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 -y

sudo apt-get install docker-compose -y

sudo systemctl start docker

sudo systemctl enable docker

sudo apt install jq

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

Thêm quyền người dùng vào group docker

sudo usermod -aG docker ${USER}

sudo apt-get update

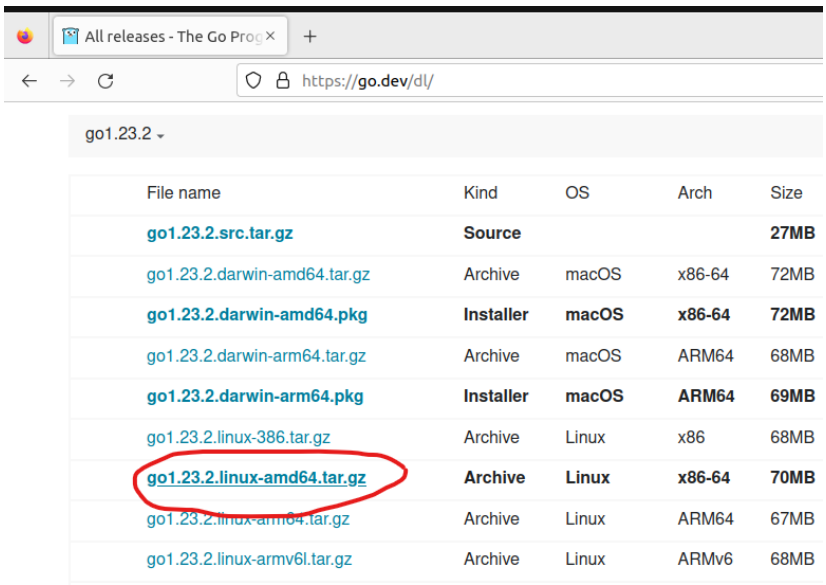
kiểm tra phiên bản nodejs: nodejs --version

sudo npm install npm@6.14.17 -g

Cài đặt ngôn ngữ Go

sudo rm -rf /usr/local/go

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



File name	Kind	OS	Arch	Size
go1.23.2.src.tar.gz	**Source**			**27MB**
go1.23.2.darwin-amd64.tar.gz	Archive	macOS	x86-64	72MB
go1.23.2.darwin-amd64.pkg	**Installer**	**macOS**	**x86-64**	**72MB**
go1.23.2.darwin-arm64.tar.gz	Archive	macOS	ARM64	68MB
go1.23.2.darwin-arm64.pkg	**Installer**	**macOS**	**ARM64**	**69MB**
go1.23.2.linux-386.tar.gz	Archive	Linux	x86	68MB
go1.23.2.linux-amd64.tar.gz	**Archive**	**Linux**	**x86-64**	**70MB**
go1.23.2.linux-arm64.tar.gz	Archive	Linux	ARM64	67MB
go1.23.2.linux-armv6l.tar.gz	Archive	Linux	ARMv6	68MB

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)

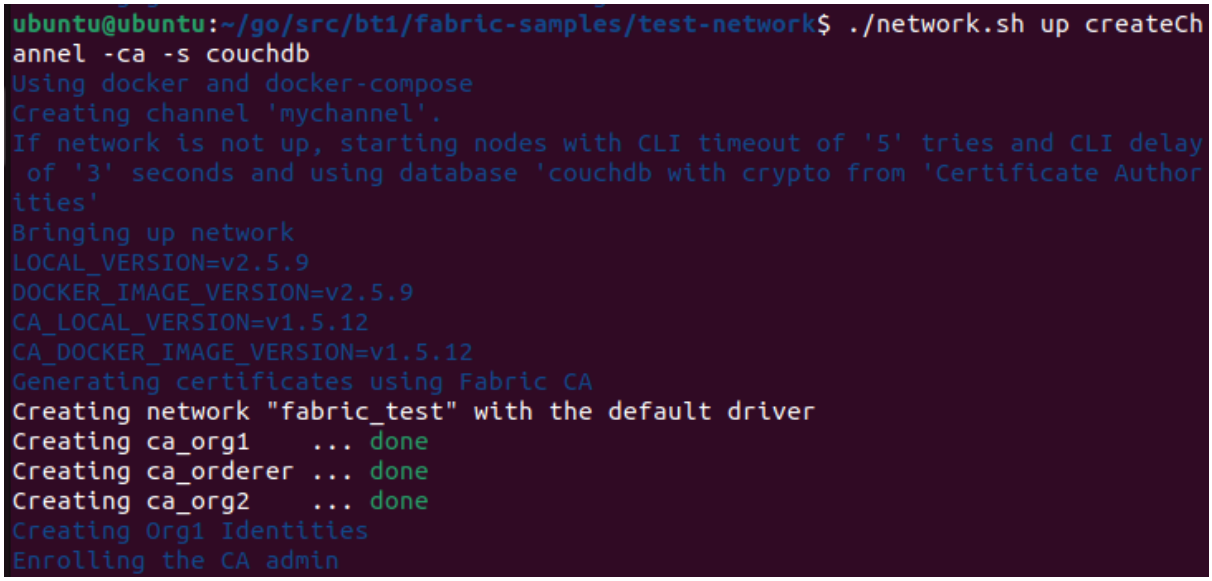
cd fabric-samples/test-network/

./network.sh down

./network.sh up

./network.sh down

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



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

```
ubuntu@ubuntu:~/go/src/bt1/fabric-samples/test-network$ ./network.sh deployCC -c
cn basic -ccp ../asset-transfer-basic/chaincode-go -ccl go
Using docker and docker-compose
deploying chaincode on channel 'mychannel'
executing with the following
- CHANNEL_NAME: mychannel
- CC_NAME: basic
- CC_SRC_PATH: ../asset-transfer-basic/chaincode-go
- CC_SRC_LANGUAGE: go
- CC_VERSION: 1.0
- CC_SEQUENCE: auto
- CC_END_POLICY: NA
- CC_COLL_CONFIG: NA
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
executing with the following
```

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)

```
ubuntu@ubuntu:~/go/src/bt1/fabric-samples/asset-transfer-basic/application-gatew
ay-javascript/src$ node app.js
channelName:       mychannel
chaincodeName:     basic
mspId:             Org1MSP
cryptoPath:        /home/ubuntu/go/src/bt1/fabric-samples/test-network/organizat
ions/peerOrganizations/org1.example.com
keyDirectoryPath:  /home/ubuntu/go/src/bt1/fabric-samples/test-network/organizat
ions/peerOrganizations/org1.example.com/users/User1@org1.example.com/msp/keystor
e
certDirectoryPath: /home/ubuntu/go/src/bt1/fabric-samples/test-network/organizat
ions/peerOrganizations/org1.example.com/users/User1@org1.example.com/msp/signcer
ts
tlsCertPath:       /home/ubuntu/go/src/bt1/fabric-samples/test-network/organizat
ions/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
peerEndpoint:      localhost:7051
peerHostAlias:     peer0.org1.example.com

--> Submit Transaction: InitLedger, function creates the initial set of assets o
n the ledger
*** Transaction committed successfully
```

Cài đặt peer để truy vấn tài sản

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

```
pt/src$ peer
Usage:
  peer [command]

Available Commands:
  chaincode   Operate a chaincode: install|instantiate|invoke|package|query|sign
package|upgrade|list.
  channel     Operate a channel: create|fetch|join|joinbysnapshot|joinbysnapshot
status|list|update|signconfigtx|getinfo.
  completion  Generate the autocompletion script for the specified shell
  help        Help about any command
  lifecycle   Perform _lifecycle operations
  node        Operate a peer node: start|reset|rollback|pause|resume|rebuild-dbs
|unjoin|upgrade-dbs.
  snapshot    Manage snapshot requests: submitrequest|cancelrequest|listpending
  version     Print fabric peer version.

Flags:
  -h, --help   help for peer

Use "peer [command] --help" for more information about a command.
```

source ~/.bashrc

Truy vấn tài sản

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

```
ubuntu@ubuntu:~/go/src/bt1/fabric-samples/asset-transfer-basic/application-gatew
ay-javascript/src$ peer chaincode query -C mychannel -n basic -c '{"Args":["Read
Asset","asset1"]}'
{"AppraisedValue":300,"Color":"blue","ID":"asset1","Owner":"Tomoko","Size":5}
```

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

```
ubuntu@ubuntu:~/fabric-samples/asset-transfer-basic/application-gateway-javascript/src$ peer chaincode query -C mychan
nel -n basic -c '{"Args":["GetAllAssets"]}'
[{"AppraisedValue":300,"Color":"blue","ID":"asset1","Owner":"Tomoko","Size":5},{"AppraisedValue":1300,"Color":"yellow"
,"ID":"asset1729904268760","Owner":"Saptha","Size":5},{"AppraisedValue":400,"Color":"red","ID":"asset2","Owner":"Brad"
,"Size":5},{"AppraisedValue":500,"Color":"green","ID":"asset3","Owner":"Jin Soo","Size":10},{"AppraisedValue":600,"Col
or":"yellow","ID":"asset4","Owner":"Max","Size":10},{"AppraisedValue":700,"Color":"black","ID":"asset5","Owner":"Adria
na","Size":15},{"AppraisedValue":800,"Color":"white","ID":"asset6","Owner":"Michel","Size":15}]
```

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

```
ubuntu@ubuntu:~/fabric-samples/asset-transfer-basic/application-gateway-javascript/src$ peer chaincode invoke -o 127.0
.0.1:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $HOME/fabric-samples/test-network/organizati
ons/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n basic --peerAddresses l
ocalhost:7051 --tlsRootCertFiles $HOME/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/pe
ers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles $HOME/fabric-samples/test-netw
ork/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"function":"CreateAs
set", "Args":["asset5","Red","200","QB","123"]}'
Error: endorsement failure during invoke. response: status:500 message:"the asset asset5 already exists"
```

→ Thêm tài sản thành công

```
ubuntu@ubuntu:~/fabric-samples/asset-transfer-basic/application-gateway-javascript/src$ peer chaincode invoke -o 127.0.0.1:7050 --ord
ererTLSHostnameOverride orderer.example.com --tls --cafile $HOME/fabric-samples/test-network/organizations/ordererOrganizations/examp
le.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 localh
ost:9051 --tlsRootCertFiles $HOME/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.examp
le.com/tls/ca.crt -c '{"function":"CreateAsset", "Args":["asset15","Red","200","QB","123"]}'
2024-10-25 21:17:59.255 EDT 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
```

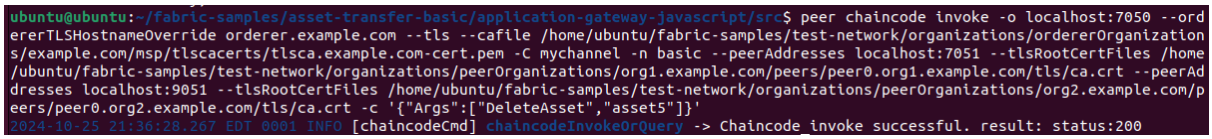
peer chaincode invoke -o 127.0.0.1:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $HOME/fabric-samples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com -cert.pem -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles $HOME/fabric-samples/test-network/organizations/peerOrganizations/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"]}'

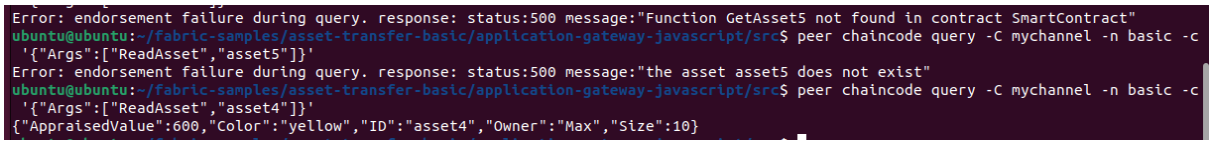
```
ubuntu@ubuntu:~/fabric-samples/asset-transfer-basic/application-gateway-javascri
pt/src$ peer chaincode invoke -o 127.0.0.1:7050 --ordererTLSHostnameOverride ord
erer.example.com --tls --cafile $HOME/fabric-samples/test-network/organizations/
ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C my
channel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles $HOME/fabric-
samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer
0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles
$HOME/fabric-samples/test-network/organizations/peerOrganizations/org2.example.c
om/peers/peer0.org2.example.com/tls/ca.crt -c '{"function":"CreateAsset", "Args"
:["asset7","Purple","10","MK","0901"]}'
2024-11-02 07:56:51.621 +07 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> C
haincode invoke successful. result: status:200
```

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

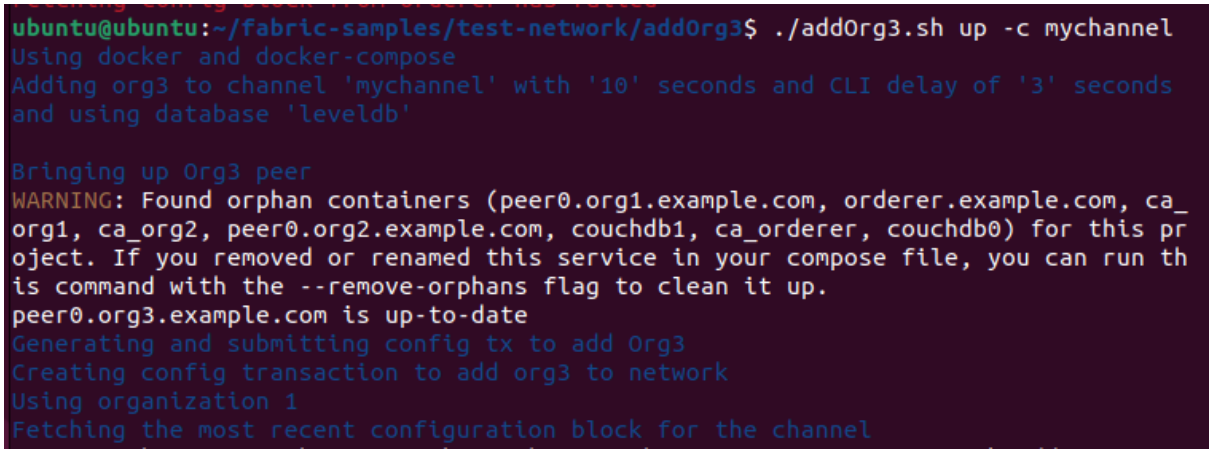
./network.sh down

./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)



```
+ configtxlator proto_encode --input /home/ubuntu/fabric-samples/test-network/addO
rg3/../channel-artifacts/config_update_in_envelope.json --type common.Envelope --o
utput /home/ubuntu/fabric-samples/test-network/addOrg3/../channel-artifacts/Org3MS
Panchors.tx
2024-11-02 08:06:01.557 +07 0001 INFO [channelCmd] InitCmdFactory -> Endorser and
orderer connections initialized
2024-11-02 08:06:01.584 +07 0002 INFO [channelCmd] update -> Successfully submitte
d channel update
Anchor peer set for org 'Org3MSP' on channel 'mychannel'
Channel 'mychannel' joined
Org3 peer successfully added to network
```

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

```
ubuntu@ubuntu:~/fabric-samples/test-network/addOrg3$ peer lifecycle chaincode querycom
mitted --channelID mychannel --name basic
Committed chaincode definition for chaincode 'basic' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approval
s: [Org1MSP: true, Org2MSP: true, Org3MSP: false]
```

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

```go
package main
import (
    "fmt"
    "github.com/hyperledger/fabric-contract-api-go/contractapi"
)

// SmartContract cung cấp các hàm của chaincode
type SmartContract struct {
    contractapi.Contract
}

// InitLedger khởi tạo ledger với dữ liệu mẫu
func (s *SmartContract) InitLedger(ctx contractapi.TransactionContextInterface) error {
    return nil
}

// Set lưu trữ một cặp key-value vào ledger
func (s *SmartContract) Set(ctx contractapi.TransactionContextInterface, key string, value string) error {
    return ctx.GetStub().PutState(key, []byte(value))
    }

```

```
22    // Get lấy giá trị của một key từ ledger
23   func (s *SmartContract) Get(ctx contractapi.TransactionContextInterface, key string) (string, error) {
24       value, err := ctx.GetStub().GetState(key)
25       if err != nil {
26           return "", fmt.Errorf("không thể lấy dữ liệu: %s", err.Error())
27           }
28       if value == nil {
29       return "", fmt.Errorf("key không tồn tại")
30       }
31       return string(value), nil
32       }
33   func main() {
34    chaincode, err := contractapi.NewChaincode(new(SmartContract))
35    if err != nil {
36       fmt.Printf("Lỗi tạo chaincode: %s", err.Error())
37       return
38       }
39    if err := chaincode.Start(); err != nil {
40    fmt.Printf("Lỗi khởi động chaincode: %s", err.Error())
41    }
42   }
43
```

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

```go
package main

import (
        "encoding/json"
        "fmt"
        "strconv"

        "github.com/hyperledger/fabric-contract-api-go/contractapi"
)

// SmartContract cung cấp các hàm của chaincode
type SmartContract struct {
        contractapi.Contract
}

// Set lưu trữ 4 giá trị (2 chuỗi, 2 số) vào ledger
func (s *SmartContract) Set(ctx contractapi.TransactionContextInterface) error {
        args := ctx.GetStub().GetArgs()
        if len(args) < 6 { // 1 là tên hàm, 5 tham số: key, value1, value2, value3, value4
                return fmt.Errorf("thiếu tham số, cần ít nhất 5 tham số: [Set, key, value1, value2, value3, value4]")
        }

        // Nhận các giá trị từ args
        key := string(args[1])
        value1 := string(args[2])
        value2 := string(args[3])

        // Chuyển đổi value3 và value4 từ chuỗi sang số nguyên
        value3, err := strconv.Atoi(string(args[4]))
        if err != nil {
                return fmt.Errorf("value3 phải là một số hợp lệ: %s", err.Error())
        }

        value4, err := strconv.Atoi(string(args[5]))
        if err != nil {
                return fmt.Errorf("value4 phải là một số hợp lệ: %s", err.Error())
        }
```

```
38
39          // Tạo một map để lưu trữ dữ liệu
40          data := map[string]interface{}{
41                  "value1": value1,
42                  "value2": value2,
43                  "value3": value3,
44                  "value4": value4,
45          }
46
47          // Chuyển map sang JSON
48          dataJSON, err := json.Marshal(data)
49          if err != nil {
50                  return fmt.Errorf("lỗi khi chuyển đổi dữ liệu sang JSON: %s", err.Error())
51          }
52
53          // Lưu JSON vào ledger
54          return ctx.GetStub().PutState(key, dataJSON)
55  }
56
57  // Get lấy giá trị từ ledger bằng key
58  func (s *SmartContract) Get(ctx contractapi.TransactionContextInterface) (map[string]interface{}, error) {
59          args := ctx.GetStub().GetArgs()
60          if len(args) < 2 { // 1 là tên hàm, 1 là key
61                  return nil, fmt.Errorf("thiếu tham số, cần key để truy vấn dữ liệu")
62          }
63
64          key := string(args[1])
65
66          // Lấy dữ liệu từ ledger
67          dataJSON, err := ctx.GetStub().GetState(key)
68          if err != nil {
69                  return nil, fmt.Errorf("không thể lấy dữ liệu: %s", err.Error())
70          }
71          if dataJSON == nil {
72                  return nil, fmt.Errorf("key không tồn tại")
73          }
74
75          // Chuyển JSON về map
76          var data map[string]interface{}
77          err = json.Unmarshal(dataJSON, &data)
78          if err != nil {
79                  return nil, fmt.Errorf("lỗi khi chuyển đổi JSON sang map: %s", err.Error())
80          }
81
82          return data, nil
83  }
```

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 Go

go mod init mychaincode

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

go mod tidy

```
ubuntu@ubuntu:~/fabric-samples/chaincode/mychaincode$ go mod init mychaincode
go: /home/ubuntu/fabric-samples/chaincode/mychaincode/go.mod already exists
ubuntu@ubuntu:~/fabric-samples/chaincode/mychaincode$ go mod init mychaincode
go: creating new go.mod: module mychaincode
go: to add module requirements and sums:
        go mod tidy
ubuntu@ubuntu:~/fabric-samples/chaincode/mychaincode$ go mod tidy
go: finding module for package github.com/hyperledger/fabric-contract-api-go/contractapi
go: found github.com/hyperledger/fabric-contract-api-go/contractapi in github.com/hyperledger/fabric-contract-api-go v1.2.2
```

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

```
ubuntu@ubuntu:~/fabric-samples/chaincode/mychaincode$ peer lifecycle chaincode package
 mychaincode.tar.gz --path ./ --lang golang --label mychaincode_1.0
ubuntu@ubuntu:~/fabric-samples/chaincode/mychaincode$ ll
total 40
drwxrwxr-x 2 ubuntu ubuntu  4096 Thg 11  2 08:54 ./
drwxrwxr-x 3 ubuntu ubuntu  4096 Thg 11  2 08:43 ../
-rw-rw-r-- 1 ubuntu ubuntu  1434 Thg 11  2 08:45 go.mod
-rw-rw-r-- 1 ubuntu ubuntu 14744 Thg 11  2 08:45 go.sum
-rw-rw-r-- 1 ubuntu ubuntu  1263 Thg 11  2 08:44 mychaincode.go
-rw------- 1 ubuntu ubuntu  7971 Thg 11  2 08:54 mychaincode.tar.gz
```

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)

```
ubuntu@ubuntu:~/fabric-samples/chaincode/mychaincode$ peer lifecycle chaincode install
 mychaincode.tar.gz
2024-11-02 08:57:15.063 +07 0001 INFO [cli.lifecycle.chaincode] submitInstallProposal
-> Installed remotely: response:<status:200 payload:"\nPmychaincode_1.0:2cfb5eb6799113
756a6c1a2b4ffe6c30a5df7ec39c4eacc9a0aac654d9844077\022\017mychaincode_1.0" >
2024-11-02 08:57:15.063 +07 0002 INFO [cli.lifecycle.chaincode] submitInstallProposal
-> Chaincode code package identifier: mychaincode_1.0:2cfb5eb6799113756a6c1a2b4ffe6c30
a5df7ec39c4eacc9a0aac654d9844077
```

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

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

```
ubuntu@ubuntu:~/fabric-samples/test-network$ ./network.sh deployCC -ccn mychaincode -c
cp ../chaincode/mychaincode -ccl go
Using docker and docker-compose
deploying chaincode on channel 'mychannel'
executing with the following
- CHANNEL_NAME: mychannel
- CC_NAME: mychaincode
- CC_SRC_PATH: ../chaincode/mychaincode
- CC_SRC_LANGUAGE: go
- CC_VERSION: 1.0
- CC_SEQUENCE: auto
- CC_END_POLICY: NA
- CC_COLL_CONFIG: NA
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
executing with the following
```

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"]}'

```
ubuntu@ubuntu:~/fabric-samples/test-network$ peer chaincode invoke -o 127.0.0.1:7050 -
-ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/ubuntu/fabric-sam
ples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.
example.com-cert.pem -C mychannel -n mychaincode --peerAddresses localhost:7051 --tlsR
ootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/
org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:905
1 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrgani
zations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"Args":["Set","k
ey 1","value1"]}'
2024-11-02 09:31:04.466 +07 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chainco
de invoke successful. result: status:200
```

Lấy giá trị key, value

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

```
ubuntu@ubuntu:~/fabric-samples/test-network$ peer chaincode query -C mychannel -n mych
aincode -c '{"Args":["Get","key1"]}'
value1
```

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

```
1  func (s *SmartContract) Delete(ctx contractapi.TransactionContextInterface, key string) error {
2      if key == "" {
3          return fmt.Errorf("Key không được để trống") }
4      return ctx.GetStub().DelState(key) }
5
```

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"]}'

```
ubuntu@ubuntu:~/fabric-samples/test-network$ peer chaincode invoke -o 127.0.0.1:7050 -
-ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/ubuntu/fabric-sam
ples/test-network/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlsca.
example.com-cert.pem -C mychannel -n mychaincode --peerAddresses localhost:7051 --tlsR
ootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrganizations/
org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:905
1 --tlsRootCertFiles /home/ubuntu/fabric-samples/test-network/organizations/peerOrgani
zations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"Args":["Delete"
,"key 1"]}'
2024-11-02 09:53:02.638 +07 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chainco
de invoke successful. result: status:200
```

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

```go
func (s *SmartContract) GetAll(ctx contractapi.TransactionContextInterface) (string, error) {
    //lấy tất cả asset vào resultsIterator
    resultsIterator, err := ctx.GetStub().GetStateByRange("", "")
    //nếu lỗi thì handle lỗi
    if err != nil {
        return "", fmt.Errorf("không thể lấy dữ liệu: %s", err.Error())
    }
    defer resultsIterator.Close()

    //tạo vòng lập đưa asset vào array
    var allAssets []string
    for resultsIterator.HasNext() {
        queryResponse, err := resultsIterator.Next()
        if err != nil {
            return "", fmt.Errorf("lỗi khi đọc iterator: %s", err.Error())
        }
        allAssets = append(allAssets, fmt.Sprintf("%s: %s", queryResponse.Key, string(queryResponse.Value)))
    }
    return fmt.Sprintf("Tất cả tài sản: %v", allAssets), nil
}

```

→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"]}'

```
ubuntu@ubuntu:~/fabric-samples/test-network$ peer chaincode query -C mychannel -n mycha
incode -c '{"Args":["GetAll"]}'
Tất cả tài sản: [key1: value1 key5: ABCD123]
```

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

```
ubuntu@ubuntu:~/fabric-samples/test-network$ peer channel fetch newest ./newest_block.
block -o 127.0.0.1:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafil
e /home/ubuntu/fabric-samples/test-network/organizations/ordererOrganizations/example.
com/msp/tlscacerts/tlsca.example.com-cert.pem -c mychannel
2024-11-02 09:57:20.660 +07 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orde
rer connections initialized
2024-11-02 09:57:20.717 +07 0002 INFO [cli.common] readBlock -> Received block: 11
```

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.

```
ubuntu@ubuntu:~/fabric-samples/test-network$ configtxlator proto_decode --input newest
_block.block --type common.Block --output newest_block.json
ubuntu@ubuntu:~/fabric-samples/test-network$ cat newest_block.json
{
        "data": {
                "data": [
                        {
                                "payload": {
                                        "data": {
                                                "actions": [
                                                        {
                                                                "header": {
                                                                        "creator": {
                                                                                "id_by
tes": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUNvekNDQWtxZ0F3SUJBZ0lVTTMza0p5Z3ZxOUJ
GdkZwRGlib3RpbVpEc1RFd0NnWUlLb1pJemowRUF3SXcKY0RFTE1Ba0dBMVVFQmhNQ1ZWTXhGekFWQmdOVkJBZ
1REazV2Y25Sb0lFTmhjbTlzYVc1aE1ROHdEUVlEVlFRSApFd1pFZFhKb1lXMHhHVEFYQmdOVkJBb1RFRzl5Wnp
FdVpYaGhiWEJzWlM1amIyMHhIREFhQmdOVkJBTVRFMk5oCkxtOXlaekV1WlhoaGJYQnNaUzVqYjIwd0hoY05Na
lF4TVRBeU1ERTBOVEF3V2hjTk1qVXhNVEF5TURFMU1EQXcKV2pCZ01Rc3dDUVlEVlFRR0V3SlZVekVYTUJVR0E
xVUVDQk1PVG05eWRHZ2dRMkZ5YjJ4cGJtRXhGREFTQmdOVgpCQW9UQzBoNWNHVnliR1ZrWjJWeU1RNHdEQVlEV
lFRTEV3VmhaRzFwYmpFU01CQUdBMVVFQXhNSmIzSm5NV0ZrCmJXbHVNRmt3RXdZSEtvWkl6ajBDQVFZSUtvWkl
```

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.

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract ProductManagement {
    // Cấu trúc thông tin sản phẩm
    struct Product {
        uint id; // Mã hàng
        string name; // Tên hàng
        uint quantity; // Số lượng
        uint price; // Đơn giá
    }

    // Mảng lưu trữ sản phẩm
    Product[] public products;

    // Sự kiện để ghi nhận giao dịch
    event ProductAdded(uint id, string name, uint quantity, uint price);
    event ProductUpdated(uint id, string name, uint quantity, uint price);
    event ProductDeleted(uint id);
    event TransactionDetails(uint gasUsed, bytes32 txHash, uint blockNumber);

    // Thêm sản phẩm mới
    function addProduct(uint id, string memory name, uint quantity, uint price) public {    infinite gas
        // Kiểm tra xem ID đã tồn tại hay chưa
        for (uint i = 0; i < products.length; i++) {
            require(products[i].id != id, "Product ID already exists");
        }
        products.push(Product(id, name, quantity, price));
        emit ProductAdded(id, name, quantity, price);
    }
```

```
31
32      // Sửa thông tin sản phẩm
33      function updateProduct(uint id, string memory name, uint quantity, uint price) public {    infinite gas
34          bool found = false;
35          for (uint i = 0; i < products.length; i++) {
36              if (products[i].id == id) {
37                  products[i].name = name;
38                  products[i].quantity = quantity;
39                  products[i].price = price;
40                  found = true;
41                  emit ProductUpdated(id, name, quantity, price);
42                  break;
43              }
44          }
45          require(found, "Product ID does not exist");
46      }
47
48      // Xóa sản phẩm
49      function deleteProduct(uint id) public {
50          bool found = false;    infinite gas
51          for (uint i = 0; i < products.length; i++) {
52              if (products[i].id == id) {
53                  products[i] = products[products.length - 1]; // Di chuyển phần tử cuối cùng vào vị trí cần xóa
54                  products.pop(); // Xóa phần tử cuối
55                  found = true;
56                  emit ProductDeleted(id);
57                  break;
58              }
59          }
60          require(found, "Product ID does not exist");
61      }
62
63      // Xuất thông tin sản phẩm
64      function getProduct(uint id) public view returns (Product memory) {
65          for (uint i = 0; i < products.length; i++) {    infinite gas
66              if (products[i].id == id) {
67                  return products[i];
68              }
69          }
70          revert("Product ID does not exist");
71      }
72
73      // Ghi nhận chi tiết giao dịch
74      function logTransactionDetails() public {
75          emit TransactionDetails(tx.gasprice, blockhash(block.number - 1), block.number);    infinite gas
76      }
77  }
78
```

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

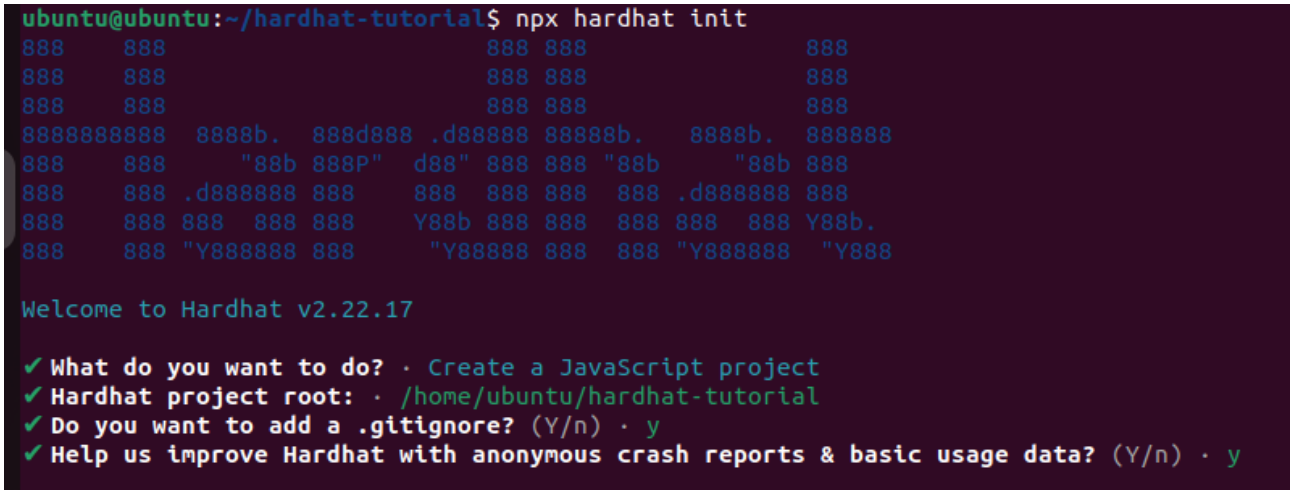
mkdir hardhat-tutorial

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() {

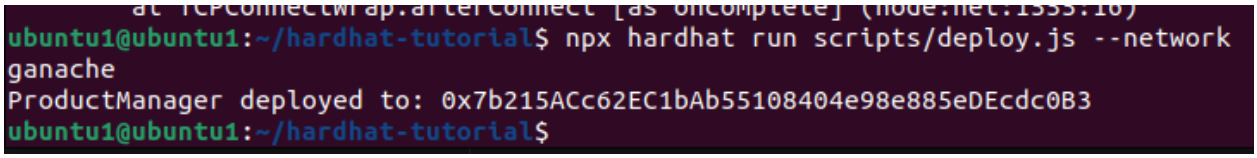
const ProductManager = await ethers.getContractFactory("P

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);

```
 await productManager.addProduct("Apple", 100, 15000);
ontractTransactionResponse {
 provider: HardhatEthersProvider {
   _hardhatProvider: LazyInitializationProviderAdapter {
     _providerFactory: [AsyncFunction (anonymous)],
     _emitter: [EventEmitter],
     _initializingPromise: [Promise],
     provider: [BackwardsCompatibilityProviderAdapter]
   },
   _networkName: 'ganache',
   _blockListeners: [],
   _transactionHashListeners: Map(0) {},
   _eventListeners: []
 },
 blockNumber: 3,
 blockHash: '0x0f763edd744c599c28ad0984fb086063e70d39326d829556fdfafdbd7c08a39e
',
 index: undefined,
 hash: '0x4daa059498b8c753fc098d9456adf77f9a6fdf8a1860b9fa360897311309d6d8',
 type: 2,
 to: '0x8c757C82550f3800129dDDE0B675A2ae251A00b4',
 from: '0x62Ae4b7191324f9244280C1fa3b96BAF21DfF428',
 nonce: 2,
 gasLimit: 138045n,
```

Cập nhật:

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

```
> await productManager.updateProduct(1, "Apple", 120, 15500);
ContractTransactionResponse {
  provider: HardhatEthersProvider {
    _hardhatProvider: LazyInitializationProviderAdapter {
      _providerFactory: [AsyncFunction (anonymous)],
      _emitter: [EventEmitter],
      _initializingPromise: [Promise],
      provider: [BackwardsCompatibilityProviderAdapter]
    },
    _networkName: 'ganache',
    _blockListeners: [],
    _transactionHashListeners: Map(0) {},
    _eventListeners: []
  },
  blockNumber: 5,
  blockHash: '0xdc6317ecd48fe2958382134132687305ddc90019ff056b1b480da241a942775b
',
  index: undefined,
  hash: '0xd79a5c1bcd26619183ca3f8682dee2ad9c99825d7f88b88dc63cb2b931ef55a5',
```

Lấy thông tin hàng hóa:

const product = await productManager.getProduct(1);

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

```
> const product = await productManager.getProduct(1);
undefined
> console.log("Product:", product);
Product: Result(4) [ 1n, 'Apple', 120n, 15500n ]
undefined
>
```

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

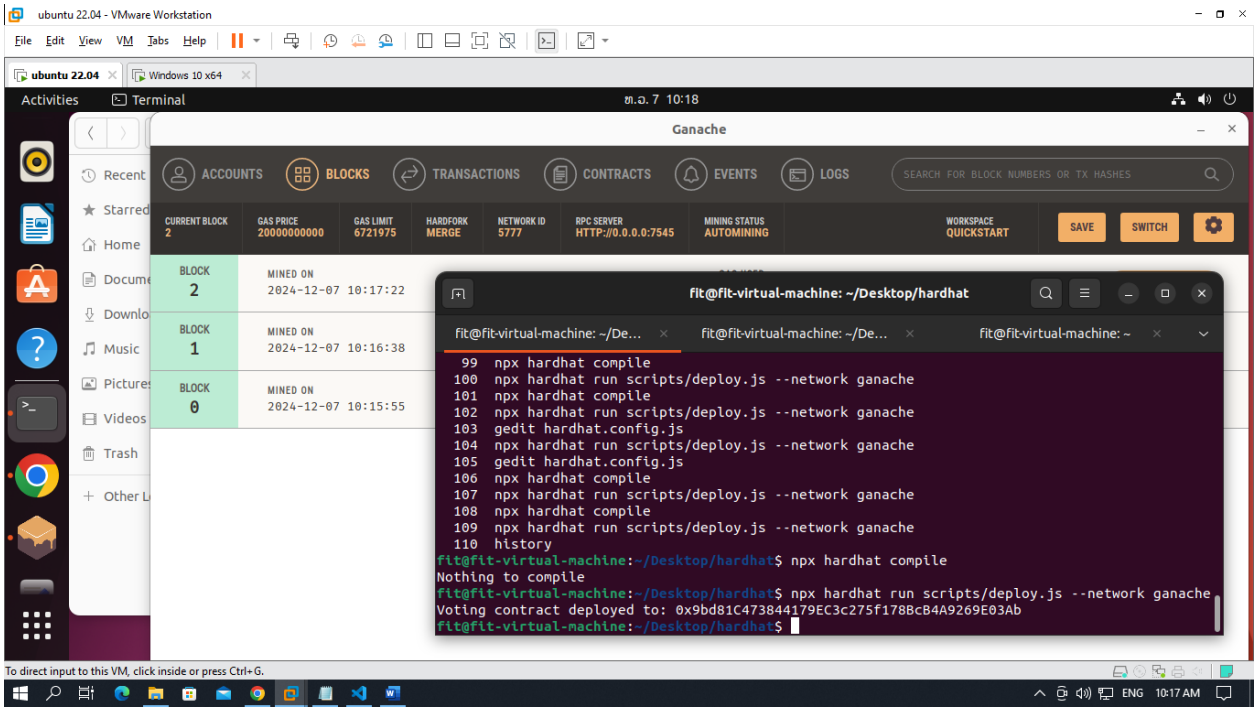
```
// Khởi tạo Web3
const GANACHE_NODE_URL = "http://<Ubuntu_IP>:7545"; // Thay <Ubuntu_IP> bằng IP Ubuntu
```

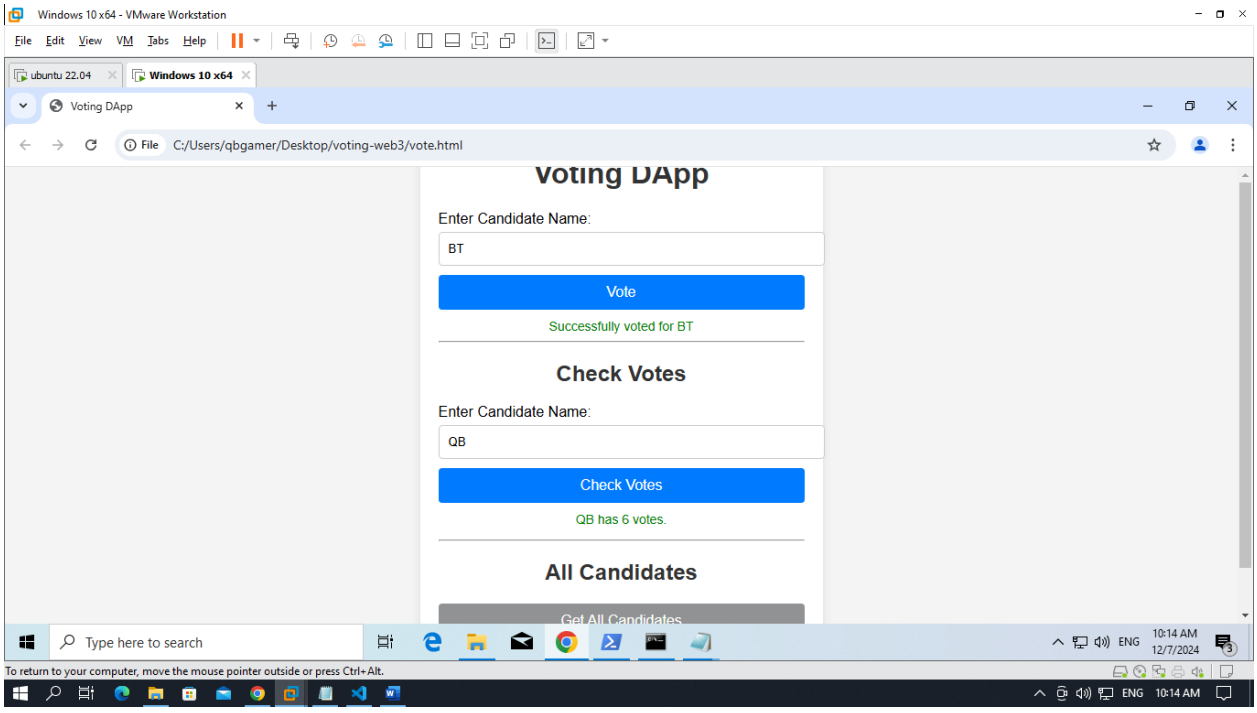
```
const web3 = new Web3(GANACHE_NODE_URL);

// 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


ubuntu 22.04 - VMware Workstation
Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
fit@fit-virtual-machine: ~/Desktop/hardhat
99 npx hardhat compile
100 npx hardhat run scripts/deploy.js --network ganache
101 npx hardhat compile
102 npx hardhat run scripts/deploy.js --network ganache
103 gedit hardhat.config.js
104 npx hardhat run scripts/deploy.js --network ganache
105 gedit hardhat.config.js
106 npx hardhat compile
107 npx hardhat run scripts/deploy.js --network ganache
108 npx hardhat compile
109 npx hardhat run scripts/deploy.js --network ganache
110 history
fit@fit-virtual-machine:~/Desktop/hardhat$ npx hardhat compile
Nothing to compile
fit@fit-virtual-machine:~/Desktop/hardhat$ npx hardhat run scripts/deploy.js --network ganache
Voting contract deployed to: 0x9bd81C473844179EC3c275f178BcB4A9269E03Ab
fit@fit-virtual-machine:~/Desktop/hardhat$


Windows 10 x64 - VMware Workstation
Voting DApp
C:/Users/qbgamer/Desktop/voting-web3/vote.html
Voting DApp
Enter Candidate Name:
BT
Vote
Successfully voted for BT
Check Votes
Enter Candidate Name:
QB
Check Votes
QB has 6 votes.
All Candidates
Get All Candidates