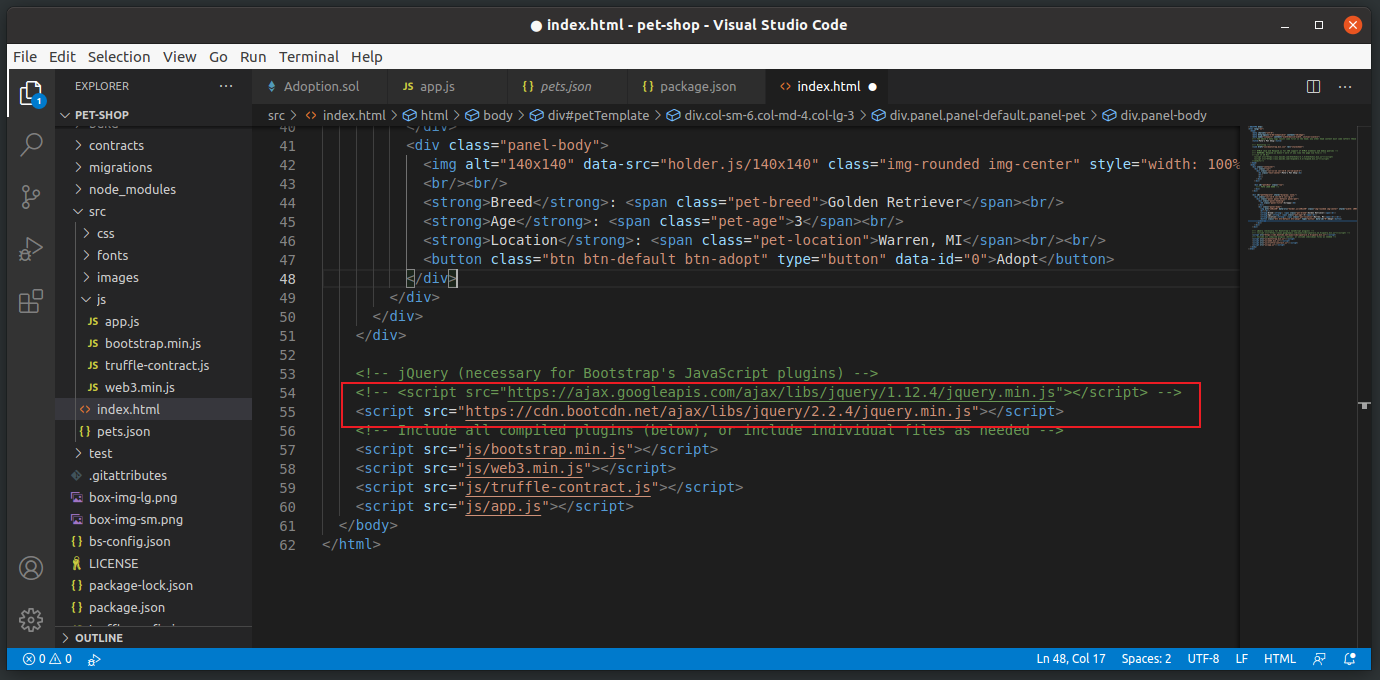


连上开发链后，账户1的比特币余额和Ganache中一致。

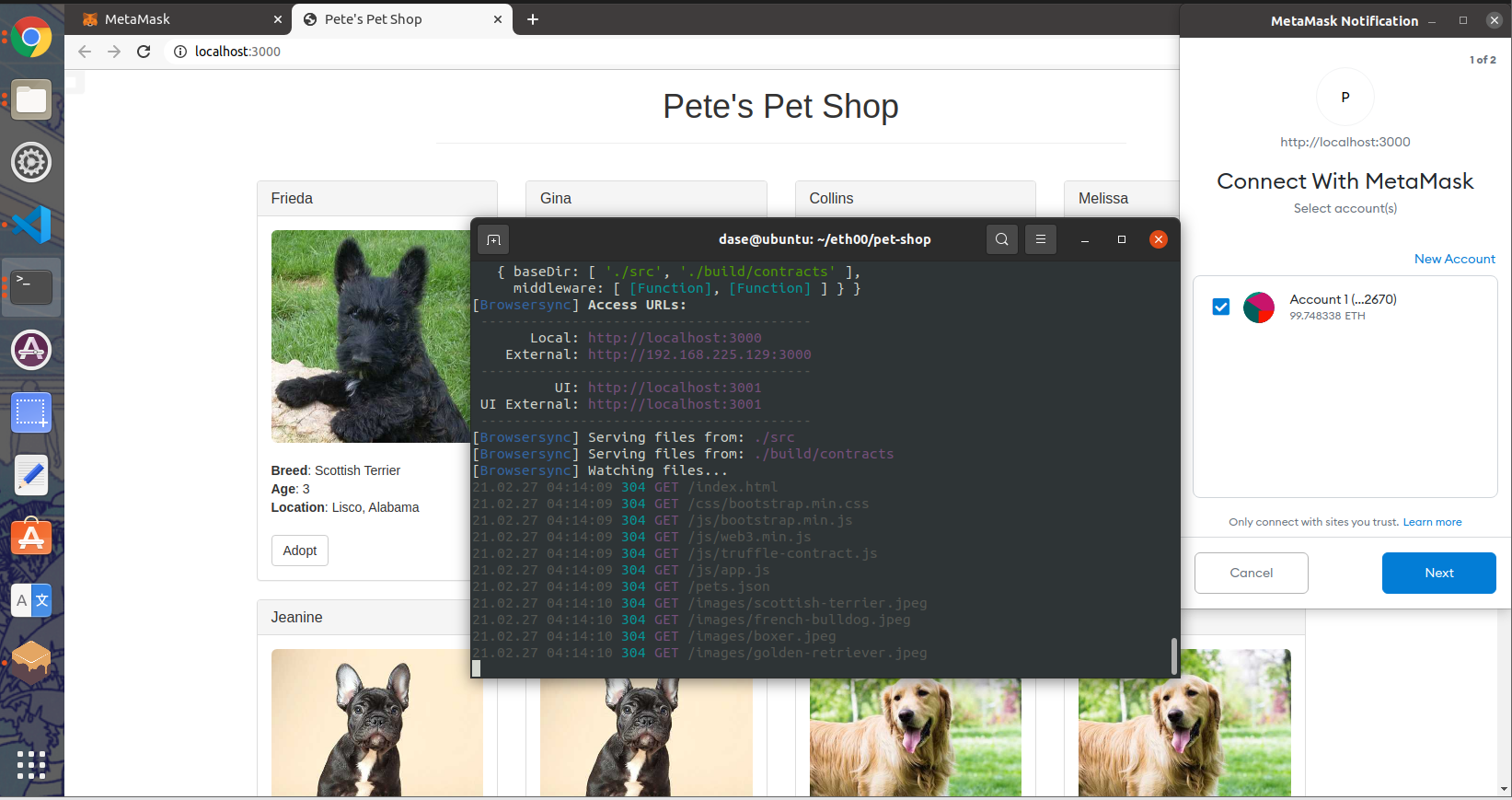


1. 启动服务

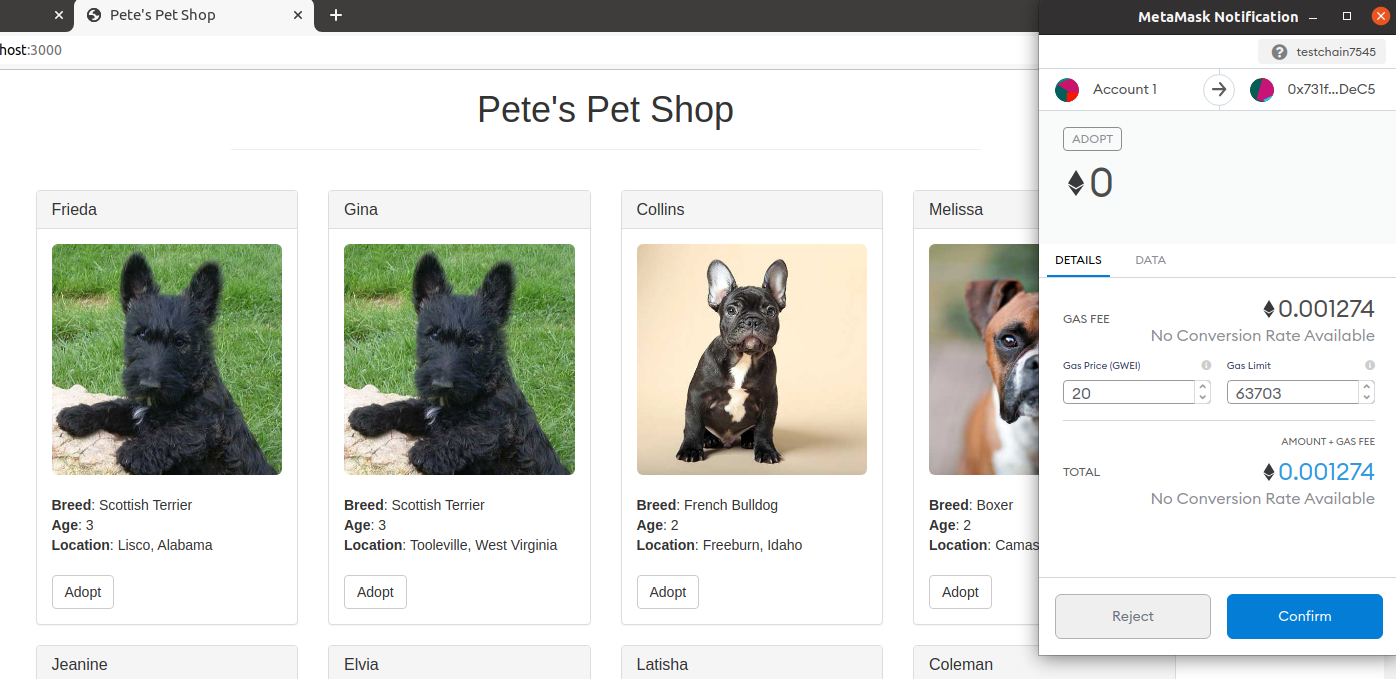
修改index.html中的代码，将jquery.min.js的cdn源改为可访问的。



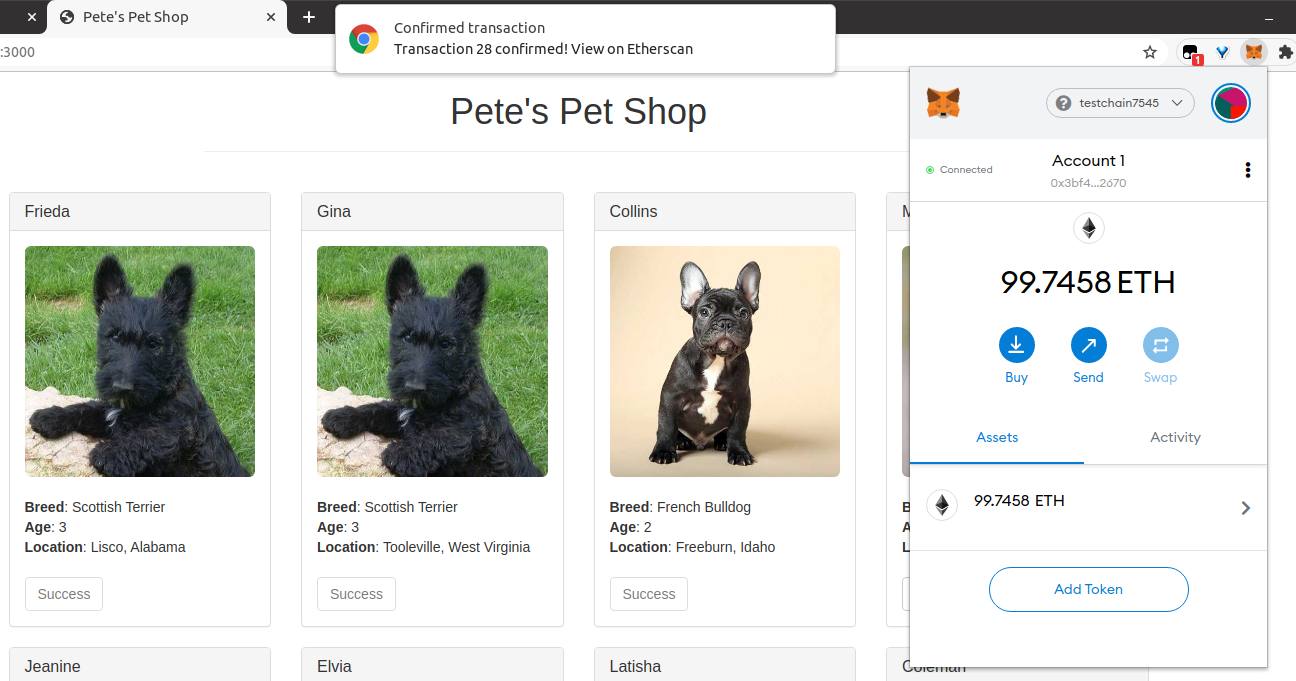
运行npm run dev启动应用：



点击上图右方的Next，将MetaMask钱包连接到本应用中，然后后面点击任一Adopt按钮，MetaMask会弹出相应的交易确认界面：



确认后将收养成功（这里我又收养了两只：）），交易记录被保存在链上，从Ganache中可查询到相关区块和交易记录。



若无法运行，尝试清除npm缓存，再运行npm run dev

|  |
| --- |
| npm cache clean –force  **rm** **-**rf node\_modules  **rm** **-**rf package-lock.json  npm install  npm run dev |

