Documentation of hyperledger-fabric

Prerequisites Setup

VM installation guide - https://www.youtube.com/watch?v=x5MhydijWmc

Hyperledger Fabric Prerequisites Setup

1. Curl Installation
2. NodeJs Installation
3. Git Installation
4. Python Installation
5. Libtool
6. Docker CE
7. Docker Compose

Curl Installation

Run below command to install Curl.

```
sudo apt-get install curl
```

Verify the installation and check the version of Curl using the below command.

```
curl --version
```

NodeJs Installation

Open the terminal window and run the below command to download and execute the node js file.

```
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
```

Then run below command.

```
sudo apt-get update
```

Run the below command to start the installation for NodeJs.

```
sudo apt-get install nodejs
```

Run the below command to check if Nodejs is successfully installed or not. This should return the version of NodeJs.

```
node --version
```

Git Installation

Open the terminal window and run below command. This will start the installation for Git.

```
sudo apt-get install git
```

Run the command below to check if Git is successfully installed or not. This should return the version of Git.

```
git --version
```

Python Installation

In the terminal window, run the command below to install Python.

```
sudo apt-get install python
```

Verify the installation by running below command and that should return the version of Python.

```
python --version
```

Lib Tools Installation

Install Lib tools using the below command.

```
sudo apt-get install libltdl-dev
```

Install Docker CE (Community Edition)

First download and then install it using below commands.

```
wget
https://download.docker.com/linux/ubuntu/dists/xenial/pool/
stable/amd64/docker-ce_18.06.3~ce~3-0~ubuntu_amd64.deb
```

```
sudo dpkg -i docker-ce_18.06.3~ce~3-0~ubuntu_amd64.deb
```

Check the version of docker using the below command and this should return the version of docker.

```
docker --version
```

Install Docker Compose

Run below commands to set up Docker compose.

```
sudo apt-get install python-pip
```

```
pip --version
```

```
sudo pip install docker-compose
```

Verify the installation and check the version from below command.

```
docker-compose version
```

Hyperledger Installation

Step 1: Run below command to download and set up Fabric.

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

You may encounter the below issue when you run the above command.

```
failed to get default registry endpoint from daemon (Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock:
```

To fix this you need to run the command below.

```
sudo chmod 666 /var/run/docker.sock
```

start your test-network and create CA

Step 1: Go to the fabric-samples folder by using the below command.

```
cd fabric-samples
```

Step 2: Go to the test-network folder by using the below command.

```
cd test-network
```

Step 3: Run below command to start your test-network and create CA container for each organization (one for orderer, one for org1 peer and one for org2 peer)

```
sudo ./network.sh up -ca
```

```
CONTAINER ID        IMAGE                               COMMAND                  CREATED              STATUS                  PO
RTS                 NAMES
67799db77f67        hyperledger/fabric-peer:latest      "peer node start"        7 seconds ago        Up Less than a second   70
51/tcp, 0.0.0.0:9051->9051/tcp   peer0.org2.example.com
10677ec49128        hyperledger/fabric-peer:latest      "peer node start"        7 seconds ago        Up 1 second             0.
0.0.0:7051->7051/tcp             peer0.org1.example.com
ea584da2f05d        hyperledger/fabric-orderer:latest   "orderer"                7 seconds ago        Up 1 second             0.
0.0.0:7050->7050/tcp             orderer.example.com
5a82a9350f0d        hyperledger/fabric-ca:latest        "sh -c 'fabric-ca-..."   38 seconds ago       Up 32 seconds           70
54/tcp, 0.0.0.0:9054->9054/tcp   ca_orderer
4166d278cbd8        hyperledger/fabric-ca:latest        "sh -c 'fabric-ca-..."   38 seconds ago       Up 32 seconds           70
54/tcp, 0.0.0.0:8054->8054/tcp   ca_org2
b039ae22ebc0        hyperledger/fabric-ca:latest        "sh -c 'fabric-ca-..."   38 seconds ago       Up 31 seconds           0.
0.0.0:7054->7054/tcp             ca_org1
```

Step 4: Create a new channel by using the below command.

```
sudo ./network.sh createChannel -c testchannel2
```

This will create a new channel with the name testchannel2.

Step 5: To stop the network, you need to run below command.

```
sudo ./network.sh down
```

Peer channel

Step 1: Go to the fabric-samples folder by using the below command.

```
cd fabric-samples
```

Step 2: Go to the test-network folder by using the below command.

```
cd test-network
```

Step 3: Run below command to start your test-network

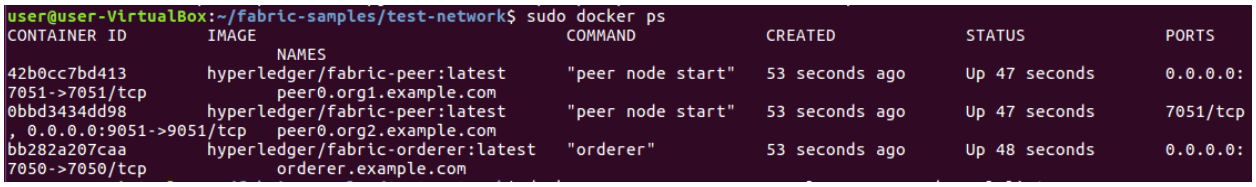
```
sudo ./network.sh up
```

This starts the network, you can run below command to check docker containers.

```
sudo docker ps
```

This shows you three docker containers

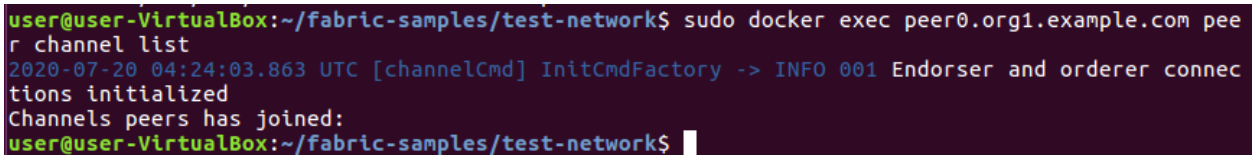
1. One for Org1 peer node
2. One for Org2 peer node
3. One for Orderer



When you start the network, you will also not get any channel by default. You can check the channel by using the below command.

```
sudo docker exec peer0.org1.example.com peer channel list
```

This command shows you that you don't have any channel created.



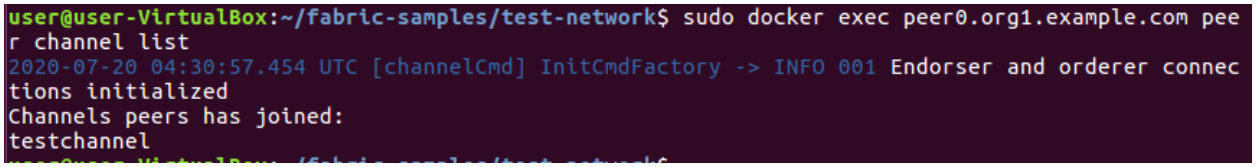
Step 4: Create a new channel by using the below command.

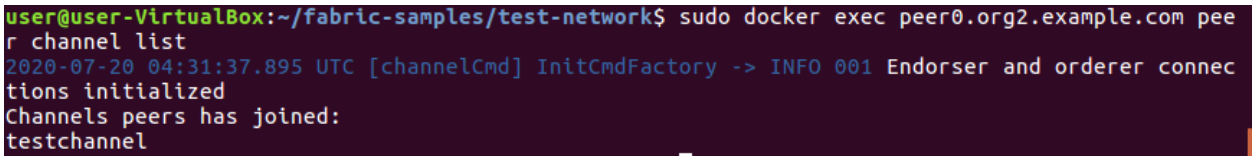
```
sudo ./network.sh createChannel -c testchannel
```

This will create a new channel with the name testchannel.

To verify this channel creation, run below command on both the peers.

```
sudo docker exec peer0.org1.example.com peer channel list
sudo docker exec peer0.org2.example.com peer channel list
```





Step 5: To stop the network, you need to run below command.

```
sudo ./network.sh down
```

Start the network and create couchDB

Step 1: Go to the fabric-samples folder by using the below command.

```
cd fabric-samples
```

Step 2: Go to the test-network folder by using the below command.

```
cd test-network
```

Step 3: Run below command to start the network and create couchDB containers as well.

```
sudo ./network.sh up -s couchdb
```

This command starts your network and creates a couchdb container for each peer as well.

Step 4: Create a new channel by using the below command.

```
sudo ./network.sh createChannel -c testchannel1
```

This will create a new channel with the name testchannel1.

Step 5: To stop the network, you need to run below command.

```
sudo ./network.sh down
```

Error and setup

step 1 - curl -sSL https://bit.ly/2ysbOFE | bash -s 2.2.1
step 2 - ./network.sh up createChannel -ca -c mychannel -s couchdb -i 2.2.1
step 3 - paste orgainization folder of /faber-sample/test-net/oraginstaion in explorer folder
step 4- give permission of orgramization file of in explorer (sudo chmod -R 777 organization).
step 4 - change and check file path in test-network.json
step 5 - up explorer 1.1.18 version (docker-compose up)
step 6 - deploy erc20 contract - (./network.sh deployCC -ccn token_erc20 -ccp ../token-erc-20/chaincode-javascript/ -ccl javascript)
step 7 - then follow all command of this url - https://github.com/hyperledger/fabric-samples/tree/main/token-erc-20

Start the restart the container

docker restart $(docker ps -a -q)

demon container stop solution

sudo systemctl restart docker.socket docker.service

Udemy I'd

ad-priti@mobiloitte.com

Pass
Mobiloitte@123

Hyperledger Fabric.....

sudo ~/fabric-samples/bin/stop.sh

sudo ~/fabric-samples/bin/clean.sh

docker rmi $(docker images | grep "^dev-peer" | awk '{print $3}')

sudo rm -rf ~/fabric-samples

Docker compose.....

sudo systemctl stop docker

sudo apt-get remove docker docker-engine docker.io containerd runc

sudo rm -rf /var/lib/docker

sudo rm /usr/local/bin/docker-compose

Image .....

```
docker rm -f $(docker ps -aq)
docker rmi -f $(docker images -q)
```
docker rmi $(docker images -q)

All image removing

sudo docker rmi -f $(sudo docker images -a -q)

Installation.....

Docker and docker-compose

sudo apt-get update && sudo apt-get install -y docker.io docker-compose

Go Lang

sudo apt-get update && sudo apt-get install -y golang

Nodejs and npm

sudo apt-get update && sudo apt-get install -y nodejs npm

Python

sudo apt-get update && sudo apt-get install -y python

Git

sudo apt-get update && sudo apt-get install -y git

All setup .....

sudo apt-get update && sudo apt-get install -y docker.io docker-compose golang nodejs npm python git

Uninstall.....

sudo apt-get remove --purge docker.io docker-compose golang nodejs npm python git

sudo apt-get remove docker.io docker-compose golang nodejs npm python git

Hyperledger-fabric docs Imp. command

Make sure the Docker daemon is running.

```
sudo systemctl start docker
```

Optional: If you want the Docker daemon to start when the system starts, use the following:

```
sudo systemctl enable docker
```

Add your user to the Docker group.

```
sudo usermod -a -G docker $USER
```

Update the following `git` configurations

```
git config --global core.autocrlf false
```

```
git config --global core.longpaths true
```

Starting a chaincode on the channel

Golang

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

javascript

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

Interacting with the network

```
export PATH=${PWD}/../bin:$PATH
export FABRIC_CFG_PATH=$PWD/../config/
```

```
# Environment variables for Org1
```

```
export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID="Org1MSP"
export
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/pe
ers/peer0.org1.example.com/tls/ca.crt
export
CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/
Admin@org1.example.com/msp
export CORE_PEER_ADDRESS=localhost:7051
```

Run the following command to initialize the ledger with assets:

```
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride
orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/ms
p/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses
localhost:7051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/
tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/
tls/ca.crt" -c '{"function":"InitLedger","Args":[]}'
```

You can now query the ledger from your CLI. Run the following command to get the list of assets that were added to your channel ledger:

```
peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'
```

Chaincodes are invoked when a network member wants to transfer or change an asset on the ledger. Use the following command to change the owner of an asset on the ledger by invoking the asset-transfer (basic) chaincode:

```
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride
orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/ms
p/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses
localhost:7051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/
tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/
tls/ca.crt" -c '{"function":"TransferAsset","Args":["asset6","Christopher"]}'
```

```
# Environment variables for Org2
```

```
export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID="Org2MSP"
export
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org2.example.com/pe
ers/peer0.org2.example.com/tls/ca.crt
export
CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org2.example.com/users/
Admin@org2.example.com/msp
export CORE_PEER_ADDRESS=localhost:9051
```

You can now query the asset-transfer (basic) chaincode running on `peer0.org2.example.com`:

```
peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAsset","asset6"]}'
```

The result will show that `"asset6"` was transferred to Christopher:

```
{"ID":"asset6","color":"white","size":15,"owner":"Christopher","appraisedValue":800}
```

Bring down the network

```
./network.sh down
```

Adding Multi Org

Running the test network

This modified test-network only supports working with exactly 4 organizations.

How to run:

1. **Copy the test-network to your fabric-samples folder, and cd to fabric-samples**

```
cd test-network
./network.sh up createChannel -s couchdb
cd addOrg3
./addOrg3.sh up -s couchdb
cd ../addOrg4
./addOrg4.sh up -s couchdb
cd ..
```

Adding Multi peer..

Creating channel

```
./network.sh up createChannel -ca -c mychannel -s couchdb
```

Installing Chaincode

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

Setting up environment variable

```
export PATH=${PWD}/../bin:$PATH
export FABRIC_CFG_PATH=${PWD}/../config

export
ORDERER_CA=${PWD}/organizations/ordererOrganizations/example.com/orderers/orde
rer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

export
CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.co
m/users/Admin@org1.example.com/msp

export CORE_PEER_ADDRESS=localhost:7051

export
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.exampl
e.com/peers/peer0.org1.example.com/tls/ca.crt

export CORE_PEER_LOCALMSPID=Org1MSP
```

Invoke CC

```
peer chaincode invoke -n basic -C mychannel -o localhost:7050
--ordererTLSHostnameOverride orderer.example.com  --tls --cafile "$ORDERER_CA"
--peerAddresses localhost:9051 --tlsRootCertFiles
${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer1.org1.examp
le.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles
${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.examp
le.com/tls/ca.crt  -c '{"Args":["CreateAsset", "100","red",
"20","aditya","100"]}'
```

Query CC

```
peer chaincode query -n basic -C mychannel -o localhost:7050
--ordererTLSHostnameOverride orderer.example.com  -c '{"Args":["ReadAsset",
"100"]}'
```

Setting up the new peer

Creating MSP Identities for peer1.org1.example.com

```
./organizations/fabric-ca/registerPeer1.sh
```

Starting up the peer container

```
docker-compose -f docker/docker-compose-peer1.yaml up -d
```

Joining existing channel

Query channel on peer0.org1.example.com

```
peer channel list
peer channel fetch -c mychannel newest
CORE_PEER_ADDRESS=localhost:8051 peer channel getinfo -c mychannel
```

Query channel on peer1.org1.example.com

```
CORE_PEER_ADDRESS=localhost:8051 peer channel list
CORE_PEER_ADDRESS=localhost:8051 peer channel fetch -c mychannel newest
CORE_PEER_ADDRESS=localhost:8051 peer channel getinfo -c mychannel
```

Join channel

```
CORE_PEER_ADDRESS=localhost:8051 peer channel join -b
./channel-artifacts/mychannel.block
```

Query channel on peer1.org1.example.com

```
CORE_PEER_ADDRESS=localhost:8051 peer channel list
CORE_PEER_ADDRESS=localhost:8051 peer channel fetch -c mychannel newest
CORE_PEER_ADDRESS=localhost:8051 peer channel getinfo -c mychannel
```

Chaincode Setup

Install CC

```
export CC_NAME=basic
```

```
CORE_PEER_ADDRESS=localhost:8051 peer lifecycle chaincode install
${CC_NAME}.tar.gz
```

Query installed CC

```
CORE_PEER_ADDRESS=localhost:8051 peer lifecycle chaincode queryinstalled
```

Invoke CC

```
CORE_PEER_ADDRESS=localhost:8051 peer chaincode invoke -n basic -C mychannel
-o localhost:7050 --ordererTLSHostnameOverride orderer.example.com  --tls
--cafile "$ORDERER_CA" --peerAddresses localhost:8051 --tlsRootCertFiles
${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer1.org1.examp
le.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles
${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.examp
le.com/tls/ca.crt  -c '{"Args":["CreateAsset", "200","red",
"20","aditya","100"]}'
```

Query CC

```
CORE_PEER_ADDRESS=localhost:8051 peer chaincode query -n basic -C mychannel -o
localhost:7050 --ordererTLSHostnameOverride orderer.example.com  -c
'{"Args":["ReadAsset", "200"]}'
```

Setup the hyperledger fabric and hyperledger explorer