**LabHyper - Build and Deploy a Hyperledger Fabric Network**

[**1.** **Giới thiệu**](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932144)

[**2.** **Triển khai máy ảo**](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932145)

[**3.** **Các bước triển khai một mạng Hyperledger Fabric**](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932146)

[3.1 Bước 1. Generate cấu hình network](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932147)

[3.2 Bước 2. Generate cấu hình channel](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932148)

[3.3 Bước 3. Replace Private Key cho các CA](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932149)

[3.4 Bước 4. Up các container](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932150)

[3.5 Bước 5. Tạo channel](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932151)

[3.6 Bước 6. Join các peer của các Org vào channel](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932152)

[3.7 Bước 7. Update Anchor Peer cho các Org](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932153)

[3.8 Bước 8. Install chaincode lên các peer](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932154)

[3.9 Bước 9. Instantiate Chaincode](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932155)

[3.10 B10. Dừng mạng Fabric](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932156)

[***4.*** **Tham khảo các bước khi chạy script *byfn.sh***](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932157)

[**5.** **Dừng mạng first-network**](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932158)

[**6.** **Tham khảo**](http://cnttbkdn.com:7777/mod/resource/tknetlab/LabHyper%20-%20Build%20a%20first-network%20With%20Hyperledger%20Fabric.htm#_Toc102932159)

# **1.** **Giới thiệu**

Nội dung bài lab nhằm triển khai hệ thống mạng blockchain first-network và tạo một mạng mới trên nền tảng Hyperledger.

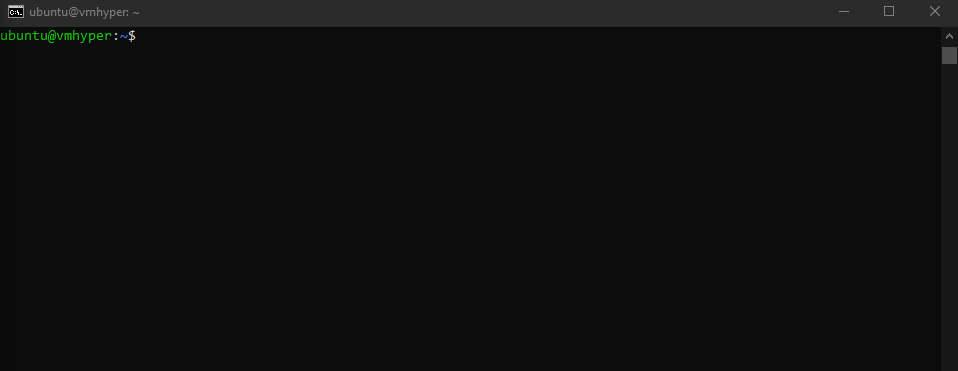
# **2.** **Triển khai máy ảo**

§ Tải máy ảo VMHyper15 tại LINK06. Giải nén file tải về tại thư mục gốc ổ đĩa C:

§ Cài đặt và chạy VMware Workstation.

§ Từ cửa sổ VMWare, chọn menu File/Open, đến thư mục VMHyper15, mở file \*.vmx và khởi động máy ảo.

§ Sử dụng công cụ SecureCRT hoặc lệnh ssh để truy cập từ xa đến máy ảo theo địa chỉ IP: 192.168.1.15, port 22, account: ubuntu, pass: 123456

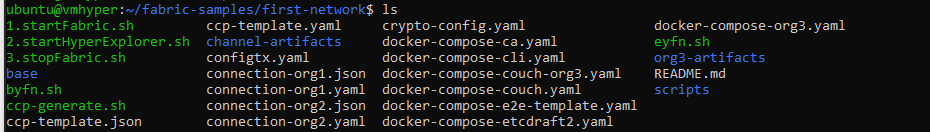


# **3.** **Các bước triển khai một mạng Hyperledger Fabric**

§ Nội dung phần này chạy mạng blockchain *first-network* bằng lệnh.

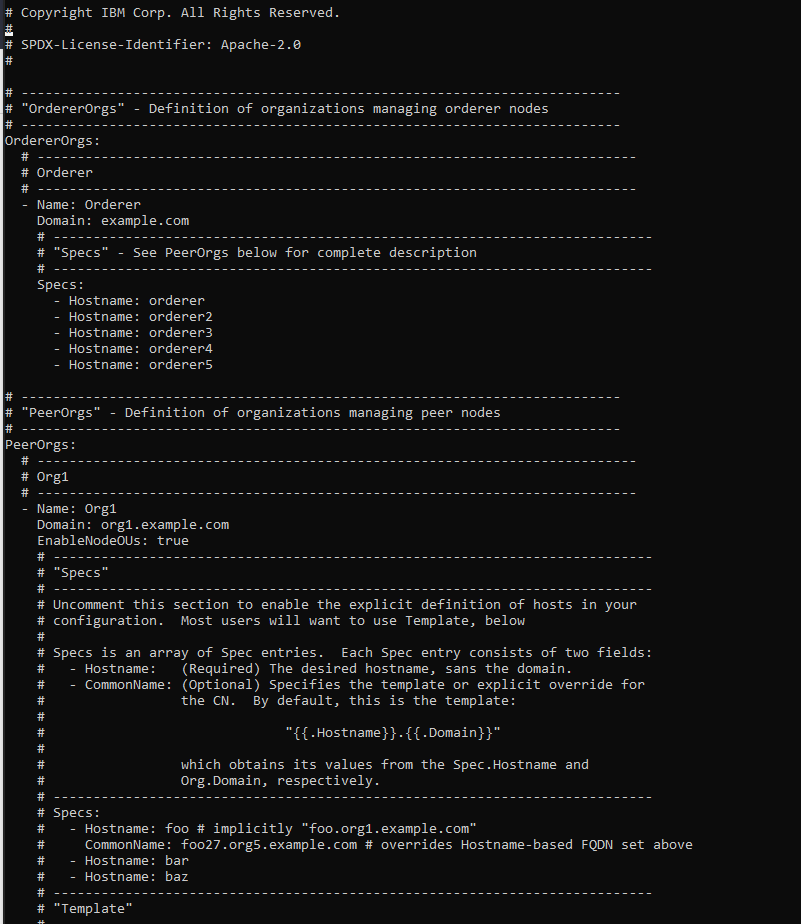
## **3.1** **Bước 1. Generate cấu hình network**

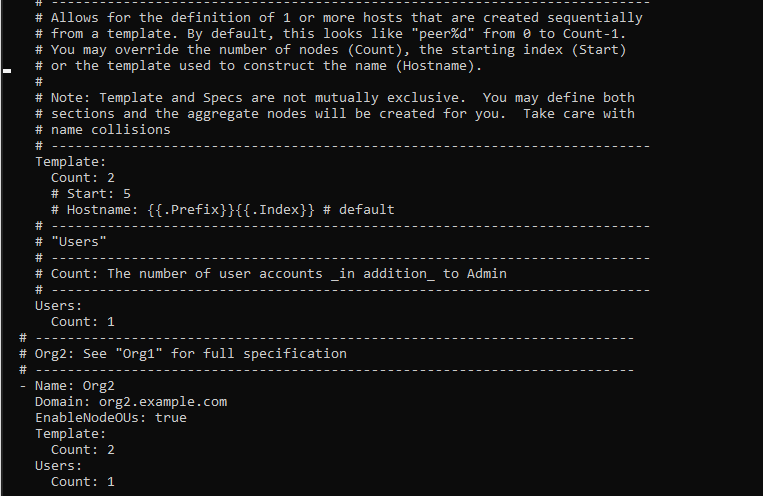
§ Chuyển vào thư mục *first-network*

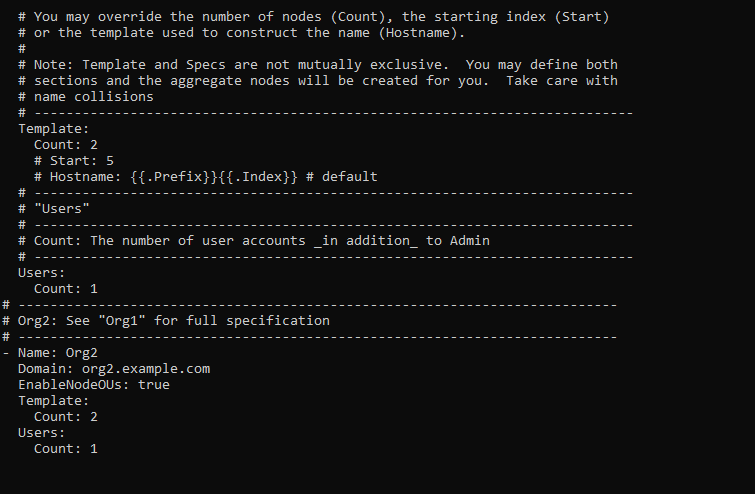
**

**ubuntu@vmhyper:~$ cd $HOME/fabric-samples-v1/first-network**

§ Xem file cấu hình *crypto-config.yaml*

**

**

**

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ cat crypto-config.yaml**

##########################################################

##### Generate certificates using cryptogen tool #########

##########################################################

§ Generate cấu hình của network và channel: Ba gồm nhiều bước để từ các file cấu hình *crypto-config.yaml* và *configtx.yaml* ta sinh ra được cấu hình cho network và channel.

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ ../bin/cryptogen generate --config=./crypto-config.yaml**

org1.example.com

Org2.example.com



## **3.2** **Bước 2. Generate cấu hình channel**

##########################################################

######### Generating Orderer Genesis block ##############

##########################################################

§ Bước này sử dụng tool ../*bin*/*configtxgen* để từ nội dung của file configtx.yaml tạo ra file genesis.block trong thư mục channel-artifacts.

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ ../bin/configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block -channelID byfn-sys-channel**

2022-09-20 01:03:30.465 PDT [common.tools.configtxgen] main -> INFO 001 Loading configuration

2022-09-20 01:03:30.552 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 002 orderer type: solo

2022-09-20 01:03:30.553 PDT [common.tools.configtxgen.localconfig] Load -> INFO 003 Loaded configuration: /home/ubuntu/fabric-samples-v1/first-network/configtx.yaml

2022-09-20 01:03:30.667 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 004 orderer type: solo

2022-09-20 01:03:30.667 PDT [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 005 Loaded configuration: /home/ubuntu/fabric-samples-v1/first-network/configtx.yaml

2022-09-20 01:03:30.677 PDT [common.tools.configtxgen] doOutputBlock -> INFO 006 Generating genesis block

2022-09-20 01:03:30.678 PDT [common.tools.configtxgen] doOutputBlock -> INFO 007 Writing genesis block

#################################################################

### Generating channel configuration transaction 'channel.tx' ###

#################################################################

§ Bước này cũng sử dụng tool ../*bin*/*configtxgen* để từ nội dung của file configtx.yaml sinh ra file *channel.tx* trong thư mục *channel-artifacts*

§ Xem file cấu hình *configtx.yaml*

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ cat configtx.yaml**

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ export CHANNEL\_NAME=mychannel**

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ ../bin/configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID $CHANNEL\_NAME**

2022-09-20 01:04:24.790 PDT [common.tools.configtxgen] main -> INFO 001 Loading configuration

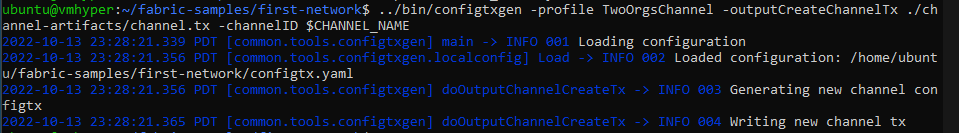
2022-09-20 01:04:24.874 PDT [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/ubuntu/fabric-samples-v1/first-network/configtx.yaml

2022-09-20 01:04:24.968 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 orderer type: solo

2022-09-20 01:04:24.968 PDT [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 004 Loaded configuration: /home/ubuntu/fabric-samples-v1/first-network/configtx.yaml

2022-09-20 01:04:24.968 PDT [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 005 Generating new channel configtx

2022-09-20 01:04:24.972 PDT [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 006 Writing new channel tx



§ Trong thư mục channel-artifacts sẽ có file *channel.tx*

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ ls channel-artifacts**

#################################################################

####### Generating anchor peer update for Org1MSP ##########

#################################################################

§ Bước này cũng sử dụng tool *../bin/configtxgen* để từ nội dung của file *configtx.yaml* tạo ra các file tương ứng trong thư mục channel-artifacts.

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ ../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org1MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org1MSP**

2022-09-20 01:04:47.536 PDT [common.tools.configtxgen] main -> INFO 001 Loading configuration

2022-09-20 01:04:47.644 PDT [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/ubuntu/fabric-samples-v1/first-network/configtx.yaml

2022-09-20 01:04:47.749 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 orderer type: solo

2022-09-20 01:04:47.749 PDT [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 004 Loaded configuration: /home/ubuntu/fabric-samples-v1/first-network/configtx.yaml

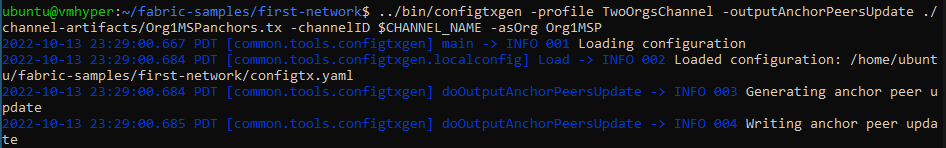
2022-09-20 01:04:47.750 PDT [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 005 Generating anchor peer update

2022-09-20 01:04:47.750 PDT [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 006 Writing anchor peer updat

#################################################################

####### Generating anchor peer update for Org2MSP ##########

#################################################################



**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ ../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org2MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org2MSP**

2022-09-20 01:05:14.603 PDT [common.tools.configtxgen] main -> INFO 001 Loading configuration

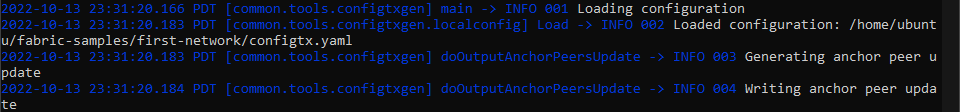
2022-09-20 01:05:14.692 PDT [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/ubuntu/fabric-samples-v1/first-network/configtx.yaml

2022-09-20 01:05:14.804 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 orderer type: solo

2022-09-20 01:05:14.805 PDT [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 004 Loaded configuration: /home/ubuntu/fabric-samples-v1/first-network/configtx.yaml

2022-09-20 01:05:14.805 PDT [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 005 Generating anchor peer update

2022-09-20 01:05:14.805 PDT [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 006 Writing anchor peer update



§ Kết quả trong thư mục channel-artifacts sẽ xuất hiện các file Org1MSPanchors.tx và Org2MSPanchors.tx



**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ ls channel-artifacts**

§ NOTE: Khi xem tutorial BYFN ta sẽ thấy quá trình này thực thi tự động các bước bằng các function generateCerts(), replacePrivateKey(), generateChannelArtifacts() trong file ./byfn.sh

## **3.3** **Bước 3. Replace Private Key cho các CA**

§ Thay thế private key đúng vào cho các container ca0 và ca1 trong file *docker-compose-e2e.yaml*

§ Xem thêm về cách thực hiện trong function replacePrivateKey() trong file ./byfn.sh

§ Bước này được tự động thực hiện theo container

## **3.4** **Bước 4. Up các container**

§ Cấu trúc của một network trong Hyperledger Fabric bao gồm Orderer, Các peer của các Org, CA của mỗi Org, CLI. Tùy vào yêu cầu của project mà cấu trúc có thể khác đi một ít.

§ Tất cả các container này đều được định nghĩa trong file *docker-compose.yaml*

#################################################################

####### Starting for channel 'mychannel' with CLI timeout of '10' seconds and CLI delay of '3' seconds ##########

#################################################################

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ sudo chmod 666 /var/run/docker.sock**

[sudo] password for ubuntu: 123456



**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ docker-compose -f docker-compose-cli.yaml up -d**

Creating network "net\_byfn" with the default driver

Creating volume "net\_peer0.org2.example.com" with default driver

Creating volume "net\_peer1.org2.example.com" with default driver

Creating volume "net\_peer1.org1.example.com" with default driver

Creating volume "net\_peer0.org1.example.com" with default driver

Creating volume "net\_orderer.example.com" with default driver

Creating peer0.org1.example.com

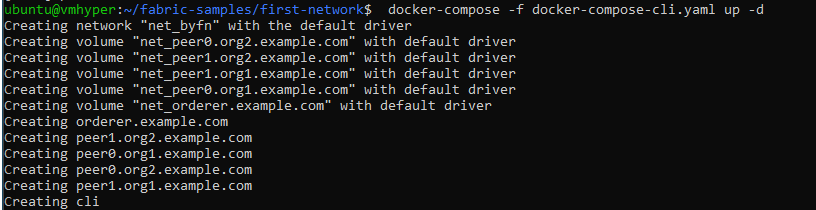
Creating peer1.org1.example.com

Creating peer0.org2.example.com

Creating orderer.example.com

Creating peer1.org2.example.com

Creating cli



§ Để xem danh sách các container đang chạy:

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ docker ps**

## **3.5** **Bước 5. Tạo channel**

§ Truy cập vào container CLI để thực hiện các lệnh tiếp theo

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ docker exec -it cli bash**

§ Đường dẫn sẽ đổi thành:

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer#**

§ Trong container CLI sẽ thực hiện lệnh tạo channel. Lệnh này sẽ lấy file channel.tx được sinh ra ở bước 2 trong mục 1.2 để tạo channel.

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# export CHANNEL\_NAME=mychannel**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem**

2022-09-20 08:07:05.200 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized

2022-09-20 08:07:05.246 UTC [cli.common] readBlock -> INFO 002 Received block: 0

## **3.6** **Bước 6. Join các peer của các Org vào channel**

§ Các lệnh sau sẽ join peer của một Org vào channel và các biến môi trường trong CLI sẽ thay đổi theo.

§ Join các peer của Org1:

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_LOCALMSPID="Org1MSP"**

****

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt**

****

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# export CHANNEL\_NAME=mychannel**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem**

===================== peer0.org1 joined channel 'mychannel' =====================

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel join -b mychannel.block**

2022-09-20 08:09:46.972 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized

2022-09-20 08:09:47.018 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel

§ Join các peer của Org2:

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_ADDRESS=peer0.org2.example.com:9051**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_LOCALMSPID="Org2MSP"**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt**

****

===================== peer0.org2 joined channel 'mychannel' =====================

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel join -b mychannel.block**

2022-09-20 07:36:23.230 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized

2022-09-20 07:36:23.275 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel

## **3.7** **Bước 7. Update Anchor Peer cho các Org**

§ Ta cũng phải thiết lập biến môi trường trong CLI tương ứng với Anchor Peer của Org sắp được update.

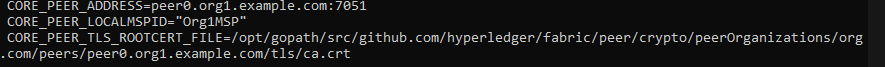
§ Update Anchor Peer cho Org1:

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_LOCALMSPID="Org1MSP"**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt**

****

===================== Anchor peers updated for org 'Org1MSP' on channel 'mychannel' =====================

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel update -o orderer.example.com:7050 -c mychannel -f ./channel-artifacts/Org1MSPanchors.tx --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem**

2022-09-20 07:39:09.438 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized

2022-09-20 07:39:09.464 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update





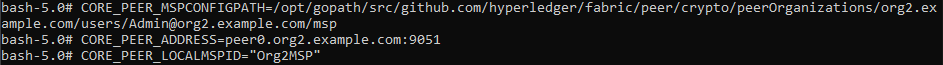
§ Update Anchor Peer cho Org2:

===================== Anchor peers updated for org 'Org2MSP' on channel 'mychannel' =====================

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_ADDRESS=peer0.org2.example.com:9051**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_LOCALMSPID="Org2MSP"**

****

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt**

===================== Anchor peers updated for org 'Org2MSP' on channel 'mychannel' =====================

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel update -o orderer.example.com:7050 -c mychannel -f ./channel-artifacts/Org2MSPanchors.tx --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem**

2022-09-20 07:40:49.054 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized

2022-09-20 07:40:49.084 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update

## **3.8** **Bước 8. Install chaincode lên các peer**

§ Bước này chỉ thực hiện được sau khi tạo xong chaincode và ở bước trên đã volumes chính xác đường dẫn đến thư mục chứa chaincode.

§ Tương tự như 2 bước trên, muốn install chaincode lên peer của tổ chức nào thì phải đặt lại biến môi trường trong CLI.

§ Install chaincode lên peer của Org1:

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_LOCALMSPID="Org1MSP"**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt**

****

===================== Chaincode is installed on peer0.org1 =====================

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode install -n mycc -v 1.0 -l golang -p github.com/chaincode/chaincode\_example02/go/**

2022-09-20 08:25:20.103 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc

2022-09-20 08:25:20.103 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc

2022-09-20 08:25:21.208 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >

§ Install chaincode lên peer của Org1:

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_ADDRESS=peer0.org2.example.com:9051**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_LOCALMSPID="Org2MSP"**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt**

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode install -n mycc -v 1.0 -l golang -p github.com/chaincode/chaincode\_example02/go/**

2022-09-20 08:27:27.261 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc

2022-09-20 08:27:27.261 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc

2022-09-20 08:27:27.544 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >

## **3.9** **Bước 9. Instantiate Chaincode**

§ Sau khi chaincode được install, nó cần phải được instantiate thì mới có thể hoạt động được trên channel.

Instantiating chaincode on peer0.org2...

===================== Chaincode is instantiated on peer0.org2 on channel 'mychannel' =====================

**root@ee4e88eaf9c0:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode instantiate -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n mycc -l golang -v 1.0 -c '{"Args":["init","a","100","b","200"]}' -P 'AND ('\''Org1MSP.peer'\'','\''Org2MSP.peer'\'')'**

2022-09-20 07:44:57.328 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc

2022-09-20 07:44:57.329 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc

§ hoặc

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode instantiate -o orderer.certificate.com:7050 -C $CHANNEL\_NAME -n $CHAINCODE\_NAME -v 1.0 -c '{"Args":[]}' -P "OR ('Org1.member','Org2.member')"**

## **3.10** **Bước 10. Querying on peer0.org1 on channel 'mychannel'**

===================== Querying on peer0.org1 on channel 'mychannel'... =====================

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C mychannel -n mycc -c '{"Args":["query","a"]}'**

100

Sending invoke transaction on peer0.org1 peer0.org2...

**root@010c91b66b68:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n mycc --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"Args":["invoke","a","b","10"]}'**

2022-09-20 08:29:53.750 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200

## **3.11** **Bước 11. Dừng mạng Fabric**

**root@aeed554b44da:/opt/gopath/src/github.com/hyperledger/fabric/peer# exit**

**ubuntu@vmhyper:~/fabric-samples-v1/first-network$ ./byfn.sh down**

Nhấn phím Y

§ Sau khi thực hiện lần lượt đầy đủ các bước trên là ta sẽ được một network. Tổng hợp các bước này được trong file script ./byfn.sh để thực hiện tự động 10 bước trên.

# **4.** **Dừng mạng first-network**

**ubuntu@vmhyper~ $ cd $HOME/fabric-samples-v1/first-network**

**ubuntu@vmhyper:~/ fabric-samples-v1/first-network $ ./byfn.sh down**