trường hợp update bị lỗi--> search source list --> .... dán etc

nếu bị đụng tiến trình --> sudo kill -9 <tiến trình>

# BT: Cài đặt môi trường và xây dựng hệ thống mạng fabric-samples

sudo apt update

sudo apt install -y curl

curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E bash -

sudo apt install -y nodejs

sudo apt-get update

sudo apt-get install git -ysudo apt-get install docker-compose -y

sudo systemctl start docker

sudo systemctl enable dockersudo apt install jqsudo curl -sSL https://bit.ly/2ysbOFE | bash -s

Thêm quyền người dùng vào group docker

sudo usermod -aG docker ${USER}

sudo apt-get update

kiểm tra phiên bản nodejs: nodejs --version

sudo npm install npm@6.14.17 -gCài đặt ngôn ngữ Go

sudo rm -rf /usr/local/go

Truy cập trên máy ảo : <https://go.dev/dl/>

A screenshot of a computer

Description automatically generated

Chuyển đến thư mục Download: cd Downloads

sudo tar -C /usr/local -xzf go1.23.2.linux-amd64.tar.gz

sudo gedit ~/.bashrc

🡪Dán đoạn code dưới vào file bashrc, save

project1: thay đổi thành tên thư mục đã tạo

export PATH=$PATH:/home/ubuntu/.local/bin

export GOROOT=/usr/local/go

export PATH=$GOROOT/bin:$PATH

export FABRIC\_CFG\_PATH=$HOME/go/src/project1/fabric-samples/config/

export CORE\_PEER\_LOCALMSPID="Org1MSP"

export CORE\_PEER\_MSPCONFIGPATH=$HOME/go/src/project1/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

export CORE\_PEER\_ADDRESS=localhost:7051

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=$HOME/go/src/project1/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

export CORE\_PEER\_TLS\_ENABLED=true

export PATH=$PATH:$HOME/go/src/ project1/fabric-samples/bin

reboot

sudo systemctl enable docker

Tạo thư mục: mkdir -p $HOME/go/src/ project1 (CÓ THỂ KO CẦN TẠO THƯ MỤC)

cd $HOME/go/src/ project1

Tải fabric sample vào thư mục vừa tạo

curl -sSLO https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh && chmod +x install-fabric.sh

sudo ./install-fabric.sh

Remove tất cả các tài nguyên đã có để tạo các tài nguyên mới (tránh 2 ứng dụng khác nhau cùng chia sẻ cùng một tài nguyên)

cd fabric-samples/test-network/

./network.sh down

./network.sh up

./network.sh down

./network.sh up createChannel -ca -s couchdb

A computer screen with text

Description automatically generated

./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-go -ccl go

A computer screen shot of a program

Description automatically generated

cd ..

cd asset-transfer-basic/application-gateway-javascript/src/

npm install @grpc/grpc-js

npm install @hyperledger/fabric-gateway

node app.js

Cài vscode: sudo snap install code –classic

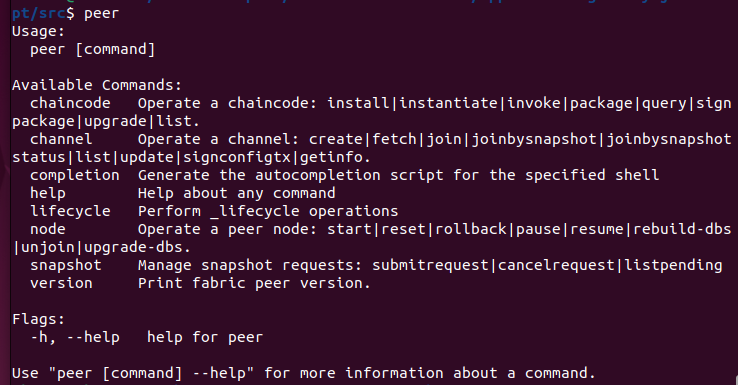
code . tenfile 🡪 vd: code . node app.js (KHÔNG QUAN TRỌNG CHỈ ĐỂ XEM FILE)

A computer screen shot of a program

Description automatically generated

Cài đặt peer để truy vấn tài sản

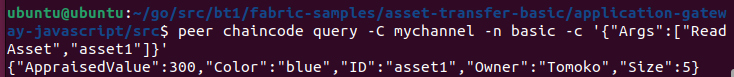
npm install --save-dev --ignore-scripts install-peers



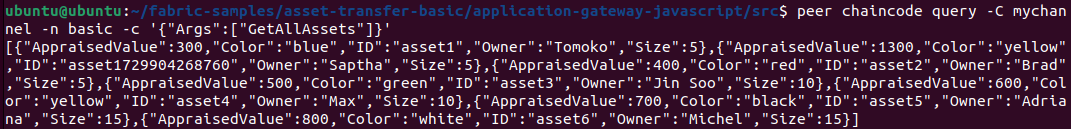
source ~/.bashrc

Truy vấn tài sản

peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAsset","asset1"]}'



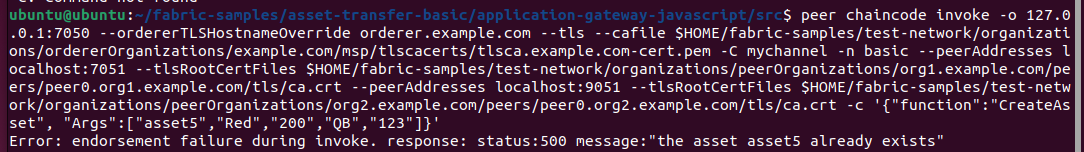
🡪Xem toàn bộ tài sản: peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'



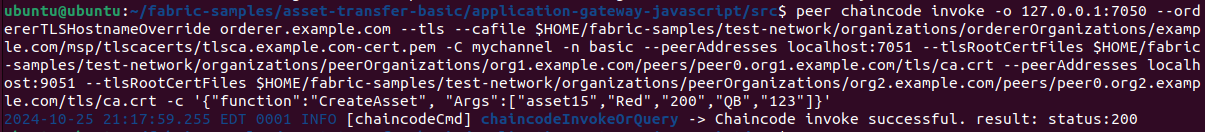
Thêm tài sản

peer chaincode invoke -o 127.0.0.1:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $HOME/go/src/project1/fabric-samples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles $HOME/go/src/project1/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles $HOME/go/src/project1/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"function":"CreateAsset", "Args":["asset20","Red","200","QB","123"]}'

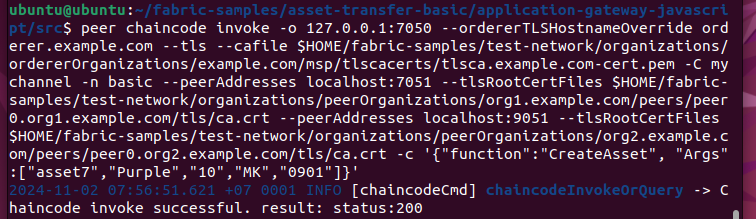
🡪 Nếu tài sản tồn tại thì sẽ có thông báo



🡪 Thêm tài sản thành công

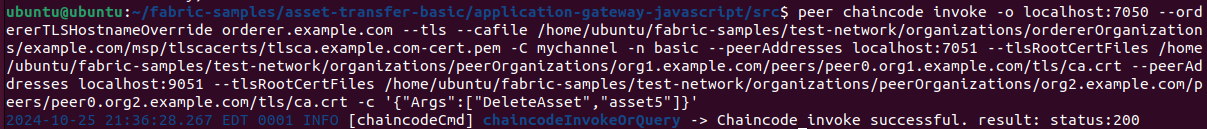


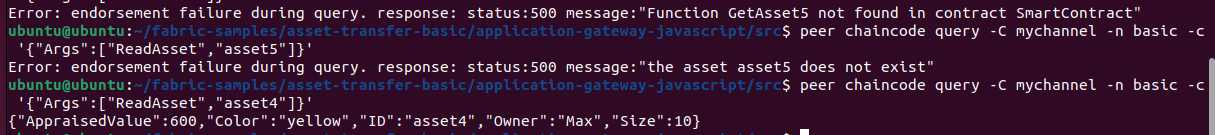
peer chaincode invoke -o 127.0.0.1:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $HOME/fabric-samples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles $HOME/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles $HOME/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"function":"CreateAsset", "Args":["asset7","Purple","10","MK","0901"]}'



Xóa tài sản

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/ubuntu/fabric-samples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"Args":["DeleteAsset","asset5"]}'





Thêm org3

Chuyển thư mục đế vị trí thư mục test-network

cd

cd fabric-samples/test-network

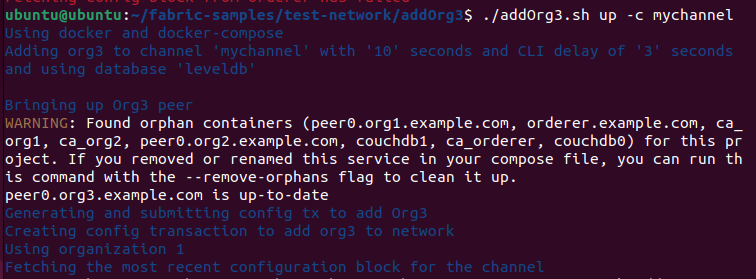
./network.sh down

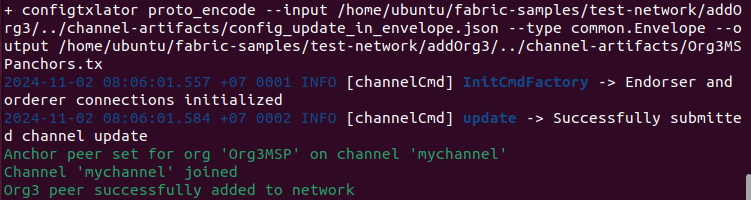
./network.sh up createChannel -c mychannel

./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-go -ccl go

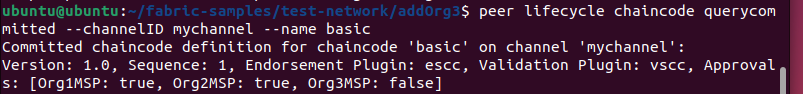
cd addOrg3

./addOrg3.sh up -c mychannel (chỉnh thành tên channel đã tạo)





peer lifecycle chaincode querycommitted --channelID mychannel --name basic



Viết Chaincode trên Hyperledger đã cài đặt

Trong thư mục fabric-sample tạo thư mục mychaincode

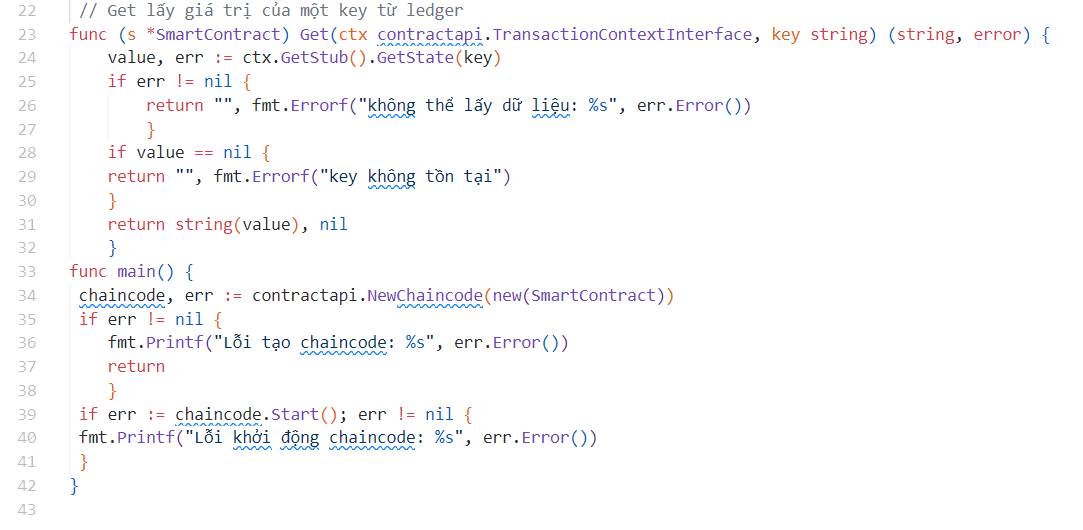
mkdir -p fabric-samples/chaincode/mychaincode (TẠI VỊ TRÍ THƯ MỤC GỐC)

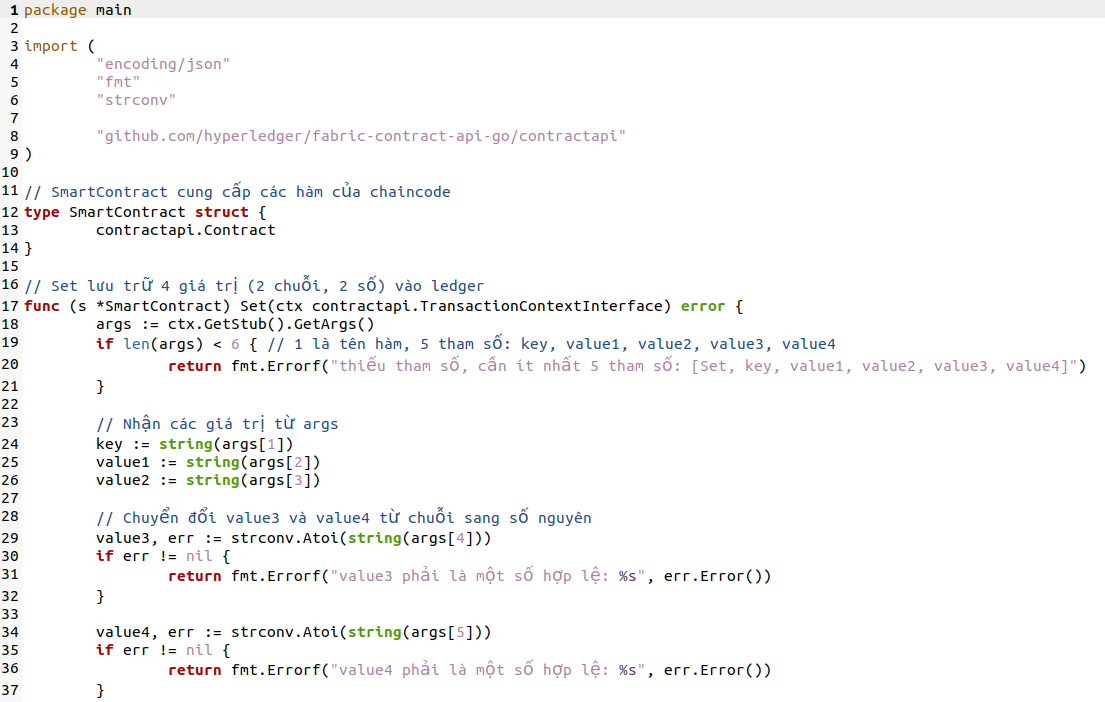
cd fabric-samples/chaincode/mychaincode/

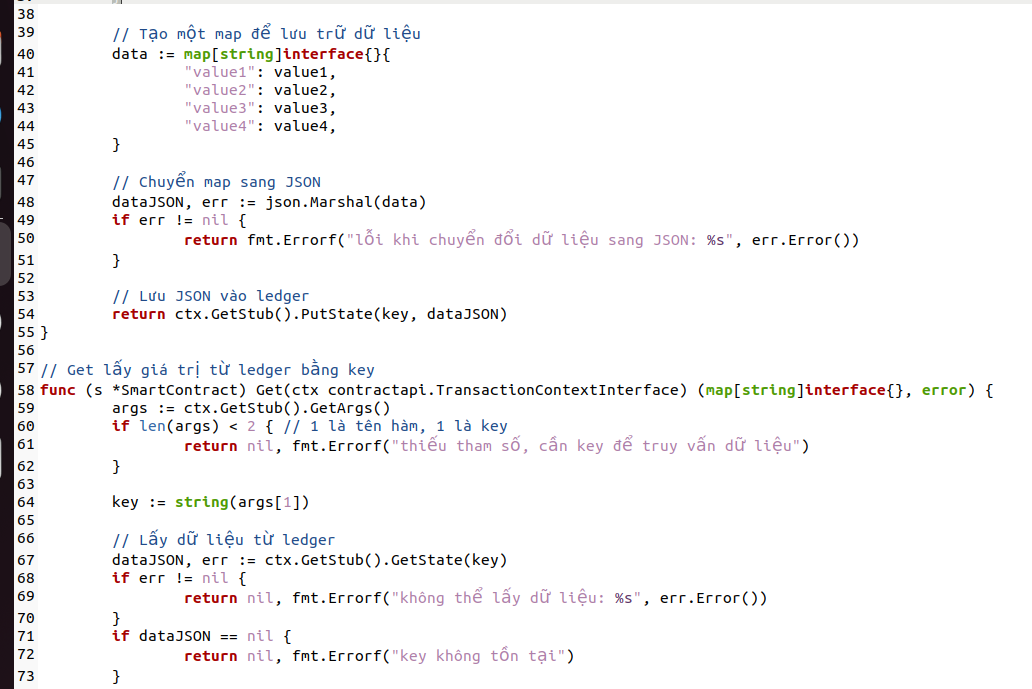
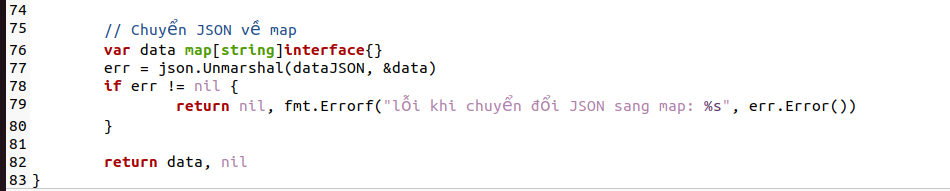
Tạo file mychaincode.go: touch mychaincode.go

Định dạng code trong file mychaincode.go: gofmt -w mychaincode.go





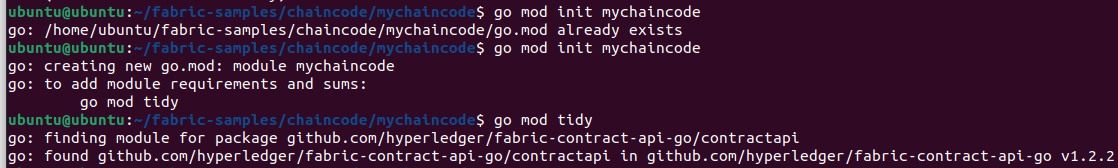
* Chỉnh sửa file mychaincode.go để quản lý thông tin sản phẩm: mã hàng, tên hàng, số lượng đơn giá. [“TH1383”, “Coca”, 10, 15000]



Tại thư mục fabric chạy lệnh: sudo fabric -R 777 (Thêm all quyền cho thư mục fabric)

Di chuyển vào thư mục mychaincode tạo module Gogo mod init mychaincode

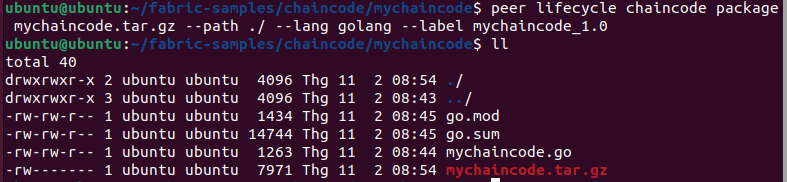
Chạy lệnh sau để tải và quản lý các dependencies cho dự án của bạn:

go mod tidy

Đóng gói chaincode

Lệnh này sẽ tạo file mychaincode.tar.gz cho chaincode của bạn.

peer lifecycle chaincode package mychaincode.tar.gz --path ./ --lang golang --label mychaincode\_1.0

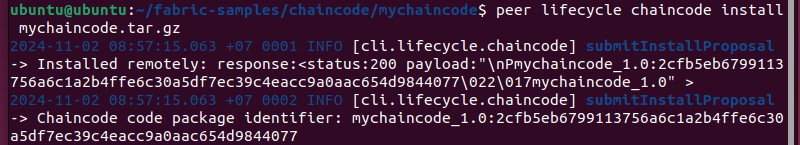


Tiếp theo, cài đặt chaincode lên peer như trước đây: (Tại vị trí ~/fabric-samples/test-network)

./network.sh down

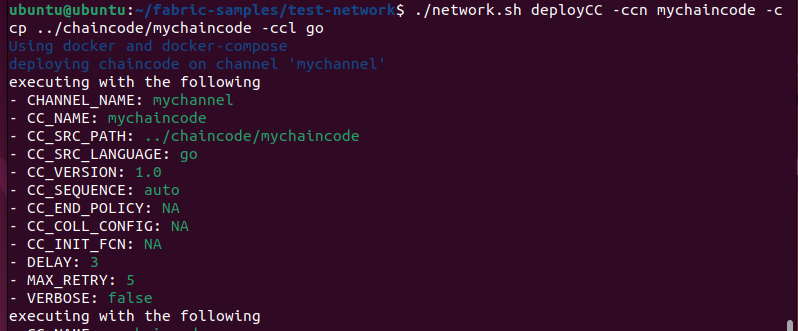
./network.sh up createChannel -c mychannel -ca

peer lifecycle chaincode install mychaincode.tar.gz (chạy ngay vị trí fabric-samples/chaincode/mychaincode chứa file mychaincode.tar.gz )



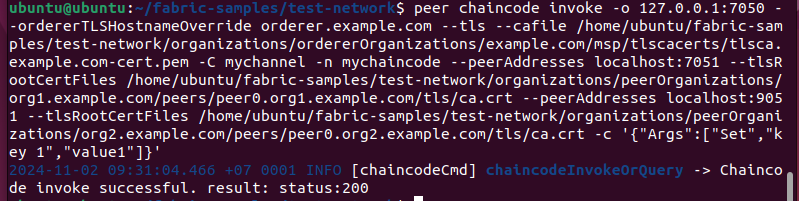
Tại vị trí thư mục test-network chạy lệnh sau:

./network.sh deployCC -ccn mychaincode -ccp ../chaincode/mychaincode -ccl go



Thêm asset mới

peer chaincode invoke -o 127.0.0.1:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/ubuntu/fabric-samples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n mychaincode --peerAddresses localhost:7051 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"Args":["Set","key1","value1"]}'



Lấy giá trị key, value

peer chaincode query -C mychannel -n mychaincode -c '{"Args":["Get","key1"]}'

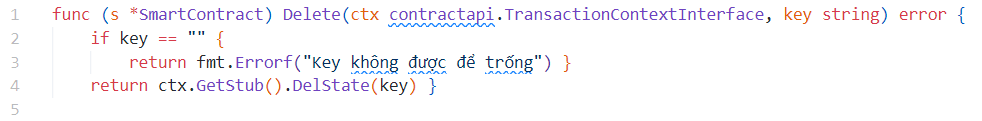


Chuyển đến đường dẫn ~/fabric-samples/chaincode/mychaincode

Chạy lệnh: sudo gedit mychaincode.go (mở file mychaincode.go để chỉnh sửa các phương thức)

Thêm phương thức Delete vào mychaincode.go

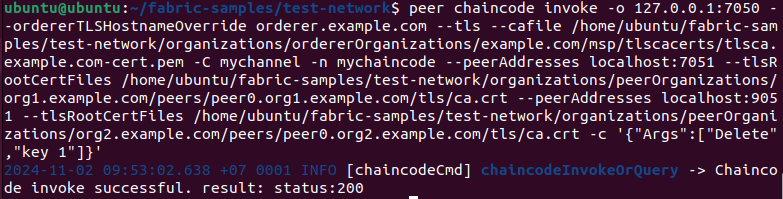
🡪Nhớ deploy (vị trí /fabric-samples/test-network): ./network.sh deployCC -ccn mychaincode -ccp ../chaincode/mychaincode -ccl go



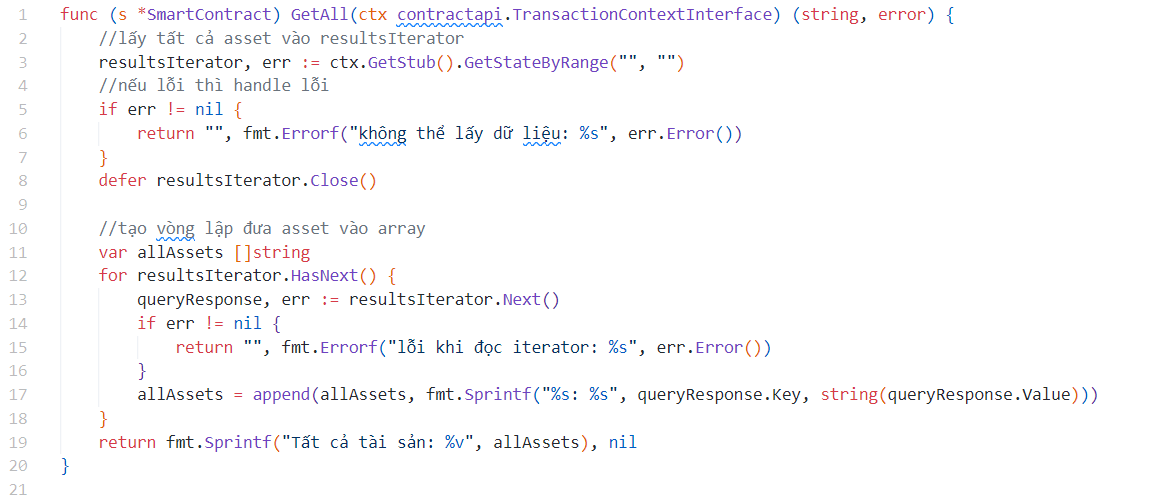
Chạy lệnh: ./network.sh deployCC -ccn mychaincode -ccp ../chaincode/mychaincode -ccl go 🡪Để cập nhật phương thức mới

Sử dụng lệnh để xóa 1 asset

peer chaincode invoke -o 127.0.0.1:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/ubuntu/fabric-samples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n mychaincode --peerAddresses localhost:7051 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"Args":["Delete","key 1"]}'



Thêm phương thức để GetAll: (Mở file mychaincode.go)



🡪Deploy tại vị trí fabric-samples/test-network

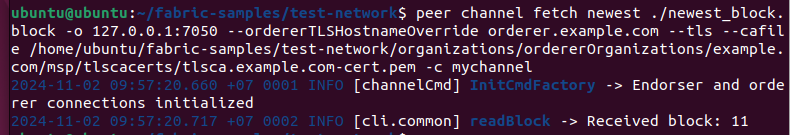
./network.sh deployCC -ccn mychaincode -ccp ../chaincode/mychaincode -ccl go

peer chaincode query -C mychannel -n mychaincode -c '{"Args":["GetAll"]}'



Để xem có tất cả bao nhiêu block

peer channel fetch newest ./newest\_block.block -o 127.0.0.1:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/ubuntu/fabric-samples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem -c mychannel



Để giải mã block

configtxlator proto\_decode --input newest\_block.block --type common.Block --output newest\_block.json

Xem thông tin giao dịch: cat newest\_block.json

Xem thông tin giao dịch: phí gas, mã hash, number\_block.



# Kết nối Metamask – Remix – Ganache

https://github.com/kecci/solidity-truffle-ganache-remix

1. Install Ganache linux (tải file ganache trước)

cd Downloads/

sudo chmod a+x ganache-2.7.1-linux-x86\_64.AppImage

sudo apt-get install fuse libfuse2 (Có thể bị đụng process: sudo kill <process >)

./ganache-2.7.1-linux-x86\_64.AppImage

🡪Chọn Quickstart

2. Install metamask linux🡪extension firefox

Tạo Network mới: Click vào network ở giao diện 🡪 Add a custom network 🡪 Tạo network mới 🡪 Save.

* Network name: Ganache
* Default RPC URL: <HTTP://127.0.0.1:7545>
* Chain ID: 1337
* Currency symbol: ETH

Tạo Account mới: 🡪Select an account🡪Add account🡪 Import account (lấy private key của một account bất kỳ ở Ganache )

3. Remix blockchain

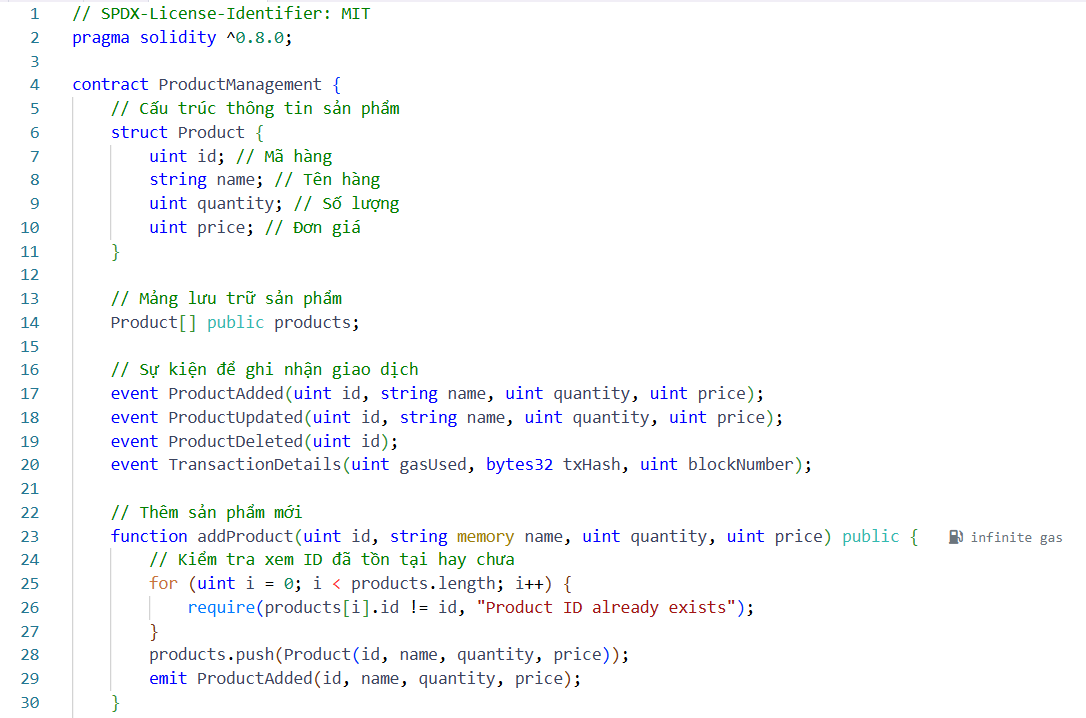
Mở remix 🡪 tạo SmartContract🡪 Deploy & Run Transaction🡪 Environment (WalletConnect)🡪 Click vào WalletConnect 🡪 MetaMask (khi hiện Account và ETH là thành công)

# Code Solidity (Remix)

1. Viết SmartContract quản lý thông tin sản phẩm bao gồm:

Khai báo mảng dữ liệu quản lý thông tin: mã hàng, tên hàng, số lượng, đơn giá.

Viết hàm thêm sản phẩm: [“TH1383”, “Coca”, 10, 15000]

Cho biết thông tin giao dịch: phí gas, mã hash, number\_block của transaction.

# Cài đặt và triển khai Hardhat

Cài đặt môi trường

sudo apt update

sudo apt install curl git

curl -fsSL https://deb.nodesource.com/setup\_22.x | sudo -E bash -

sudo apt-get install -y nodejs

Tạo dự án Hardhat mới

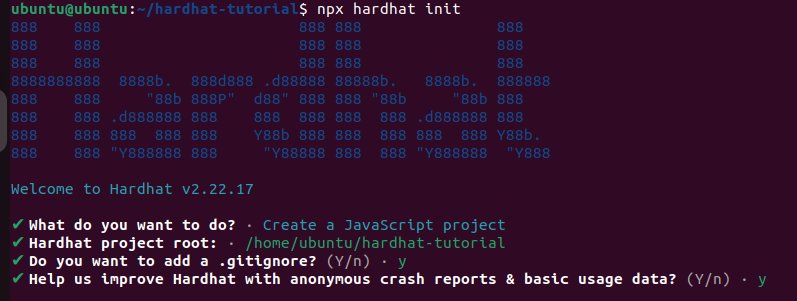
mkdir hardhat-tutorial

cd hardhat-tutorial

npm init (Enter)

npm install --save-dev hardhat

npx hardhat init (Enter)



Deploy SmartContract:

cd ignition/modules

npx hardhat ignition deploy Lock.js

Xem tài khoản: npx hardhat node

Cài đặt Ganache: <https://gist.github.com/GoodnessEzeokafor/e3a2665bb482addbb603269428424967>

Thiết lập mạng: (Thiết lập trong file hardhat-tutorial/hardhat.config.js)

require("@nomicfoundation/hardhat-toolbox");

module.exports = {

solidity: "0.8.27", // hoặc phiên bản đang sử dụng

networks: {

ganache: {

url: "http://127.0.0.1:7545", // URL của Ganache

accounts: ["<PRIVATE\_KEY>"] // Thay <PRIVATE\_KEY> của Ganache

} } };

Deploy: (tại vị trí hardhat-tutorial/ignition/modules)

npx hardhat ignition deploy Lock.js --network ganache



Tạo contract.sol ( Nằm trong thư mục hardhat-tutorial/contracts)

function addProduct(string memory \_name, uint256 \_quantity, uint256 \_price) public {

productCount++;

products[productCount] = Product(productCount, \_name, \_quantity, \_price);

emit ProductAdded(productCount, \_name, \_quantity, \_price);

}

function updateProduct(uint256 \_productId, string memory \_name, uint256 \_quantity, uint256 \_price) public {

require(products[\_productId].productId != 0, "Product does not exist");

products[\_productId] = Product(\_productId, \_name, \_quantity, \_price);

emit ProductUpdated(\_productId, \_name, \_quantity, \_price);

}

function getProduct(uint256 \_productId) public view returns (Product memory) {

require(products[\_productId].productId != 0, "Product does not exist");

return products[\_productId];

} }Tạo folder hardhat-tutorial/scripts

Tạo file deploy.js (

async function main() {

const ProductManager = await ethers.getContractFactory("P

roductManager");

const productManager = await ProductManager.deploy();

await productManager.waitForDeployment();

// Lay dia chi trien khai SC

console.log("ProductManager deployed to:", productManage

r.target);

}

main()

.then(() => process.exit(0))

.catch((error) => {

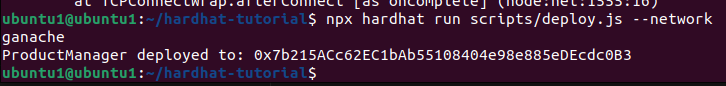
console.error(error);

process.exit(1);

});

Deploy contract:

npx hardhat run scripts/deploy.js --network ganache



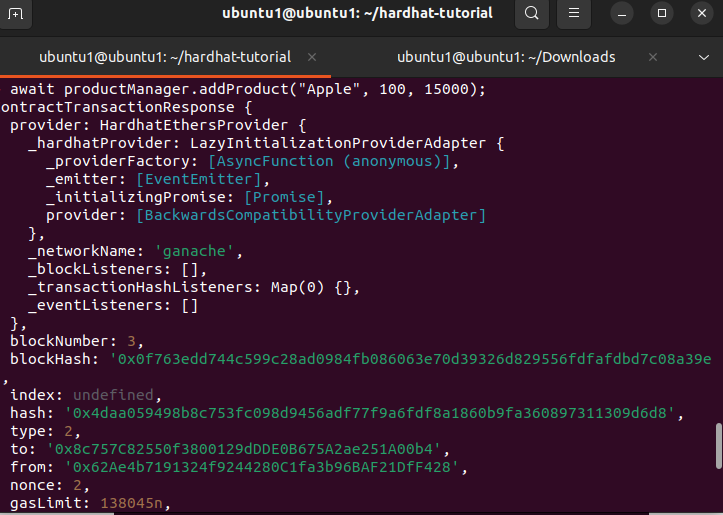
Thực hiện tương tác:

npx hardhat console --network ganache

const ProductManager = await ethers.getContractFactory("ProductManager");

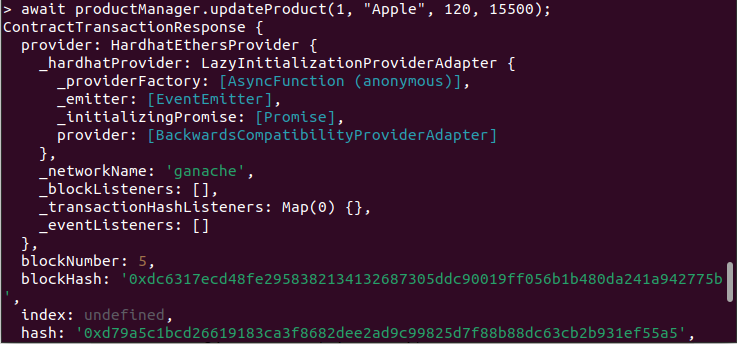
const productManager = await ProductManager.attach("Dia chi contract da deploy");

await productManager.addProduct("Apple", 100, 15000);



Cập nhật:

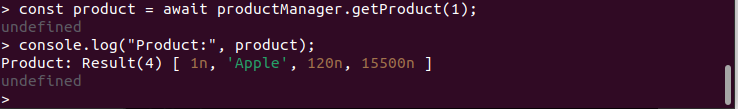
await productManager.updateProduct(1, "Apple", 120, 15500);



Lấy thông tin hàng hóa:

const product = await productManager.getProduct(1);

console.log("Product:", product);



# Xây dựng ứng dụng DAPPs -WEB3 (bỏ phiếu bầu)

Cài curl

sudo apt install curl

Cài wget

Sudo apt install wget

Cài nodejs

# installs fnm (Fast Node Manager)  
curl -fsSL https://fnm.vercel.app/install | bash

# activate fnm  
source ~/.bashrc

# download and install Node.js  
fnm use --install-if-missing 22

Cài ganache

# Cài ngoài desktop cho dễ

cd $HOME/Desktop

Wget <https://github.com/trufflesuite/ganache-ui/releases/download/v2.7.1/ganache-2.7.1-linux-x86_64.AppImage>

Sudo chmod a+x ganache-2.7.1-linux-x86\_64.AppImage

Sudo apt install fuse libfuse2 # Có thể cần

./ganache-2.7.1-linux-x86\_64.AppImage

# Vào setting -> server -> hostname chọn 0.0.0.0

Cài hardhat

mkdir hardhat

cd hardhat

npm init –yes

npm install --save-dev hardhat

npx hardhat

#Chọn js

# Ubuntu: Viết và triển khai hợp đồng thông minh

# Trong thư mục contracts/, tạo file Voting.sol có code:  
pragma solidity ^0.8.18;  
contract Voting {  
 mapping(string => uint256) public votes;  
 function vote(string memory candidate) public {  
 votes[candidate]++;  
 }  
 function getVotes(string memory candidate) public view returns (uint256) {  
 return votes[candidate];  
 }  
}

# Trong thư mục hardhat  
npx hardhat compile

# Về thư mục hardhat sau đó edit file config  
require("@nomiclabs/hardhat-ethers");  
module.exports = {  
 networks: {  
 ganache: {  
 url: "http://127.0.0.1:7545",  
 accounts: ["0x<Private\_Key\_From\_Ganache>"], // Thay private key của Ganache  
 },  
 },  
 solidity: "0.8.28",  
};

# Tạo deploy.js(để dễ quản lý tạo folder scripts -> cd vào scripts)  
touch deploy.js

Gedit deploy.js

# ghi code vào  
const hre = require("hardhat");  
async function main() {  
 const Voting = await hre.ethers.getContractFactory("Voting");  
 const voting = await Voting.deploy();  
 await voting.deployed();  
 console.log("Voting contract deployed to:", voting.address);  
}

main().catch((error) => {  
 console.error(error);  
 process.exitCode = 1;  
});

# Triển khai  
npm install --save-dev @nomiclabs/hardhat-ethers ethers # Tránh lỗi  
npx hardhat run scripts/deploy.js --network ganache

# Windows 10: Tương tác với hợp đồng

# Tạo thư mục vote ngoài Desktop cho dễ quản lý  
-> tạo file html  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Voting DApp</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<h1>Voting DApp</h1>

<div class="form">

<label for="candidate">Enter Candidate Name:</label>

<input type="text" id="candidate" placeholder="e.g., Alice">

<button id="voteBtn">Vote</button>

</div>

<div id="message"></div>

<hr>

<div class="results">

<h2>Check Votes</h2>

<label for="checkCandidate">Enter Candidate Name:</label>

<input type="text" id="checkCandidate" placeholder="e.g., Alice">

<button id="checkVotesBtn">Check Votes</button>

<p id="votesResult"></p>

</div>

</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/web3/1.8.2/web3.min.js"></script>

<script src="script.js"></script>

</body>

</html>  
-> style.css (optional)  
body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.container {

background: white;

border-radius: 8px;

padding: 20px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

width: 400px;

}

h1, h2 {

text-align: center;

color: #333;

}

label {

display: block;

margin: 10px 0 5px;

}

input {

width: 100%;

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

width: 100%;

padding: 10px;

background-color: #007BFF;

border: none;

color: white;

font-size: 16px;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

#message, #votesResult {

margin-top: 10px;

font-size: 14px;

color: green;

text-align: center;

}

-> script.js  
ip của ubuntu lấy bằng cách vào ubuntu chạy lệnh ip a -> 192.168.xxx.xxx

// Khởi tạo Web3

const GANACHE\_NODE\_URL = "http://<Ubuntu\_IP>:7545"; // Thay <Ubuntu\_IP> bằng IP Ubuntu

const web3 = new Web3(GANACHE\_NODE\_URL);

// ABI và địa chỉ hợp đồng (cần thay bằng ABI và địa chỉ thực tế)

const abi = [

{

"inputs": [{ "internalType": "string", "name": "candidate", "type": "string" }],

"name": "vote",

"outputs": [],

"stateMutability": "nonpayable",

"type": "function"

},

{

"inputs": [{ "internalType": "string", "name": "candidate", "type": "string" }],

"name": "getVotes",

"outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }],

"stateMutability": "view",

"type": "function"

}

];

const contractAddress = "0xYourContractAddress"; // Thay bằng địa chỉ hợp đồng thực tế

const votingContract = new web3.eth.Contract(abi, contractAddress);

// Lấy tài khoản đầu tiên từ Ganache

let account;

web3.eth.getAccounts().then(accounts => {

account = accounts[0];

});

// Xử lý sự kiện nút "Vote"

document.getElementById("voteBtn").addEventListener("click", async () => {

const candidate = document.getElementById("candidate").value;

if (candidate) {

try {

await votingContract.methods.vote(candidate).send({ from: account });

document.getElementById("message").innerText = `Successfully voted for ${candidate}`;

} catch (error) {

document.getElementById("message").innerText = `Error: ${error.message}`;

}

} else {

document.getElementById("message").innerText = "Please enter a candidate name.";

}

});

// Xử lý sự kiện nút "Check Votes"

document.getElementById("checkVotesBtn").addEventListener("click", async () => {

const candidate = document.getElementById("checkCandidate").value;

if (candidate) {

try {

const votes = await votingContract.methods.getVotes(candidate).call();

document.getElementById("votesResult").innerText = `${candidate} has ${votes} votes.`;

} catch (error) {

document.getElementById("votesResult").innerText = `Error: ${error.message}`;

}

} else {

document.getElementById("votesResult").innerText = "Please enter a candidate name.";

}

});

# Ảnh test

