NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications
Bishakh Chandra Ghosh

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 35: Hyperledger Fabric 2

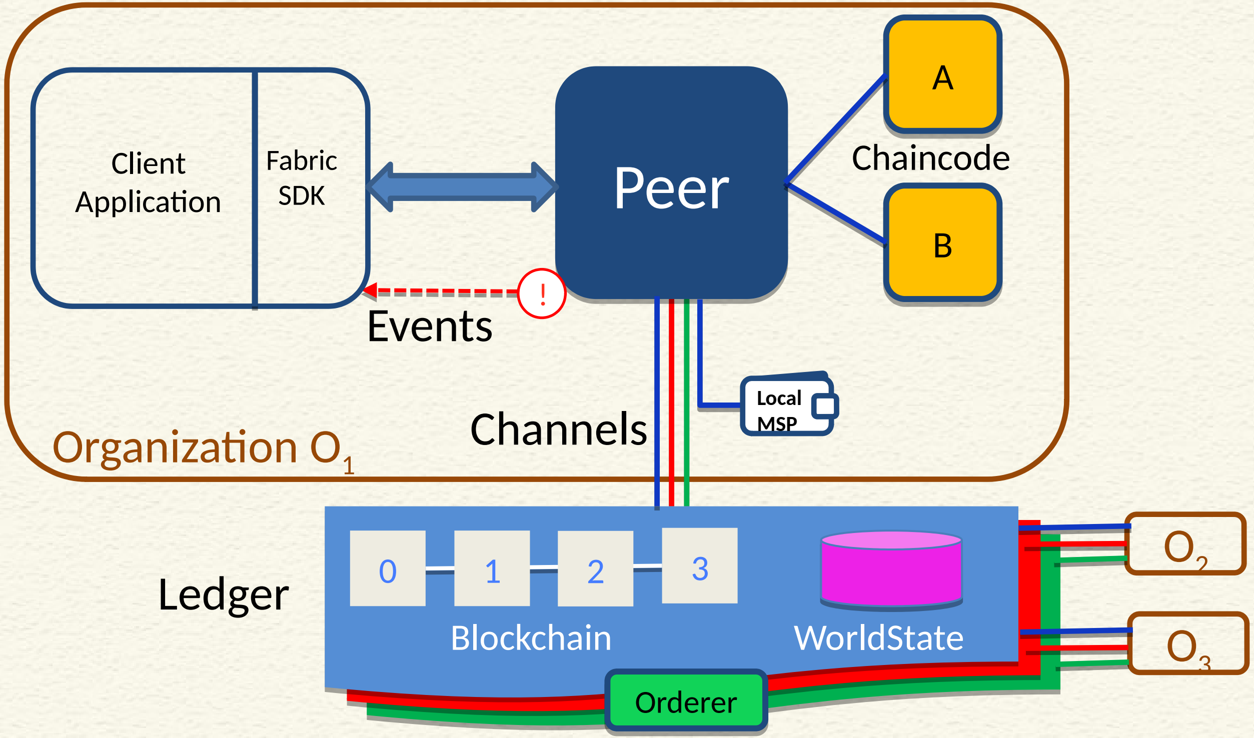
CONCEPTS COVERED

- Fabric Architecture
- Fabric Test Network
- Sample Chaincode
- Invoke and Query Chaincode

KEYWORDS

- **Fabric Test Network**
- **Chaincode Transactions**

Fabric Architecture



Fabric Test Network

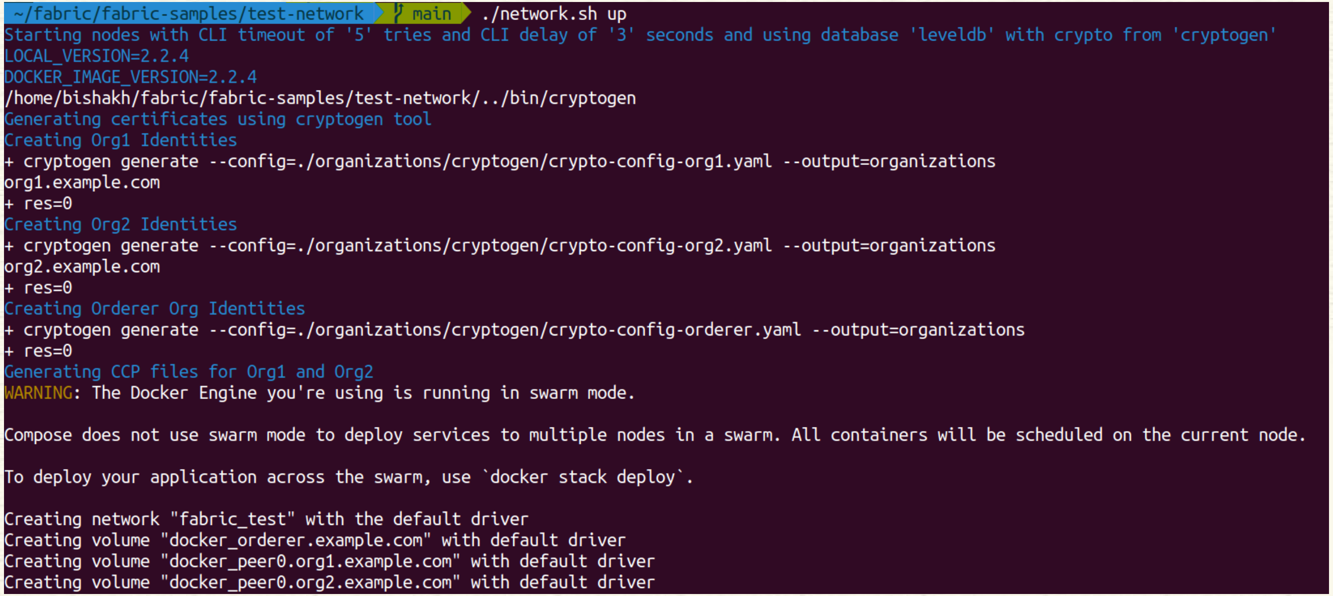
- Real network consists of multiple organizations. Each maintain their own set of:
  - Peers
  - Client Applications
  - Optionally Orderers
  - MSP

- **Test Network**:
  - All organizations in a single system
  - Development and testing purposes
  - 2 Orgs, each having 1 peer and optionally one CA
  - 1 orderer
  - All components are containerized

Start Test Network

Navigate to the directory where you have installed fabric samples.

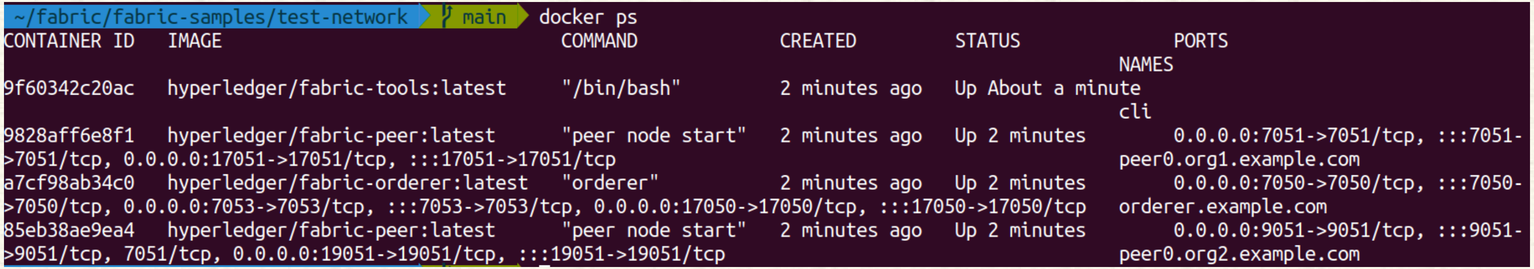
```
cd fabric-samples
cd test-network
./network.sh up
```



```
~/fabric/fabric-samples/test-network  main  ./network.sh up
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb' with crypto from 'cryptogen'
LOCAL_VERSION=2.2.4
DOCKER_IMAGE_VERSION=2.2.4
/home/bishakh/fabric/fabric-samples/test-network/../bin/cryptogen
Generating certificates using cryptogen tool
Creating Org1 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organizations
org1.example.com
+ res=0
Creating Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
org2.example.com
+ res=0
Creating Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
Generating CCP files for Org1 and Org2
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm. All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

Creating network "fabric_test" with the default driver
Creating volume "docker_orderer.example.com" with default driver
Creating volume "docker_peer0.org1.example.com" with default driver
Creating volume "docker_peer0.org2.example.com" with default driver
```

Monitor Containers

```
docker ps
```



```
~/fabric/fabric-samples/test-network  main  docker ps
CONTAINER ID   IMAGE                               COMMAND             CREATED         STATUS              PORTS
                                                                                                      NAMES
9f60342c20ac   hyperledger/fabric-tools:latest     "/bin/bash"         2 minutes ago   Up About a minute
                                                                                                      cli
9828aff6e8f1   hyperledger/fabric-peer:latest      "peer node start"   2 minutes ago   Up 2 minutes        0.0.0.0:7051->7051/tcp, :::7051-
>7051/tcp, 0.0.0.0:17051->17051/tcp, :::17051->17051/tcp                                              peer0.org1.example.com
a7cf98ab34c0   hyperledger/fabric-orderer:latest   "orderer"           2 minutes ago   Up 2 minutes        0.0.0.0:7050->7050/tcp, :::7050-
>7050/tcp, 0.0.0.0:7053->7053/tcp, :::7053->7053/tcp, 0.0.0.0:17050->17050/tcp, :::17050->17050/tcp   orderer.example.com
85eb38ae9ea4   hyperledger/fabric-peer:latest      "peer node start"   2 minutes ago   Up 2 minutes        0.0.0.0:9051->9051/tcp, :::9051-
>9051/tcp, 7051/tcp, 0.0.0.0:19051->19051/tcp, :::19051->19051/tcp                                    peer0.org2.example.com
```

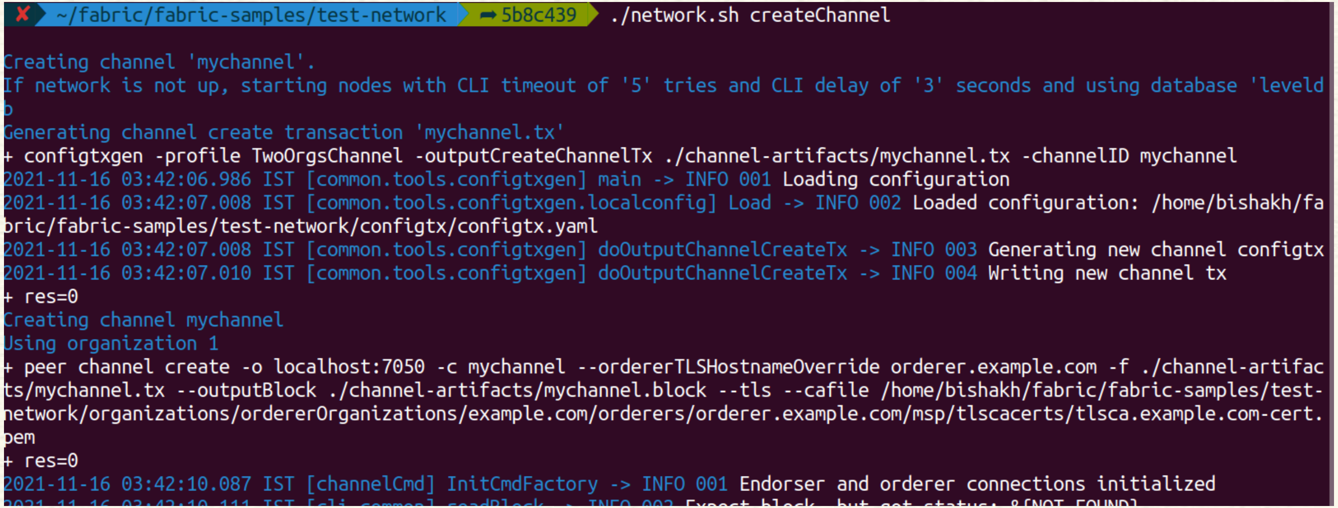
- 2 fabric-peer containers, - 1 per organization.
- 1 fabric-orderer container
- 1 fabric-tools container

Create Channel

```
./network.sh createChannel
```

- Creates a channel with name "**mychannel**"

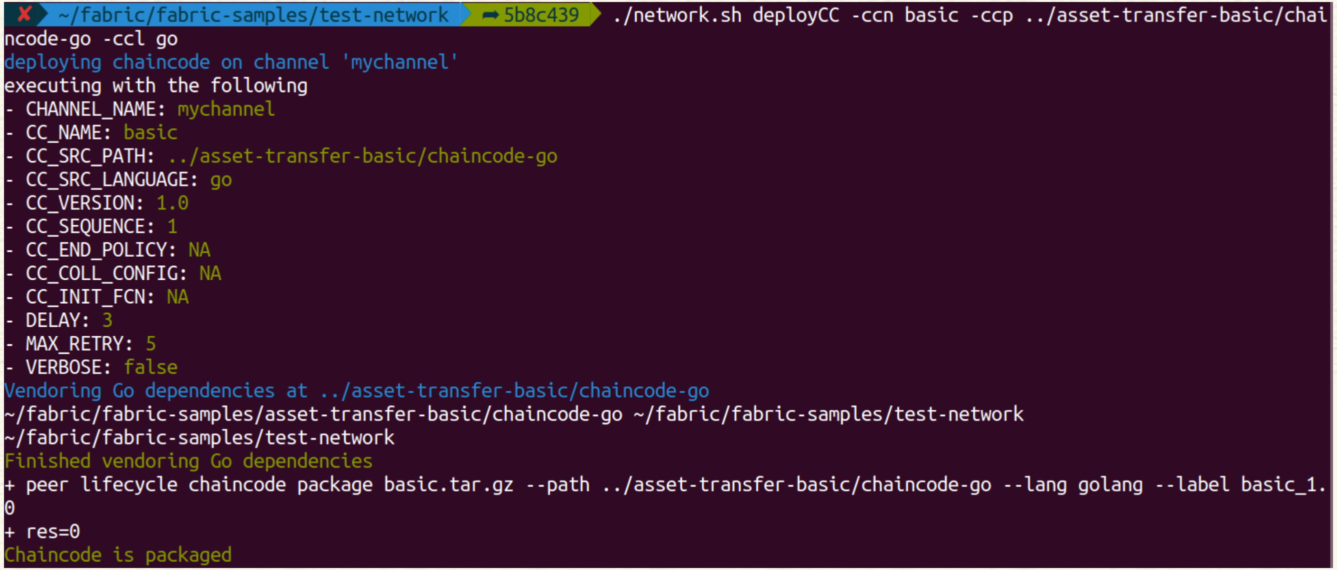
```
./network.sh createChannel -c <channel name>
```



```
✘ ~/fabric/fabric-samples/test-network ➦ 5b8c439  ./network.sh createChannel

Creating channel 'mychannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveld
b
Generating channel create transaction 'mychannel.tx'
+ configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/mychannel.tx -channelID mychannel
2021-11-16 03:42:06.986 IST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-11-16 03:42:07.008 IST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/bishakh/fa
bric/fabric-samples/test-network/configtx/configtx.yaml
2021-11-16 03:42:07.008 IST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
2021-11-16 03:42:07.010 IST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
+ res=0
Creating channel mychannel
Using organization 1
+ peer channel create -o localhost:7050 -c mychannel --ordererTLSHostnameOverride orderer.example.com -f ./channel-artifac
ts/mychannel.tx --outputBlock ./channel-artifacts/mychannel.block --tls --cafile /home/bishakh/fabric/fabric-samples/test-
network/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.
pem
+ res=0
2021-11-16 03:42:10.087 IST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2021-11-16 03:42:10.111 IST [cli.common] readBlock -> INFO 002 Expect block, but got status: &{NOT_FOUND}
```

Install Chaincode

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



```
✘ ~/fabric/fabric-samples/test-network ➦ 5b8c439 ./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chai
ncode-go -ccl go
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: 1
- CC_END_POLICY: NA
- CC_COLL_CONFIG: NA
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
Vendoring Go dependencies at ../asset-transfer-basic/chaincode-go
~/fabric/fabric-samples/asset-transfer-basic/chaincode-go ~/fabric/fabric-samples/test-network
~/fabric/fabric-samples/test-network
Finished vendoring Go dependencies
+ peer lifecycle chaincode package basic.tar.gz --path ../asset-transfer-basic/chaincode-go --lang golang --label basic_1.
0
+ res=0
Chaincode is packaged
```

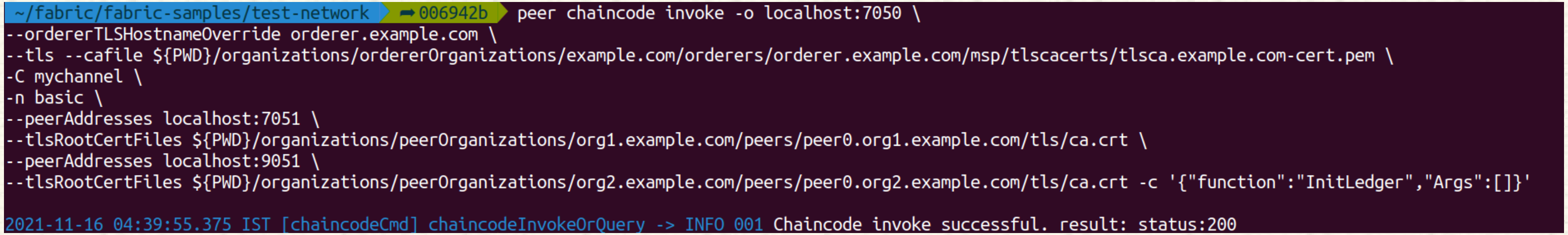
Invoke Chaincode – Configure Peer

```
export FABRIC_CFG_PATH=$PWD/../config/

# Set 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/peers/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
```

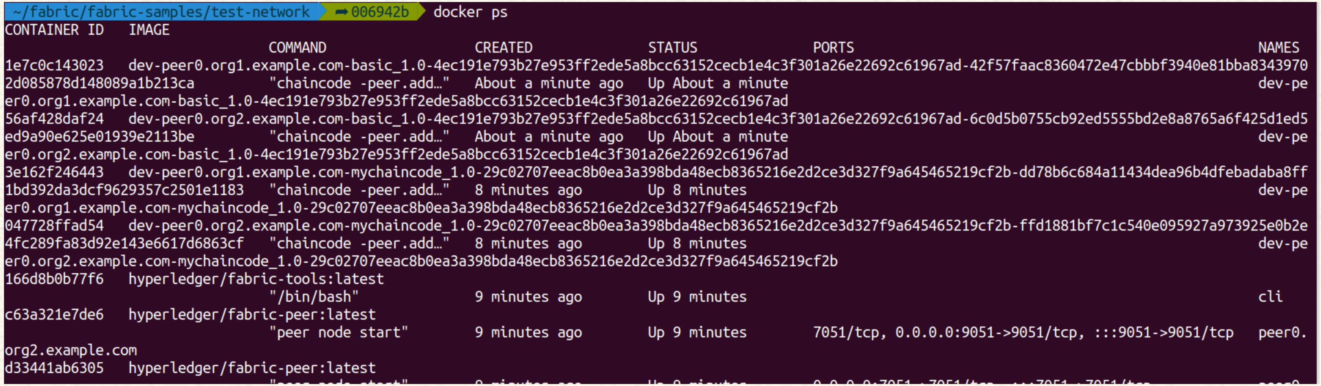
Invoke Chaincode

```
peer chaincode invoke -o localhost:7050 \
--ordererTLSHostnameOverride orderer.example.com \
--tls --cafile
${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/
msp/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":[]}'
```

Invoke Chaincode



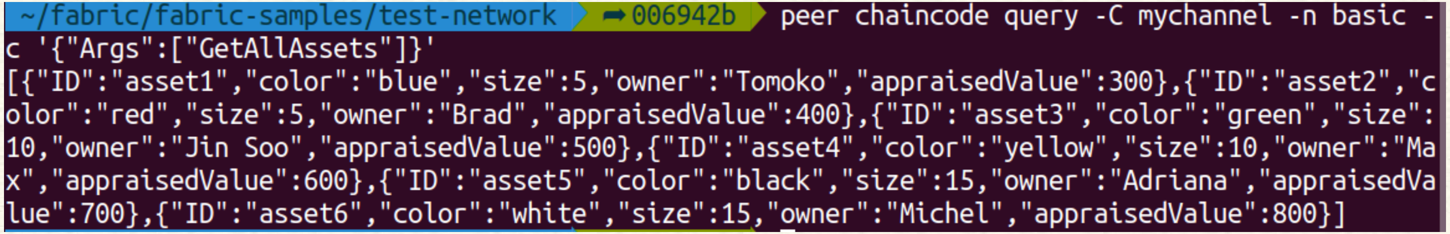
```
 ~/fabric/fabric-samples/test-network  ➡006942b  peer chaincode invoke -o localhost:7050 \
--ordererTLSHostnameOverride orderer.example.com \
--tls --cafile ${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem \
-C mychannel \
-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":[]}'

2021-11-16 04:39:55.375 IST [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
```



```
 ~/fabric/fabric-samples/test-network  ➡006942b  docker ps
CONTAINER ID   IMAGE
                                COMMAND                  CREATED               STATUS               PORTS                                                  NAMES
1e7c0c143023   dev-peer0.org1.example.com-basic_1.0-4ec191e793b27e953ff2ede5a8bcc63152cecb1e4c3f301a26e22692c61967ad-42f57faac8360472e47cbbbf3940e81bba8343970
2d085878d148089a1b213ca         "chaincode -peer.add…"   About a minute ago   Up About a minute                                                         dev-pe
er0.org1.example.com-basic_1.0-4ec191e793b27e953ff2ede5a8bcc63152cecb1e4c3f301a26e22692c61967ad
56af428daf24   dev-peer0.org2.example.com-basic_1.0-4ec191e793b27e953ff2ede5a8bcc63152cecb1e4c3f301a26e22692c61967ad-6c0d5b0755cb92ed5555bd2e8a8765a6f425d1ed5
ed9a90e625e01939e2113be         "chaincode -peer.add…"   About a minute ago   Up About a minute                                                         dev-pe
er0.org2.example.com-basic_1.0-4ec191e793b27e953ff2ede5a8bcc63152cecb1e4c3f301a26e22692c61967ad
3e162f246443   dev-peer0.org1.example.com-mychaincode_1.0-29c02707eeac8b0ea3a398bda48ecb8365216e2d2ce3d327f9a645465219cf2b-dd78b6c684a11434dea96b4dfebadaba8ff
1bd392da3dcf9629357c2501e1183   "chaincode -peer.add…"   8 minutes ago        Up 8 minutes                                                              dev-pe
er0.org1.example.com-mychaincode_1.0-29c02707eeac8b0ea3a398bda48ecb8365216e2d2ce3d327f9a645465219cf2b
047728ffad54   dev-peer0.org2.example.com-mychaincode_1.0-29c02707eeac8b0ea3a398bda48ecb8365216e2d2ce3d327f9a645465219cf2b-ffd1881bf7c1c540e095927a973925e0b2e
4fc289fa83d92e143e6617d6863cf   "chaincode -peer.add…"   8 minutes ago        Up 8 minutes                                                              dev-pe
er0.org2.example.com-mychaincode_1.0-29c02707eeac8b0ea3a398bda48ecb8365216e2d2ce3d327f9a645465219cf2b
166d8b0b77f6   hyperledger/fabric-tools:latest
                                "/bin/bash"              9 minutes ago        Up 9 minutes                                                              cli
c63a321e7de6   hyperledger/fabric-peer:latest
                                "peer node start"        9 minutes ago        Up 9 minutes        7051/tcp, 0.0.0.0:9051->9051/tcp, :::9051->9051/tcp   peer0.
org2.example.com
d33441ab6305   hyperledger/fabric-peer:latest
                                "peer node start"        9 minutes ago        Up 9 minutes        0.0.0.0:7051->7051/tcp, :::7051->7051/tcp             peer0.
```

Query Chaincode

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



```
~/fabric/fabric-samples/test-network  ➡006942b  peer chaincode query -C mychannel -n basic -
c '{"Args":["GetAllAssets"]}'
[{"ID":"asset1","color":"blue","size":5,"owner":"Tomoko","appraisedValue":300},{"ID":"asset2","c
olor":"red","size":5,"owner":"Brad","appraisedValue":400},{"ID":"asset3","color":"green","size":
10,"owner":"Jin Soo","appraisedValue":500},{"ID":"asset4","color":"yellow","size":10,"owner":"Ma
x","appraisedValue":600},{"ID":"asset5","color":"black","size":15,"owner":"Adriana","appraisedVa
lue":700},{"ID":"asset6","color":"white","size":15,"owner":"Michel","appraisedValue":800}]
```

Conclusion

- Fabric Test Network
- Query and Invoke Transactions

Thank you