The Chinese University of Hong Kong

Department of Information Engineering

FTEC5520 –Applied Blockchain & Cryptocurrency

Lab2-1 Report

Hyperledger Fabric Setup & Practice on AWS(Task1,2,3)

Please edit this file directly but submitted a PDF version to blackboard finally.

1 Questions Answering:

There will be 3 questions only for **task1,2 and 3**.

Q1: Basic concepts and knowledge about Hyperledger Fabric. (2 mark)

i. What's the difference between member and peer node in Hyperledger fabric?

In Hyperledger fabric, member is a legally separate entity that owns a unique root certificate for the network. Network components such as peer nodes and application clients will be linked to a member.

Peer node is a network entity that maintains a ledger and runs chaincode containers in order to perform read-write operations to the ledger. Peers are owned and maintained by members.

ii. What's the channel in Hyperledger Fabric? Why do we need channel?

A Hyperledger Fabric channel is a private "subnet" of communication between two or more specific network members, for the purpose of conducting private and confidential transactions. A channel is defined by members (organizations), anchor peers per member, the shared ledger, chaincode application(s) and the ordering service node(s).

Hyperledger Fabric enables confidentiality through its channel architecture. Basically, participants on a Fabric network can establish a "channel" between the subset of participants that should be granted visibility to a particular set of transactions. Only those nodes that participate in a channel have access to the smart contract (chaincode) and data transacted, preserving the privacy and confidentiality of both.

iii. What's chaincode in Hyperledger Fabric? What did you do in task 1~3 about chaincode?

Chaincode is software defining an asset or assets, and the transaction instructions for modifying the asset(s); in other words, it's the business logic. Chaincode enforces the rules for reading or altering key-value pairs or other state database information.

In task 3 **Configure Blockchain Network**, after enrolling an admin identity with the Fabric CA, we install chaincode on the peer node, instantiate the chaincode on the channel and query the chaincode on peer.

iv. What's Membership Service Provider (MSP)? Why do we need it? What did you do about this component in task1~3?

Membership Service Provider (MSP) is a component that aims to offer an abstraction of a membership operation architecture. A Hyperledger Fabric blockchain network can be governed by one or more MSPs. This provides modularity of membership operations, and interoperability across different membership standards and architectures.

We need MSP because it provides credentials to clients, and peers for them to participate in a Hyperledger Fabric network. Clients use these credentials to authenticate their transactions, and peers use these credentials to authenticate transaction processing results (endorsements).

In task3, some elements to an MSP are setup, such as Intermediate Cas, Root CAs, Organizational Units (OUs) and Node Identity. These are listed in the config.yaml file , whose members are considered to be part of the organization represented by this MSP.

Hint: *You may refer to this chapter of official docs of Hyperledger Fabric.*

Q2: Please read step 1 ~ 7 of the AmazonManagedBlockchain official documents and answer the following questions. (2 mark)

Note: Please describe the procedure in your own words and list the commands.

1) How to create blockchain network, member and node using AWS CLI instead of GUI?

Use the **`create-network`** command to create a new blockchain network and member

```
[ec2-user@ip-192-0-2-17 ~]$ aws managedblockchain create-network \
--network-configuration '{"Name":"ngo", \
"Description":"MyTaigaNetDescription", \
"Framework":"HYPERLEDGER_FABRIC","FrameworkVersion": "1.2", \
"FrameworkConfiguration": {"Fabric": {"Edition": "STARTER"}}, \
"VotingPolicy": {"ApprovalThresholdPolicy": {"ThresholdPercentage": 50, \
"ProposalDurationInHours": 24, \
"ThresholdComparator": "GREATER_THAN"}}}' \
--member-configuration '{"Name":"member", \
"Description":"first member of network",\
"FrameworkConfiguration":{"Fabric":
{"AdminUsername":"admin","AdminPassword":"Adminpwd1!"}}}'
```

The command returns the `Network ID` and the `Member ID`

```
{
    "NetworkId": "n-MWY63ZJZU5HGNCMBQER7IN6OIU",
    "MemberId": "m-K46ICRRXJRCGRNNS4ES4XUUS5A"
}
```

Use the **`create-node`** command and Replace the value of **`--network-id, --member-id,`** and AvailabilityZone as.

```
[ec2-user@ip-192-0-2-17 ~]$ aws managedblockchain create-node \
--node-configuration '{"InstanceType":"bc.t3.small","AvailabilityZone":"us-east-
1a"}' \
--network-id n-MWY63ZJZU5HGNCMBQER7IN6OIU \
--member-id m-K46ICRRXJRCGRNNS4ES4XUUS5A
```

2) **What's the CAEndpoint of here? How can we get the value in AmazonManagedBlockchain service dashboard? How to enroll an Admin User with admin permissions to your member's CA? Hint: `fabric-ca-client enroll`**

It is the values of **`--network-id and --member-id.`** We can use the get-member command to get the CA endpoint for the member:

```
[ec2-user@ip-192-0-2-17 ~]$ aws managedblockchain get-member \
--network-id n-MWY63ZJZU5HGNCMBQER7IN6OIU \
--member-id m-K46ICRRXJRCGRNNS4ES4XUUS5A
```

Use the CA endpoint, administrator profile, and the certificate file to enroll the member administrator using the **`fabric-ca-client enroll`** command:

```
[ec2-user@ip-192-0-2-17 ~]$ fabric-ca-client enroll \
-u https://admin:Admin1!@ca.m-K46ICRRXJRCGRNNS4ES4XUUS5A.n-MWY63ZJZU5HGNCMBQER7IN6OIU.managedblockchain.us-east-1.amazonaws.com:30002 \
--tls.certfiles /home/ec2-user/managedblockchain-tls-chain.pem -M /home/ec2-user/admin-msp
```

3) **How to create a channel and join your peer node to the channel your created?**
4) **How to install, instantiate, query and invoke the Chaincode? Please use the example we used in task3.**

Q3: What's the difference between Hyperledger Fabric and Ethereum after you learnt about both the two frameworks including the hands-on lab session? (1 mark)

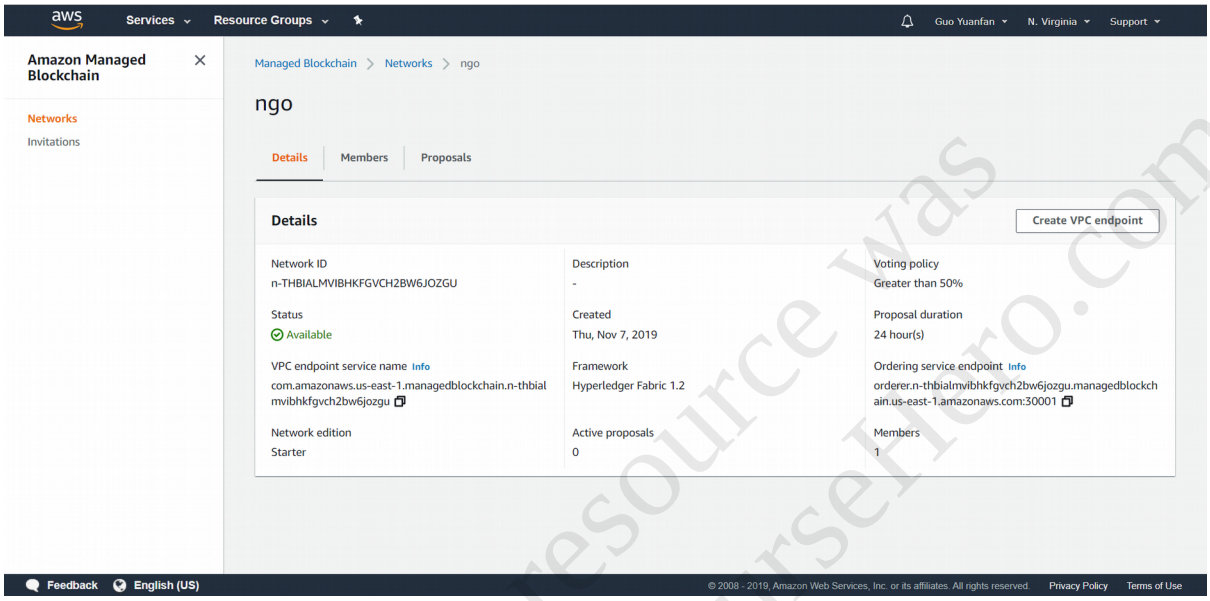
Note: Detailed answer expected but no more than 1 page. Please refer to lecture slides including guest speaker from Oracle, the article-1, article-2 and combine the contents of previous questions if necessary.

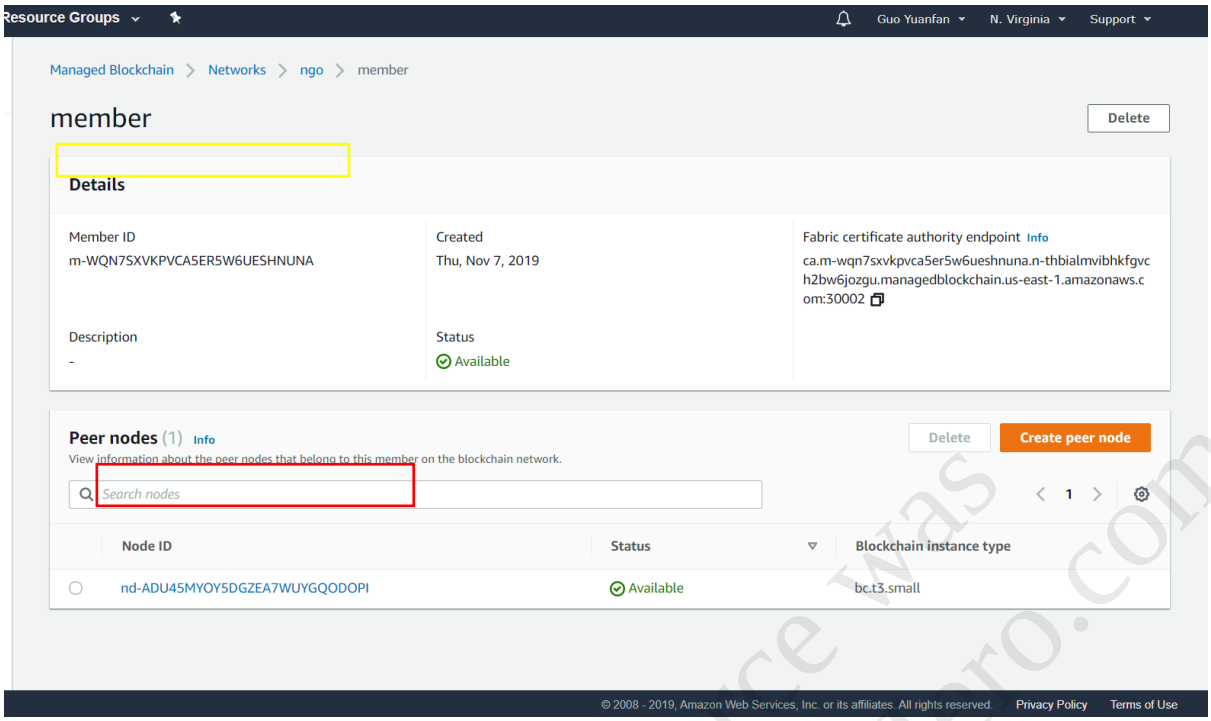
2 Screen Capture of Main Steps:

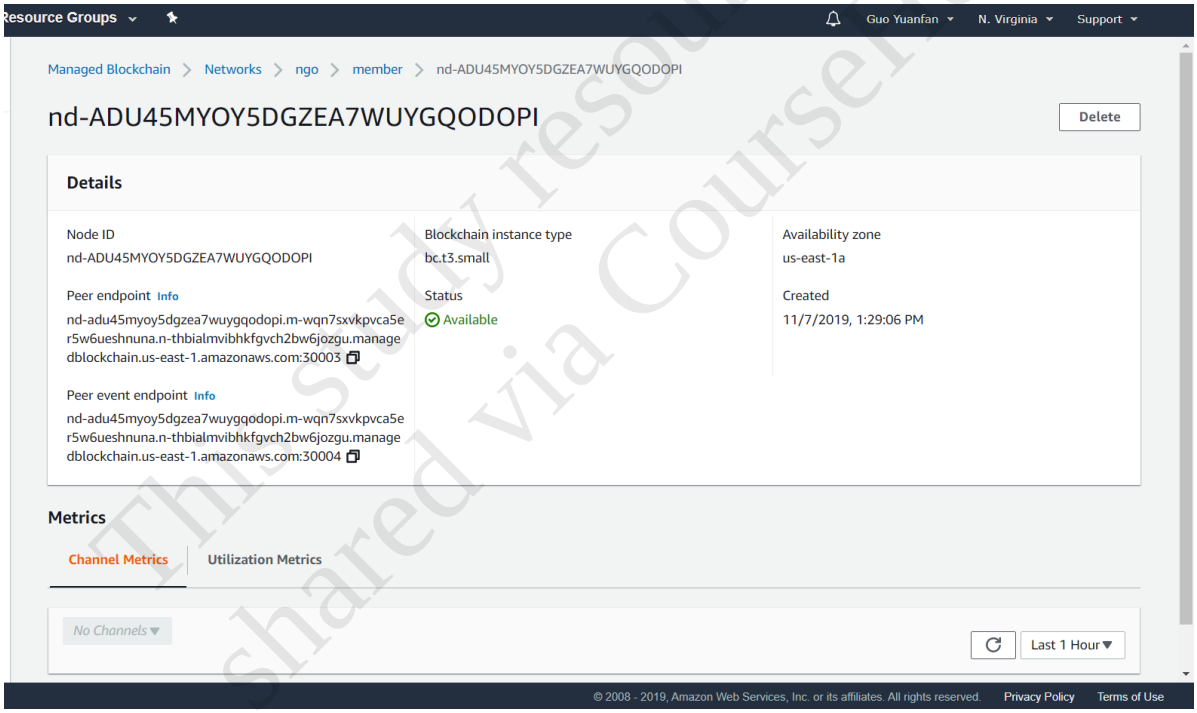
Please replace the sample photos with your own results.

Your screenshots must include these parts at least and detailed description of each screenshots:

1 Your Hyperledger Fabric blockchain network on AmazonManagedBlockchain dashboard:







2 SSH from the Cloud9 IDE to peer node. You should know they are two VMs.

```
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - ngo-fabric-client-node
ec2-user:~/non-profit-blockchain/ngo-fabric (master) $ cd
ec2-user:~ $ ssh -i ~/ngo-keypair.pem ec2-54-221-67-222.compute-1.amazonaws.com
The authenticity of host 'ec2-54-221-67-222.compute-1.amazonaws.com (54.221.67.222)' can't be established.
ECDSA key fingerprint is SHA256:+kX1pN3Hbxo9Puj+R0hG/3PP152b5NP1P8e4O1YDh8M.
ECDSA key fingerprint is MD5:e7:b5:81:87:e1:e2:2d:f7:bf:e5:77:5a:c5:82:c4:32.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-221-67-222.compute-1.amazonaws.com,54.221.67.222' (ECDSA) to the list of known hosts.
Last login: Tue Nov 27 21:57:11 2018 from 72-21-198-66.amazon.com

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
23 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
Starting cli ... done
[ec2-user@ip-10-0-22-178 ~]$
```

3 The content of **fabric-exports.sh** after insert your blockchain network information

ssh - "ec2-user@ × Immediate ×

```
#!/bin/bash

# Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License.
# A copy of the License is located at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# or in the "license" file accompanying this file. This file is distributed
# on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
# express or implied. See the License for the specific language governing
# permissions and limitations under the License.

# Update these values, then `source` this script
export REGION=us-east-1
export NETWORKNAME=ngo
export MEMBERNAME=member
export NETWORKVERSION=1.2
export ADMINUSER=admin
export ADMINPWD=Adminpwd1!
export NETWORKID=n-THBIALMVIBHKFGVCH2BW6JOZGU
export MEMBERID=m-WQN7SXVKPVCA5ER5W6UESHNUMA

# No need to change anything below here
echo Updating AWS CLI to the latest version
sudo pip install awscli --upgrade
cd ~

VpcEndpointServiceName=$(aws managedblockchain get-network --region $REGION --network-id $NETWORKID --query 'Network.VpcEndpointServiceName' --out
OrderingServiceEndpoint=$(aws managedblockchain get-network --region $REGION --network-id $NETWORKID --query 'Network.FrameworkAttributes.Fabric.O
CaEndpoint=$(aws managedblockchain get-member --region $REGION --network-id $NETWORKID --member-id $MEMBERID --query 'Member.FrameworkAttributes.F
nodeID=$(aws managedblockchain list-nodes --region $REGION --network-id $NETWORKID --member-id $MEMBERID --query 'Nodes[0].Id' --output text)
peerEndpoint=$(aws managedblockchain get-node --region $REGION --network-id $NETWORKID --member-id $MEMBERID --node-id $nodeID --query 'Node.Frame
peerEventEndpoint=$(aws managedblockchain get-node --region $REGION --network-id $NETWORKID --member-id $MEMBERID --node-id $nodeID --query 'Node.
export ORDERINGSERVICEENDPOINT=$OrderingServiceEndpoint
export ORDERINGSERVICEENDPOINTNOPORT=${ORDERINGSERVICEENDPOINT::-6}
export VPCENDPOINTSERVICENAME=$VpcEndpointServiceName
export CASERVICEENDPOINT=$CaEndpoint
export PEERNODEID=$nodeID
```

4 The useful information after you ran the command in task3 step 2.16

```
source fabric-exports.sh
```

```
Useful information used in Cloud9
REGION: us-east-1
NETWORKNAME: ngo
NETWORKVERSION: 1.2
ADMINUSER: admin
ADMINPWD: Adminpwd1!
MEMBERNAME: member
NETWORKID: n-THBIALMVIBHKFGVCH2BW6JOZGU
MEMBERID: m-WQN7SXVKPVCA5ER5W6UESHNUNA
ORDERINGSERVICEENDPOINT: orderer.n-thbialmvibhkfgvch2bw6jozgu.managedblockchain.us-east-1.amazonaws.com:30001
ORDERINGSERVICEENDPOINTNOPORT: orderer.n-thbialmvibhkfgvch2bw6jozgu.managedblockchain.us-east-1.amazonaws.com
VPCENDPOINTSERVICENAME: com.amazonaws.us-east-1.managedblockchain.n-thbialmvibhkfgvch2bw6jozgu
CASERVICEENDPOINT: ca.m-wqn7sxvkpvca5er5w6ueshnuna.n-thbialmvibhkfgvch2bw6jozgu.managedblockchain.us-east-1.amazonaws.com:30002
PEERNODEID: nd-ADU45MYOY5DGZEA7WUYGQODOPI
PEERSERVICEENDPOINT: nd-adu45myoy5dgzea7wuygqodopi.m-wqn7sxvkpvca5er5w6ueshnuna.n-thbialmvibhkfgvch2bw6jozgu.managedblockchain.us-east-1.amazonaws.com:30003
PEERSERVICEENDPOINTNOPORT: nd-adu45myoy5dgzea7wuygqodopi.m-wqn7sxvkpvca5er5w6ueshnuna.n-thbialmvibhkfgvch2bw6jozgu.managedblockchain.us-east-1.amazonaws.com
PEEREVENTENDPOINT: nd-adu45myoy5dgzea7wuygqodopi.m-wqn7sxvkpvca5er5w6ueshnuna.n-thbialmvibhkfgvch2bw6jozgu.managedblockchain.us-east-1.amazonaws.com:30004
[ec2-user@ip-10-0-22-178 ~]$
```

5 Use this command to check the peer export information:

```
cat ~/peer-exports.sh
```

```
ssh - "ec2-user@ × Immediate ×
[ec2-user@ip-10-0-22-178 ~]$ cat ~/peer-exports.sh
export MSP_PATH=/opt/home/admin-msp
export MSP=m-WQN7SXVKPVCA5ER5W6UESHNUNA
export ORDERER=orderer.n-thbialmvibhkfgvch2bw6jozgu.managedblockchain.us-east-1.amazonaws.com:30001
export PEER=nd-adu45myoy5dgzea7wuygqodopi.m-wqn7sxvkpvca5er5w6ueshnuna.n-thbialmvibhkfgvch2bw6jozgu.managedbloc
kchain.us-east-1.amazonaws.com:30003
export CHANNEL=mychannel
export CAFILE=/opt/home/managedblockchain-tls-chain.pem
export CHAINCODENAME=mycc
export CHAINCODEVERSION=v0
export CHAINCODEDIR=github.com/chaincode_example02/go
[ec2-user@ip-10-0-22-178 ~]$
```

6 List the generated configuration using

```
ls -lt ~/$CHANNEL.pb
```

```
ssh - "ec2-user@ × Immediate ×
[ec2-user@ip-10-0-22-178 ngo-fabric]$ docker exec cli configtxgen -outputCreateChannelTx /opt/home/$CHANNEL.pb
-profile OneOrgChannel -channelID $CHANNEL --configPath /opt/home/
2019-11-07 06:09:57.478 UTC [common/tools/configtxgen] main -> INFO 001 Loading configuration
2019-11-07 06:09:57.484 UTC [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 002 Generating new chann
el configtx
2019-11-07 06:09:57.484 UTC [common/tools/configtxgen/encoder] NewApplicationGroup -> WARN 003 Default policy e
mission is deprecated, please include policy specificiations for the application group in configtx.yaml
2019-11-07 06:09:57.485 UTC [common/tools/configtxgen/encoder] NewApplicationOrgGroup -> WARN 004 Default polic
y emission is deprecated, please include policy specificiations for the application org group m-WQN7SXVKPVCA5ER
5W6UESHNUNA in configtx.yaml
2019-11-07 06:09:57.486 UTC [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 005 Writing new channel
tx
[ec2-user@ip-10-0-22-178 ngo-fabric]$ ls -lt ~/$CHANNEL.pb
-rw-r--r-- 1 root root 327 Nov  7 06:09 /home/ec2-user/mychannel.pb
[ec2-user@ip-10-0-22-178 ngo-fabric]$
```

7 List the mychannel.block generated fabric channel in task3 step 4.1

```
ls -lt /home/ec2-user/fabric-samples/chaincode/hyperledger/fabric/peer
```

```
ssh - "ec2-user@ ×   Immediate ×
[ec2-user@ip-10-0-22-178 ngo-fabric]$ ls -lt /home/ec2-user/fabric-samples/chaincode/hyperledger/fabric/peer
total 16
-rw-r--r-- 1 root root 13114 Nov  7 06:16 mychannel.block
[ec2-user@ip-10-0-22-178 ngo-fabric]$
```

8 Query the chaincode on peer in the task3 step 8

```
docker exec -e "CORE_PEER_TLS_ENABLED=true" -e
"CORE_PEER_TLS_ROOTCERT_FILE=/opt/home/managedblockchain-tls-chain.pem"
-e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e
"CORE_PEER_MSPCONFIGPATH=$MSP_PATH" cli peer chaincode query -C $CHANNEL
-n $CHAINCODENAME -c '{"Args":["query","a"]}'
```

```
[ec2-user@ip-10-0-22-178 ngo-fabric]$ docker exec -e "CORE_PEER_TLS_ENABLED=true" -e "CORE_PEER_TLS_ROOTCERT_FI
LE=/opt/home/managedblockchain-tls-chain.pem" \
>     -e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e "CORE_PEER_MSPCONFIGPATH=$MSP_PATH" \
>     cli peer chaincode query -C $CHANNEL -n $CHAINCODENAME -c '{"Args":["query","a"]}'
100
[ec2-user@ip-10-0-22-178 ngo-fabric]$
```

9 Invoke a transaction to transfer 10 dollars from Account a to Account b:

```
docker exec -e "CORE_PEER_TLS_ENABLED=true" -e
"CORE_PEER_TLS_ROOTCERT_FILE=/opt/home/managedblockchain-tls-chain.pem"
\
    -e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e
"CORE_PEER_MSPCONFIGPATH=$MSP_PATH" \
    cli peer chaincode invoke -o $ORDERER -C $CHANNEL -n $CHAINCODENAME
\
    -c '{"Args":["invoke","a","b","10"]}' --cafile $CAFILE --tls
```

```
[ec2-user@ip-10-0-22-178 ngo-fabric]$ docker exec -e "CORE_PEER_TLS_ENABLED=true" -e "CORE_PEER_TLS_ROOTCERT_FI
LE=/opt/home/managedblockchain-tls-chain.pem" \
>     -e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e "CORE_PEER_MSPCONFIGPATH=$MSP_PATH" \
>     cli peer chaincode invoke -o $ORDERER -C $CHANNEL -n $CHAINCODENAME \
>     -c '{"Args":["invoke","a","b","10"]}' --cafile $CAFILE --tls
2019-11-07 06:22:22.124 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. resu
lt: status:200
[ec2-user@ip-10-0-22-178 ngo-fabric]$
```

10 Query the balance of Account named ‘a’ again after transaction:

```
docker exec -e "CORE_PEER_TLS_ENABLED=true" -e
"CORE_PEER_TLS_ROOTCERT_FILE=/opt/home/managedblockchain-tls-chain.pem"
\
    -e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e
"CORE_PEER_MSPCONFIGPATH=$MSP_PATH" \
    cli peer chaincode query -C $CHANNEL -n $CHAINCODENAME -c '{"Args":
["query","a"]}'
```

```
[ec2-user@ip-10-0-22-178 ngo-fabric]$ docker exec -e "CORE_PEER_TLS_ENABLED=true" -e "CORE_PEER_TLS_ROOTCERT_FI
LE=/opt/home/managedblockchain-tls-chain.pem" \
>     -e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e "CORE_PEER_MSPCONFIGPATH=$MSP_PATH" \
>     cli peer chaincode query -C $CHANNEL -n $CHAINCODENAME -c '{"Args":["query","a"]}'
90
[ec2-user@ip-10-0-22-178 ngo-fabric]$
```