The Chinese University of Hong Kong

Department of Financial Technology

FTEC5520 –Applied Blockchain & Cryptocurrency

Lab2-part1-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)

1. What’s the difference between member and peer node in Hyperledger fabric?

***Answer:***

***In a Hyperledger fabric network, there are one or more members that are unique identities in the network. Each member can run one or more peer nodes.***

***Peer nodes are linked to members, peer node is the network entity and maintain a ledger, run chaincode.***

1. What’s the channel in Hyperledger Fabric? Why do we need it?

***Answer:***

***Channel is a private sub-network of communication between members.***

***The purpose of channel is to conduct private & confidential transactions between members.***

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

***Answer:***

***Chaincode is the ‘smart contract’ which runs on peers and creates transactions, namely, it’s the business logic, and enforces the rules.***

***Task 3 is to prepare client node, enroll an identity for creating channels, then instantiate chaincode on the channel, and query chaincode on peer.***

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

***Answer:***

***MSP is a modular component of Hyperledger Fabric.***

***It offers an abstraction of membership operations, to manage identities on the blockchain network and authenticate clients who want to join the blockchain network.***

***In Task 3, some elements related to MSP are initialized, which are all listed in ‘config.yaml’ file.***

**Hint:** You may refer to this [chapter of official docs of Hyperledger Fabric](https://hyperledger-fabric.readthedocs.io/en/release-1.4/key_concepts.html).

**Q2:** Please read step 1 ~ 7 of the [AmazonManagedBlockchain official documents](https://docs.aws.amazon.com/managed-blockchain/latest/managementguide/managed-blockchain-get-started-tutorial.html) 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?

***Answer:***

***a) Firstly, create the network & member with ‘create-network’ CLI:***

[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!"}}}'

***This CLI will return the network ID & member ID:***

{

"NetworkId": "n-MWY63ZJZU5HGNCMBQER7IN6OIU",

"MemberId": "m-K46ICRRXJRCGRNNS4ES4XUUS5A"

}

***b) Then, create the peer node with ‘create-node’ CLI:***

[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

1. 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

***Answer:***

***CAEndpoint: it is the values of ‘--network-id and --member-id’.***

***How to get the value: use ‘get-member’ command’:***

aws managedblockchain get-member \

--network-id n-MWY63ZJZU5HGNCMBQER7IN6OIU \

--member-id m-K46ICRRXJRCGRNNS4ES4XUUS5A

***How to enroll: Use the CAEndpoint, admin profile, and the certificate file to enroll the member admin, CLI:***

fabric-ca-client enroll \

-u 'https://AdminUsername:AdminPassword@$CASERVICEENDPOINT' \

--tls.certfiles /home/ec2-user/managedblockchain-tls-chain.pem -M /home/ec2-user/admin-msp

1. How to create a channel and join your peer node to the channel your created?

***Answer:***

1. ***Use the configtx.yaml file, but to replace the ‘MemberID’ with the MemberID you returned previously. At the same time, ‘MSPDir’ also needs to be set the same directory location that you established before.***

***Generate the configtx peer block with CLI:***

docker exec cli configtxgen \

-outputCreateChannelTx /opt/home/mychannel.pb \

-profile OneOrgChannel -channelID mychannel \

--configPath /opt/home/

1. ***Set ENV for Orderer with CLI:***

export ORDERER=orderer.n-MWY63ZJZU5HGNCMBQER7IN6OIU.managedblockchain.amazonaws.com:30001

1. ***Create the channel & join your node with CLI:***

docker exec cli peer channel create -c mychannel \

-f /opt/home/mychannel.pb -o $ORDERER \

--cafile /opt/home/managedblockchain-tls-chain.pem –tls

docker exec cli peer channel join -b mychannel.block \

-o $ORDERER --cafile /opt/home/managedblockchain-tls-chain.pem --tls

1. How to install, instantiate, query and invoke the Chaincode? Please use the example we used in task3.

***Answer:***

1. ***Install chaincode on Fabric peer with CLI:***

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 install -n $CHAINCODENAME -v $CHAINCODEVERSION -p $CHAINCODEDIR

1. ***Instantiate with CLI:***

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 instantiate -o $ORDERER -C $CHANNEL -n $CHAINCODENAME -v $CHAINCODEVERSION \

-c '{"Args":["init","a","100","b","200"]}' --cafile $CAFILE --tls

1. ***Query with CLI:***

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"]}'

1. ***Invoke a transaction with CLI:***

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

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

***Answer:***

1. ***They have different purpose, Ethereum is for running smart contracts on EVM for the mass consumption of DApps, while Hyperledger is to facilitate the creation of cross-industry Blockchain tools and applications and encourage seamless collaboration between businesses and developers working with Distributed Ledger Technology (DLT).***
2. ***They have different mode of accessibility, Ethereum is a permission-free, public Blockchain platform, but Hyperledger has strict control over accessibility and only authorized members can access and use the platform.***
3. ***They have different Consensus mechanisms. Ethereum is based on PoW (proof of work), however, Hyperledger provides 2 choices: no-op (no consensus needed) and Practical Byzantine Fault Tolerance (PBFT).***
4. ***Ethereum is public, so entirely transparent. Conversely, Hyperledger is permissioned Blockchain platform, therefore, it is highly secured.***
5. ***Ethereum supports more programming languages such as Ruby, Javascript, Python, Kotlin, C++, Go, and Typescript. In Hyperledger, ‘chaincode’ are always coded in Java / Golang.***
6. ***Ethereum has an inbuild token, namely, cryptocurrency. But Hyperledger does not have.***

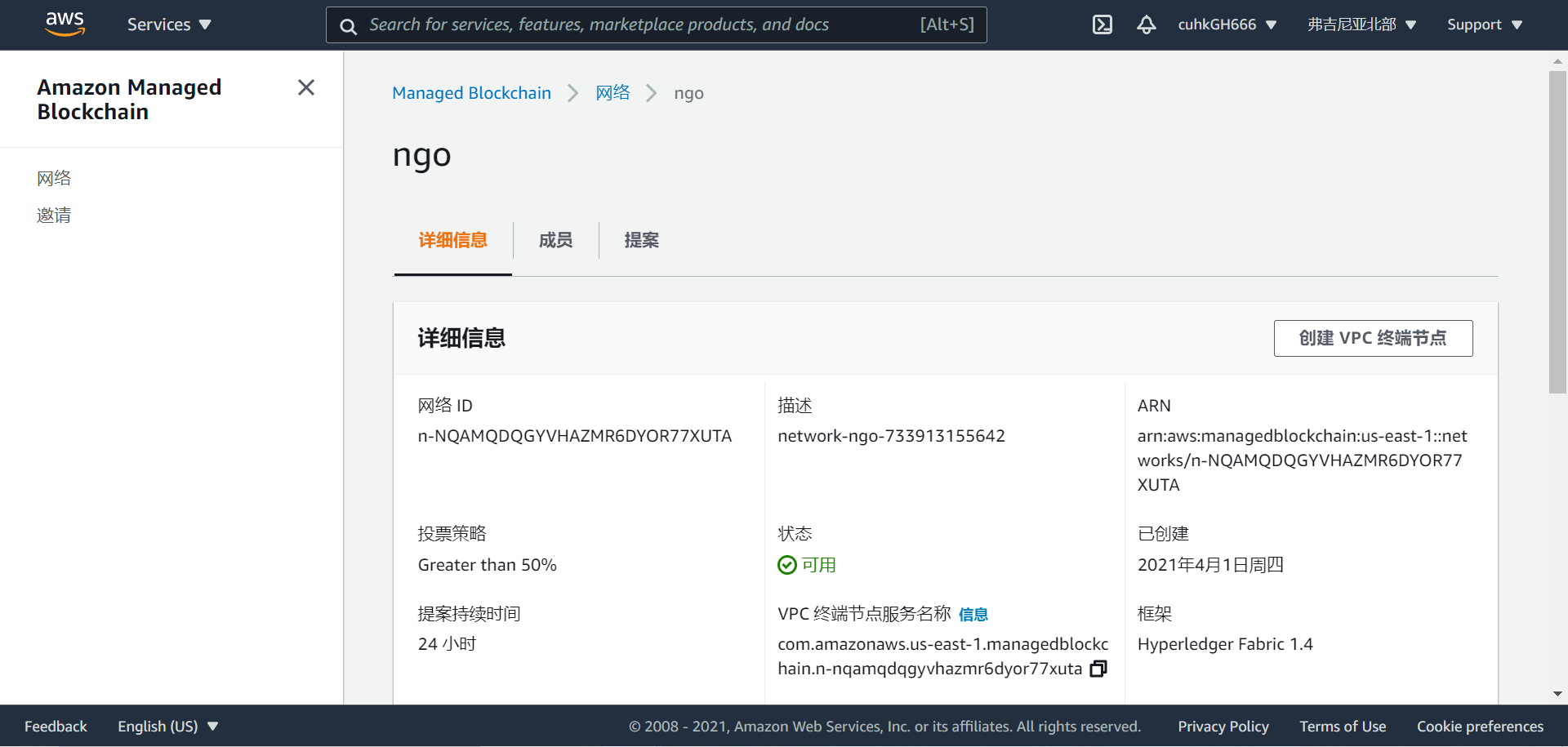
**Note:** Detailed answer expected but no more than 1 page. Please refer to the lecture slides the [article-1](https://medium.com/@philippsandner/comparison-of-ethereum-hyperledger-fabric-and-corda-21c1bb9442f6), [article-2](https://medium.com/quillhash/ethereum-or-hyperledger-fabric-259f3c9b8da6) and try to merge some contents of previous questions if necessary.

# 2 Screen Capture of Main Steps:

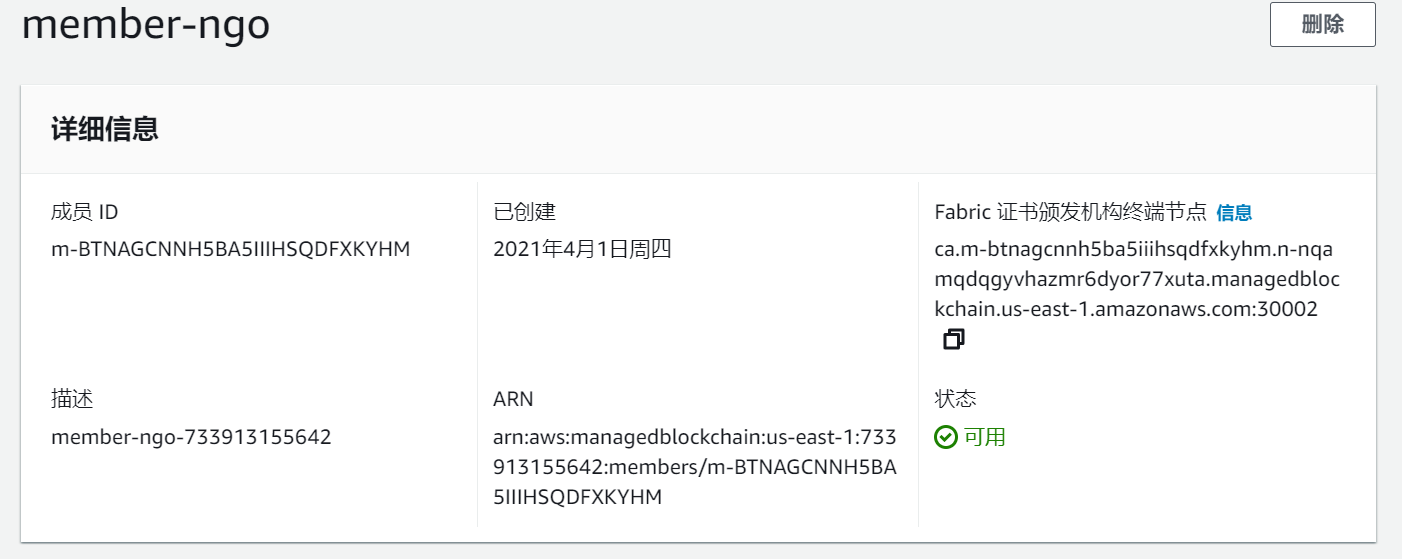
Please replace the sample photos with your own results.

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

1. Your Hyperledger Fabric blockchain network on AmazonManagedBlockchain dashboard:
2. Detail info about the Managed Blockchain – ngo:



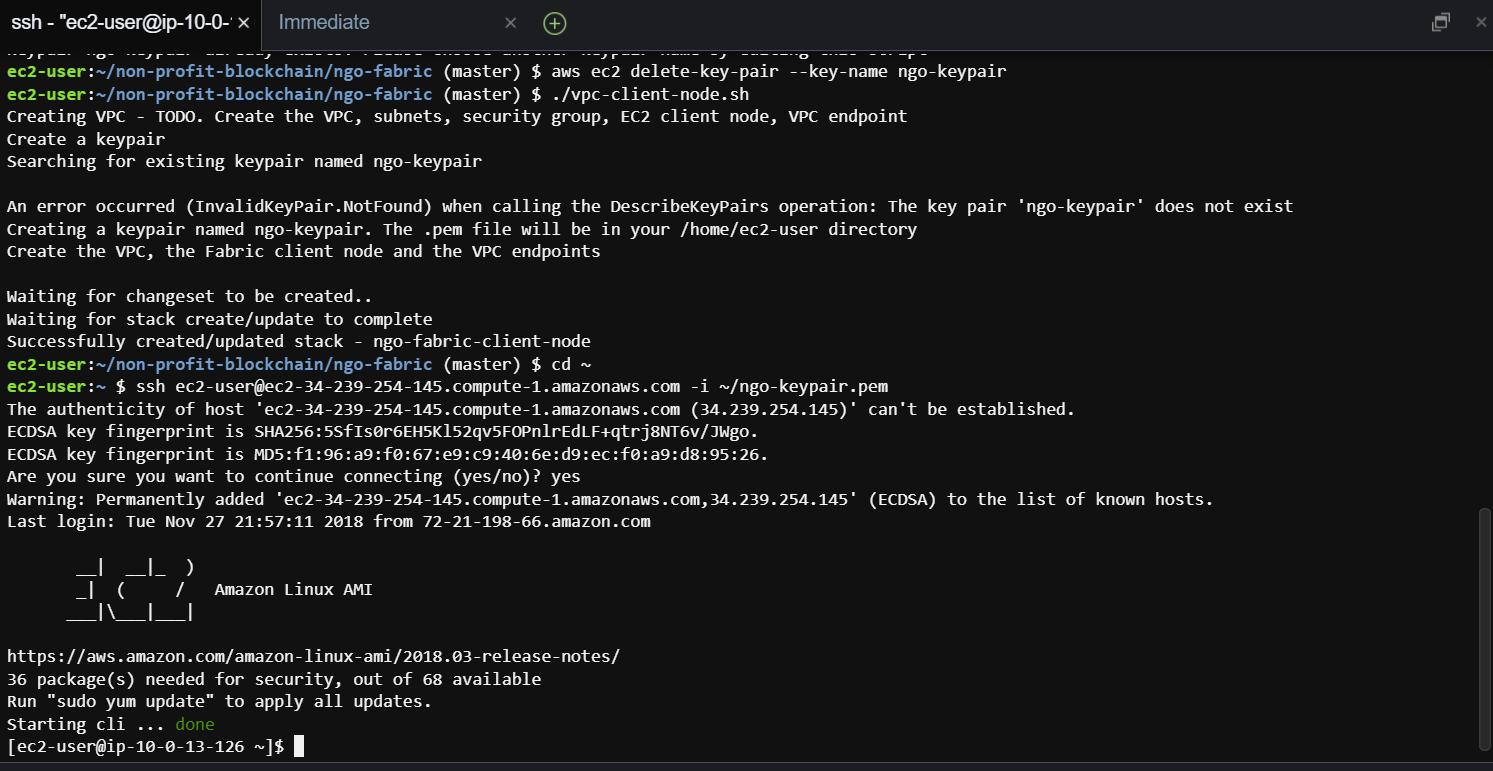
1. Detail info about the member of ngo:

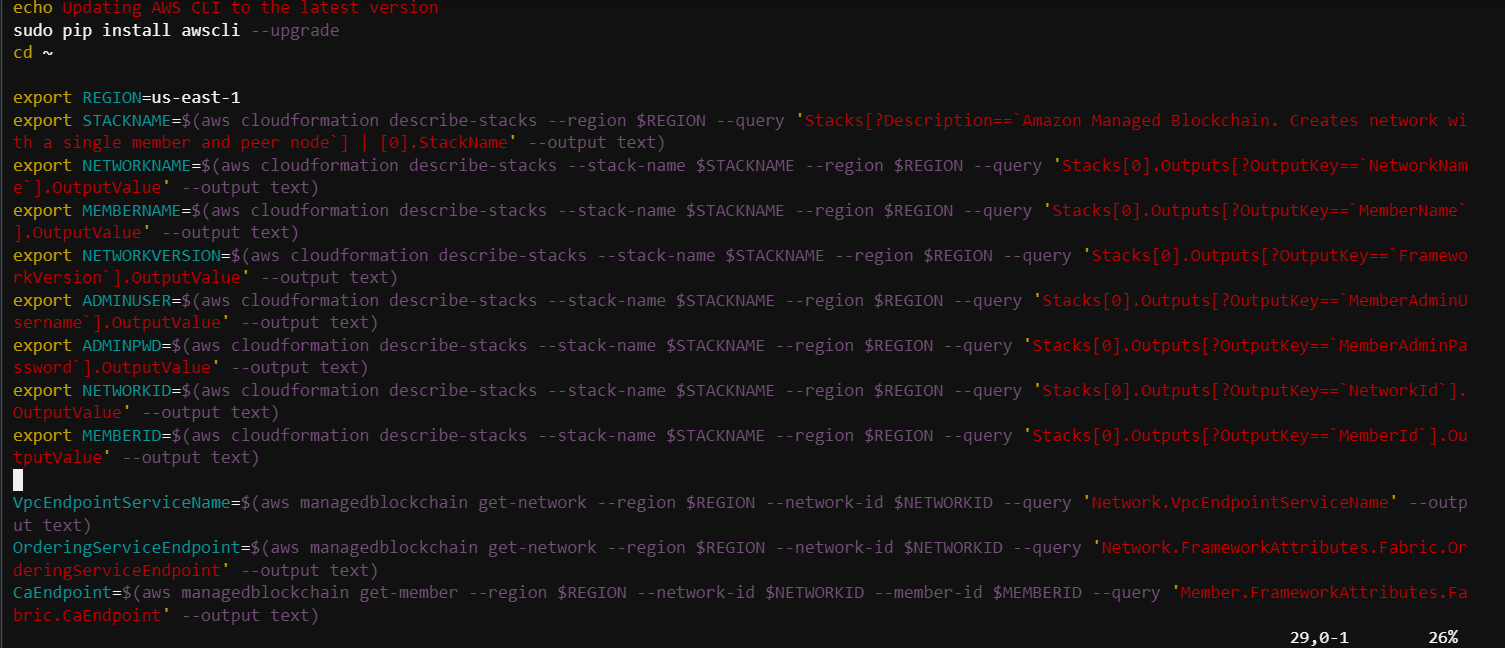
1. Detail info about the peer node:



1. SSH from the Cloud9 IDE to peer node. You should know they are two VMs.
2. Connect to client node with the key:



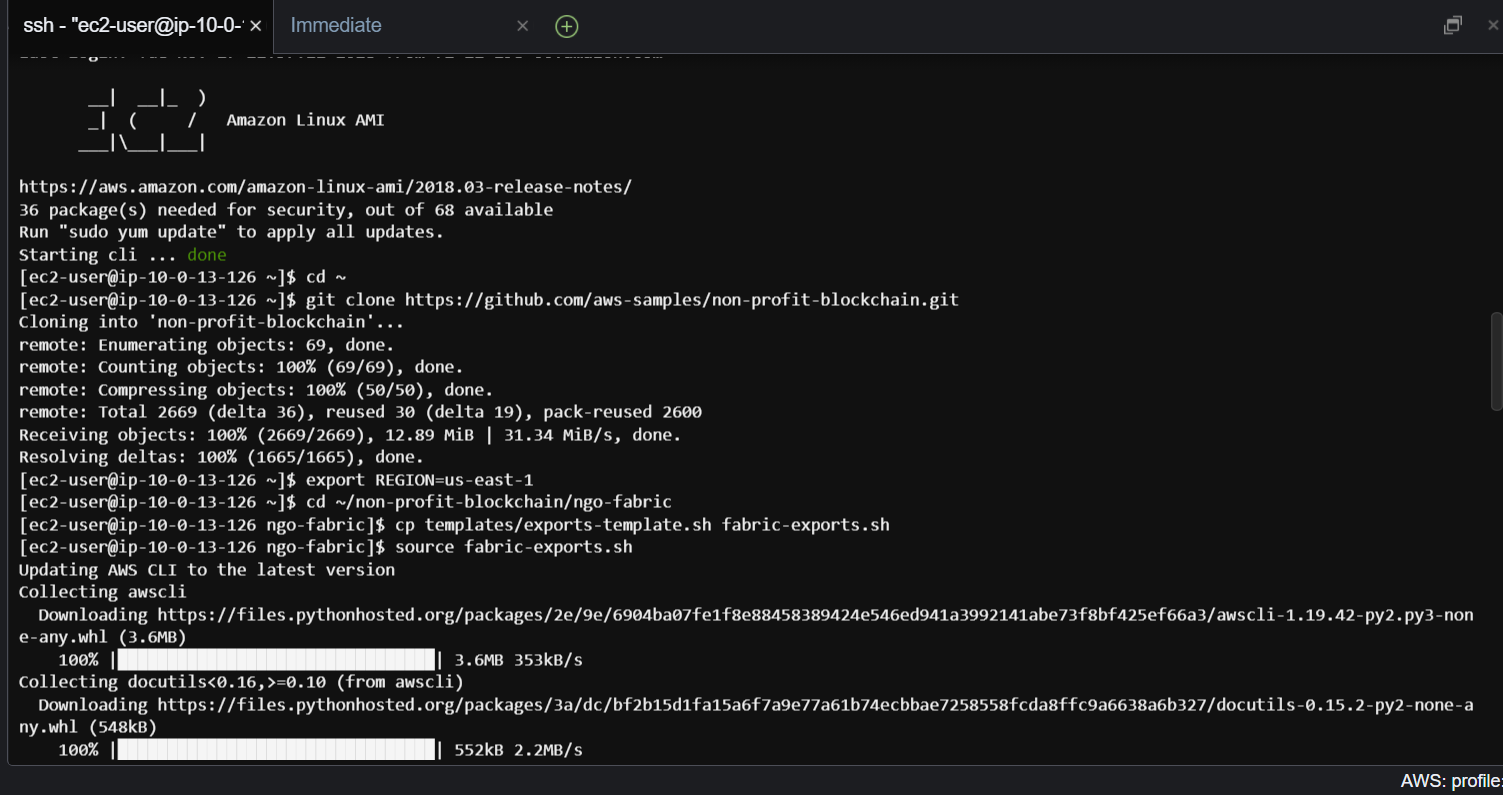
1. The content of **fabric-exports.sh** after insert your blockchain network information
2. Show the content of file fabric-exports.sh to be sourced, fabric-exports.sh records:



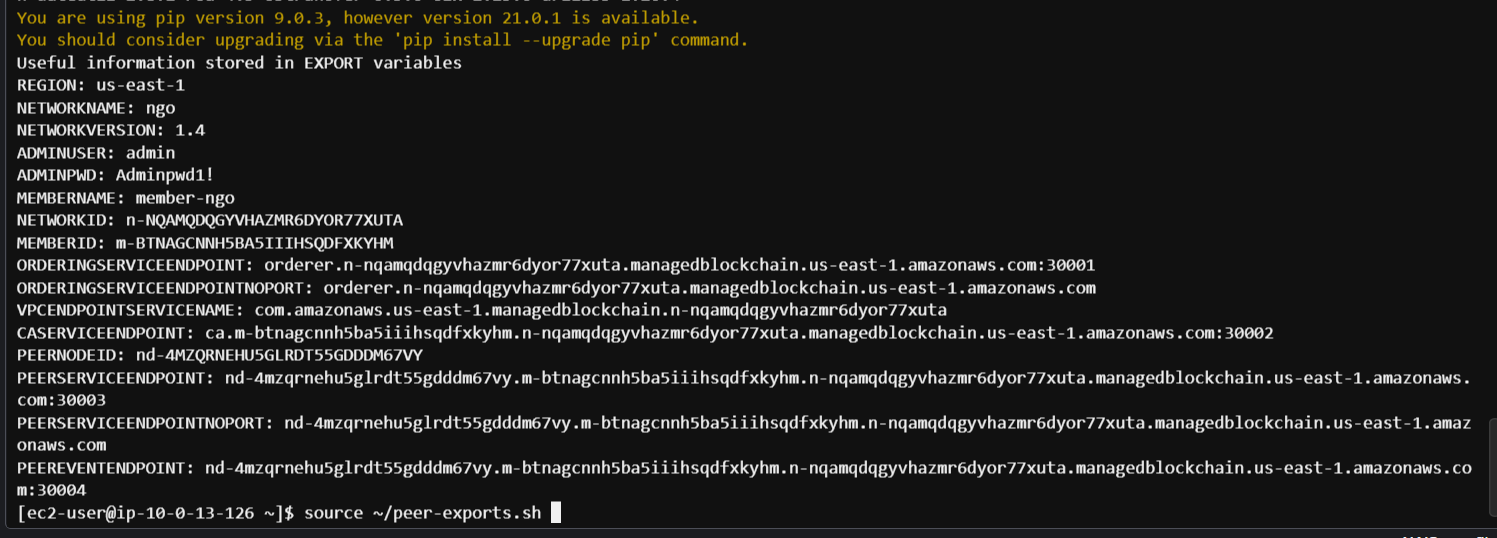
1. The useful information after you ran the command in task3 step 2.16

source fabric-exports.sh

1. Installation records:



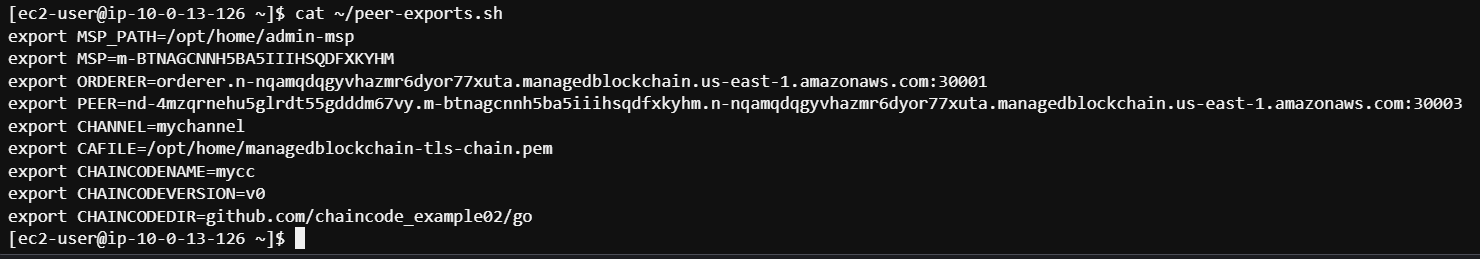
1. Then it shows the EXPORT info stored:



1. Use this command to check the peer export information:

cat ~/peer-exports.sh

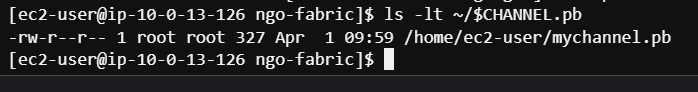
1. Check the peer-exports file:



1. List the generated configuration using

ls -lt ~/$CHANNEL.pb

1. Check the channel configuration:



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

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

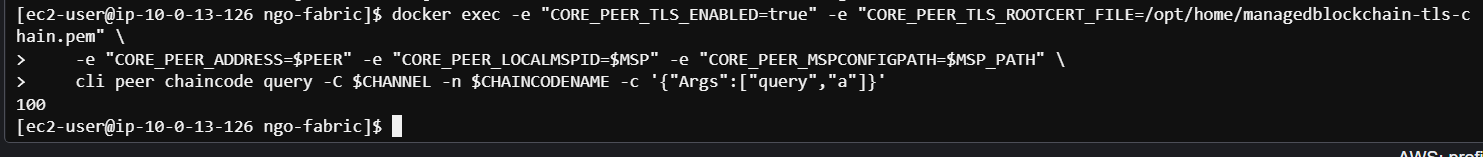
1. See the block file generated:



1. 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"]}'

1. Query the chaincode on fabric peer:



1. Invoke a transaction to transfer 10 dollars from Account a to Account b:

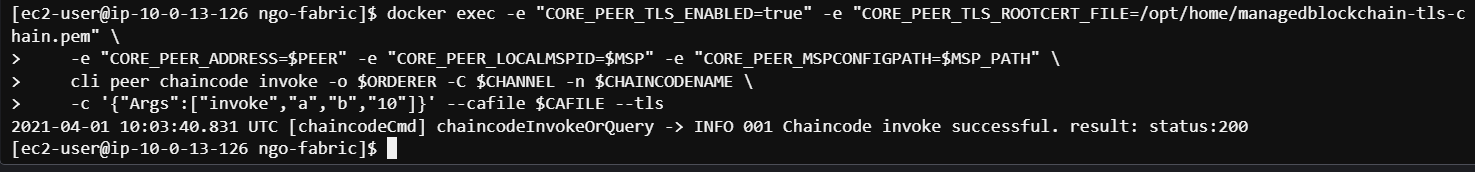
docker exec -e "CORE\_PEER\_TLS\_ENABLED=true" -e "CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/home/managedblockchain-tls-chain.pem" **\**

-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

a) Invoke a fabric transaction:



1. 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"]}'

1. Check the change of ‘a’ after above transaction:

