

연구 보고서

작성자	김한호	작성일자	2021.05.02.
-----	-----	------	-------------

1. 연구 계획

- Hyperledger fabric asset-transfer-private-data 테스트 프로젝트 구현시작
- 공개키 기반 구조 체인코드에 적용

2. 논문 연구 진행

```
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic$ cd chaincode-go/
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/chaincode-go$ ls
META-INF README.md chaincode collections_config.json go.mod go.sum main.go vendor
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/chaincode-go$ cd chaincode/
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/chaincode-go/chaincode$ ls
enc_rsa.go mocks
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/chaincode-go/chaincode$ cd ..
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/chaincode-go$ ls
META-INF README.md chaincode collections_config.json go.mod go.sum main.go vendor
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/chaincode-go$ cd ..
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic$ ls
application-javascript chaincode-go
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic$ cd application-javascript/
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/application-javascript$ ls
app.js app.js.bak node_modules package-lock.json package.json wallet
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/application-javascript$ vi app.js
h2kim@h2kim-VBox:~/fabric-samples/e2-list-basic/application-javascript$
```

asset-transfer-private-data javascript-applicationin app.js에서 작성했던 공개키 암호화 테스트를 e2-list-basic으로 경량화해서 테스트하기로 함.

```
pki := &PKI{
    ID:      pkiID,
    PrivateKey: privateKey,
    PublicKey: publicKey,
}
return pki, nil
}

/*
func (s *SmartContract) GetPubPrivKey(ctx contractapi.TransactionContextInterface, privStr string
, pubStr string) (*rsa.PrivateKey, *rsa.PublicKey, error) {
    privKey, err := ParseRsaPrivateKeyFromPemStr(privStr)
    if err != nil {
        return nil, nil, fmt.Errorf("Error: func:GetPubPrivKey Get privateKey")
    }
    pubKey, err := ParseRsaPublicKeyFromPemStr(pubStr)
    if err != nil {
        return nil, nil, fmt.Errorf("Error: func:GetPubPrivKey Get publicKey")
    }
    return privKey, pubKey, nil
}
*/

func (s *SmartContract) EncryptTest(ctx contractapi.TransactionContextInterface, pkiID string,
message string) error {
    pki, err := s.ReadPKI(ctx, pkiID)
    if err != nil {
        return err
    }
    log.Println("id: %v, pub: %v, priv: %v", pki.ID, pki.PrivateKey, pki.PublicKey)
```

공개키를 생성한 후 읽어오고 개인키와 공개키를 출력하는 테스트함수 작성함.

```

h2kim@h2kim-VBox: ~/fabric-samples/e2-list-basic/application-javascript$ docker logs -f bd491d7605ef
h2kim@h2kim-VBox: ~/fabric-samples/e2-list-basic/chaincode-go

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

91653658390016544053895016204897120396322463871125205811043661957015866622248985189641047576885865273
60580636205589780773302731425250070096212256189053628860434801764283990213914702773911520396476716748
26348894317790949840272854833147083254534316634299056247866940522329177295143330844512818014583007699
89975086961284051735140111578368479313197628582373082875177064844197581650221394008461044669632517314
81790988077820063283066150182055963953341737435211349987666553270431963824236650636760994644233154817
305623121269858117513039101307899867060545989 65537}
2021/04/23 08:08:35 id: %v, pub: %v, priv: %v pki1 -----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAwx2owDejjdsGNrfZUSl7nvRvYJefCj5Gy6wVhk28/pGZmJ3
SpV3ijk2v7DtdL72RDrC/n6vEf/xQgdtQn8pQ2hj+PrThh2g6IV/Jd5MBfmQtReK
0IP81F3WgV4KOPsJUjzLJ7LLAch9311BXVi4/IVNfmPFuXFLzJncLgLV3+i7IOg
P0C1MUUs95BjI3i7699Tboj+8j6Ir8rt1Xqm0JBw1FxRtTHC6Nn49PEKL4utae78
iePH9wzxfm5a8+uF8fU4we2p1kKEWu46Xlnopvb/ifiGtdU44ySJRhPDv40awrxM
yjkU8Xvg6yrPeez3hhN0H68nqQbI7QEFWypxQIDAQABaoIBABaLu2gJP6D/oGuf
HTDU465QwLxvSa4iNNt8avfb7YkFDjY3rnerwqbJ8bRE3ICx69Qj4W0X+4KM/Yzs
mfT7ndvxwRabb7+sP0JTL2nQNOh8GVhJvPYWamcpvkonJZ0NuIhtGTblMVjBlTt
joyvVbsojC+gOCJHaanniwiELwkSz6kaVrU27GyLx9zBvFTXG2KcER6osIMmr2i9
huM8R2j4EXPFMFtvtvvg40zAOWyMXhMLzMfHS3hGbQT5vbKeiUh4EefjyyXML6j+V
279wuiB7xHimZcG0F2KV8+e/7gvb8PPDLb6EyyBjswfIHUXiUGOLgP0Lb2TJCZA
olhtqEEcgYEAwHvcbUu+2gaPtbKBSbcoSVFv5hj0tLciqBLG08kBIkeCvses25hg
LN3X3iX3PmnQ6ufLNVYxvTGHwmyVojl1asxGeVNzOJTEf2PJwRsZdPdvThX7XG2At
Z9uGm2R5GakMrTtQwUi84H3CWh+QsF90gPrEfmpPzgodQUEIYJ0ItECgYEA6M/D
IxKil1a9N6+F5FyXIY1ua1dueFW0AL5PigcKeRyavS9UspVKKWN7PPYscCrX7i50
TXos6dnkFdDVQuqzWD/TMBwVxu9LsjqToExvVf1Vu1uVVWj0/3E0w/JxJjo6hv0Z
QYbdkj0ESauLq9w3aiou5f09fPDVEFzV0tnKTLUCgYEAAR35Z3eLpE3obL0QYmFP
438qrZkBN/7KNzQqbQpIcFUYW3v89Nf8D/5Ki2s8Dq9fDhcY6bC2MxvcDTPZ7jwe
WhWLGKnC4mJnUyjjMFryQhN3D4a/8sgyrkkbrKCy0o9fb1pC9L0+NotN2GskXXFT
mKwqhHL6FPvuLykeyQCqCtECgYAA3DCKPufBH6HKFID0Ac4QootBhASNvWKT7WmFk
OZX4XRMZeH+GWLZvWh5h0700MhLSQBmv2BuWj7amkeJbGD0fwjelotti9ATp93/A
fcjBhE0pJ9Z/5c1MC0d83W5zo4ED0gUmwUTSt0uqn7Nkl4Viz0aZ6K/9QtSeMdU9
LdHvoQK8gQCZKRQ5Sg52MxD0vQr6Pki+1Pa3qeQzNTP6UZKD/eAIhfQ0EV2wtw4fp
I7M2x4l8t2dL3E6mZi+cLYxIq9LEoBEQo2oRcL6f+izGjui7LBdKdKXishmNu6X
ND2cn7SNXGvX84zAh9g7U6sPP5JbzClyRA/2/7+jKx4MEkseNFtzYQ==
-----END RSA PRIVATE KEY-----
-----BEGIN RSA PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArwx2owDejjdsGNrfZUSl
7nvRvYJefCj5Gy6wVhk28/pGZmJ3SpV3ijk2v7DtdL72RDrC/n6vEf/xQgdtQn8p
Q2hj+PrThh2g6IV/Jd5MBfmQtReK0IP81F3WgV4KOPsJUjzLJ7LLAch9311BXVi4
/IVNfmPFuXFLzJncLgLV3+i7IOgP0C1MUUs95BjI3i7699Tboj+8j6Ir8rt1Xqm
0JBw1FxRtTHC6Nn49PEKL4utae78iePH9wzxfm5a8+uF8fU4we2p1kKEWu46Xlno
pvb/ifiGtdU44ySJRhPDv40awrxMyjkU8Xvg6yrPeez3hhN0H68nqQbI7QEFWyp
xQIDAQAB
-----END RSA PUBLIC KEY-----

```

도커 컨테이너에 남아있는 체인코드에서 출력된 결과를 확인함.

```

        break // fall through
    }
    return nil, fmt.Errorf("Key type is not RSA")
}

func (s *SmartContract) GenPubPrivKey(ctx contractapi.TransactionContextInterface, pkiID string)
(*PKI, error) {
    privKey, pubKey := generateKeyPair(2048)
    log.Printf("Private Key: %v\n", privKey)
    privateKey := ExportRsaPrivateKeyAsPemStr(privKey)
    log.Printf("Public Key: %v\n", pubKey)
    publicKey, err := ExportRsaPublicKeyAsPemStr(pubKey)
    if err != nil {
        return nil, fmt.Errorf("Error: fun:ExportRsaPublicKeyAsPemStr")
    }
    pki := &PKI{
        ID:          pkiID,
        PrivateKey:   privateKey,
        PublicKey:    publicKey,
    }
    return pki, nil
}

func (s *SmartContract) EnDecryptTest(ctx contractapi.TransactionContextInterface, pkiID string,
message string) error {
    pki, err := s.ReadPKI(ctx, pkiID)
    if err != nil {
        return err
    }
    log.Println("id: ", pki.ID, "\npub:", pki.PublicKey, "\npriv:", pki.PrivateKey)
    privKey, err := ParseRsaPrivateKeyFromPemStr(pki.PrivateKey)
    if err != nil {
        return err
    }
    pubKey, err := ParseRsaPublicKeyFromPemStr(pki.PublicKey)
    if err != nil {
        return err
    }
    cipherText, _ := rsa.EncryptOAEP(sha256.New(), rand.Reader, pubKey, []byte(message), nil)
    log.Printf("exchange message to ciphertext")
    fmt.Printf("Encrypted: %v\n", cipherText)

    decMessage, _ := rsa.DecryptOAEP(sha256.New(), rand.Reader, privKey, cipherText, nil)
    log.Printf("exchange ciphertext to message")
    fmt.Printf("Original: %s\n", string(decMessage))
    return nil
}

```

공개키와 개인키를 테스트 출력 뒤에 개인키와 공개키를 체인코드에서 불러와서 암호화와 복호화를 하고 그 과정을 로그를 남겨서 확인하기로 함.

```

2021/04/23 08:46:13 exchange message to ciphertext
Encrypted: [84 156 217 110 104 156 83 110 19 74 175 146 198 250 195 153 51 126 14 134 163 169 145 12
45 119 171 185 113 204 48 106 90 65 58 152 82 159 44 134 252 36 71 170 5 146 156 138 105 168 56 18 14
4 90 208 49 129 85 252 246 218 84 192 135 29 87 53 108 10 254 93 75 201 91 22 186 44 131 222 179 242
24 56 9 86 172 32 171 1 114 149 22 192 79 5 159 57 210 218 209 93 194 32 19 152 237 114 191 12 49 126
216 49 50 200 169 143 236 22 136 53 177 80 237 135 21 173 94 145 196 136 222 120 204 55 238 213 207
185 131 70 20 91 122 75 146 148 131 236 222 245 253 162 218 17 47 164 42 235 255 16 201 183 205 163 6
2 233 149 86 58 148 134 145 228 94 3 66 14 228 116 140 58 118 223 161 215 241 162 95 152 111 90 10 19
9 74 114 22 40 135 51 129 133 148 144 209 59 153 76 212 151 34 202 78 232 247 1 158 20 111 3 164 186
196 146 57 64 175 23 29 208 168 233 224 117 140 181 117 94 171 164 94 85 105 121 244 80 155 129 145 1
90 139 25 181 25 1]
2021/04/23 08:46:13 exchange ciphertext to message
Original: Test Something Encrypt Decrypt
^C

```

암호화한 텍스트가 출력된 이후에 복호화된 문자가 출력되는 것을 확인할 수 있음.

```

h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ./network.sh createChannel -s couchdb
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 'couchdb with crypto from 'cryptogen'
Bringing up network
LOCAL_VERSION=2.2.0
DOCKER_IMAGE_VERSION=2.2.0
/home/h2kim/fabric-samples/e2-network/./bin/cryptogen
Generate certificates using cryptogen tool
Create Org1 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organiza
tions
org1.example.com
+ res=0
Create Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organiza
tions
org2.example.com
+ res=0
Create Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organ
izations
+ res=0
Generate CCP files for Org1 and Org2
awk: cannot open organizations/peerOrganizations/control.example.com/tlsca/tlsca.control.example.
com-cert.pem (No such file or directory)
awk: cannot open organizations/peerOrganizations/control.example.com/ca/ca.control.example.com-ce
rt.pem (No such file or directory)
./organizations/ccp-generate.sh: 줄 41: organizations/peerOrganizations/control.example.com/conne
ction-control.json: 그런 파일이나 디렉터리가 없습니다
awk: cannot open organizations/peerOrganizations/control.example.com/tlsca/tlsca.control.example.
com-cert.pem (No such file or directory)
awk: cannot open organizations/peerOrganizations/control.example.com/ca/ca.control.example.com-ce
rt.pem (No such file or directory)
./organizations/ccp-generate.sh: 줄 42: organizations/peerOrganizations/control.example.com/conne
ction-control.yaml: 그런 파일이나 디렉터리가 없습니다
awk: cannot open organizations/peerOrganizations/civil.example.com/tlsca/tlsca.civil.example.com-
cert.pem (No such file or directory)
awk: cannot open organizations/peerOrganizations/civil.example.com/ca/ca.civil.example.com-cert.p
em (No such file or directory)
./organizations/ccp-generate.sh: 줄 52: organizations/peerOrganizations/civil.example.com/connect
ion-civil.json: 그런 파일이나 디렉터리가 없습니다
awk: cannot open organizations/peerOrganizations/civil.example.com/tlsca/tlsca.civil.example.com-
cert.pem (No such file or directory)
awk: cannot open organizations/peerOrganizations/civil.example.com/ca/ca.civil.example.com-cert.p
em (No such file or directory)
./organizations/ccp-generate.sh: 줄 53: organizations/peerOrganizations/civil.example.com/connect

```

테스트 프로젝트를 설계한 것과 같이 구현하기 위해 설정파일을 변경하고 채널 생성을 하여 설정을 변경할 부분을 확인함.

```

h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ./network.sh createChannel -s couchdb
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 'couchdb'
Generating channel create transaction 'mychannel.tx'
+ configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/mychannel.tx -ch
annelID mychannel
2021-05-01 13:43:58.223 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-05-01 13:43:58.279 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded config
uration: /home/h2kim/fabric-samples/e2-network/configtx/configtx.yaml
2021-05-01 13:43:58.279 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Genera
ting new channel configtx
2021-05-01 13:43:58.279 KST [common.tools.configtxgen] main -> FATA 004 Error on outputChannelCre
ateTx: could not generate default config template: error parsing configuration: could not create
application group: failed to create application org: 1 - Error loading MSP configuration for org
controlMSP: could not load a valid ca certificate from directory /home/h2kim/fabric-samples/e2-ne
twork/organizations/peerOrganizations/control.example.com/msp/cacerts: stat /home/h2kim/fabric-sa
mples/e2-network/organizations/peerOrganizations/control.example.com/msp/cacerts: no such file or
directory
+ res=1
Failed to generate channel configuration transaction...
Create channel failed
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ls
README.md channel-artifacts docker organizations scripts
addOrg3 configtx network.sh scriptUtils.sh system-genesis-block
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ls
README.md channel-artifacts docker organizations scripts
addOrg3 configtx network.sh scriptUtils.sh system-genesis-block
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ cd organizations/
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations$ s
s: 명령을 찾을 수 없습니다
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations$ ls
ccp-generate.sh ccp-template.yaml fabric-ca peerOrganizations
ccp-template.json cryptogen ordererOrganizations
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations$ cd ..
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ls
README.md channel-artifacts docker organizations scripts
addOrg3 configtx network.sh scriptUtils.sh system-genesis-block
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ cd organizations/
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations$ ls
ccp-generate.sh ccp-template.yaml fabric-ca peerOrganizations
ccp-template.json cryptogen ordererOrganizations
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations$ cd peerOrganizations/
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ ls
org1.example.com org2.example.com

```

설정을 변경한 이후 설정한 폴더를 찾을 수 없기 때문에 명령이 실행되지 않을 것을 확인함.


```

h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ cp org2.example.com
shop.example.com
cp: -r not specified; omitting directory 'org2.example.com'
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ cp -r org2.example.
com shop.example.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ ls
org1.example.com  org2.example.com  shop.example.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ mv org1.example.com
control.example.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ ls
control.example.com  org2.example.com  shop.example.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ mv org2.example.com
civil.example.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ ls
civil.example.com  control.example.com  shop.example.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ cd civil.example.co
m/
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations/civil.example.com$ l
s
ca  msp  peers  tlsca  users
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations/civil.example.com$ c
d ca
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations/civil.example.com/ca
$ ls
ca.org2.example.com-cert.pem  priv_sk
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations/civil.example.com/ca
$ cd ..
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations/civil.example.com$ l
s
ca  msp  peers  tlsca  users
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations/civil.example.com$ c
d ..
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ ls
civil.example.com  control.example.com  shop.example.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ rm -rf civil.exampl
e.com/
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ rm -rf control.exam
ple.com/
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ rm -rf shop.example
.com/
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ ls
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ mkdir control.examp
le.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ mkdir civil.example
.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ mkdir shop.example.
com
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/peerOrganizations$ cd

```

파일을 이동하여 적용하는 방법을 생각하기 보다 새롭게 폴더를 생성하여 진행함.

```

h2kim@h2kim-VBox:~/fabric-samples/e2-network$ mkdir -p ./organizations/peerOrganizations/control.
example.com/msp/cacerts
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ mkdir -p ./organizations/peerOrganizations/civil.ex
ample.com/msp/cacerts
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ mkdir -p ./organizations/peerOrganizations/shop.exa
mple.com/msp/cacerts
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ./network.sh up createChannel -s couchdb
Creating channel 'mychannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds a
nd using database 'couchdb'
Generating channel create transaction 'mychannel.tx'
+ configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/mychannel.tx -ch
annelID mychannel
2021-05-01 13:48:15.407 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-05-01 13:48:15.466 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded config
uration: /home/h2kim/fabric-samples/e2-network/configtx/configtx.yaml
2021-05-01 13:48:15.466 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Genera
ting new channel configtx
2021-05-01 13:48:15.466 KST [common.tools.configtxgen] func1 -> PANI 004 proto: Marshal called wi
th nil
panic: proto: Marshal called with nil [recovered]
      panic: proto: Marshal called with nil

goroutine 1 [running]:
go.uber.org/zap/zapcore.(*CheckedEntry).Write(0xc0000abe40, 0x0, 0x0, 0x0)
    /__w/1/go/src/github.com/hyperledger/fabric/vendor/go.uber.org/zap/zapcore/entry.go:230 +
0x545
go.uber.org/zap.(*SugaredLogger).log(0xc0000102f0, 0x4, 0xc000037760, 0x1e, 0x0, 0x0, 0x0, 0x0, 0
x0, 0x0)
    /__w/1/go/src/github.com/hyperledger/fabric/vendor/go.uber.org/zap/sugar.go:234 +0x100
go.uber.org/zap.(*SugaredLogger).Panicf(...)
    /__w/1/go/src/github.com/hyperledger/fabric/vendor/go.uber.org/zap/sugar.go:159
github.com/hyperledger/fabric/common/flogging.(*FabricLogger).Panic(0xc0000102f8, 0xc000313818, 0
x1, 0x1)
    /__w/1/go/src/github.com/hyperledger/fabric/common/flogging/zap.go:73 +0x99
main.main.func1()
    /__w/1/go/src/github.com/hyperledger/fabric/cmd/configtxgen/main.go:256 +0x176
panic(0xac7c80, 0xc00006e360)
    /usr/local/go/src/runtime/panic.go:969 +0x166
github.com/hyperledger/fabric/protoutil.MarshalOrPanic(...)
    /__w/1/go/src/github.com/hyperledger/fabric/protoutil/commonutils.go:26
github.com/hyperledger/fabric/internal/configtxgen/encoder.addValue(0xc0001414a0, 0xc97280, 0xc00
01c2120, 0xbaaac5, 0x6)
    /__w/1/go/src/github.com/hyperledger/fabric/internal/configtxgen/encoder/encoder.go:59 +0

```

crypto-config 파일을 읽어오면서 에러가 발생한 것을 확인함.

```

h2kim@h2kim-VBox:~/fabric-samples/e2-network$ find ./ ! -type d | xargs grep -n "crypto-config"
./network.sh:201: cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml
l --output="organizations"
./network.sh:211: cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml
l --output="organizations"
./network.sh:221: cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.
yaml --output="organizations"
./.gitignore:6:org3-artifacts/crypto-config/*
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ls
README.md channel-artifacts docker organizations scripts
addOrg3 configtx network.sh scriptUtils.sh system-genesis-block
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ cd organizations/cryptogen/
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ ls
crypto-config-orderer.yaml crypto-config-org1.yaml crypto-config-org2.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ mv crypto-config-org1.yaml
crypto-config-conrol.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ ls
crypto-config-conrol.yaml crypto-config-orderer.yaml crypto-config-org2.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ mv crypto-config-org2.yaml
crypto-config-civil.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ ls
crypto-config-civil.yaml crypto-config-conrol.yaml crypto-config-orderer.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ vi crypto-config-conrol.yam
l
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ ls
crypto-config-civil.yaml crypto-config-conrol.yaml crypto-config-orderer.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ vi crypto-config-civil.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ cp crypto-config-civil.yaml
crypto-config-shop.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ ci crypto-config-shop.yaml

Command 'ci' not found, but can be installed with:

sudo apt install rcs

h2kim@h2kim-VBox:~/fabric-samples/e2-network/organizations/cryptogen$ vi crypto-config-shop.yaml

```

프로젝트 폴더내에서 crypto-config 내용을 담고 있는 코드 부분을 확인하여 설정에 맞도록 수정함.


```

ERROR: The Compose file is invalid because:
Service peer0.control.example.com has neither an image nor a build context specified. At least one must be provided.
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS
PORTS              NAMES
Generating channel create transaction 'mychannel.tx'
+ configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/mychannel.tx -channelID mychannel
2021-05-01 14:03:41.441 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-05-01 14:03:41.491 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/h2kim/fabric-samples/e2-network/configtx/configtx.yaml
2021-05-01 14:03:41.491 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
2021-05-01 14:03:41.494 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
+ res=0
Generating anchor peer update transactions
Generating anchor peer update transaction for controlMSP
+ configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/controlMSPanchors.tx -channelID mychannel -asOrg controlMSP
2021-05-01 14:03:41.525 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-05-01 14:03:41.583 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/h2kim/fabric-samples/e2-network/configtx/configtx.yaml
2021-05-01 14:03:41.583 KST [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Generating anchor peer update
2021-05-01 14:03:41.585 KST [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 004 Writing anchor peer update
+ res=0
Generating anchor peer update transaction for civilMSP
+ configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/civilMSPanchors.tx -channelID mychannel -asOrg civilMSP
2021-05-01 14:03:41.614 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-05-01 14:03:41.677 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/h2kim/fabric-samples/e2-network/configtx/configtx.yaml
2021-05-01 14:03:41.677 KST [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Generating anchor peer update
2021-05-01 14:03:41.678 KST [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 004 Writing anchor peer update
+ res=0
Generating anchor peer update transaction for shopMSP
+ configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/shopMSPanchors.tx -channelID mychannel -asOrg shopMSP
2021-05-01 14:03:41.704 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-05-01 14:03:41.770 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/h2kim/fabric-samples/e2-network/configtx/configtx.yaml
2021-05-01 14:03:41.771 KST [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Generating anchor peer update
2021-05-01 14:03:41.772 KST [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 004 Writing anchor peer update
+ res=0

```

configtx 설정을 읽어오는 것은 확인할 수 있음. 도커에서 컨테이너가 생성되지 않음.

```

./organizations/fabric-ca/registerEnroll.sh:33: info!n "Register peer0"
./organizations/fabric-ca/registerEnroll.sh:35: fabric-ca-client register --caname ca-org1 --id.name peer0 --id.secret peer0pw --id.type peer --tls.certfiles ${PWD}/organizations/fabric-ca/org1/tls-cert.pem
./organizations/fabric-ca/registerEnroll.sh:49: mkdir -p organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com
./organizations/fabric-ca/registerEnroll.sh:51: info!n "Generate the peer0 msp"
./organizations/fabric-ca/registerEnroll.sh:53: fabric-ca-client enroll -u https://peer0:peer0pw@localhost:7054 --caname ca-org1 -M ${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp --csr.hosts peer0.org1.example.com --tls.certfiles ${PWD}/organizations/fabric-ca/org1/tls-cert.pem
./organizations/fabric-ca/registerEnroll.sh:56: cp ${PWD}/organizations/peerOrganizations/org1.example.com/msp/config.yaml ${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp/config.yaml
./organizations/fabric-ca/registerEnroll.sh:58: info!n "Generate the peer0-tls certificates"
./organizations/fabric-ca/registerEnroll.sh:60: fabric-ca-client enroll -u https://peer0:peer0pw@localhost:7054 --caname ca-org1 -M ${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls --enrollment.profile tls --csr.hosts peer0.org1.example.com --csr.hosts localhost --tls.certfiles ${PWD}/organizations/fabric-ca/org1/tls-cert.pem
./organizations/fabric-ca/registerEnroll.sh:63: cp ${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/tlscacerts/* ${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
./organizations/fabric-ca/registerEnroll.sh:64: cp ${PWD}/organizations/peerOrganizations/org1.e

```

```

./addOrg3/docker/docker-compose-org3.yaml:36: - CORE_PEER_LISTENADDRESS=0.0.0.0:11051
./addOrg3/docker/docker-compose-org3.yaml:39: - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org3.example.com:11051
./addOrg3/docker/docker-compose-org3.yaml:40: - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org3.example.com:11051
./addOrg3/docker/docker-compose-org3.yaml:50: - 11051:11051
./addOrg3/docker/docker-compose-org3.yaml:65: - CORE_PEER_ADDRESS=peer0.org3.example.com:11051
./addOrg3/configtx.yaml:45: Port: 11051
./configtx/configtx.yaml:80: Port: 7051
./configtx/configtx.yaml:114: Port: 9051
./configtx/configtx.yaml:148: Port: 11051
./scripts/org3-scripts/envVarCLI.sh:28: CORE_PEER_ADDRESS=peer0.org1.example.com:7051
./scripts/org3-scripts/envVarCLI.sh:33: CORE_PEER_ADDRESS=peer0.org2.example.com:9051
./scripts/org3-scripts/envVarCLI.sh:38: CORE_PEER_ADDRESS=peer0.org3.example.com:11051
./scripts/envVar.sh:37: export CORE_PEER_ADDRESS=localhost:7051
./scripts/envVar.sh:42: export CORE_PEER_ADDRESS=localhost:8051
./scripts/envVar.sh:48: export CORE_PEER_ADDRESS=localhost:10051
./organizations/ccp-generate.sh:35:P0PORT=7051
./organizations/ccp-generate.sh:46:P0PORT=9051
./organizations/ccp-generate.sh:57:P0PORT=11051
grep: ./organizations/fabric-ca/control/msp/keystore/9e790cf991cf2b8269a93fc6a63de770fb662657e9a970a55e40eaa08d4791d9_sk: 허가 거부
grep: ./organizations/fabric-ca/control/msp/keystore/112d3a40d1fd71b2748ac0fb636e8b11753dd641105e09d8ebd62cdf408785e4_sk: 허가 거부
grep: ./organizations/fabric-ca/shop/msp/keystore/167c0f20661af2c1bf22c2d8d606df6f832945b663d17b71d5e8d4d69a9a1d64_sk: 허가 거부
grep: ./organizations/fabric-ca/shop/msp/keystore/990bd1575086fed60a065b8c5ae837c29ba0e94df2ee1e63447581192bf2e4af_sk: 허가 거부
grep: ./organizations/fabric-ca/civil/msp/keystore/95593ef7ea49ca3380273cfdcb9b8a45f96f188e0d13f521208d599693495b10_sk: 허가 거부
grep: ./organizations/fabric-ca/civil/msp/keystore/11c12612624a881ce187a298ed89cfb8dd766d4d8f84b3abc2c7b8a7a81c31dd_sk: 허가 거부
./docker/docker-compose-test-net.yaml:70: - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
./docker/docker-compose-test-net.yaml:71: - CORE_PEER_LISTENADDRESS=0.0.0.0:7051
./docker/docker-compose-test-net.yaml:74: - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org1.example.com:7051
./docker/docker-compose-test-net.yaml:75: - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.example.com:7051
./docker/docker-compose-test-net.yaml:85: - 7051:7051
./docker/docker-compose-test-net.yaml:108: - CORE_PEER_ADDRESS=peer0.org2.example.com:9051
./docker/docker-compose-test-net.yaml:109: - CORE_PEER_LISTENADDRESS=0.0.0.0:9051
./docker/docker-compose-test-net.yaml:112: - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org2.example.com:9051
./docker/docker-compose-test-net.yaml:113: - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org2.example.com:9051
./docker/docker-compose-test-net.yaml:123: - 9051:9051

```

peer0 설정을 검색하여 계획에 맞도록 수정하여 다시 진행함.

```

h2kim@h2kim-VBox:~/fabric-samples/e2-network$ find ./ ! -type d | xargs grep -n "7051"
./configtx/configtx.yaml:80: Port: 7051
./scripts/org3-scripts/envVarCLI.sh:28: CORE_PEER_ADDRESS=peer0.org1.example.com:7051
./scripts/envVar.sh:37: export CORE_PEER_ADDRESS=localhost:7051
./organizations/ccp-generate.sh:35:P0PORT=7051
grep: ./organizations/fabric-ca/control/msp/keystore/9e790cf991cf2b8269a93fc6a63de770fb662657e9a970a55e40eaa08d4791d9_sk: 허가 거부
grep: ./organizations/fabric-ca/control/msp/keystore/112d3a40d1fd71b2748ac0fb636e8b11753dd641105e09d8ebd62cdf408785e4_sk: 허가 거부
grep: ./organizations/fabric-ca/shop/msp/keystore/167c0f20661af2c1bf22c2d8d606df6f832945b663d17b71d5e8d4d69a9a1d64_sk: 허가 거부
grep: ./organizations/fabric-ca/shop/msp/keystore/990bd1575086fed60a065b8c5ae837c29ba0e94df2ee1e63447581192bf2e4af_sk: 허가 거부
grep: ./organizations/fabric-ca/civil/msp/keystore/95593ef7ea49ca3380273cfdcb9b8a45f96f188e0d13f521208d599693495b10_sk: 허가 거부
grep: ./organizations/fabric-ca/civil/msp/keystore/11c12612624a881ce187a298ed89cfb8dd766d4d8f84b3abc2c7b8a7a81c31dd_sk: 허가 거부
./docker/docker-compose-test-net.yaml:70: - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
./docker/docker-compose-test-net.yaml:71: - CORE_PEER_LISTENADDRESS=0.0.0.0:7051
./docker/docker-compose-test-net.yaml:74: - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org1.example.com:7051
./docker/docker-compose-test-net.yaml:75: - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.example.com:7051
./docker/docker-compose-test-net.yaml:85: - 7051:7051

```

control조직의 peer0도 실행되지 않아서 포트번호의 통신할 때 설정을 복사붙여넣기 하다 문제가 발생했는지 확인함.

```
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ./network.sh up createChannel -ca -s couchdb
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 'couchdb with crypto from 'Certificate Authorities'
Bringing up network
LOCAL_VERSION=2.2.0
DOCKER_IMAGE_VERSION=2.2.0
CA_LOCAL_VERSION=1.4.8
CA_DOCKER_IMAGE_VERSION=1.4.8
Generate certificates using Fabric CA's
Recreating ca_civil ...
Recreating ca_civil
Creating ca_shop ...
Creating ca_control ...
Creating ca_orderer ...
Creating ca_shop
Creating ca_control
Creating ca_control ... done
```

네트워크를 생성하고 진행상황에 문제가 있는지 실행함.

```
90 peer0.org2.example.com:
91 container_name: peer0.org2.example.com
92 image: hyperledger/fabric-peer:$IMAGE_TAG
93 environment:
94   #Generic peer variables
95   - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
96   # the following setting starts chaincode containers on the same
97   # bridge network as the peers
98   # https://docs.docker.com/compose/networking/
99   - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_test
100   - FABRIC_LOGGING_SPEC=INFO
101   #- FABRIC_LOGGING_SPEC=DEBUG
102   - CORE_PEER_TLS_ENABLED=true
103   - CORE_PEER_PROFILE_ENABLED=true
104   - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
105   - CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
106   - CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
107   # Peer specific variables
108   - CORE_PEER_ID=peer0.org2.example.com
109   - CORE_PEER_ADDRESS=peer0.org2.example.com:9051
110   - CORE_PEER_LISTENADDRESS=0.0.0.0:9051
111   - CORE_PEER_CHAINCODEADDRESS=peer0.org2.example.com:9052
112   - CORE_PEER_CHAINCODELISTENADDRESS=0.0.0.0:9052
113   - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org2.example.com:9051
114   - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org2.example.com:9051
115   - CORE_PEER_LOCALMSPID=Org2MSP
116 volumes:
117   - /var/run:/host/var/run/
118   - ../organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/ms
119     p:/etc/hyperledger/fabric/msp
120   - ../organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tl
121     s:/etc/hyperledger/fabric/tls
122   - peer0.org2.example.com:/var/hyperledger/production
123 working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
124 command: peer node start
125 ports:
126   - 9051:9051
127 networks:
128   - test
129 %s/org2/civil/g
```

yaml 파일을 vim 명령어 모드에서 치환하는 %s명령을 사용하여 기존 설정을 변경함.


```

h2kim@h2kim-VBox:~/fabric-samples/e2-network$ vi docker/docker-compose-e2-net.yaml
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ vi network.sh
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ./network.sh up
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb'
LOCAL_VERSION=2.2.0
DOCKER_IMAGE_VERSION=2.2.0
Recreating orderer.example.com ...
Recreating peer0.shop.example.com ...
Recreating orderer.example.com
Recreating peer0.control.example.com ...
Recreating peer0.shop.example.com
Recreating peer0.civil.example.com ...
Recreating peer0.control.example.com
Recreating orderer.example.com ... done
CONTAINER ID          IMAGE                                COMMAND                  CREATED              STATUS
TATUS                PORTS
9a0e710f50b9         hyperledger/fabric-peer:latest     "peer node start"       3 seconds ago       U
p 1 second          7051/tcp, 0.0.0.0:10051->10051/tcp  peer0.shop.example.com
e0c82d8df3ff         hyperledger/fabric-peer:latest     "peer node start"       3 seconds ago       U
p 1 second          0.0.0.0:7051->7051/tcp              peer0.control.example.com
ec9c8813f574         hyperledger/fabric-peer:latest     "peer node start"       4 seconds ago       U
p 1 second          7051/tcp, 0.0.0.0:8051->8051/tcp    peer0.civil.example.com
fb21a1e30ebc         hyperledger/fabric-orderer:latest  "orderer"               4 seconds ago       U
p Less than a second 0.0.0.0:7050->7050/tcp              orderer.example.com
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ./network.sh down
Stopping network
WARNING: Some networks were defined but are not used by any service: test
Stopping peer0.shop.example.com ... done
Stopping peer0.control.example.com ... done
Stopping peer0.civil.example.com ... done
Stopping orderer.example.com ... done
Removing peer0.shop.example.com ... done
Removing peer0.control.example.com ... done
Removing peer0.civil.example.com ... done
Removing orderer.example.com ... done
Removing network net_e2
Removing volume net_peer0.control.example.com
Removing volume net_peer0.shop.example.com
Removing volume net_peer0.civil.example.com
Removing volume net_orderer.example.com
Removing network net_test
Removing volume net_peer0.org3.example.com
WARNING: Volume net_peer0.org3.example.com not found.
No containers available for deletion
No images available for deletion
h2kim@h2kim-VBox:~/fabric-samples/e2-network$

```

네트워크 컨테이너 구동하는 것을 확인할 수 있었음. 미진하게 설정이 테스트 네트워크로 되어 있는 부분도 있어서 WARNING로그가 나오는 것도 확인할 수 있었음.

```

h2kim@h2kim-VBox:~/fabric-samples/e2-network$ ./network.sh up 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 'leveldb with crypto from 'cryptogen'
Bringing up network
LOCAL_VERSION=2.2.0
DOCKER_IMAGE_VERSION=2.2.0
/home/h2kim/fabric-samples/e2-network/./bin/cryptogen
Generate certificates using cryptogen tool
Create control Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-control.yaml --output=organizations
control.example.com
+ res=0
Create civil Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-civil.yaml --output=organizations
civil.example.com
+ res=0
Create shop Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-shop.yaml --output=organizations
shop.example.com
+ res=0
Create Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
Generate CCP files for Org1 and Org2
/home/h2kim/fabric-samples/e2-network/./bin/configtxgen
Generating Orderer Genesis block
+ configtxgen -profile TwoOrgsOrdererGenesis -channelID system-channel -outputBlock ./system-genesis-block/genesis.block
2021-05-01 16:13:44.689 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-05-01 16:13:44.737 KST [common.tools.configtxgen.localconfig] completeInitialization -> INFO 002 orderer type: etcdraft
2021-05-01 16:13:44.737 KST [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 Orderer.EtcdRaft.Options unset, setting to tick_interval:"500ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2021-05-01 16:13:44.737 KST [common.tools.configtxgen.localconfig] Load -> INFO 004 Loaded configuration: /home/h2kim/fabric-samples/e2-network/configtx/configtx.yaml
2021-05-01 16:13:44.739 KST [common.tools.configtxgen] doOutputBlock -> INFO 005 Generating genesis block
2021-05-01 16:13:44.740 KST [common.tools.configtxgen] doOutputBlock -> INFO 006 Writing genesis block
+ res=0

```

네트워크를 생성하고 결과를 확인함.


```

2021-05-01 16:13:56.964 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2021-05-01 16:13:57.022 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
Join Org2 peers to the channel...
Using organization 2
+ peer channel join -b ./channel-artifacts/mychannel.block
+ res=0
2021-05-01 16:14:00.077 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2021-05-01 16:14:00.130 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
Updating anchor peers for org1...
Using organization 1
+ peer channel update -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c mychannel -f ./channel-artifacts/controlMSPanchors.tx --tls --cafile /home/h2kim/fabric-samples/e2-network/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
+ res=0
2021-05-01 16:14:03.221 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2021-05-01 16:14:03.235 KST [channelCmd] update -> INFO 002 Successfully submitted channel update
Anchor peers updated for org 'controlMSP' on channel 'mychannel'
Updating anchor peers for org2...
Using organization 2
+ peer channel update -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c mychannel -f ./channel-artifacts/civilMSPanchors.tx --tls --cafile /home/h2kim/fabric-samples/e2-network/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
+ res=0
2021-05-01 16:14:09.304 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2021-05-01 16:14:09.316 KST [channelCmd] update -> INFO 002 Successfully submitted channel update
Anchor peers updated for org 'civilMSP' on channel 'mychannel'
Channel successfully joined
h2kim@h2kim-VBox:~/fabric-samples/e2-network$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
7051/tcp, 0.0.0.0:8051->8051/tcp
2276017e1b07   hyperledger/fabric-peer:latest     "peer node start"       58 seconds ago Up
0.0.0.0:7051->7051/tcp
102f7a6e8155   hyperledger/fabric-orderer:latest  "orderer"               58 seconds ago Up
0.0.0.0:7050->7050/tcp
aa16a99d645e   hyperledger/fabric-peer:latest     "peer node start"       58 seconds ago Up
7051/tcp, 0.0.0.0:10051->10051/tcp
h2kim@h2kim-VBox:~/fabric-samples/e2-network$

```

\$docker ps 명령어를 확인하여 설정된 컨테이너로 생성하는 것을 확인함.

```

+ cryptogen generate --config=./organizations/cryptogen/crypto-config-shop.yaml --output=organiza
tions
shop.example.com
+ res=0
Create Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organ
izations
+ res=0
Generate CCP files for Org1 and Org2
/home/h2kim/fabric-samples/e2-network/./bin/configtxgen
Generating Orderer Genesis block
+ configtxgen -profile TwoOrgsOrdererGenesis -channelID system-channel -outputBlock ./system-gene
sis-block/genesis.block
2021-05-01 17:03:07.542 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-05-01 17:03:07.592 KST [common.tools.configtxgen.localconfig] completeInitialization -> INFO
002 orderer type: etcdraft
2021-05-01 17:03:07.592 KST [common.tools.configtxgen.localconfig] completeInitialization -> INFO
003 Orderer.EtcdRaft.Options unset, setting to tick_interval:"500ms" election_tick:10 heartbeat_
tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2021-05-01 17:03:07.592 KST [common.tools.configtxgen.localconfig] Load -> INFO 004 Loaded config
uration: /home/h2kim/fabric-samples/e2-network/configtx/configtx.yaml
2021-05-01 17:03:07.594 KST [common.tools.configtxgen] doOutputBlock -> INFO 005 Generating genes
is block
2021-05-01 17:03:07.594 KST [common.tools.configtxgen] doOutputBlock -> INFO 006 Writing genesis
block
+ res=0
Creating network "net_e2" with the default driver
Creating volume "net_peer0.control.example.com" with default driver
Creating volume "net_peer0.shop.example.com" with default driver
Creating volume "net_peer0.civil.example.com" with default driver
Creating volume "net_orderer.example.com" with default driver
Creating couchdb1 ...
Creating couchdb0 ...
Creating couchdb2 ...
Creating orderer.example.com ...
Creating couchdb1
Creating couchdb0
Creating couchdb2
Creating orderer.example.com ... done
Creating peer0.control.example.com ...
Creating couchdb2 ... done
Creating peer0.shop.example.com ...
Creating couchdb1 ... done

```

데이터베이스를 couchdb로 생성하는 것도 실행함.

```

h2kim@h2kim-VBox:~/fabric-samples/e2-network$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED
STATUS        PORTS
ffb6623bccca   hyperledger/fabric-peer:latest      "peer node start"       41 seconds ago
Up 36 seconds  7051/tcp, 0.0.0.0:10051->10051/tcp    peer0.shop.example.com
5f7a4cf2680d   hyperledger/fabric-peer:latest      "peer node start"       42 seconds ago
Up 37 seconds  7051/tcp, 0.0.0.0:8051->8051/tcp      peer0.civil.example.com
243a8ac547ed   hyperledger/fabric-peer:latest      "peer node start"       42 seconds ago
Up 37 seconds  0.0.0.0:7051->7051/tcp                peer0.control.example.com
eef5794c6152   couchdb:3.1                         "tini -- /docker-ent..." 44 seconds ago
Up 40 seconds  4369/tcp, 9100/tcp, 0.0.0.0:8984->5984/tcp couchdb2
0f8ee468e6d5   hyperledger/fabric-orderer:latest   "orderer"                44 seconds ago
Up 40 seconds  0.0.0.0:7050->7050/tcp                orderer.example.com
e93b0afdf325   couchdb:3.1                         "tini -- /docker-ent..." 44 seconds ago
Up 41 seconds  4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp couchdb1
b5a02943d044   couchdb:3.1                         "tini -- /docker-ent..." 44 seconds ago
Up 42 seconds  4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp couchdb0
h2kim@h2kim-VBox:~/fabric-samples/e2-network$

```

도커 컨테이너를 확인하면 couchdb0~2까지 확인함.

3. 차주 계획

- asset-transfer-private-data를 원본으로 프로젝트 go언어 체인코드 구현