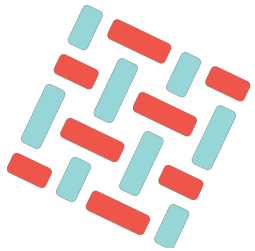
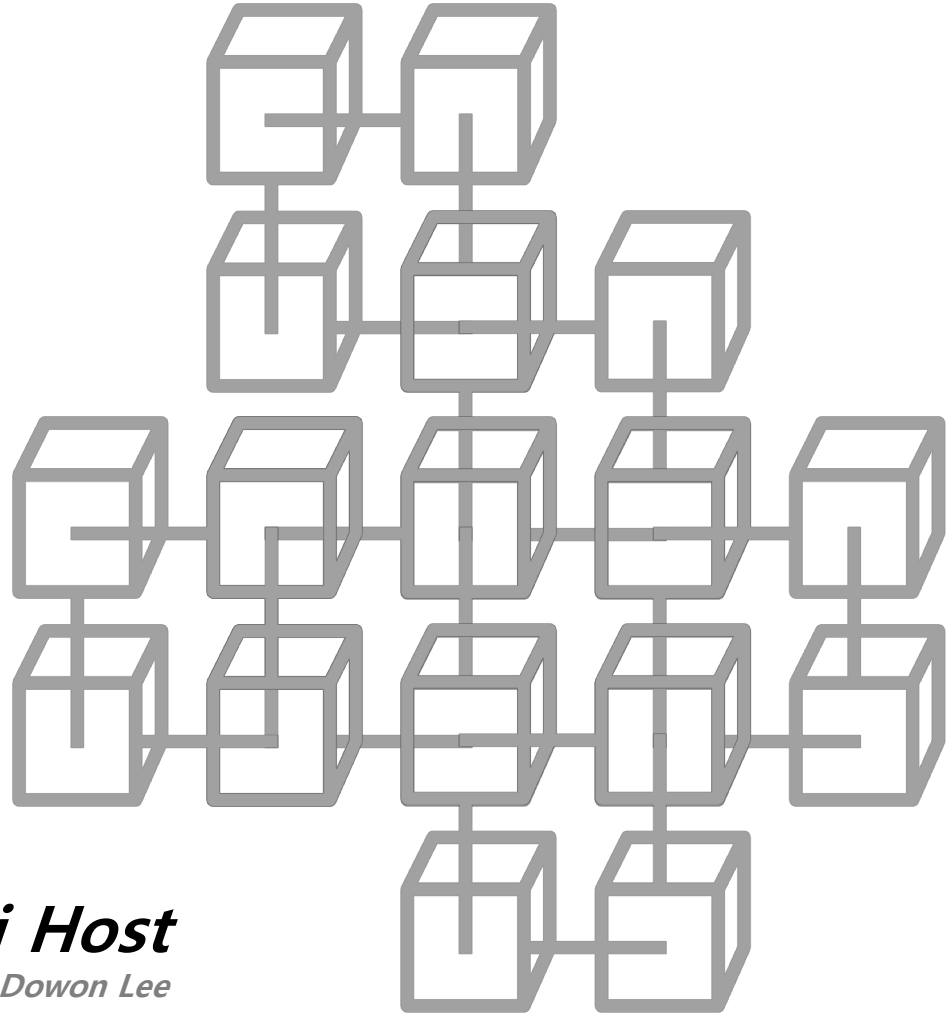


HYPERLEDGER
FABRIC



Hyperledger Fabric – Multi Host

written by Dowon Lee

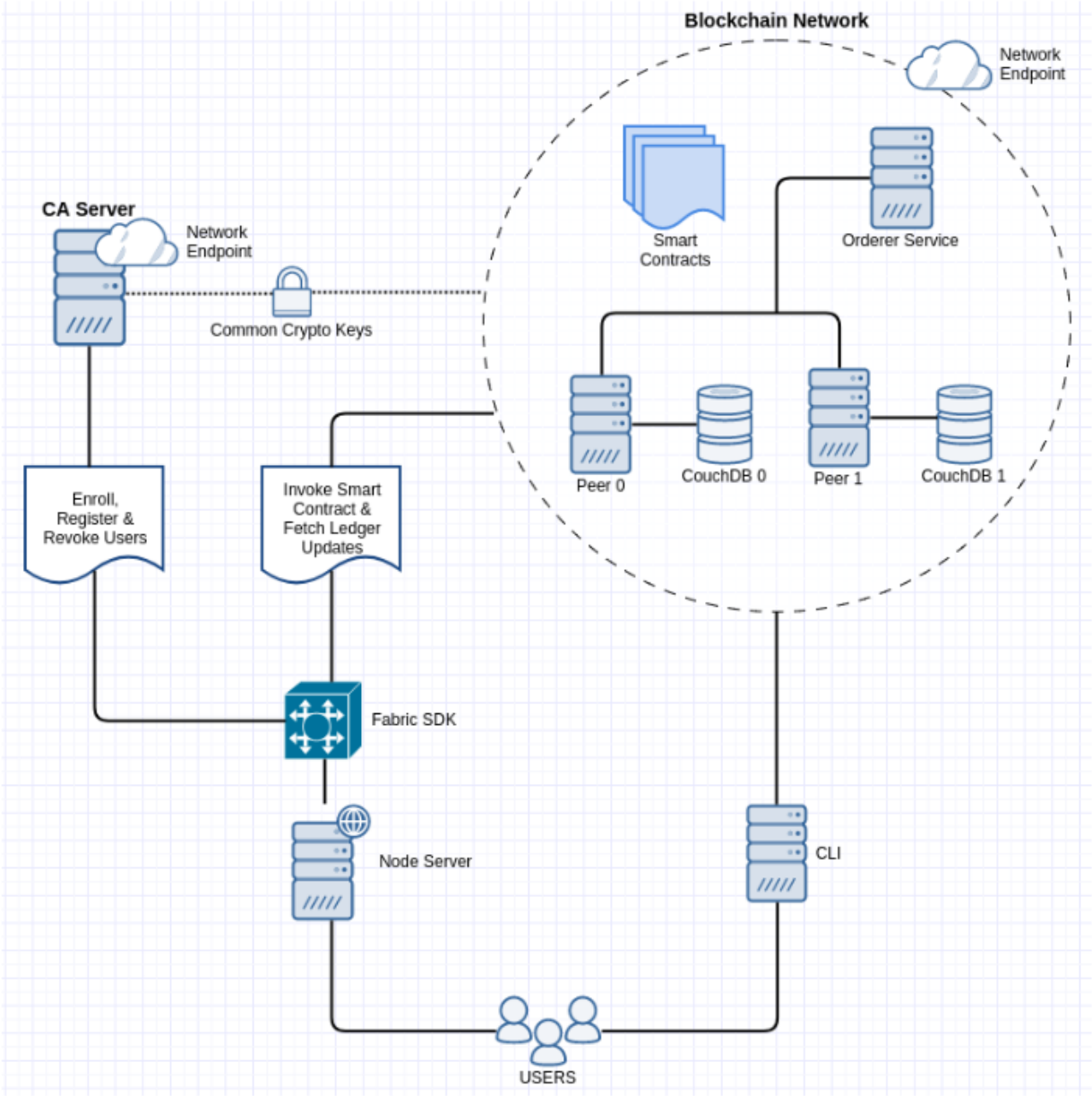
Multi Host 구성

- Build Your First Network를 사용한 멀티티어 구성
  - fabric-samples 디렉토리에서 작업

- Docker의 ***overlay*** network driver
  - 여러 Docker daemon host로 구성된 분산 네트워크를 생성
  - PC1 ➔ docker swarm 초기화
  - PC2, PC3 … ➔ "manager"로써 swarm에 참여
  - 모든 host들이 공유하기 위한 overlay 네트워크 생성 ➔ 오직 swarm을 통해서만 네트워크 상의 존재 확인

Multi Host 구성

- A Certificate Authority (CA) → PC1
- An Orderer → PC1
- 1 PEER (peer0) → PC1
- 1 PEER (peer1) → PC2
- CLI → PC2



Multi Host 구성

- Swarm 초기화

  $ docker swarm init (PC1)

  $ docker swarm join-token manager

```
docker swarm join --token SWMTKN-1-45tb4k2yqdy6j9yshji3tfhwz2w4a1nlky96k7vlfenh020koi-al3lv0i8mbyzafpeyu64tei30 10.0.2.15:237
```

- Join Swarm

  $ docker swarm join --token SWMTKN-1-45tb4k2yqdy6j9yshji3tfhwz2w4a1nlky96k7vlfenh020koi-

  al3lv0i8mbyzafpeyu64tei30 10.0.2.15:237 (PC2)

Multi Host 구성

- Network 생성

  $ docker network create --attachable --driver overlay my-net (PC1)

- Clone repository on PC1, PC2

  $ git clone https://github.com/joneconsulting/hlf.git

  $ cd hlf/multi_host/

  $ ./bmhn.sh

  - ***crypto-config, channel-artifacts 디렉토리 생성***
  - ***생성된 디렉토리를 다른 호스트(PC2, PC3 ...)에 복사***

Multi Host 구성 – PC1

1. CA Server

- ***{put the name of secret key} 에는 자신의 key 값을 입력***
- ***/crypto-config/peerOrganizations/org1.example.com/ca/***

```
$ docker run --rm -it --network="my-net" --name ca.example.com -p 7054:7054
          -e FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
          -e FABRIC_CA_SERVER_CA_NAME=ca.example.com
          -e FABRIC_CA_SERVER_CA_CERTFILE=/etc/hyperledger/fabric-ca-server-config/ca.org1.example.com-cert.pem
          -e FABRIC_CA_SERVER_CA_KEYFILE=/etc/hyperledger/fabric-ca-server-config/{put the name of secret key}
          -v $(pwd)/crypto-config/peerOrganizations/org1.example.com/ca/:/etc/hyperledger/fabric-ca-server-config
          -e CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=hyp-net hyperledger/fabric-ca
          sh -c 'fabric-ca-server start -b admin:adminpw -d'
```

Multi Host 구성 – PC1

2. Orderer

```
$ docker run --rm -it --network="my-net" --name orderer.example.com -p 7050:7050
            -e ORDERER_GENERAL_LOGLEVEL=debug -e ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
            -e ORDERER_GENERAL_LISTENPORT=7050 -e ORDERER_GENERAL_GENESISMETHOD=file
            -e ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/orderer.genesis.block
            -e ORDERER_GENERAL_LOCALMSPID=OrdererMSP
            -e ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
            -e ORDERER_GENERAL_TLS_ENABLED=false
            -e CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=my-net
            -v $(pwd)/channel-artifacts/genesis.block:/var/hyperledger/orderer/orderer.genesis.block
            -v $(pwd)/crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/msp:
                                                                    /var/hyperledger/orderer/msp
            -w /opt/gopath/src/github.com/hyperledger/fabric hyperledger/fabric-orderer orderer
```

Multi Host 구성 – PC1

3. CouchDB 0 – for Peer 0

```
$ docker run --rm -it --network="my-net" --name couchdb0 -p 5984:5984
             -e COUCHDB_USER=
             -e COUCHDB_PASSWORD=
             -e CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=my-net hyperledger/fabric-couchdb
```

Multi Host 구성 – PC1

4. Peer 0

```
$ docker run --rm -it --link orderer.example.com:orderer.example.com --network="my-net"
            --name peer0.org1.example.com -p 8051:7051 -p 8053:7053
            -e CORE_LEDGER_STATE_STATEDATABASE=CouchDB
            -e CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb0:5984
            -e CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
            -e CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
            -e CORE_PEER_ADDRESSAUTODETECT=true  -e CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
            -e CORE_LOGGING_LEVEL=DEBUG  -e CORE_PEER_NETWORKID=peer0.org1.example.com
            -e CORE_NEXT=true -e CORE_PEER_ENDORSER_ENABLED=true
            -e CORE_PEER_ID=peer0.org1.example.com  -e CORE_PEER_PROFILE_ENABLED=true
            -e CORE_PEER_COMMITTER_LEDGER_ORDERER=orderer.example.com:7050
            -e CORE_PEER_GOSSIP_IGNORESECURITY=true
            -e CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=my-net
            -e CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.example.com:7051
            -e CORE_PEER_TLS_ENABLED=false -e CORE_PEER_GOSSIP_USELEADERELECTION=false
            -e CORE_PEER_GOSSIP_ORGLEADER=true
            -e CORE_PEER_LOCALMSPID=Org1MSP -v /var/run/:/host/var/run/
            -v $(pwd)/crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp
                                    :/etc/hyperledger/fabric/msp
            -w /opt/gopath/src/github.com/hyperledger/fabric/peer hyperledger/fabric-peer peer node start
```

Multi Host 구성 – PC2

5. CouchDB 1 – for Peer 1

```
$ docker run --rm -it --network="my-net" --name couchdb1 -p 6984:5984
             -e COUCHDB_USER=
             -e COUCHDB_PASSWORD=
             -e CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=my-net hyperledger/fabric-couchdb
```

Multi Host 구성 – PC2

6. Peer 1

```
$ docker run --rm -it --link orderer.example.com:orderer.example.com --network="my-net"
            --link peer0.org1.example.com:peer0.org1.example.com
            --name peer1.org1.example.com -p 9051:7051 -p 9053:7053
            -e CORE_LEDGER_STATE_STATEDATABASE=CouchDB
            -e CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb1:5984
            -e CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
            -e CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
            -e CORE_PEER_ADDRESSAUTODETECT=true  -e CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
            -e CORE_LOGGING_LEVEL=DEBUG  -e CORE_PEER_NETWORKID=peer1.org1.example.com
            -e CORE_NEXT=true -e CORE_PEER_ENDORSER_ENABLED=true
            -e CORE_PEER_ID=peer1.org1.example.com  -e CORE_PEER_PROFILE_ENABLED=true
            -e CORE_PEER_COMMITTER_LEDGER_ORDERER=orderer.example.com:7050
            -e CORE_PEER_GOSSIP_IGNORESECURITY=true
            -e CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=my-net
            -e CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org1.example.com:7051
            -e CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer1.org1.example.com:7051
            -e CORE_PEER_TLS_ENABLED=false -e CORE_PEER_GOSSIP_USELEADERELECTION=false
            -e CORE_PEER_GOSSIP_ORGLEADER=true
            -e CORE_PEER_LOCALMSPID=Org1MSP -v /var/run/:/host/var/run/
            -v $(pwd)/crypto-config/peerOrganizations/org1.example.com/peers/peer1.org1.example.com/msp
                                    :/etc/hyperledger/fabric/msp
            -w /opt/gopath/src/github.com/hyperledger/fabric/peer hyperledger/fabric-peer peer node start
```

Multi Host 구성 – PC2

7. CLI

```
$ docker run --rm -it --network="my-net" --name cli
            --link orderer.example.com:orderer.example.com
            --link peer0.org1.example.com:peer0.org1.example.com
            --link peer1.org1.example.com:peer1.org1.example.com
            -p 12051:7051 -p 12053:7053
            -e GOPATH=/opt/gopath
            -e CORE_PEER_LOCALMSPID=Org1MSP
            -e CORE_PEER_TLS_ENABLED=false -e CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
            -e CORE_LOGGING_LEVEL=DEBUG -e CORE_PEER_ID=cli
            -e CORE_PEER_ADDRESS=peer0.org1.example.com:7051 -e CORE_PEER_NETWORKID=cli
            -e CORE_PEER_MSPCONFIGPATH=
                        /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/
                        org1.example.com/users/Admin@org1.example.com/msp
            -e CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=my-net
            -v /var/run/:/host/var/run/
            -v $(pwd)/chaincode/:/opt/gopath/src/github.com/hyperledger/fabric/examples/chaincode/go
            -v $(pwd)/crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
            -v $(pwd)/scripts:/opt/gopath/src/github.com/hyperledger/fabric/peer/scripts/
            -v $(pwd)/channel-artifacts:/opt/gopath/src/github.com/hyperledger/fabric/peer/channel-artifacts
            -w /opt/gopath/src/github.com/hyperledger/fabric/peer hyperledger/fabric-tools /bin/bash
            -c './scripts/script.sh'
```

Multi Host 구성 – PC2

```
2018-02-14 08:10:35.481 UTC [chaincodeCmd] install -> DEBU 00d Installed remotely response:<status:200 payload:"OK" >
2018-02-14 08:10:35.481 UTC [main] main -> INFO 00e Exiting.....
===================== Chaincode is installed on remote peer PEER0 =====================

Install chaincode on org2/peer2...
CORE_PEER_NETWORKID=cli
CORE_PEER_LOCALMSPID=Org1MSP
CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
CORE_PEER_TLS_ENABLED=false
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=my-net
CORE_PEER_ID=cli
CORE_LOGGING_LEVEL=DEBUG
CORE_PEER_ADDRESS=peer1.org1.example.com:7051
2018-02-14 08:10:35.556 UTC [msp] GetLocalMSP -> DEBU 001 Returning existing local MSP
2018-02-14 08:10:35.556 UTC [msp] GetDefaultSigningIdentity -> DEBU 002 Obtaining default signing identity
2018-02-14 08:10:35.556 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 003 Using default escc
2018-02-14 08:10:35.556 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 004 Using default vscc
2018-02-14 08:10:35.645 UTC [golang-platform] getCodeFromFS -> DEBU 005 getCodeFromFS github.com/hyperledger/fabric/examples/chaincode/go/chaincode_example02
2018-02-14 08:10:35.894 UTC [golang-platform] func1 -> DEBU 006 Discarding GOROOT package fmt
2018-02-14 08:10:35.894 UTC [golang-platform] func1 -> DEBU 007 Discarding provided package github.com/hyperledger/fabric/core/chaincode/shim
2018-02-14 08:10:35.894 UTC [golang-platform] func1 -> DEBU 008 Discarding provided package github.com/hyperledger/fabric/protos/peer
2018-02-14 08:10:35.894 UTC [golang-platform] func1 -> DEBU 009 Discarding GOROOT package strconv
2018-02-14 08:10:35.895 UTC [golang-platform] GetDeploymentPayload -> DEBU 00a done
2018-02-14 08:10:35.898 UTC [msp/identity] Sign -> DEBU 00b Sign: plaintext: 0A86070A5C08031A0C08FBDD8FD40510...175DFF090000FFFF7C012598002C0000
2018-02-14 08:10:35.898 UTC [msp/identity] Sign -> DEBU 00c Sign: digest: 8B421BBA045DDFE01576F0BB411D0DAB98950AF242D85A836CC28AEB8AFAC640
2018-02-14 08:10:35.905 UTC [chaincodeCmd] install -> DEBU 00d Installed remotely response:<status:200 payload:"OK" >
2018-02-14 08:10:35.905 UTC [main] main -> INFO 00e Exiting.....
===================== Chaincode is installed on remote peer PEER1 =====================


========= All GOOD, BMHN execution completed ===========


 _____   _   _   ____
| ____| | \ | | |  _ \
|  _|   |  \| | | | | |
| |___  | |\  | | |_| |
|_____| |_| \_| |____/
```

$ ./scripts/script.sh

- mychannel 채널 생성
- peer0, peer1를 채널에 join
- anchor peer update (peer0)
- Chaincode 설치 (peer0, peer1)

Multi Host 구성 – PC2

8. CLI Bash

```
$ docker run --rm -it --network="my-net" --name cli
            --link orderer.example.com:orderer.example.com
            --link peer0.org1.example.com:peer0.org1.example.com
            --link peer1.org1.example.com:peer1.org1.example.com
            -p 12051:7051 -p 12053:7053 -e GOPATH=/opt/gopath
            -e CORE_PEER_LOCALMSPID=Org1MSP -e CORE_PEER_TLS_ENABLED=false
            -e CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
            -e CORE_LOGGING_LEVEL=DEBUG -e CORE_PEER_ID=cli
            -e CORE_PEER_ADDRESS=peer0.org1.example.com:7051
            -e CORE_PEER_NETWORKID=cli
            -e CORE_PEER_MSPCONFIGPATH=
                        /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/
                        org1.example.com/users/Admin@org1.example.com/msp
            -e CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=my-net   -v /var/run/:/host/var/run/
            -v $(pwd)/chaincode/:/opt/gopath/src/github.com/hyperledger/fabric/examples/chaincode/go
            -v $(pwd)/crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
            -v $(pwd)/scripts:/opt/gopath/src/github.com/hyperledger/fabric/peer/scripts/
            -v $(pwd)/channel-artifacts:/opt/gopath/src/github.com/hyperledger/fabric/peer/channel-artifacts
            -w /opt/gopath/src/github.com/hyperledger/fabric/peer hyperledger/fabric-tools /bin/bash
```

root@a14d67c2dbb5:/opt/gopath/src/github.com/hyperledger/fabric/peer#

Multi Host 구성 – PC2

9. Instantiate Chaincode

```
# Environment variables for PEER0

CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
CORE_PEER_LOCALMSPID="Org1MSP"
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
CORE_PEER_ADDRESS=peer0.org1.example.com:7051
```

```
$ peer chaincode instantiate -o orderer.example.com:7050 -C mychannel -n mycc -v 1.0 -c
'{"Args":["init","a","100","b","200"]}' -P "OR ('Org1MSP.member','Org2MSP.member')"
```

Multi Host 구성 – PC1, PC2

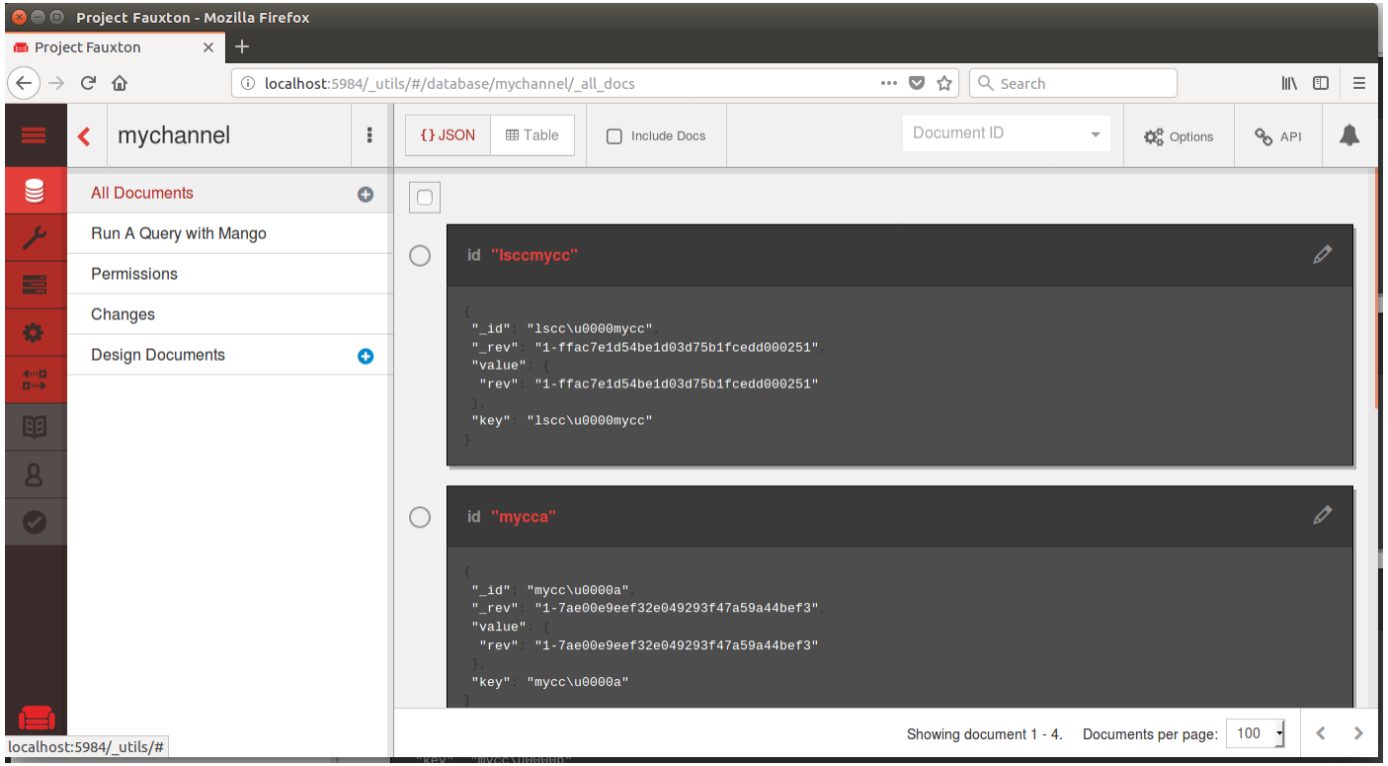
10. Check CouchDB (Peer0, Peer1)

- Peer 0 (PC1):

  http://localhost:5984/_utils/#/database/mychannel_/_all_docs

- Peer 1 (PC2):

  http://localhost:6984/_utils/#/database/mychannel_/_all_docs



Multi Host 구성 – PC2

11. Query Chaincode

```
# Environment variables for PEER1

CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
CORE_PEER_LOCALMSPID="Org1MSP"
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
CORE_PEER_ADDRESS=peer1.org1.example.com:7051
```

```
$ peer chaincode query -C mychannel -n mycc -c '{"Args":["query","a"]}'
```

```
Query Result: 100
```

Multi Host 구성 – PC2

12. Invoke Chaincode

```
# Environment variables for PEER0

CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
CORE_PEER_LOCALMSPID="Org1MSP"
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
CORE_PEER_ADDRESS=peer0.org1.example.com:7051
```

```
$ peer chaincode invoke -o orderer.example.com:7050 -C mychannel -n mycc -c '{"Args":["invoke","a","b","10"]}'
```

Multi Host 구성 – PC2

13. Query Chaincode

```
# be sure to set the -C and -n flags appropriately
$ peer chaincode query -C mychannel -n mycc -c '{"Args":["query","a"]}'
```

```
Query Result: 90
```