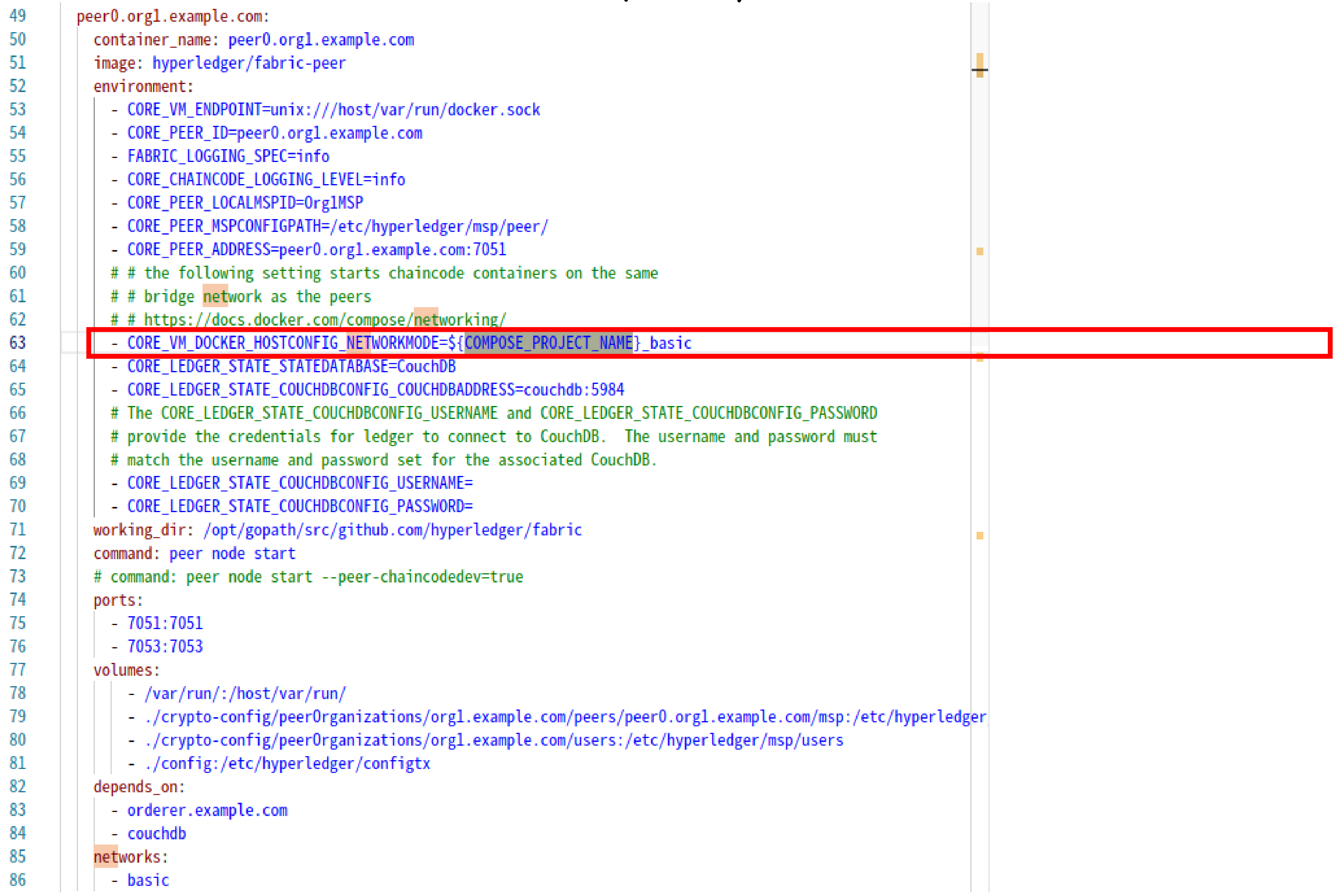
docker network naming

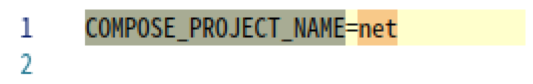
- basic_natwork/docker-compose.yaml



```yaml
  peer0.org1.example.com:
    container_name: peer0.org1.example.com
    image: hyperledger/fabric-peer
    environment:
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - CORE_PEER_ID=peer0.org1.example.com
      - FABRIC_LOGGING_SPEC=info
      - CORE_CHAINCODE_LOGGING_LEVEL=info
      - CORE_PEER_LOCALMSPID=Org1MSP
      - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/peer/
      - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
      # # the following setting starts chaincode containers on the same
      # # bridge network as the peers
      # # https://docs.docker.com/compose/networking/
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_basic
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb:5984
      # The CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME and CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD
      # provide the credentials for ledger to connect to CouchDB.  The username and password must
      # match the username and password set for the associated CouchDB.
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric
    command: peer node start
    # command: peer node start --peer-chaincodedev=true
    ports:
      - 7051:7051
      - 7053:7053
    volumes:
        - /var/run/:/host/var/run/
        - ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp:/etc/hyperledger
        - ./crypto-config/peerOrganizations/org1.example.com/users:/etc/hyperledger/msp/users
        - ./config:/etc/hyperledger/configtx
    depends_on:
      - orderer.example.com
      - couchdb
    networks:
      - basic
```

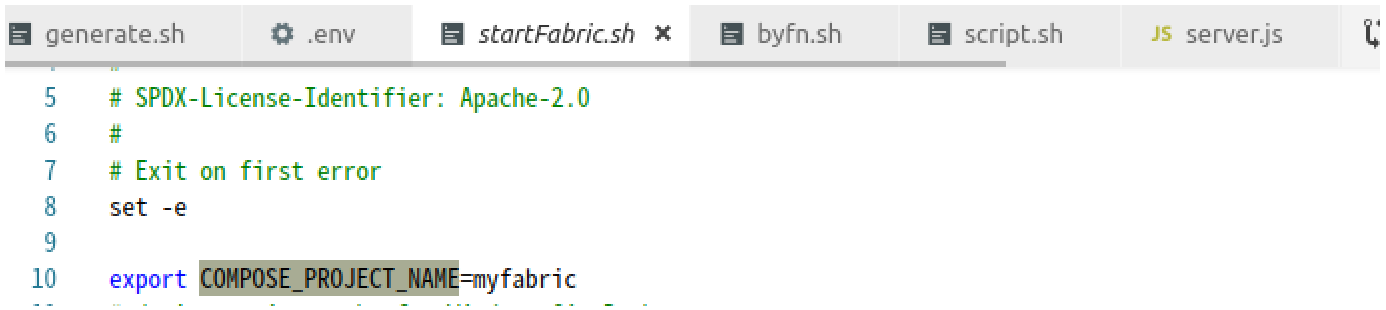
env설정

- basic_network/.env



```
COMPOSE_PROJECT_NAME=net
```

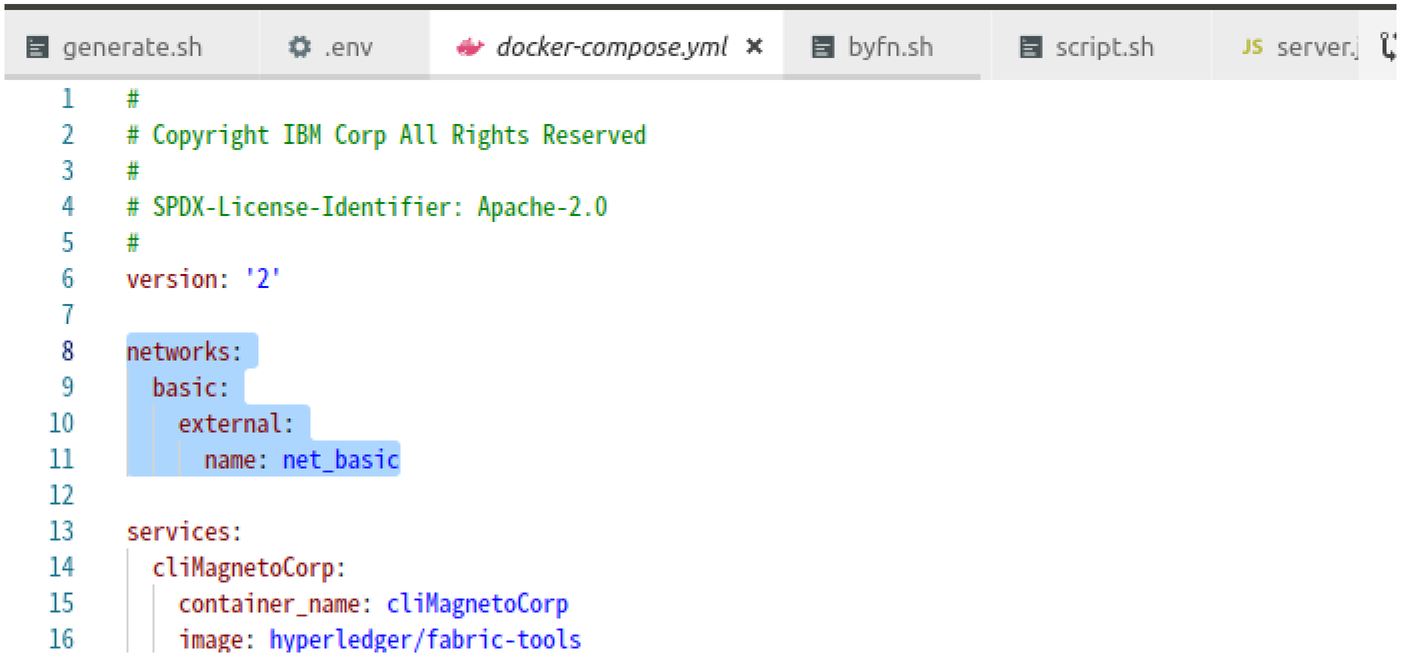
- network 복사 수정시에 COMPOSE_PROJECT_NAME 설정



```
# SPDX-License-Identifier: Apache-2.0
#
# Exit on first error
set -e

export COMPOSE_PROJECT_NAME=myfabric
```

외부참조 network사용



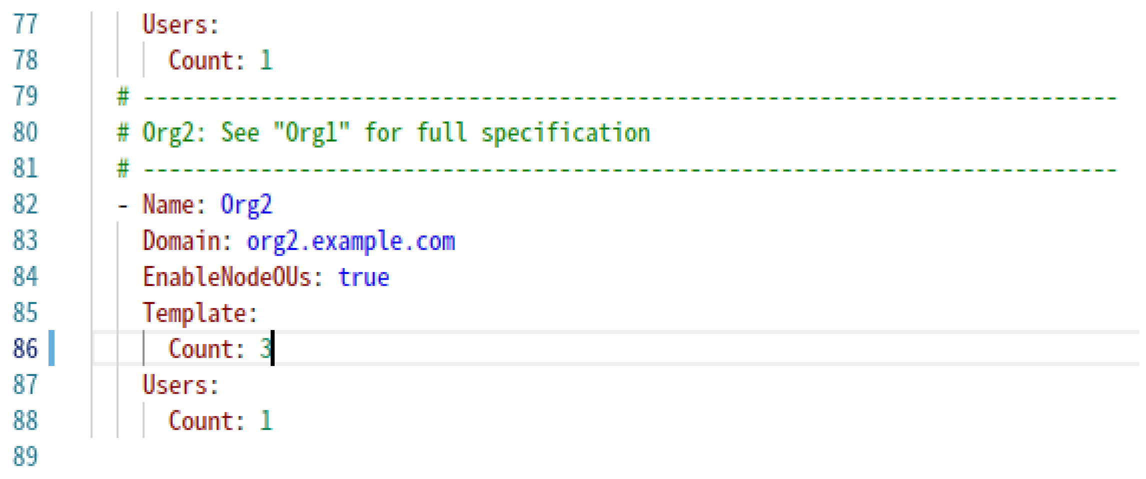
```
#
# Copyright IBM Corp All Rights Reserved
#
# SPDX-License-Identifier: Apache-2.0
#
version: '2'

networks:
  basic:
    external:
      name: net_basic

services:
  cliMagnetoCorp:
    container_name: cliMagnetoCorp
    image: hyperledger/fabric-tools
```

실시간 peer 추가

- first_network사용
- crypto-config.yaml 탬플릿 수 증가(peer수)

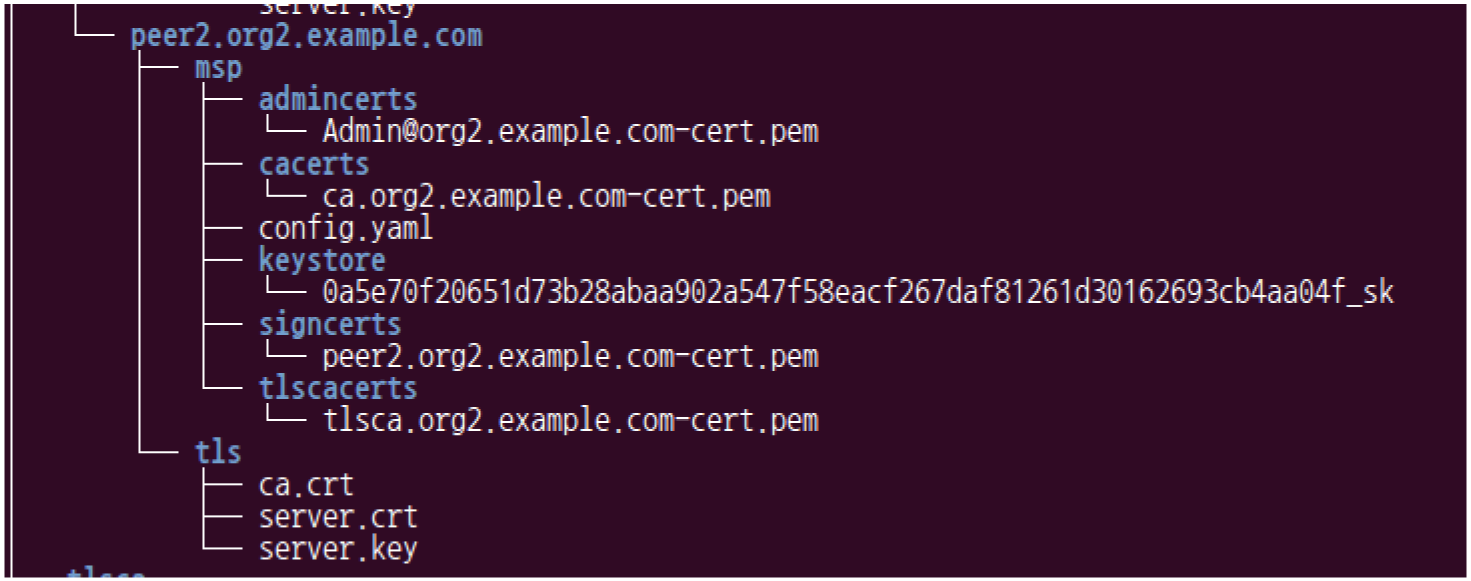


```
    Users:
      Count: 1
  # ---------------------------------------------------------------------------
  # Org2: See "Org1" for full specification
  # ---------------------------------------------------------------------------
  - Name: Org2
    Domain: org2.example.com
    EnableNodeOUs: true
    Template:
      Count: 3
    Users:
      Count: 1

```

peer2.org2 crypto-config추가

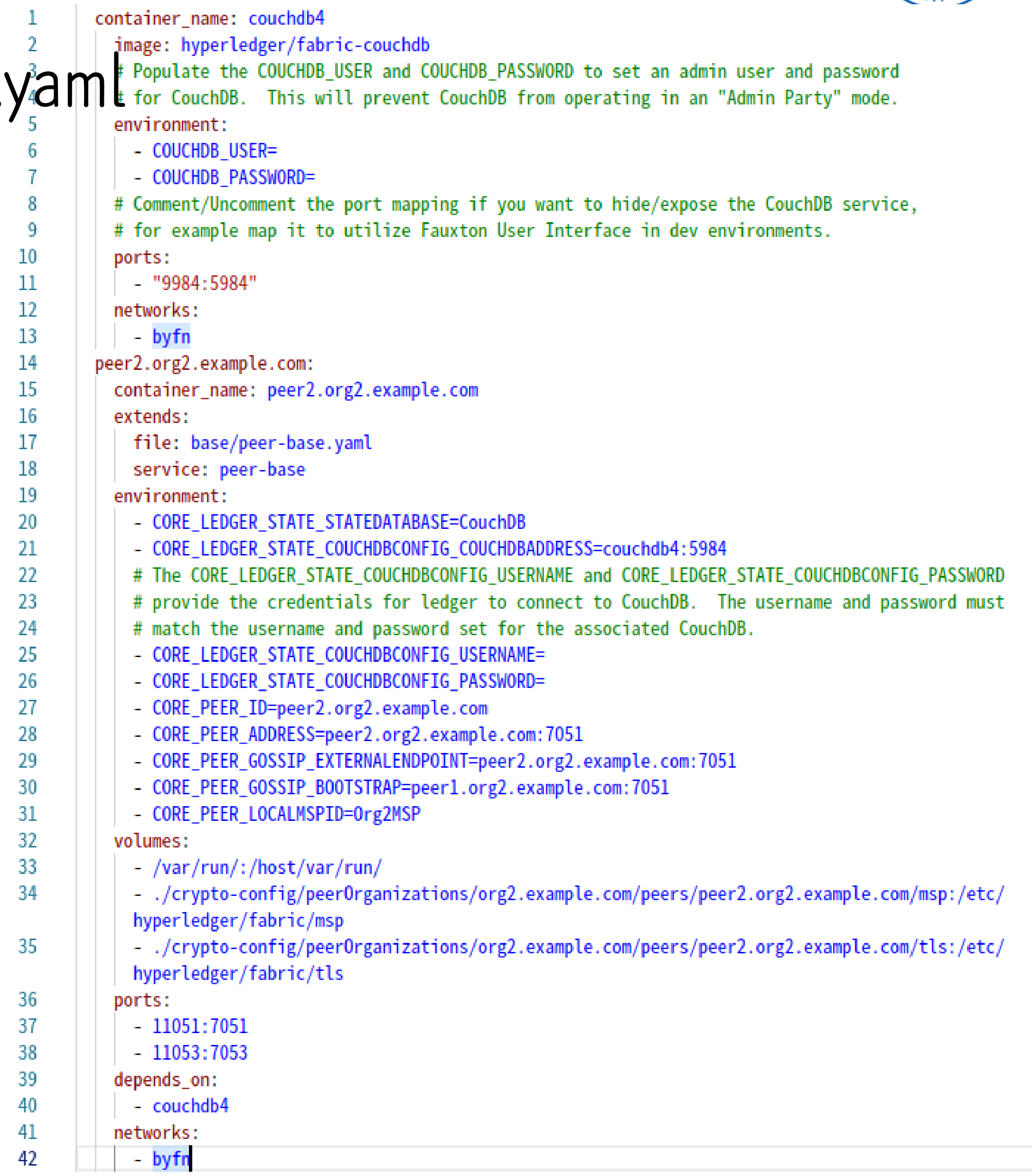
- ../bin/cryptogen extend --config=./crypto-config.yaml



```
└── peer2.org2.example.com
    ├── msp
    │   ├── admincerts
    │   │   └── Admin@org2.example.com-cert.pem
    │   ├── cacerts
    │   │   └── ca.org2.example.com-cert.pem
    │   ├── config.yaml
    │   ├── keystore
    │   │   └── 0a5e70f20651d73b28abaa902a547f58eacf267daf81261d30162693cb4aa04f_sk
    │   ├── signcerts
    │   │   └── peer2.org2.example.com-cert.pem
    │   └── tlscacerts
    │       └── tlsca.org2.example.com-cert.pem
    └── tls
        ├── ca.crt
        ├── server.crt
        └── server.key
```

peer, couchdb 컨테이너 추가

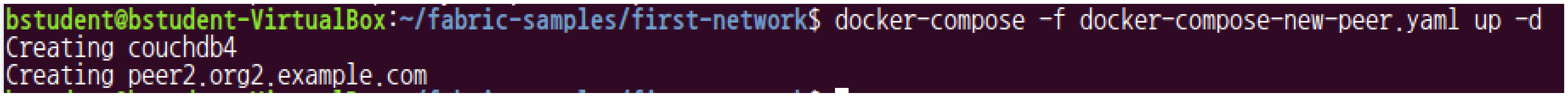
- docker-compose-new-peer.yaml



```yaml
container_name: couchdb4
  image: hyperledger/fabric-couchdb
  # Populate the COUCHDB_USER and COUCHDB_PASSWORD to set an admin user and password
  # for CouchDB.  This will prevent CouchDB from operating in an "Admin Party" mode.
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  # Comment/Uncomment the port mapping if you want to hide/expose the CouchDB service,
  # for example map it to utilize Fauxton User Interface in dev environments.
  ports:
    - "9984:5984"
  networks:
    - byfn
peer2.org2.example.com:
  container_name: peer2.org2.example.com
  extends:
    file: base/peer-base.yaml
    service: peer-base
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb4:5984
    # The CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME and CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD
    # provide the credentials for ledger to connect to CouchDB.  The username and password must
    # match the username and password set for the associated CouchDB.
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    - CORE_PEER_ID=peer2.org2.example.com
    - CORE_PEER_ADDRESS=peer2.org2.example.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer2.org2.example.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=peer1.org2.example.com:7051
    - CORE_PEER_LOCALMSPID=Org2MSP
  volumes:
    - /var/run/:/host/var/run/
    - ./crypto-config/peerOrganizations/org2.example.com/peers/peer2.org2.example.com/msp:/etc/hyperledger/fabric/msp
    - ./crypto-config/peerOrganizations/org2.example.com/peers/peer2.org2.example.com/tls:/etc/hyperledger/fabric/tls
  ports:
    - 11051:7051
    - 11053:7053
  depends_on:
    - couchdb4
  networks:
    - byfn
```

컨테이너 실행

$ docker-compose -f docker-compose-new-peer.yaml up -d



```
bstudent@bstudent-VirtualBox:~/fabric-samples/first-network$ docker-compose -f docker-compose-new-peer.yaml up -d
Creating couchdb4
Creating peer2.org2.example.com
```

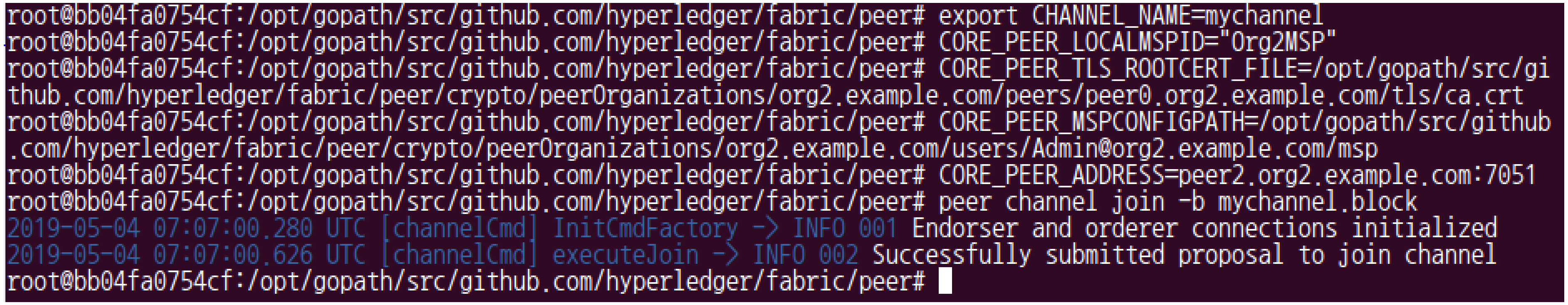
채널에 join

$ docker exec –it cli bash

피어2 명령 수행을 위한 환경변수 지정

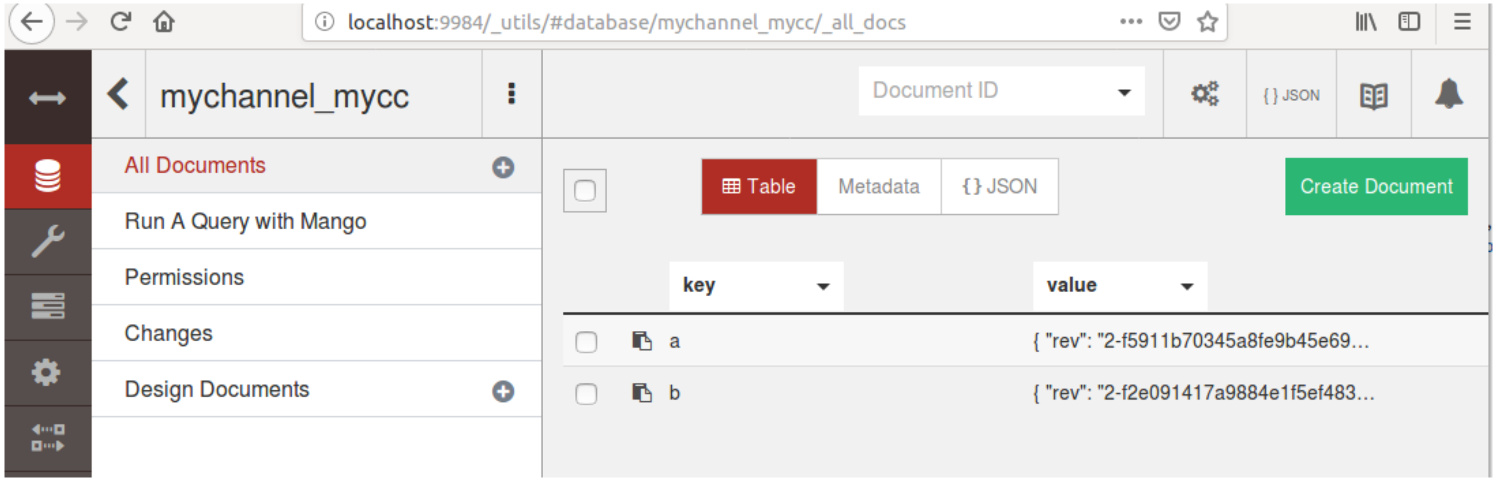
```
export CHANNEL_NAME=mychannel
CORE_PEER_LOCALMSPID="Org2MSP"
CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
CORE_PEER_ADDRESS=peer2.org2.example.com:7051
```

$ peer channel join -b mychannel.block



```
root@bb04fa0754cf:/opt/gopath/src/github.com/hyperledger/fabric/peer# export CHANNEL_NAME=mychannel
root@bb04fa0754cf:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE_PEER_LOCALMSPID="Org2MSP"
root@bb04fa0754cf:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/gi
thub.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
root@bb04fa0754cf:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github
.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
root@bb04fa0754cf:/opt/gopath/src/github.com/hyperledger/fabric/peer# CORE_PEER_ADDRESS=peer2.org2.example.com:7051
root@bb04fa0754cf:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel join -b mychannel.block
2019-05-04 07:07:00.280 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2019-05-04 07:07:00.626 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
root@bb04fa0754cf:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

http://localhost:9984/_utils/#database/mychannel_mycc/_all_docs



key	value
a	{ "rev": "2-f5911b70345a8fe9b45e69...
b	{ "rev": "2-f2e091417a9884e1f5ef483...