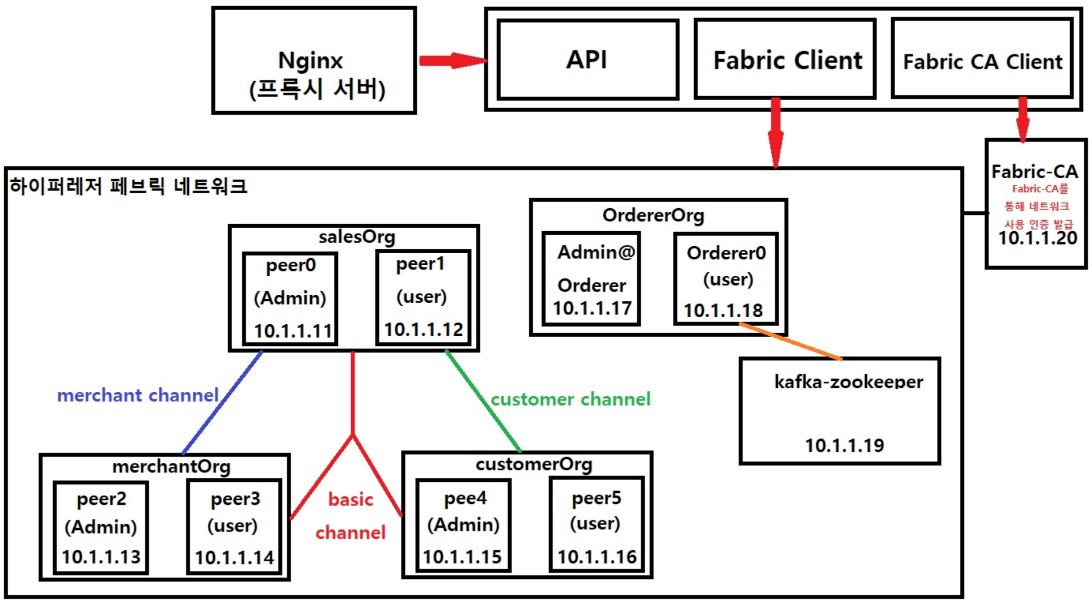
하이퍼레저 Fabric-CA 를 이용하여 멀티 호스트 네트워크 구축

published by HippoMans

[네트워크 구성도]



hosts 파일에 호스트 등록

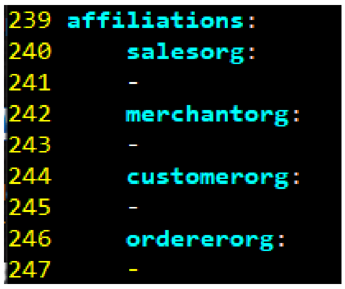
```
# vim /etc/hosts
10.1.1.11   peer0
10.1.1.12   peer1
10.1.1.13   peer2
10.1.1.14   peer3
10.1.1.15   peer4
10.1.1.16   peer5
10.1.1.17   admin-orderer
10.1.1.18   orderer0
10.1.1.19   kafka-zookeeper
10.1.1.20   fabric-ca

# vim /etc/hostname
peer0
```

Fabric-CA-server 실행

1. Fabric-ca server 노드 구동하기 (fabric-ca 노드에서 실시)

```
# vim /etc/profile
export FABRIC_CA_SERVER_HOME=/root/testnet
# source /etc/profile
# mkdir testnet
# cd /root/testnet
# fabric-ca-server start -b admin:adminpw --cfg.affiliations.allowremove --cfg.identities.allowremove -d
```

※ fabric-ca-server 를 stop 하고 fabric-ca-server-config.yaml 을 수정



```
affiliations:
    salesorg:
    -
    merchantorg:
    -
    customerorg:
    -
    ordererorg:
    -
```

2. Fabric-ca 서버을 통해 운영자의 MSP 생성하기(admin-orderer 노드 실행)

-> admin-orderer 에게 네트워크 사용 권한 부여

```
# vim /etc/profile
export FABRIC_CA_CLIENT_HOME=/root/testnet
# source /etc/profile
# mkdir testnet
# cd /root/testnet
# fabric-ca-client enroll -u http://admin:adminpw@10.1.1.20:7054          // fabric-ca 주소
```

```
2020/03/10 00:03:15 [INFO] Created a default configuration file at /root/testnet/fabric-ca-client-config.yaml
2020/03/10 00:03:15 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 00:03:15 [INFO] encoded CSR
2020/03/10 00:03:15 [INFO] Stored client certificate at /root/testnet/msp/signcerts/cert.pem
2020/03/10 00:03:15 [INFO] Stored root CA certificate at /root/testnet/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 00:03:15 [INFO] Stored Issuer public key at /root/testnet/msp/IssuerPublicKey
2020/03/10 00:03:15 [INFO] Stored Issuer revocation public key at /root/testnet/msp/IssuerRevocationPublicKey
```

fabic-ca 에 admin-orderer 을 등록

```
# ls
fabric-ca-client-config.yaml   msp
```

3. 조직 생성 및 조직 운영자 MSP 생성(admin-orderer 노드에서 실시)

```
# fabric-ca-client affiliation list
```

```
affiliation: .
   affiliation: salesorg
   affiliation: customerorg
   affiliation: ordererorg
   affiliation: merchantorg
```

3.1 peer0 에서 실행: salesorg 조직의 admin 인증서 저장하기

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/msp
# vim /etc/profile
export FABRIC_CA_CLIENT_HOME=/root/testnet
# source /etc/profile
# fabric-ca-client getcacert -u http://10.1.1.20:7054 -M /root/testnet/crypto-config/peerOrganizations/salesorg/msp
```

```
2020/03/10 00:19:21 [INFO] Configuration file location: /root/testnet/fabric-ca-client-config.yaml
2020/03/10 00:19:21 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/salesorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 00:19:21 [INFO] Stored Issuer public key at /root/testnet/crypto-
```

```
config/peerOrganizations/salesorg/msp/IssuerPublicKey
2020/03/10 00:19:21 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/salesorg/msp/IssuerRevocationPublicKey
```

3.2 peer2 에서 실행: merchantorg 조직의 admin 인증서 저장하기

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/msp
# vim /etc/profile
export FABRIC_CA_CLIENT_HOME=/root/testnet
# source /etc/profile
# fabric-ca-client getcacert -u http://10.1.1.20:7054 -M /root/testnet/crypto-
config/peerOrganizations/merchantorg/msp
```

```
2020/03/10 00:28:46 [INFO] Configuration file location: /root/testnet/fabric-ca-client-config.yaml
2020/03/10 00:28:46 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/merchantorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 00:28:46 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/merchantorg/msp/IssuerPublicKey
2020/03/10 00:28:46 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/merchantorg/msp/IssuerRevocationPublicKey
```

3.3 peer4 에서 실행: customerorg 조직의 admin 인증서 저장하기

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/msp
# vim /etc/profile
export FABRIC_CA_CLIENT_HOME=/root/testnet
# source /etc/profile
# fabric-ca-client getcacert -u http://10.1.1.20:7054 -M /root/testnet/crypto-
config/peerOrganizations/customerorg/msp
```

```
2020/03/10 00:41:23 [INFO] Configuration file location: /root/testnet/fabric-ca-client-config.yaml
2020/03/10 00:41:23 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/customerorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 00:41:23 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/customerorg/msp/IssuerPublicKey
2020/03/10 00:41:23 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/customerorg/msp/IssuerRevocationPublicKey
```

3.4 admin-orderer 에서 실행: ordererorg 조직의 admin 인증서 저장하기

```
# mkdir -p /root/testnet/crypto-config/ordererOrganizations/ordererorg0/msp
# fabric-ca-client getcacert -u http://10.1.1.20:7054 -M /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/msp
```

```
2020/03/10 00:42:39 [INFO] Configuration file location: /root/testnet/fabric-ca-client-config.yaml
2020/03/10 00:42:39 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 00:42:39 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/msp/IssuerPublicKey
2020/03/10 00:42:39 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/msp/IssuerRevocationPublicKey
```

인증서 이름 변경하기

4.1 peer0 에서 실행: ca.crt 인증서 이름 변경하기

```
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/msp/cacerts
# mv 10-1-1-20-7054.pem ca.crt
# cat ca.crt
```

```
-----BEGIN CERTIFICATE-----
MIICFjCCAb2gAwIBAgIUIaQ2ktL1FrEvLIlQC0CWtMNDQB0wCgYIKoZIzj0EAwIw
aDELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQK
EwtIeXBlcmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRkwFwYDVQQDExBmYWJyaWMt
Y2Etc2VydmVyMB4XDTIwMDMwOTE0MzcwMFoXDTM1MDMwNjE0MzcwMFowaDELMAkG
A1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQKEwtIeXBl
cmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRkwFwYDVQQDExBmYWJyaWMtY2Etc2Vy
dmVyMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEDPxs9pfmp/hoMf5Gm1Gdr91p
pFQSgjTk+HmykfpUVJfavhcTqvOh/LSJQlr+kGfC9rfn1n9RIUDpyvFszPWzjqNF
MEMwDgYDVR0PAQH/BAQDAgEGMBIGA1UdEwEB/wQIMAYBAf8CAQEwHQYDVR0OBBYE
FK7g/VhVfQJsB1kcZTsaS/sK6115MAoGCCqGSM49BAMCA0cAMEQCIGBv7RwOgGsm
1ga/SOQkUSRqrK939nWXMfphwwEnSt7tAiA+shlsU1di5bLg5Om06uRwi8njP2c3
ueEY/nqHHSTNrw==
-----END CERTIFICATE-----
```

4.2 peer2 에서 실행: ca.crt 인증서 이름 변경하기

```
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/msp/cacerts
# mv 10-1-1-20-7054.pem ca.crt
# cat ca.crt
```

```
-----BEGIN CERTIFICATE-----
MIICFjCCAb2gAwIBAgIUIaQ2ktL1FrEvLIlQC0CWtMNDQB0wCgYIKoZIzj0EAwIw
aDELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQK
EwtIeXBlcmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRkwFwYDVQQDExBmYWJyaWMt
Y2Etc2VydmVyMB4XDTIwMDMwOTE0MzcwMFoXDTM1MDMwNjE0MzcwMFowaDELMAkG
A1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQKEwtIeXBl
cmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRkwFwYDVQQDExBmYWJyaWMtY2Etc2Vy
dmVyMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEDPxs9pfmp/hoMf5Gm1Gdr91p
pFQSgjTk+HmykfpUVJfavhcTqvOh/LSJQlr+kGfC9rfn1n9RIUDpyvFszPWzjqNF
MEMwDgYDVR0PAQH/BAQDAgEGMBIGA1UdEwEB/wQIMAYBAf8CAQEwHQYDVR0OBBYE
FK7g/VhVfQJsB1kcZTsaS/sK6115MAoGCCqGSM49BAMCA0cAMEQCIGBv7RwOgGsm
1ga/SOQkUSRqrK939nWXMfphwwEnSt7tAiA+shlsU1di5bLg5Om06uRwi8njP2c3
ueEY/nqHHSTNrw==
-----END CERTIFICATE-----
```

4.3 peer4 에서 실행: ca.crt 인증서 이름 변경하기

```
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/msp/cacerts
# mv 10-1-1-20-7054.pem ca.crt
# cat ca.crt
```

```
-----BEGIN CERTIFICATE-----
MIICFjCCAb2gAwIBAgIUIaQ2ktL1FrEvLIlQC0CWtMNDQB0wCgYIKoZIzj0EAwIw
aDELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQK
EwtIeXBlcmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRkwFwYDVQQDExBmYWJyaWMt
```

```
Y2Etc2VydmVyMB4XDTIwMDMwOTE0MzcwMFoXDTM1MDMwNjE0MzcwMFowaDELMAkG
A1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQKEwtIeXBl
cmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRkwFwYDVQQDExBmYWJyaWMtY2Etc2Vy
dmVyMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEDPxs9pfmp/hoMf5Gm1Gdr91p
pFQSgjTk+HmykfpUVJfavhcTqvOh/LSJQlr+kGfC9rfn1n9RIUDpyvFszPWzjqNF
MEMwDgYDVR0PAQH/BAQDAgEGMBIGA1UdEwEB/wQIMAYBAf8CAQEwHQYDVR0OBBYE
FK7g/VhVfQJsB1kcZTsaS/sK6115MAoGCCqGSM49BAMCA0cAMEQCIGBv7RwOgGsm
1ga/SOQkUSRqrK939nWXMfphwwEnSt7tAiA+shlsU1di5bLg5Om06uRwi8njP2c3
ueEY/nqHHSTNrw==
-----END CERTIFICATE-----
```

4.4 admin-orderer 에서 실행: ca.crt 인증서 이름 변경하기

```
# cd /root/testnet/crypto-config/ordererOrganizations/ordererorg0/msp/cacerts
# mv 10-1-1-20-7054.pem ca.crt
# cat ca.crt
```

```
-----BEGIN CERTIFICATE-----
MIICFjCCAb2gAwIBAgIUIaQ2ktL1FrEvLIlQC0CWtMNDQB0wCgYIKoZIzj0EAwIw
aDELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQK
EwtIeXBlcmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRkwFwYDVQQDExBmYWJyaWMt
Y2Etc2VydmVyMB4XDTIwMDMwOTE0MzcwMFoXDTM1MDMwNjE0MzcwMFowaDELMAkG
A1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQKEwtIeXBl
cmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRkwFwYDVQQDExBmYWJyaWMtY2Etc2Vy
dmVyMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEDPxs9pfmp/hoMf5Gm1Gdr91p
pFQSgjTk+HmykfpUVJfavhcTqvOh/LSJQlr+kGfC9rfn1n9RIUDpyvFszPWzjqNF
MEMwDgYDVR0PAQH/BAQDAgEGMBIGA1UdEwEB/wQIMAYBAf8CAQEwHQYDVR0OBBYE
FK7g/VhVfQJsB1kcZTsaS/sK6115MAoGCCqGSM49BAMCA0cAMEQCIGBv7RwOgGsm
1ga/SOQkUSRqrK939nWXMfphwwEnSt7tAiA+shlsU1di5bLg5Om06uRwi8njP2c3
ueEY/nqHHSTNrw==
-----END CERTIFICATE-----
```

5 각 조직의 운영자 계정 등록하기 (admin-orderer 에서 실행)

5.1 Admin@salesorg 계정 등록

```
# vim /root/testnet/fabric-ca-client-config.yaml
id:
  name: Admin@salesorg
  type: client
  affiliation: salesorg
  maxenrollments: 0
  attributes:
    - name: hf.Registrar.Roles
      value: client, orderer, peer, user
    - name: hf.Registrar.DelegateRoles
      value: client, orderer, peer, user
    - name: hf.Registrar.Attributes
      value: "*"
    - name: hf.GenCRL
      value: true
    - name: hf.Revoker
      value: true
    - name: hf.AffiliationMgr
      value: true
    - name: hf.IntermediateCA
      value: true
    - name: role
      value: admin
      ecert: true

# fabric-ca-client register --id.secret=salesorgpassword
```

```
2020/03/10 01:10:29 [INFO] Configuration file location: /root/testnet/fabric-ca-client-config.yaml
Password: salesorgpassword
```

Fabric-ca 서버 변화

```
2020/03/10 01:10:29 [DEBUG] Checking if registrar can register attribute: role
2020/03/10 01:10:29 [DEBUG] Performing authorization check...
2020/03/10 01:10:29 [DEBUG] Registering user id: Admin@salesorg
2020/03/10 01:10:29 [DEBUG] Max enrollment value verification - User specified max enrollment: 0, CA max enrollment: -1
2020/03/10 01:10:29 [DEBUG] DB: Getting identity Admin@salesorg
2020/03/10 01:10:29 [DEBUG] DB: Add identity Admin@salesorg
2020/03/10 01:10:29 [DEBUG] Successfully added identity Admin@salesorg to the database
2020/03/10 01:10:29 [INFO] 10.1.1.17:49942 POST /register 201 0 "OK
```

5.2 Admin@merchantorg 계정 등록

```
# vim /root/testnet/fabric-ca-client-config.yaml
id:
  name: Admin@merchantorg
  type: client
  affiliation: merchantorg
  maxenrollments: 0
  attributes:
    - name: hf.Registrar.Roles
      value: client, orderer, peer, user
    - name: hf.Registrar.DelegateRoles
      value: client, orderer, peer, user
    - name: hf.Registrar.Attributes
      value: "*"
    - name: hf.GenCRL
      value: true
    - name: hf.Revoker
      value: true
    - name: hf.AffiliationMgr
      value: true
    - name: hf.IntermediateCA
      value: true
    - name: role
      value: admin
```

```
        ecert: true

# fabric-ca-client register --id.secret=merchantorgpassword
```

```
2020/03/10 01:14:04 [INFO] Configuration file location: /root/testnet/fabric-ca-client-config.yaml
Password: merchantorgpassword
```

Fabric-ca 서버 변화

```
2020/03/10 01:14:04 [DEBUG] Checking if registrar can register attribute: role
2020/03/10 01:14:04 [DEBUG] Performing authorization check...
2020/03/10 01:14:04 [DEBUG] Registering user id: Admin@merchantorg
2020/03/10 01:14:04 [DEBUG] Max enrollment value verification - User specified max enrollment: 0, CA max enrollment: -1
2020/03/10 01:14:04 [DEBUG] DB: Getting identity Admin@merchantorg
2020/03/10 01:14:04 [DEBUG] DB: Add identity Admin@merchantorg
2020/03/10 01:14:04 [DEBUG] Successfully added identity Admin@merchantorg to the database
2020/03/10 01:14:04 [INFO] 10.1.1.17:49944 POST /register 201 0 "OK"
```

5.3 Admin@customerorg 계정 등록

```
# vim /root/testnet/fabric-ca-client-config.yaml
id:
  name: Admin@customerorg
  type: client
  affiliation: customerorg
  maxenrollments: 0
  attributes:
    - name: hf.Registrar.Roles
      value: client, orderer, peer, user
    - name: hf.Registrar.DelegateRoles
      value: client, orderer, peer, user
    - name: hf.Registrar.Attributes
      value: "*"
    - name: hf.GenCRL
      value: true
    - name: hf.Revoker
      value: true
    - name: hf.AffiliationMgr
      value: true
    - name: hf.IntermediateCA
      value: true
    - name: role
      value: admin
      ecert: true

# fabric-ca-client register --id.secret=customerorgpassword
```

```
2020/03/10 01:16:37 [INFO] Configuration file location: /root/testnet/fabric-ca-client-config.yaml
Password: customerorgpassword
```

Fabric-ca 서버 변화

```
2020/03/10 01:16:38 [DEBUG] Checking if registrar can register attribute: role
2020/03/10 01:16:38 [DEBUG] Performing authorization check...
2020/03/10 01:16:38 [DEBUG] Registering user id: Admin@customerorg
2020/03/10 01:16:38 [DEBUG] Max enrollment value verification - User specified max enrollment: 0, CA max enrollment: -1
2020/03/10 01:16:38 [DEBUG] DB: Getting identity Admin@customerorg
2020/03/10 01:16:38 [DEBUG] DB: Add identity Admin@customerorg
2020/03/10 01:16:38 [DEBUG] Successfully added identity Admin@customerorg to the database
2020/03/10 01:16:38 [INFO] 10.1.1.17:49946 POST /register 201 0 "OK"
```

5.4 Admin@ordererorg 계정 등록

```
# vim /root/testnet/fabric-ca-client-config.yaml
id:
  name: Admin@ordererorg
  type: client
  affiliation: ordererorg
  maxenrollments: 0
  attributes:
    - name: hf.Registrar.Roles
      value: client, orderer, peer, user
    - name: hf.Registrar.DelegateRoles
      value: client, orderer, peer, user
    - name: hf.Registrar.Attributes
      value: "*"
    - name: hf.GenCRL
      value: true
    - name: hf.Revoker
      value: true
    - name: hf.AffiliationMgr
      value: true
    - name: hf.IntermediateCA
      value: true
    - name: role
      value: admin
      ecert: true

# fabric-ca-client register --id.secret=ordererorgpassword
```

```
2020/03/10 01:19:03 [INFO] Configuration file location: /root/testnet/fabric-ca-client-config.yaml
Password: ordererorgpassword
```

Fabric-ca 서버 변화

```
2020/03/10 01:19:03 [DEBUG] Checking if registrar can register attribute: role
2020/03/10 01:19:03 [DEBUG] Performing authorization check...
2020/03/10 01:19:03 [DEBUG] Registering user id: Admin@ordererorg
2020/03/10 01:19:03 [DEBUG] Max enrollment value verification - User specified max enrollment: 0, CA max enrollment: -1
2020/03/10 01:19:03 [DEBUG] DB: Getting identity Admin@ordererorg
2020/03/10 01:19:03 [DEBUG] DB: Add identity Admin@ordererorg
```

```
2020/03/10 01:19:03 [DEBUG] Successfully added identity Admin@ordererorg to the database
2020/03/10 01:19:03 [INFO] 10.1.1.17:49948 POST /register 201 0 "OK"
```

6. Identity 확인(admin-orderer 노드에서 실시)

```
# fabric-ca-client identity list
```

```
Name: admin, Type: client, Affiliation: , Max Enrollments: -1, Attributes: [{Name:hf.Registrar.Roles Value:* ECert:false}
{Name:hf.Registrar.DelegateRoles Value:* ECert:false} {Name:hf.Revoker Value:1 ECert:false} {Name:hf.IntermediateCA
Value:1 ECert:false} {Name:hf.GenCRL Value:1 ECert:false} {Name:hf.Registrar.Attributes Value:* ECert:false}
{Name:hf.AffiliationMgr Value:1 ECert:false}]
Name: Admin@salesorg, Type: client, Affiliation: salesorg, Max Enrollments: -1, Attributes: [{Name:hf.Registrar.Roles
Value:client, orderer, peer, user ECert:false} {Name:hf.Registrar.DelegateRoles Value:client, orderer, peer, user ECert:false}
{Name:hf.Registrar.Attributes Value:"*" ECert:false} {Name:hf.GenCRL Value:1 ECert:false} {Name:hf.Revoker Value:1
ECert:false} {Name:hf.AffiliationMgr Value:1 ECert:false} {Name:hf.IntermediateCA Value:1 ECert:false} {Name:role
Value:admin ECert:true} {Name:hf.EnrollmentID Value:Admin@salesorg ECert:true} {Name:hf.Type Value:client ECert:true}
{Name:hf.Affiliation Value:salesorg ECert:true}]
Name: Admin@merchantorg, Type: client, Affiliation: merchantorg, Max Enrollments: -1, Attributes:
[{Name:hf.Registrar.Roles Value:client, orderer, peer, user ECert:false} {Name:hf.Registrar.DelegateRoles Value:client, orderer,
peer, user ECert:false} {Name:hf.Registrar.Attributes Value:"*" ECert:false} {Name:hf.GenCRL Value:1 ECert:false}
{Name:hf.Revoker Value:1 ECert:false} {Name:hf.AffiliationMgr Value:1 ECert:false} {Name:hf.IntermediateCA Value:1
ECert:false} {Name:role Value:admin ECert:true} {Name:hf.EnrollmentID Value:Admin@merchantorg ECert:true}
{Name:hf.Type Value:client ECert:true} {Name:hf.Affiliation Value:merchantorg ECert:true}]
Name: Admin@customerorg, Type: client, Affiliation: customerorg, Max Enrollments: -1, Attributes:
[{Name:hf.Registrar.Roles Value:client, orderer, peer, user ECert:false} {Name:hf.Registrar.DelegateRoles Value:client, orderer,
peer, user ECert:false} {Name:hf.Registrar.Attributes Value:"*" ECert:false} {Name:hf.GenCRL Value:1 ECert:false}
{Name:hf.Revoker Value:1 ECert:false} {Name:hf.AffiliationMgr Value:1 ECert:false} {Name:hf.IntermediateCA Value:1
ECert:false} {Name:role Value:admin ECert:true} {Name:hf.EnrollmentID Value:Admin@customerorg ECert:true}
{Name:hf.Type Value:client ECert:true} {Name:hf.Affiliation Value:customerorg ECert:true}]
Name: Admin@ordererorg, Type: client, Affiliation: ordererorg, Max Enrollments: -1, Attributes: [{Name:hf.Registrar.Roles
Value:client, orderer, peer, user ECert:false} {Name:hf.Registrar.DelegateRoles Value:client, orderer, peer, user ECert:false}
{Name:hf.Registrar.Attributes Value:"*" ECert:false} {Name:hf.GenCRL Value:1 ECert:false} {Name:hf.Revoker Value:1
ECert:false} {Name:hf.AffiliationMgr Value:1 ECert:false} {Name:hf.IntermediateCA Value:1 ECert:false} {Name:role
Value:admin ECert:true} {Name:hf.EnrollmentID Value:Admin@ordererorg ECert:true} {Name:hf.Type Value:client ECert:true}
{Name:hf.Affiliation Value:ordererorg ECert:true}]
```

7 각 조직의 운영자 노드에 MSP 생성하기

7.1 salesorg MSP 생성(peer0 에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg
# fabric-ca-client enroll -u http://Admin@salesorg:salesorgpassword@10.1.1.20:7054 -H /root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg
```

```
2020/03/10 01:29:40 [INFO] Created a default configuration file at /root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg/fabric-ca-client-config.yaml
2020/03/10 01:29:40 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 01:29:40 [INFO] encoded CSR
2020/03/10 01:29:40 [INFO] Stored client certificate at /root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg/msp/signcerts/cert.pem
2020/03/10 01:29:40 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg/msp/cacerts/10-1-1-20-7054.pem
```

```
2020/03/10 01:29:40 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg/msp/IssuerPublicKey
2020/03/10 01:29:40 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp/cacerts/
# mv 10-1-1-20-7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

7.2 merchantorg MSP 생성(peer2 에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/users/Admin@merchantorg
# fabric-ca-client enroll -u http://Admin@merchantorg:merchantorgpassword@10.1.1.20:7054 -H
/root/testnet/crypto-config/peerOrganizations/merchantorg/users/Admin@merchantorg
```

```
2020/03/10 01:36:46 [INFO] Created a default configuration file at /root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/fabric-ca-client-config.yaml
2020/03/10 01:36:46 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 01:36:46 [INFO] encoded CSR
2020/03/10 01:36:46 [INFO] Stored client certificate at /root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/signcerts/cert.pem
2020/03/10 01:36:46 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 01:36:46 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/IssuerPublicKey
2020/03/10 01:36:46 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/cacerts/
# mv 10-1-1-20-7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

7.3 customerorg MSP 생성(peer4 에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/users/Admin@customerorg
# fabric-ca-client enroll -u http://Admin@customerorg:customerorgpassword@10.1.1.20:7054 -H
/root/testnet/crypto-config/peerOrganizations/customerorg/users/Admin@customerorg
```

```
2020/03/10 01:42:29 [INFO] Created a default configuration file at /root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/fabric-ca-client-config.yaml
2020/03/10 01:42:29 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 01:42:29 [INFO] encoded CSR
2020/03/10 01:42:29 [INFO] Stored client certificate at /root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp/signcerts/cert.pem
2020/03/10 01:42:29 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 01:42:29 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp/IssuerPublicKey
```

```
2020/03/10 01:42:29 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-config/peerOrganizations/customerorg/users/Admin@customerorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/users/Admin@customerorg/msp/cacerts/
# mv 10-1-1-20-7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

7.4 ordererorg MSP 생성(admin-orderer 에서 실행)

```
# mkdir -p /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg
# fabric-ca-client enroll -u http://Admin@ordererorg:ordererorgpassword@10.1.1.20:7054 -H /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg
```

```
2020/03/10 01:47:55 [INFO] Created a default configuration file at /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/fabric-ca-client-config.yaml
2020/03/10 01:47:55 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 01:47:55 [INFO] encoded CSR
2020/03/10 01:47:55 [INFO] Stored client certificate at /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/msp/signcerts/cert.pem
2020/03/10 01:47:55 [INFO] Stored root CA certificate at /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 01:47:55 [INFO] Stored Issuer public key at /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/msp/IssuerPublicKey
2020/03/10 01:47:55 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/msp/IssuerRevocationPublicKey
```

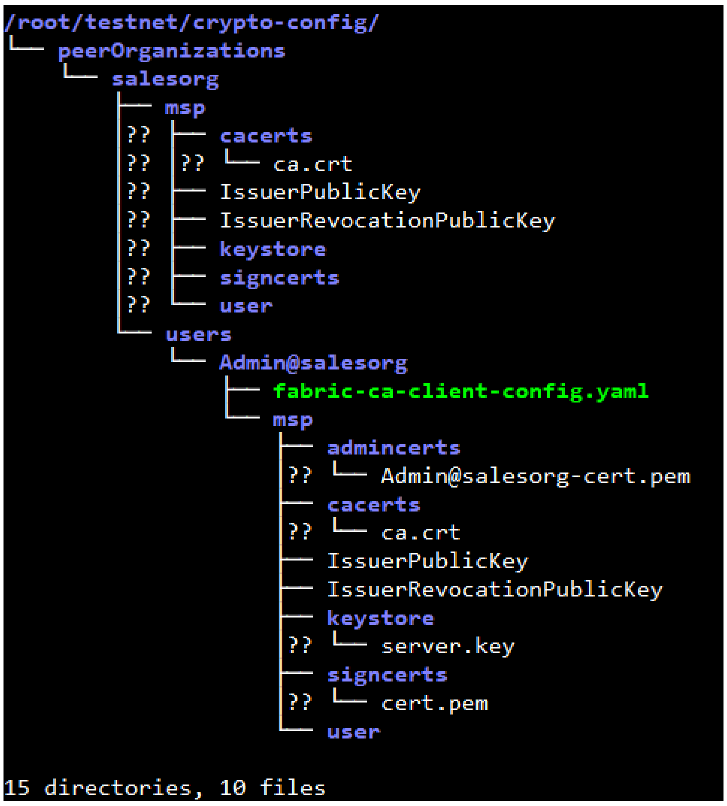
fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/msp/cacerts/
# mv 10-1-1-20-7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

8. 각 조직의 운영자 MSP 디렉토리에 admincerts 디렉토리 생성 후 signcerts 디렉토리의 공개키 파일을 (Admin@조직이름-cert.pem 으로 복사하기)

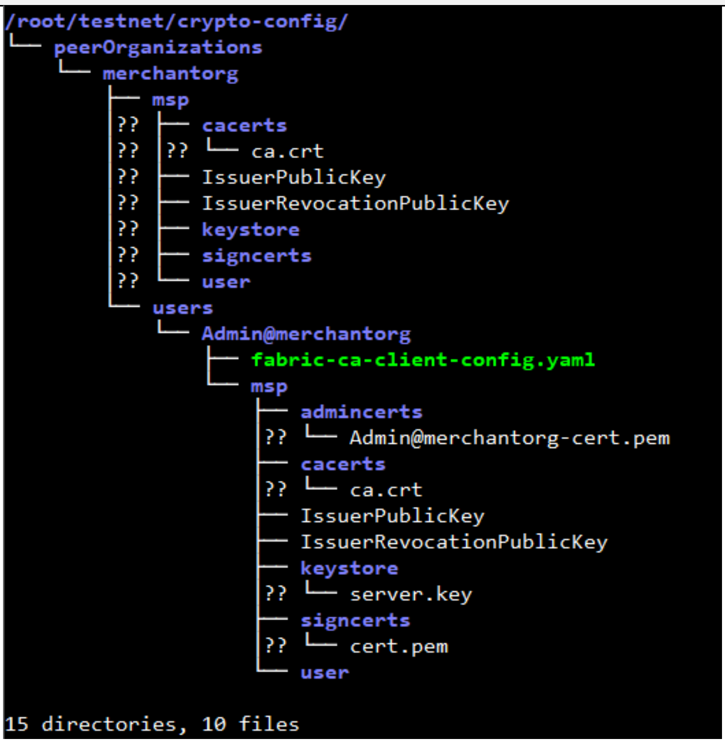
8.1 peer0 노드에서 실행 (salesorg 조직)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp/admincerts
# cp /root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp/signcerts/cert.pem /root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp/admincerts/Admin@salesorg-cert.pem
# tree /root/testnet/crypto-config/
```


```
/root/testnet/crypto-config/
└── peerOrganizations
    └── salesorg
        ├── msp
        ??  ├── cacerts
        ??  ??  └── ca.crt
        ??  ├── IssuerPublicKey
        ??  ├── IssuerRevocationPublicKey
        ??  ├── keystore
        ??  ├── signcerts
        ??  └── user
        └── users
            └── Admin@salesorg
                ├── fabric-ca-client-config.yaml
                └── msp
                    ├── admincerts
                    ??  └── Admin@salesorg-cert.pem
                    ├── cacerts
                    ??  └── ca.crt
                    ├── IssuerPublicKey
                    ├── IssuerRevocationPublicKey
                    ├── keystore
                    ??  └── server.key
                    ├── signcerts
                    ??  └── cert.pem
                    └── user

15 directories, 10 files
```

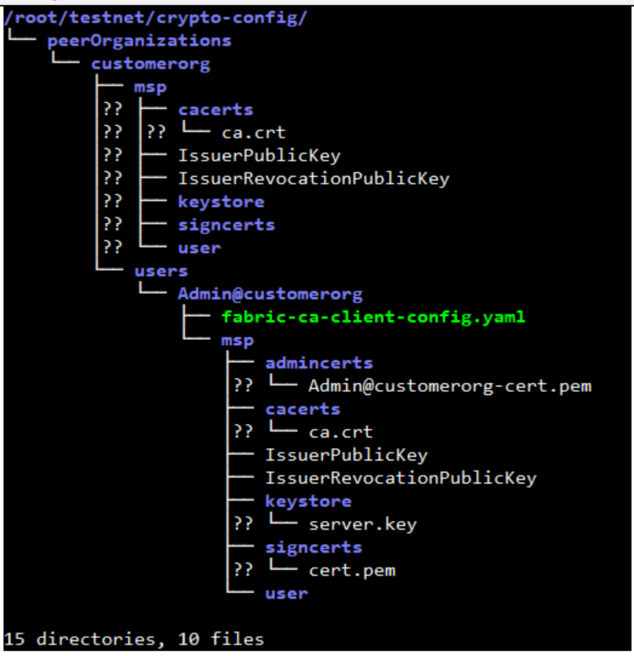
8.2 peer2 노드에서 실행 (merchantorg 조직)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/admincerts
# cp /root/testnet/crypto-config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/signcerts/cert.pem /root/testnet/crypto-config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/admincerts/Admin@merchantorg-cert.pem
# tree /root/testnet/crypto-config/
```


```
/root/testnet/crypto-config/
└── peerOrganizations
    └── merchantorg
        ├── msp
        ??  ├── cacerts
        ??  ??  └── ca.crt
        ??  ├── IssuerPublicKey
        ??  ├── IssuerRevocationPublicKey
        ??  ├── keystore
        ??  ├── signcerts
        ??  └── user
        └── users
            └── Admin@merchantorg
                ├── fabric-ca-client-config.yaml
                └── msp
                    ├── admincerts
                    ??  └── Admin@merchantorg-cert.pem
                    ├── cacerts
                    ??  └── ca.crt
                    ├── IssuerPublicKey
                    ├── IssuerRevocationPublicKey
                    ├── keystore
                    ??  └── server.key
                    ├── signcerts
                    ??  └── cert.pem
                    └── user

15 directories, 10 files
```

8.3 peer4 노드에서 실행 (customerorg 조직)

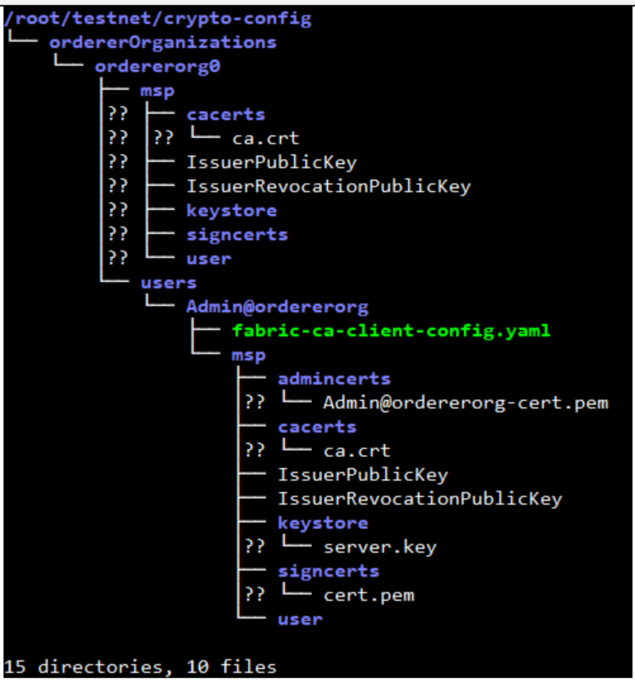
```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/users/Admin@customerorg/msp/admincerts
# cp /root/testnet/crypto-config/peerOrganizations/customerorg/users/Admin@customerorg/msp/signcerts/cert.pem /root/testnet/crypto-config/peerOrganizations/customerorg/users/Admin@customerorg/msp/admincerts/Admin@customerorg-cert.pem
# tree /root/testnet/crypto-config/
```



```
/root/testnet/crypto-config/
└── peerOrganizations
    └── customerorg
        ├── msp
        ?? ├── cacerts
        ?? ?? └── ca.crt
        ?? ├── IssuerPublicKey
        ?? ├── IssuerRevocationPublicKey
        ?? ├── keystore
        ?? ├── signcerts
        ?? └── user
        └── users
            └── Admin@customerorg
                ├── fabric-ca-client-config.yaml
                └── msp
                    ├── admincerts
                    ?? └── Admin@customerorg-cert.pem
                    ├── cacerts
                    ?? └── ca.crt
                    ├── IssuerPublicKey
                    ├── IssuerRevocationPublicKey
                    ├── keystore
                    ?? └── server.key
                    ├── signcerts
                    ?? └── cert.pem
                    └── user

15 directories, 10 files
```

8.4 admin-orderer 노드에서 실행 (ordererorg 조직)

```
# mkdir -p /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/msp/admincerts
# cd /root/testnet/crypto-config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/msp
# cp signcerts/cert.pem   admincerts/Admin@ordererorg-cert.pem
# tree /root/testnet/crypto-config
```



```
/root/testnet/crypto-config
└── ordererOrganizations
    └── ordererorg0
        ├── msp
        ?? ├── cacerts
        ?? ?? └── ca.crt
        ?? ├── IssuerPublicKey
        ?? ├── IssuerRevocationPublicKey
        ?? ├── keystore
        ?? ├── signcerts
        ?? └── user
        └── users
            └── Admin@ordererorg
                ├── fabric-ca-client-config.yaml
                └── msp
                    ├── admincerts
                    ?? └── Admin@ordererorg-cert.pem
                    ├── cacerts
                    ?? └── ca.crt
                    ├── IssuerPublicKey
                    ├── IssuerRevocationPublicKey
                    ├── keystore
                    ?? └── server.key
                    ├── signcerts
                    ?? └── cert.pem
                    └── user

15 directories, 10 files
```

Peer 및 Orderer 노드 MSP 생성하기

원래는 peer 의 조직 admin 에서 MSP 를 생성하는 것이 원리에서는 맞지만 admin 에서 등록이 안되기 때문에 admin-orderer 에서 실행한다. fabric-ca 서버에 저장하는 것이 목적이기 때문에 admin-orderer 에서 해도 조직의 admin 과 같은 결과를 가진다.

1.1 admin-orderer 노드에서 실행 (peer0 등록)

```
# vim testnet/fabric-ca-client-config.yaml
id:
   name: peer0
   type: peer
   affiliation: salesorg
   maxenrollments: 0
   attributes:
     - name: role
       value: peer
       ecert: true

# fabric-ca-client register -d --id.name peer0 --id.secret peer0password --id.type peer
```

```
2020/03/10 02:45:45 [DEBUG] Received response
statusCode=201 (201 Created)
2020/03/10 02:45:45 [DEBUG] Response body result: map[secret:peer0password]
2020/03/10 02:45:45 [DEBUG] The register request completed successfully
Password: peer0password
```

fabric-ca 에 peer0 을 등록한 내용을 확인

```
# fabric-ca-client identity list | grep peer0
```

```
Name: peer0, Type: peer, Affiliation: salesorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer0 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:salesorg
ECert:true}]
```

1.2 admin-orderer 노드에서 실행 (peer1 등록)

```
# vim testnet/fabric-ca-client-config.yaml
id:
   name: peer1
   type: peer
   affiliation: salesorg
   maxenrollments: 0
   attributes:
     - name: role
       value: peer
       ecert: true

# fabric-ca-client register -d --id.name peer1 --id.secret peer1password --id.type peer
```

```
2020/03/10 02:47:46 [DEBUG] Received response
statusCode=201 (201 Created)
2020/03/10 02:47:46 [DEBUG] Response body result: map[secret:peer1password]
2020/03/10 02:47:46 [DEBUG] The register request completed successfully
Password: peer1password
```

fabric-ca 에 peer1 을 등록한 내용을 확인

```
# fabric-ca-client identity list | grep peer1
```

```
Name: peer1, Type: peer, Affiliation: salesorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer1 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:salesorg
ECert:true}]
```

2.1 admin-orderer 노드에서 실행 (peer2 등록)

```
# vim testnet/fabric-ca-client-config.yaml
id:
   name: peer2
   type: peer
   affiliation: merchantorg
   maxenrollments: 0
   attributes:
     - name: role
       value: peer
       ecert: true

# fabric-ca-client register -d --id.name peer2 --id.secret peer2password --id.type peer
```

```
2020/03/10 02:51:23 [DEBUG] Received response
statusCode=201 (201 Created)
2020/03/10 02:51:23 [DEBUG] Response body result: map[secret:peer2password]
2020/03/10 02:51:23 [DEBUG] The register request completed successfully
Password: peer2password
```

fabric-ca 에 peer2 을 등록한 내용을 확인

```
# fabric-ca-client identity list | grep peer2
```

```
Name: peer2, Type: peer, Affiliation: merchantorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer2 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:merchantorg
ECert:true}]
```

2.2 admin-orderer 노드에서 실행 (peer3 등록)

```
# vim testnet/fabric-ca-client-config.yaml
id:
   name: peer3
   type: peer
   affiliation: merchantorg
   maxenrollments: 0
   attributes:
     - name: role
       value: peer
       ecert: true

# fabric-ca-client register -d --id.name peer3 --id.secret peer3password --id.type peer
```

```
2020/03/10 02:52:23 [DEBUG] Received response
statusCode=201 (201 Created)
2020/03/10 02:52:23 [DEBUG] Response body result: map[secret:peer3password]
```

```
2020/03/10 02:52:23 [DEBUG] The register request completed successfully
Password: peer3password
```

fabric-ca 에 peer3 을 등록한 내용을 확인

```
# fabric-ca-client identity list | grep peer3
```

```
Name: peer3, Type: peer, Affiliation: merchantorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer3 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:merchantorg
ECert:true}]
```

3.1 admin-orderer 노드에서 실행 (peer4 등록)

```
# vim testnet/fabric-ca-client-config.yaml
id:
   name: peer4
   type: peer
   affiliation: customerorg
   maxenrollments: 0
   attributes:
     - name: role
       value: peer
       ecert: true

# fabric-ca-client register -d --id.name peer4 --id.secret peer4password --id.type peer
```

```
2020/03/10 02:53:40 [DEBUG] Received response
statusCode=201 (201 Created)
2020/03/10 02:53:40 [DEBUG] Response body result: map[secret:peer4password]
2020/03/10 02:53:40 [DEBUG] The register request completed successfully
Password: peer4password
```

fabric-ca 에 peer4 을 등록한 내용을 확인

```
# fabric-ca-client identity list | grep peer4
```

```
Name: peer4, Type: peer, Affiliation: customerorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer4 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:customerorg
ECert:true}]
```

3.2 admin-orderer 노드에서 실행 (peer5 등록)

```
# vim testnet/fabric-ca-client-config.yaml
id:
   name: peer5
   type: peer
   affiliation: customerorg
   maxenrollments: 0
   attributes:
     - name: role
       value: peer
       ecert: true

# fabric-ca-client register -d --id.name peer5 --id.secret peer5password --id.type peer
```

```
2020/03/10 02:54:33 [DEBUG] Received response
statusCode=201 (201 Created)
2020/03/10 02:54:33 [DEBUG] Response body result: map[secret:peer5password]
2020/03/10 02:54:33 [DEBUG] The register request completed successfully
Password: peer5password
```

fabric-ca 에 peer5 을 등록한 내용을 확인

```
# fabric-ca-client identity list | grep peer5
```

```
Name: peer5, Type: peer, Affiliation: customerorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer5 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:customerorg
ECert:true}]
```

3.2 admin-orderer 노드에서 실행 (orderer0 등록)

```
# vim testnet/fabric-ca-client-config.yaml
id:
   name: orderer0
   type: orderer
   affiliation: ordererorg
   maxenrollments: 0
   attributes:
     - name: role
       value: orderer
       ecert: true

# fabric-ca-client register -d --id.name orderer0 --id.secret orderer0password --id.type orderer
```

```
2020/03/10 02:56:12 [DEBUG] Received response
statusCode=201 (201 Created)
2020/03/10 02:56:12 [DEBUG] Response body result: map[secret:orderer0password]
2020/03/10 02:56:12 [DEBUG] The register request completed successfully
Password: orderer0password
```

fabric-ca 에 orderer0 을 등록한 내용을 확인

```
# fabric-ca-client identity list | grep orderer0
```

```
Name: orderer0, Type: orderer, Affiliation: ordererorg, Max Enrollments: -1, Attributes: [{Name:role Value:orderer ECert:true}
{Name:hf.EnrollmentID Value:orderer0 ECert:true} {Name:hf.Type Value:orderer ECert:true} {Name:hf.Affiliation
Value:ordererorg ECert:true}]
```

4. Identity 확인(admin-orderer 노드에서 실시)

```
# fabric-ca-client identity list
```

```
Name: admin, Type: client, Affiliation: , Max Enrollments: -1, Attributes: [{Name:hf.Registrar.Roles Value:* ECert:false}
{Name:hf.Registrar.DelegateRoles Value:* ECert:false} {Name:hf.Revoker Value:1 ECert:false} {Name:hf.IntermediateCA
Value:1 ECert:false} {Name:hf.GenCRL Value:1 ECert:false} {Name:hf.Registrar.Attributes Value:* ECert:false}
{Name:hf.AffiliationMgr Value:1 ECert:false}]
Name: Admin@salesorg, Type: client, Affiliation: salesorg, Max Enrollments: -1, Attributes: [{Name:hf.Registrar.Roles
Value:client, orderer, peer, user ECert:false} {Name:hf.Registrar.DelegateRoles Value:client, orderer, peer, user ECert:false}
{Name:hf.Registrar.Attributes Value:"*" ECert:false} {Name:hf.GenCRL Value:1 ECert:false} {Name:hf.Revoker Value:1
ECert:false} {Name:hf.AffiliationMgr Value:1 ECert:false} {Name:hf.IntermediateCA Value:1 ECert:false} {Name:role
Value:admin ECert:true} {Name:hf.EnrollmentID Value:Admin@salesorg ECert:true} {Name:hf.Type Value:client ECert:true}
{Name:hf.Affiliation Value:salesorg ECert:true}]
Name: Admin@merchantorg, Type: client, Affiliation: merchantorg, Max Enrollments: -1, Attributes:
[{Name:hf.Registrar.Roles Value:client, orderer, peer, user ECert:false} {Name:hf.Registrar.DelegateRoles Value:client, orderer,
peer, user ECert:false} {Name:hf.Registrar.Attributes Value:"*" ECert:false} {Name:hf.GenCRL Value:1 ECert:false}
{Name:hf.Revoker Value:1 ECert:false} {Name:hf.AffiliationMgr Value:1 ECert:false} {Name:hf.IntermediateCA Value:1
ECert:false} {Name:role Value:admin ECert:true} {Name:hf.EnrollmentID Value:Admin@merchantorg ECert:true}
{Name:hf.Type Value:client ECert:true} {Name:hf.Affiliation Value:merchantorg ECert:true}]
Name: Admin@customerorg, Type: client, Affiliation: customerorg, Max Enrollments: -1, Attributes:
[{Name:hf.Registrar.Roles Value:client, orderer, peer, user ECert:false} {Name:hf.Registrar.DelegateRoles Value:client, orderer,
peer, user ECert:false} {Name:hf.Registrar.Attributes Value:"*" ECert:false} {Name:hf.GenCRL Value:1 ECert:false}
{Name:hf.Revoker Value:1 ECert:false} {Name:hf.AffiliationMgr Value:1 ECert:false} {Name:hf.IntermediateCA Value:1
ECert:false} {Name:role Value:admin ECert:true} {Name:hf.EnrollmentID Value:Admin@customerorg ECert:true}
{Name:hf.Type Value:client ECert:true} {Name:hf.Affiliation Value:customerorg ECert:true}]
Name: Admin@ordererorg, Type: client, Affiliation: ordererorg, Max Enrollments: -1, Attributes: [{Name:hf.Registrar.Roles
Value:client, orderer, peer, user ECert:false} {Name:hf.Registrar.DelegateRoles Value:client, orderer, peer, user ECert:false}
{Name:hf.Registrar.Attributes Value:"*" ECert:false} {Name:hf.GenCRL Value:1 ECert:false} {Name:hf.Revoker Value:1
ECert:false} {Name:hf.AffiliationMgr Value:1 ECert:false} {Name:hf.IntermediateCA Value:1 ECert:false} {Name:role
Value:admin ECert:true} {Name:hf.EnrollmentID Value:Admin@ordererorg ECert:true} {Name:hf.Type Value:client ECert:true}
{Name:hf.Affiliation Value:ordererorg ECert:true}]
Name: peer0, Type: peer, Affiliation: salesorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer0 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:salesorg
ECert:true}]
Name: peer1, Type: peer, Affiliation: salesorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer1 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:salesorg
ECert:true}]
Name: peer2, Type: peer, Affiliation: merchantorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer2 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:merchantorg
ECert:true}]
Name: peer3, Type: peer, Affiliation: merchantorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer3 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:merchantorg
ECert:true}]
Name: peer4, Type: peer, Affiliation: customerorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer4 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:customerorg
ECert:true}]
Name: peer5, Type: peer, Affiliation: customerorg, Max Enrollments: -1, Attributes: [{Name:role Value:peer ECert:true}
{Name:hf.EnrollmentID Value:peer5 ECert:true} {Name:hf.Type Value:peer ECert:true} {Name:hf.Affiliation Value:customerorg
ECert:true}]
Name: orderer0, Type: orderer, Affiliation: ordererorg, Max Enrollments: -1, Attributes: [{Name:role Value:orderer ECert:true}
{Name:hf.EnrollmentID Value:orderer0 ECert:true} {Name:hf.Type Value:orderer ECert:true} {Name:hf.Affiliation
Value:ordererorg ECert:true}]
```

※ 만약 실수하였을 경우

```
# fabric-ca-client identity remove peer1
```

5. peer 와 orderer 의 MSP 생성하기

5.1 peer0 MSP 생성 (peer0 노드에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/
# fabric-ca-client enroll -u http://peer0:peer0password@10.1.1.20:7054 -H /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer0.salesorg/
```

```
2020/03/10 03:06:33 [INFO] Created a default configuration file at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer0.salesorg/fabric-ca-client-config.yaml
2020/03/10 03:06:33 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 03:06:33 [INFO] encoded CSR
2020/03/10 03:06:33 [INFO] Stored client certificate at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/signcerts/cert.pem
2020/03/10 03:06:33 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 03:06:33 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/IssuerPublicKey
2020/03/10 03:06:33 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/cacerts
# mv 10-1-1-20:7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

5.2 peer1 MSP 생성 (peer1 노드에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/
# fabric-ca-client enroll -u http://peer1:peer1password@10.1.1.20:7054 -H /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer1.salesorg/
```

```
2020/03/10 03:10:17 [INFO] Created a default configuration file at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer1.salesorg/fabric-ca-client-config.yaml
2020/03/10 03:10:17 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 03:10:17 [INFO] encoded CSR
2020/03/10 03:10:18 [INFO] Stored client certificate at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/signcerts/cert.pem
2020/03/10 03:10:18 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 03:10:18 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/IssuerPublicKey
2020/03/10 03:10:18 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/cacerts
# mv 10-1-1-20:7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

5.3 peer2 MSP 생성 (peer2 노드에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/
# fabric-ca-client enroll -u http://peer2:peer2password@10.1.1.20:7054 -H /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/
```

```
2020/03/10 03:14:25 [INFO] Created a default configuration file at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/fabric-ca-client-config.yaml
2020/03/10 03:14:25 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 03:14:25 [INFO] encoded CSR
2020/03/10 03:14:25 [INFO] Stored client certificate at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/signcerts/cert.pem
2020/03/10 03:14:25 [INFO] Stored root CA certificate at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 03:14:25 [INFO] Stored Issuer public key at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/IssuerPublicKey
2020/03/10 03:14:25 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/cacerts
# mv 10-1-1-20:7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

5.4 peer3 MSP 생성 (peer3 노드에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/
# fabric-ca-client enroll -u http://peer3:peer3password@10.1.1.20:7054 -H /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/
```

```
2020/03/10 03:21:34 [INFO] Created a default configuration file at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/fabric-ca-client-config.yaml
2020/03/10 03:21:34 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 03:21:34 [INFO] encoded CSR
2020/03/10 03:21:34 [INFO] Stored client certificate at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/signcerts/cert.pem
2020/03/10 03:21:34 [INFO] Stored root CA certificate at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 03:21:34 [INFO] Stored Issuer public key at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/IssuerPublicKey
2020/03/10 03:21:34 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/cacerts
# mv 10-1-1-20:7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

5.5 peer4 MSP 생성 (peer4 노드에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/
# fabric-ca-client enroll -u http://peer4:peer4password@10.1.1.20:7054 -H /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer4.customerorg/
```

```
2020/03/10 03:24:49 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 03:24:49 [INFO] encoded CSR
2020/03/10 03:24:50 [INFO] Stored client certificate at /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/signcerts/cert.pem
2020/03/10 03:24:50 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 03:24:50 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/IssuerPublicKey
2020/03/10 03:24:50 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/cacerts
# mv 10-1-1-20:7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

5.6 peer5 MSP 생성 (peer5 노드에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/
# fabric-ca-client enroll -u http://peer5:peer5password@10.1.1.20:7054 -H /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer5.customerorg/
```

```
2020/03/10 03:33:11 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 03:33:11 [INFO] encoded CSR
2020/03/10 03:33:11 [INFO] Stored client certificate at /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/signcerts/cert.pem
2020/03/10 03:33:11 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 03:33:11 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/IssuerPublicKey
2020/03/10 03:33:11 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/cacerts
# mv 10-1-1-20:7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

5.7 Orderer0 MSP 생성 (Orderer0 노드에서 실행)

```
# mkdir -p /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/
# fabric-ca-client enroll -u http://orderer0:orderer0password@10.1.1.20:7054 -H /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg
```

```
2020/03/10 03:43:35 [INFO] Created a default configuration file at /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/fabric-ca-client-config.yaml
2020/03/10 03:43:35 [INFO] generating key: &{A:ecdsa S:256}
2020/03/10 03:43:35 [INFO] encoded CSR
2020/03/10 03:43:35 [INFO] Stored client certificate at /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/signcerts/cert.pem
2020/03/10 03:43:35 [INFO] Stored root CA certificate at /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/cacerts/10-1-1-20-7054.pem
2020/03/10 03:43:35 [INFO] Stored Issuer public key at /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/IssuerPublicKey
2020/03/10 03:43:35 [INFO] Stored Issuer revocation public key at /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/IssuerRevocationPublicKey
```

fabic-ca 의 ca 키와 등록한 msp 의 비밀키 이름을 변경

```
# cd /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/cacerts
# mv 10-1-1-20:7054.pem ca.crt
# cd ../keystore
# mv 비밀키 server.key
```

Orderer 노드 구동하기

1. admin-orderer 노드에서 파일 생성 실행 (configtx.yaml 파일 수정)

```
# vim /root/testnet/configtx.yaml
Organizations:
  - &OrdererOrg
     Name: OrdererOrg
     ID: OrdererOrgMSP
     MSPDir: crypto-config/ordererOrganizations/ordererorg0/msp/

  - &SalesOrg
     Name: SalesOrgMSP
     ID: SalesOrgMSP
     MSPDir: crypto-config/peerOrganizations/salesorg/msp/
     AnchorPeers:
        - Host: peer0
          Port: 7051

  - &MerchantOrg
     Name: MerchantOrgMSP
     ID: MerchantOrgMSP
     MSPDir: crypto-config/peerOrganizations/merchantorg/msp/
     AnchorPeers:
        - Host: peer2
          Port: 7051
  - &CustomerOrg
     Name: CustomerOrgMSP
     ID: CustomerOrgMSP
```

```
        MSPDir: crypto-config/peerOrganizations/customerorg/msp/
        AnchorPeers:
            - Host: peer4
              Port: 7051

Orderer: &OrdererDefaults
    OrdererType: kafka
    Addresses:
        - orderer0:7050
    BatchTimeout: 1s
    BatchSize:
        MaxMessageCount: 30
        AbsoluteMaxBytes: 99 MB
        PreferredMaxBytes: 512 KB
    Kafka:
        Brokers:
            - kafka-zookeeper:9092
    Organizations:

Application: &ApplicationDefaults
    Organizations:

Profiles:

    OrgsOrdererGenesis:
        Orderer:
            <<: *OrdererDefaults
            Organizations:
                - *OrdererOrg
        Consortiums:
            SampleConsortium:
                Organizations:
                    - *SalesOrg
                    - *MerchantOrg
                    - *CustomerOrg

    BasicChannel:
        Consortium: SampleConsortium
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *SalesOrg
                - *MerchantOrg
                - *CustomerOrg

    MerchantChannel:
        Consortium: SampleConsortium
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *SalesOrg
                - *MerchantOrg

    CustomerChannel:
        Consortium: SampleConsortium
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *SalesOrg
                - *CustomerOrg
```

1.1 조직들의 공개키를 받기 위한 디렉토리를 미리 생성(admin-orderer 노드에서 실행)

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/msp/admincerts/
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/msp/cacerts/
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/msp/admincerts/
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/msp/cacerts/
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/msp/admincerts/
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/msp/cacerts/
# mkdir -p /root/testnet/crypto-config/ordererOrganizations/ordererorg0/msp/admincerts/
```

1.2 salesorg 공개키 복사 (peer0 노드에서 실행)

```
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp/admincerts/
# scp Admin@salesorg-cert.pem root@admin-orderer:/root/testnet/crypto-
config/peerOrganizations/salesorg/msp/admincerts/Admin@salesorg-cert.pem
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/msp/cacerts
# scp ca.crt root@admin-orderer:/root/testnet/crypto-config/peerOrganizations/salesorg/msp/cacerts
```

1.3 merchantorg 공개키 복사 (peer2 노드에서 실행)

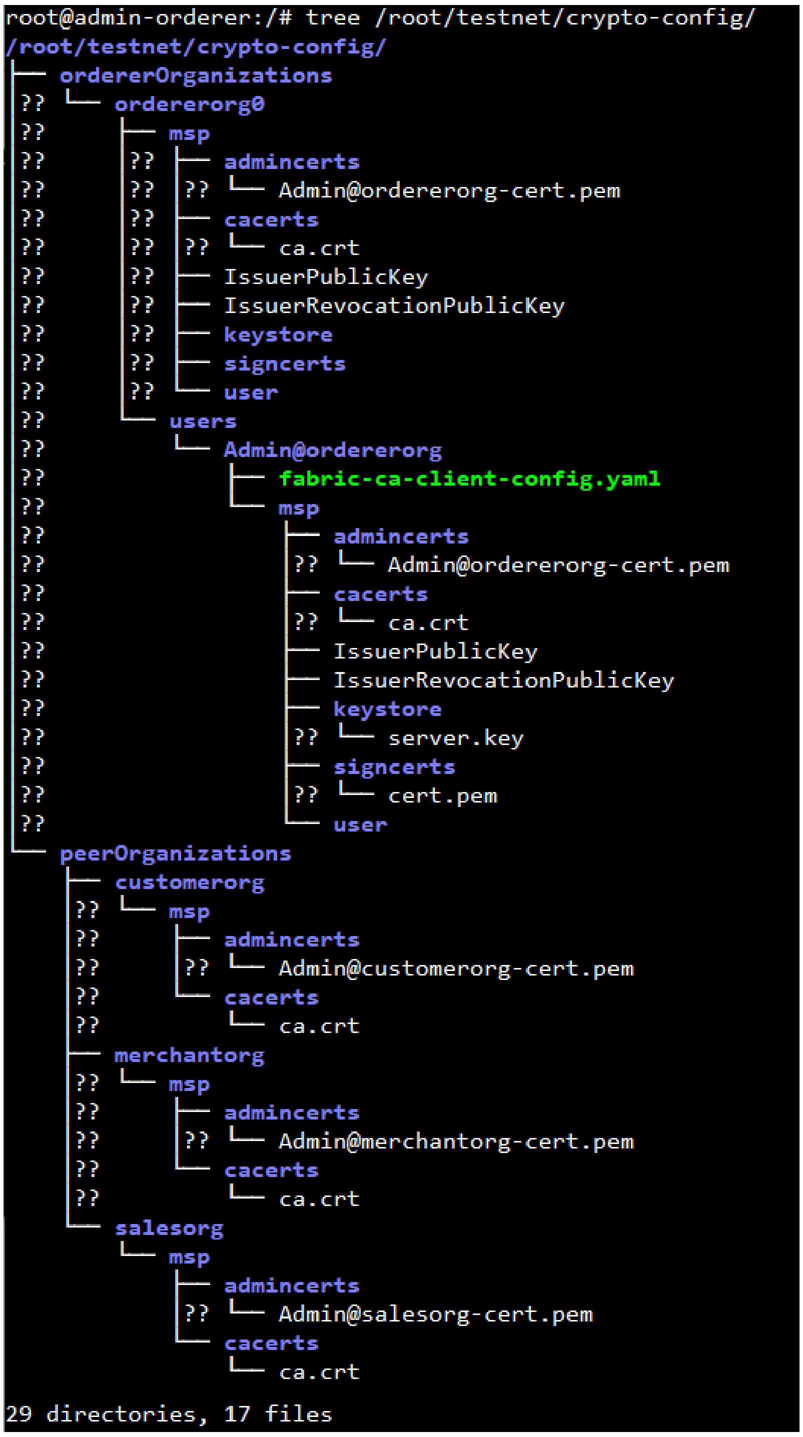
```
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/admincerts/
# scp Admin@merchantorg-cert.pem root@admin-orderer:/root/testnet/crypto-
config/peerOrganizations/merchantorg/msp/admincerts/Admin@merchantorg-cert.pem
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/msp/cacerts
# scp ca.crt root@admin-orderer:/root/testnet/crypto-config/peerOrganizations/merchantorg/msp/cacerts
```

1.4 customerorg 공개키 복사 (peer4 노드에서 실행)

```
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/users/Admin@customerorg/msp/admincerts/
# scp Admin@customerorg-cert.pem root@admin-orderer:/root/testnet/crypto-
config/peerOrganizations/customerorg/msp/admincerts/Admin@customerorg-cert.pem
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/msp/cacerts
# scp ca.crt root@admin-orderer:/root/testnet/crypto-config/peerOrganizations/customerorg/msp/cacerts
```

1.5 ordererorg 공개키 확인 (admin-orderer 노드에서 실행)

```
# cp /root/testnet/crypto-
config/ordererOrganizations/ordererorg0/users/Admin@ordererorg/msp/admincerts/Admin@ordererorg-
cert.pem   /root/testnet/crypto-config/ordererOrganizations/ordererorg0/msp/admincerts/
# tree /root/testnet/crypto-config/
```



```
root@admin-orderer:/# tree /root/testnet/crypto-config/
/root/testnet/crypto-config/
├── ordererOrganizations
│?? └── ordererorg0
│??     ├── msp
│??     │?? ├── admincerts
│??     │?? │?? └── Admin@ordererorg-cert.pem
│??     │?? ├── cacerts
│??     │?? │?? └── ca.crt
│??     │?? ├── IssuerPublicKey
│??     │?? ├── IssuerRevocationPublicKey
│??     │?? ├── keystore
│??     │?? ├── signcerts
│??     │?? └── user
│??     └── users
│??         └── Admin@ordererorg
│??             ├── fabric-ca-client-config.yaml
│??             └── msp
│??                 ├── admincerts
│??                 │?? └── Admin@ordererorg-cert.pem
│??                 ├── cacerts
│??                 │?? └── ca.crt
│??                 ├── IssuerPublicKey
│??                 ├── IssuerRevocationPublicKey
│??                 ├── keystore
│??                 │?? └── server.key
│??                 ├── signcerts
│??                 │?? └── cert.pem
│??                 └── user
└── peerOrganizations
    ├── customerorg
    │?? └── msp
    │??     ├── admincerts
    │??     │?? └── Admin@customerorg-cert.pem
    │??     └── cacerts
    │??         └── ca.crt
    ├── merchantorg
    │?? └── msp
    │??     ├── admincerts
    │??     │?? └── Admin@merchantorg-cert.pem
    │??     └── cacerts
    │??         └── ca.crt
    └── salesorg
        └── msp
            ├── admincerts
            │?? └── Admin@salesorg-cert.pem
            └── cacerts
                └── ca.crt

29 directories, 17 files
```

2. Genesis.block 생성하기(admin-orderer 노드에서 실행)

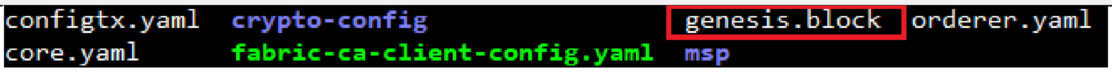
```
# configtxgen -profile ThreeOrgsOrdererGenesis -outputBlock genesis.block -channelID allchannel
```

```
2020-03-10 05:00:42.580 KST [common/tools/configtxgen] main -> WARN 001 Omitting the channel ID for configtxgen for output operations is deprecated.  Explicitly passing the channel ID will be required in the future, defaulting to 'testchainid'.
2020-03-10 05:00:42.580 KST [common/tools/configtxgen] main -> INFO 002 Loading configuration
2020-03-10 05:00:42.668 KST [common/tools/configtxgen/encoder] NewChannelGroup -> WARN 003 Default policy emission is deprecated, please include policy specifications for the channel group in configtx.yaml
2020-03-10 05:00:42.668 KST [common/tools/configtxgen/encoder] NewOrdererGroup -> WARN 004 Default policy
```

```
emission is deprecated, please include policy specifications for the orderer group in configtx.yaml
2020-03-10 05:00:42.670 KST [common/tools/configtxgen/encoder] NewOrdererOrgGroup -> WARN 005 Default policy
emission is deprecated, please include policy specifications for the orderer org group OrdererOrg in configtx.yaml
2020-03-10 05:00:42.671 KST [common/tools/configtxgen/encoder] NewOrdererOrgGroup -> WARN 006 Default policy
emission is deprecated, please include policy specifications for the orderer org group SalesOrgMSP in configtx.yaml
2020-03-10 05:00:42.671 KST [common/tools/configtxgen/encoder] NewOrdererOrgGroup -> WARN 007 Default policy
emission is deprecated, please include policy specifications for the orderer org group MerchantOrgMSP in configtx.yaml
2020-03-10 05:00:42.672 KST [common/tools/configtxgen/encoder] NewOrdererOrgGroup -> WARN 008 Default policy
emission is deprecated, please include policy specifications for the orderer org group CustomerOrgMSP in configtx.yaml
2020-03-10 05:00:42.672 KST [common/tools/configtxgen] doOutputBlock -> INFO 009 Generating genesis block
2020-03-10 05:00:42.784 KST [common/tools/configtxgen] doOutputBlock -> INFO 00a Writing genesis block
```

생성된 genesis.block 과 내용을 확인

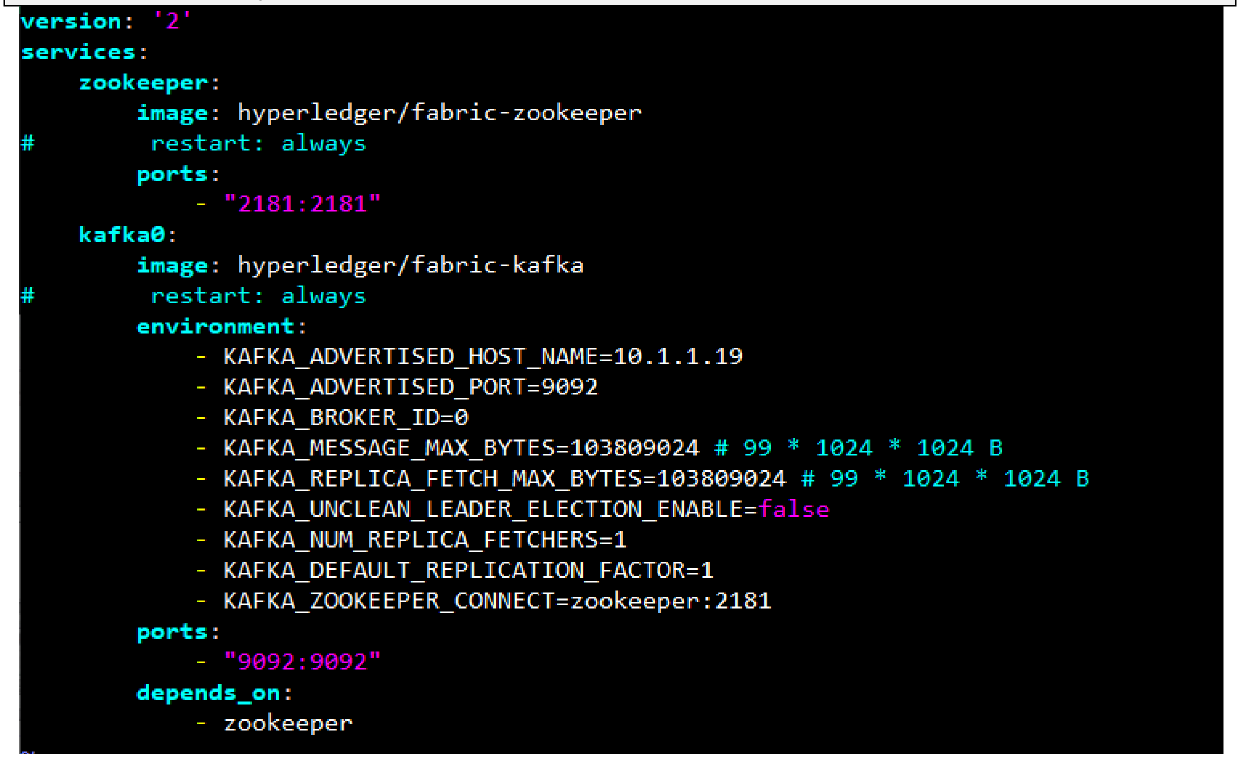
```
# ls
# configtxgen -inspectBlock genesis.block
```



```
configtx.yaml   crypto-config                 genesis.block   orderer.yaml
core.yaml       fabric-ca-client-config.yaml  msp
```

3. Kafka-zookeeper 설정 후 실행 (Kafka-zookeeper 노드에서 실행)

```
# mkdir /root/testnet
# cd testnet
# vim docker-compose.yaml
```



```
version: '2'
services:
    zookeeper:
        image: hyperledger/fabric-zookeeper
#         restart: always
        ports:
            - "2181:2181"
    kafka0:
        image: hyperledger/fabric-kafka
#         restart: always
        environment:
            - KAFKA_ADVERTISED_HOST_NAME=10.1.1.19
            - KAFKA_ADVERTISED_PORT=9092
            - KAFKA_BROKER_ID=0
            - KAFKA_MESSAGE_MAX_BYTES=103809024 # 99 * 1024 * 1024 B
            - KAFKA_REPLICA_FETCH_MAX_BYTES=103809024 # 99 * 1024 * 1024 B
            - KAFKA_UNCLEAN_LEADER_ELECTION_ENABLE=false
            - KAFKA_NUM_REPLICA_FETCHERS=1
            - KAFKA_DEFAULT_REPLICATION_FACTOR=1
            - KAFKA_ZOOKEEPER_CONNECT=zookeeper:2181
        ports:
            - "9092:9092"
        depends_on:
            - zookeeper
```

4. admin-orderer 에서 생성한 genesis.block 파일 orderer0 로 전송

```
# scp genesis.block linux@orderer0:
```

5. Orderer0 노드에서 genesis.block 파일 디렉토리로 이동 후 orderer0 노드 시작

```
# mv /home/linux/genesis.block   /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/genesis.block
# mkdir -p /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/tls
# cd /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/tls
# cp /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/cacerts/ca.crt ./
# cp /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/signcerts/cert.pem ./server.crt
# cp /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/keystore/server.key  ./
# ls /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/tls
# mkdir -p /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/admincerts
# cd /root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp/admincerts
# scp root@admin-orderer:/root/testnet/crypto-config/ordererOrganizations/ordererorg0/msp/admincerts/Admin@ordererorg-cert.pem   ./
# cd /root/testnet
# export FABRIC_CFG_PATH=$PWD
```

orderer0 을 실행할 쉘스크립트 runOrderer0.sh 을 생성

```
# vim runOrderer0.sh
export ORDERER_GENERAL_LOGLEVEL=debug
export ORDERER_GENERAL_LISTENADDRESS=orderer0
export ORDERER_GENERAL_GENESISMETHOD=file
export ORDERER_GENERAL_GENESISFILE=/root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/genesis.block
export ORDERER_GENERAL_LOCALMSPID=OrdererOrgMSP
export ORDERER_GENERAL_LOCALMSPDIR=/root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/msp
export ORDERER_GENERAL_TLS_ENABLED=false
export ORDERER_GENERAL_TLS_PRIVATEKEY=/root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/tls/server.key
export ORDERER_GENERAL_TLS_CERTIFICATE=/root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/tls/server.crt
export ORDERER_GENERAL_TLS_ROOTCAS=[/root/testnet/crypto-config/ordererOrganizations/ordererorg0/orderers/orderer0.ordererorg/tls/ca.crt,/root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/tls/ca.crt,/root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/tls/ca.crt,/root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/tls/ca.crt]
export CONFIGTX_ORDERER_BATCHTIMEOUT=1s
export CONFIGTX_ORDERER_ORDERERTYPE=kafka
export CONFIGTX_ORDERER_KAFKA_BROKERS=[kafka-zookeeper:9092]
orderer
```

runOrderer0.sh 실행

```
# chmod 777 runOrder0.sh
# ./runOrderer0.sh
```

Peer 노드 구동하기

1. Peer0 노드에서 실행

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/tls
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/tls
# cp /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/cacerts/ca.crt ./
# cp /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/keystore/server.key ./
# cp /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/signcerts/cert.pem ./server.crt
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/admincerts
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/msp/admincerts
# cp /root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp/admincerts/Admin@salesorg-
cert.pem ./
# cd /root/testnet/
# export FABRIC_CFG_PATH=$PWD
```

peer0 을 실행할 쉘스크립트 runpeer0.sh 을 생성

```
# vim runPeer0.sh
 export CORE_PEER_ENDORSER_ENABLED=true
 export CORE_PEER_PROFILE_ENABLED=true

 export CORE_PEER_ADDRESS=10.1.1.11:7051

 export CORE_PEER_CHAINCODELISTENADDRESS=10.1.1.11:7052

 export CORE_PEER_ID=salesorg-peer0

 export CORE_PEER_LOCALMSPID=SalesOrgMSP

 export CORE_PEER_GOSSIP_EXTERNALENDPOINT=10.1.1.11:7051

 export CORE_PEER_GOSSIP_USELEADERELECTION=true

 export CORE_PEER_GOSSIP_ORGLEADER=false

 export CORE_PEER_TLS_ENABLED=false

 export CORE_PEER_TLS_KEY_FILE=/root/testnet/crypto-
 config/peerOrganizations/salesorg/peers/peer0.salesorg/tls/server.key

 export CORE_PEER_TLS_CERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/salesorg/peers/peer0.salesorg/tls/server.crt

 export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/salesorg/peers/peer0.salesorg/tls/ca.crt

 export CORE_PEER_TLS_SERVERHOSTOVERRIDE=10.1.1.11

 export CORE_VM_DOCKER_ATTACHSTDOUT=true

 export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer0.salesorg/msp
peer node start
```

runPeer0.sh 실행

```
# chmod 777 runPeer0.sh
# ./runPeer0.sh
```

2. Peer1 노드에서 실행

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/tls
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/tls
# cp /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/cacerts/ca.crt ./
# cp /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/keystore/server.key ./
# cp /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/signcerts/cert.pem ./server.crt
# mkdir -p /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/admincerts
# cd /root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/msp/admincerts
# scp root@peer0:/root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg/msp/admincerts/Admin@salesorg-cert.pem ./
# cd /root/testnet/
# export FABRIC_CFG_PATH=$PWD
```

peer1 을 실행할 쉘스크립트 runpeer1.sh 을 생성

```
# vim runPeer1.sh
 export CORE_PEER_ENDORSER_ENABLED=true
 export CORE_PEER_PROFILE_ENABLED=true

 export CORE_PEER_ADDRESS=10.1.1.12:7051

 export CORE_PEER_CHAINCODELISTENADDRESS=10.1.1.12:7052

 export CORE_PEER_ID=salesorg-peer1

 export CORE_PEER_LOCALMSPID=SalesOrgMSP

 export CORE_PEER_GOSSIP_EXTERNALENDPOINT=10.1.1.12:7051

 export CORE_PEER_GOSSIP_USELEADERELECTION=true

 export CORE_PEER_GOSSIP_ORGLEADER=false

 export CORE_PEER_TLS_ENABLED=false

 export CORE_PEER_TLS_KEY_FILE=/root/testnet/crypto-
 config/peerOrganizations/salesorg/peers/peer1.salesorg/tls/server.key

 export CORE_PEER_TLS_CERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/salesorg/peers/peer1.salesorg/tls/server.crt

 export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/salesorg/peers/peer.salesorg/tls/ca.crt

 export CORE_PEER_TLS_SERVERHOSTOVERRIDE=10.1.1.12

 export CORE_VM_DOCKER_ATTACHSTDOUT=true

 export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/peers/peer1.salesorg/msp

peer node start
```

runPeer1.sh 실행

```
# chmod 777 runPeer1.sh
# ./runPeer1.sh
```

3. Peer2 노드에서 실행

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/tls
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/tls
# cp /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/cacerts/ca.crt ./
# cp /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/keystore/server.key ./
# cp /root/testnet/crypto-
config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/signcerts/cert.pem ./server.crt
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/admincerts
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp/admincerts
# cp /root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/admincerts/Admin@merchantorg-cert.pem ./
# cd /root/testnet/
# export FABRIC_CFG_PATH=$PWD
```

peer2 을 실행할 쉘스크립트 runpeer2.sh 을 생성

```
# vim runPeer2.sh
 export CORE_PEER_ENDORSER_ENABLED=true
 export CORE_PEER_PROFILE_ENABLED=true

 export CORE_PEER_ADDRESS=10.1.1.13:7051

 export CORE_PEER_CHAINCODELISTENADDRESS=10.1.1.13:7052

 export CORE_PEER_ID=merchantorg-peer2

 export CORE_PEER_LOCALMSPID=MerchantOrgMSP

 export CORE_PEER_GOSSIP_EXTERNALENDPOINT=10.1.1.13:7051

 export CORE_PEER_GOSSIP_USELEADERELECTION=true

 export CORE_PEER_GOSSIP_ORGLEADER=false

 export CORE_PEER_TLS_ENABLED=false

 export CORE_PEER_TLS_KEY_FILE=/root/testnet/crypto-
 config/peerOrganizations/merchantorg/peers/peer2.merchantorg/tls/server.key

 export CORE_PEER_TLS_CERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/merchantorg/peers/peer2.merchantorg/tls/server.crt

 export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/merchantorg/peers/peer2.merchantorg/tls/ca.crt

 export CORE_PEER_TLS_SERVERHOSTOVERRIDE=10.1.1.13

 export CORE_VM_DOCKER_ATTACHSTDOUT=true

 export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
 config/peerOrganizations/merchantorg/peers/peer2.merchantorg/msp
```

```
peer node start
```

runPeer2.sh 실행

```
# chmod 777 runPeer2.sh
# ./runPeer2.sh
```

4. Peer3 노드에서 실행

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/tls
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/tls
# cp /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/cacerts/ca.crt ./
# cp /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/keystore/server.key ./
# cp /root/testnet/crypto-
config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/signcerts/cert.pem ./server.crt
# mkdir -p /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/admincerts
# cd /root/testnet/crypto-config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp/admincerts
# scp root@peer2:/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp/admincerts/Admin@merchantorg-cert.pem ./
# cd /root/testnet/
# export FABRIC_CFG_PATH=$PWD
```

peer3 을 실행할 쉘스크립트 runpeer3.sh 을 생성

```
# vim runPeer3.sh
export CORE_PEER_ENDORSER_ENABLED=true
export CORE_PEER_PROFILE_ENABLED=true

export CORE_PEER_ADDRESS=10.1.1.14:7051

export CORE_PEER_CHAINCODELISTENADDRESS=10.1.1.14:7052

export CORE_PEER_ID=merchantorg-peer3

export CORE_PEER_LOCALMSPID=MerchantOrgMSP

export CORE_PEER_GOSSIP_EXTERNALENDPOINT=10.1.1.14:7051

export CORE_PEER_GOSSIP_USELEADERELECTION=true

export CORE_PEER_GOSSIP_ORGLEADER=false

export CORE_PEER_TLS_ENABLED=false

export CORE_PEER_TLS_KEY_FILE=/root/testnet/crypto-
config/peerOrganizations/merchantorg/peers/peer3.merchantorg/tls/server.key

export CORE_PEER_TLS_CERT_FILE=/root/testnet/crypto-
config/peerOrganizations/merchantorg/peers/peer3.merchantorg/tls/server.crt

export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
config/peerOrganizations/merchantorg/peers/peer3.merchantorg/tls/ca.crt

export CORE_PEER_TLS_SERVERHOSTOVERRIDE=10.1.1.14

export CORE_VM_DOCKER_ATTACHSTDOUT=true
```

```
 export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
 config/peerOrganizations/merchantorg/peers/peer3.merchantorg/msp

peer node start
```

runpeer3.sh 실행

```
# chmod 777 runPeer3.sh
# ./runPeer3.sh
```

5. Peer4 노드에서 실행

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/tls
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/tls
# cp /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/cacerts/ca.crt ./
# cp /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/keystore/server.key ./
# cp /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/signcerts/cert.pem ./server.crt
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/admincerts
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/msp/admincerts
# cp /root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp/admincerts/Admin@customerorg-cert.pem ./
# cd /root/testnet/
# export FABRIC_CFG_PATH=$PWD
```

peer4 을 실행할 쉘스크립트 runpeer4.sh 을 생성

```
# vim runPeer4.sh
 export CORE_PEER_ENDORSER_ENABLED=true
 export CORE_PEER_PROFILE_ENABLED=true

 export CORE_PEER_ADDRESS=10.1.1.15:7051

 export CORE_PEER_CHAINCODELISTENADDRESS=10.1.1.15:7052

 export CORE_PEER_ID=customerorg-peer4

 export CORE_PEER_LOCALMSPID=CustomerOrgMSP

 export CORE_PEER_GOSSIP_EXTERNALENDPOINT=10.1.1.15:7051

 export CORE_PEER_GOSSIP_USELEADERELECTION=true

 export CORE_PEER_GOSSIP_ORGLEADER=false

 export CORE_PEER_TLS_ENABLED=false

 export CORE_PEER_TLS_KEY_FILE=/root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer4.customerorg/tls/se

 export CORE_PEER_TLS_CERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/customerorg/peers/peer4.customerorg/tls/server.crt

 export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/customerorg/peers/peer4.customerorg/tls/ca.crt

 export CORE_PEER_TLS_SERVERHOSTOVERRIDE=10.1.1.15
 export CORE_VM_DOCKER_ATTACHSTDOUT=true

 export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
 config/peerOrganizations/customerorg/peers/peer4.customerorg/msp

peer node start
```

runpeer4.sh 실행

```
# chmod 777 runPeer4.sh
# ./runPeer4.sh
```

6. Peer5 노드에서 실행

```
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/tls
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/tls
# cp /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/cacerts/ca.crt ./
# cp /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/keystore/server.key ./
# cp /root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/signcerts/cert.pem ./server.crt
# mkdir -p /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/admincerts
# cd /root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/msp/admincerts
# scp root@peer4:/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp/admincerts/Admin@customerorg-cert.pem ./
# cd /root/testnet/
# export FABRIC_CFG_PATH=$PWD
```

peer5 을 실행할 쉘스크립트 runpeer5.sh 을 생성

```
# vim runPeer5.sh
 export CORE_PEER_ENDORSER_ENABLED=true
 export CORE_PEER_PROFILE_ENABLED=true

 export CORE_PEER_ADDRESS=10.1.1.16:7051

 export CORE_PEER_CHAINCODELISTENADDRESS=10.1.1.16:7052

 export CORE_PEER_ID=customerorg-peer5

 export CORE_PEER_LOCALMSPID=CustomerOrgMSP

 export CORE_PEER_GOSSIP_EXTERNALENDPOINT=10.1.1.16:7051

 export CORE_PEER_GOSSIP_USELEADERELECTION=true

 export CORE_PEER_GOSSIP_ORGLEADER=false

 export CORE_PEER_TLS_ENABLED=false

 export CORE_PEER_TLS_KEY_FILE=/root/testnet/crypto-config/peerOrganizations/customerorg/peers/peer5.customerorg/tls/se

 export CORE_PEER_TLS_CERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/customerorg/peers/peer5.customerorg/tls/server.crt

 export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/customerorg/peers/peer5.customerorg/tls/ca.crt
```

```
export CORE_PEER_TLS_SERVERHOSTOVERRIDE=10.1.1.16

export CORE_VM_DOCKER_ATTACHSTDOUT=true

export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/peers/peer5.customerorg/msp

peer node start
```

runpeer5.sh 실행

```
# chmod 777 runPeer5.sh
# ./runPeer5.sh
```

채널 생성

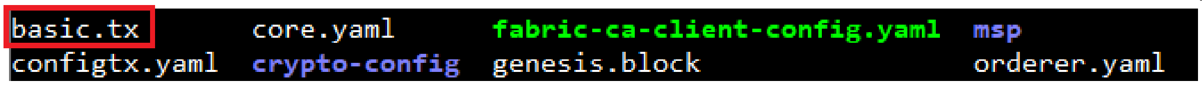
1. admin-orderer 에서 allchannel.tx 생성

```
# cd /root/testnet
# export FABRIC_CFG_PATH=$PWD
# configtxgen -profile BasicChannel -outputCreateChannelTx allchannel.tx -channelID allchannel
```

```
2020-03-10 08:17:09.148 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 08:17:09.169 KST [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 002 Generating new
channel configtx
2020-03-10 08:17:09.174 KST [common/tools/configtxgen/encoder] NewApplicationGroup -> WARN 003 Default policy
emission is deprecated, please include policy specifications for the application group in configtx.yaml
2020-03-10 08:17:09.174 KST [common/tools/configtxgen/encoder] NewApplicationOrgGroup -> WARN 004 Default policy
emission is deprecated, please include policy specifications for the application org group SalesOrgMSP in configtx.yaml
2020-03-10 08:17:09.174 KST [common/tools/configtxgen/encoder] NewApplicationOrgGroup -> WARN 005 Default policy
emission is deprecated, please include policy specifications for the application org group MerchantOrgMSP in
configtx.yaml
2020-03-10 08:17:09.174 KST [common/tools/configtxgen/encoder] NewApplicationOrgGroup -> WARN 006 Default policy
emission is deprecated, please include policy specifications for the application org group CustomerOrgMSP in
configtx.yaml
2020-03-10 08:17:09.175 KST [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 007 Writing new channel tx
```

생성된 채널 확인

```
# ls
# scp allchannel.tx root@peer0:
```



```
basic.tx       core.yaml      fabric-ca-client-config.yaml  msp
configtx.yaml  crypto-config  genesis.block                 orderer.yaml
```

2. admin-orderer 에서 merchantchannel.tx 생성

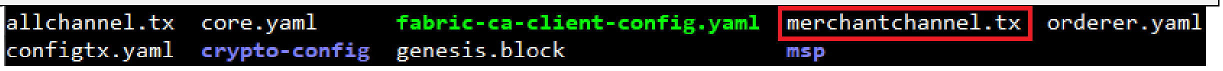
```
# cd /root/testnet
# configtxgen -profile MerchantChannel -outputCreateChannelTx merchantchannel.tx -channelID merchantchannel
```

```
2020-03-10 08:24:45.870 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 08:24:45.894 KST [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 002 Generating new
channel configtx
2020-03-10 08:24:45.913 KST [common/tools/configtxgen/encoder] NewApplicationGroup -> WARN 003 Default policy
```

```
emission is deprecated, please include policy specifications for the application group in configtx.yaml
2020-03-10 08:24:45.914 KST [common/tools/configtxgen/encoder] NewApplicationOrgGroup -> WARN 004 Default policy
emission is deprecated, please include policy specifications for the application org group SalesOrgMSP in configtx.yaml
2020-03-10 08:24:45.914 KST [common/tools/configtxgen/encoder] NewApplicationOrgGroup -> WARN 005 Default policy
emission is deprecated, please include policy specifications for the application org group MerchantOrgMSP in
configtx.yaml
2020-03-10 08:24:45.914 KST [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 006 Writing new channel tx
```

merchantchannel.tx 확인

```
# ls
# scp merchantchannel.tx root@peer0:
```



```
allchannel.tx  core.yaml      fabric-ca-client-config.yaml  merchantchannel.tx  orderer.yaml
configtx.yaml  crypto-config  genesis.block                 msp
```

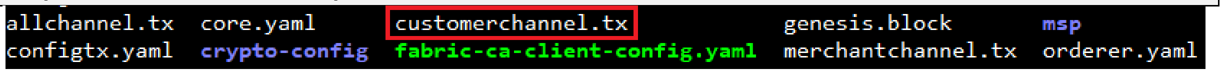
3. admin-orderer 에서 customerchannel.tx 생성

```
# cd /root/testnet
# configtxgen -profile CustomerChannel -outputCreateChannelTx customerchannel.tx -channelID customerchannel
```

```
2020-03-10 08:34:30.643 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 08:34:30.661 KST [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 002 Generating new
channel configtx
2020-03-10 08:34:30.661 KST [common/tools/configtxgen/encoder] NewApplicationGroup -> WARN 003 Default policy
emission is deprecated, please include policy specifications for the application group in configtx.yaml
2020-03-10 08:34:30.662 KST [common/tools/configtxgen/encoder] NewApplicationOrgGroup -> WARN 004 Default policy
emission is deprecated, please include policy specifications for the application org group SalesOrgMSP in configtx.yaml
2020-03-10 08:34:30.666 KST [common/tools/configtxgen/encoder] NewApplicationOrgGroup -> WARN 005 Default policy
emission is deprecated, please include policy specifications for the application org group CustomerOrgMSP in
configtx.yaml
2020-03-10 08:34:30.666 KST [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 006 Writing new channel tx
```

customerchannel.tx 확인

```
# ls
# scp customerchannel.tx root@peer0:
```



```
allchannel.tx  core.yaml      customerchannel.tx            genesis.block       msp
configtx.yaml  crypto-config  fabric-ca-client-config.yaml  merchantchannel.tx  orderer.yaml
```

채널 실행

1. peer0 노드에서 basic 채널 실행

```
# mv /root/basic.tx /root/testnet/
# cd /root/testnet
```

채널에서 채널을 실행할 쉘스크립트 create-channel.sh 을 생성

```
# vim create-channel.sh
 export CORE_PEER_LOCALMSPID="SalesOrgMSP"
 export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/salesorg/peers/peer0.salesorg/tls/ca.crt
```

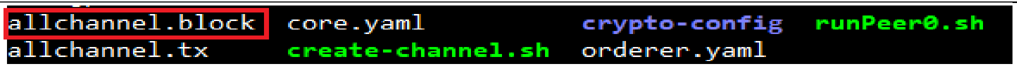
```
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg/msp

export CORE_PEER_ADDRESS=peer0:7051

peer channel create -o orderer0:7050 -c allchannel -f allchannel.tx
```

create-channel 을 실행

```
# chmod 777 create-channel.sh
# ./create-channel.sh
2020-03-10 08:18:20.873 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 08:18:21.328 KST [cli/common] readBlock -> INFO 002 Got status: &{NOT_FOUND}
2020-03-10 08:18:21.348 KST [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
2020-03-10 08:18:21.797 KST [cli/common] readBlock -> INFO 004 Got status: &{NOT_FOUND}
2020-03-10 08:18:21.802 KST [channelCmd] InitCmdFactory -> INFO 005 Endorser and orderer connections initialized
2020-03-10 08:18:22.390 KST [cli/common] readBlock -> INFO 006 Got status: &{SERVICE_UNAVAILABLE}
2020-03-10 08:18:22.423 KST [channelCmd] InitCmdFactory -> INFO 007 Endorser and orderer connections initialized
2020-03-10 08:18:22.624 KST [cli/common] readBlock -> INFO 008 Got status: &{SERVICE_UNAVAILABLE}
2020-03-10 08:18:22.626 KST [channelCmd] InitCmdFactory -> INFO 009 Endorser and orderer connections initialized
2020-03-10 08:18:22.828 KST [cli/common] readBlock -> INFO 00a Got status: &{SERVICE_UNAVAILABLE}
2020-03-10 08:18:22.829 KST [channelCmd] InitCmdFactory -> INFO 00b Endorser and orderer connections initialized
2020-03-10 08:18:23.040 KST [cli/common] readBlock -> INFO 00c Got status: &{SERVICE_UNAVAILABLE}
2020-03-10 08:18:23.042 KST [channelCmd] InitCmdFactory -> INFO 00d Endorser and orderer connections initialized
2020-03-10 08:18:23.251 KST [cli/common] readBlock -> INFO 00e Received block: 0
```

allchannel 을 실행하여 생성된 allchannel.block 을 확인

```
# ls
```



```
allchannel.block  core.yaml            crypto-config  runPeer0.sh
allchannel.tx     create-channel.sh    orderer.yaml
```

※ 채널로 연결할 admin 에게 생성한 allchannel.block 을 보낸다. allchannel.block 을 이용하는 allchannel 이름의 채널은 모든 조직에서 공유해야 하기에 각 조직의 admin 인 peer2 와 peer4 에게 allchannel.block 을 보낸다.

```
# scp allchannel.block root@peer2:
# scp allchannel.block root@peer4:
```

2. peer0 노드에서 merchant 채널 실행

```
# mv /root/basic.tx /root/testnet/
# cd /root/testnet
```

채널에서 채널을 실행할 쉘스크립트 create-channel.sh 을 생성

```
# vim create-channel.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
config/peerOrganizations/salesorg/peers/peer0.salesorg/tls/ca.crt

export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/salesorg/users/Admin@salesorg/msp

export CORE_PEER_ADDRESS=peer0:7051
```

```
peer channel create -o orderer0:7050 -c merchantchannel -f merchantchannel.tx
```

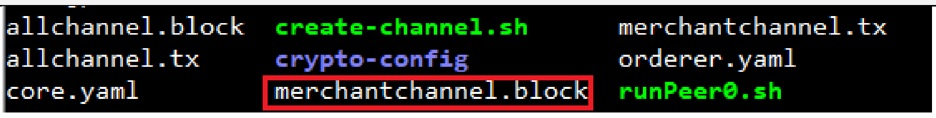
create-channel 을 실행

```
# chmod 777 create-channel.sh
# ./create-channel.sh
```

```
2020-03-10 08:27:07.243 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 08:27:07.393 KST [cli/common] readBlock -> INFO 002 Got status: &{NOT_FOUND}
2020-03-10 08:27:07.398 KST [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
2020-03-10 08:27:07.942 KST [cli/common] readBlock -> INFO 004 Got status: &{NOT_FOUND}
2020-03-10 08:27:07.946 KST [channelCmd] InitCmdFactory -> INFO 005 Endorser and orderer connections initialized
2020-03-10 08:27:08.150 KST [cli/common] readBlock -> INFO 006 Got status: &{SERVICE_UNAVAILABLE}
2020-03-10 08:27:08.152 KST [channelCmd] InitCmdFactory -> INFO 007 Endorser and orderer connections initialized
2020-03-10 08:27:08.354 KST [cli/common] readBlock -> INFO 008 Got status: &{SERVICE_UNAVAILABLE}
2020-03-10 08:27:08.356 KST [channelCmd] InitCmdFactory -> INFO 009 Endorser and orderer connections initialized
2020-03-10 08:27:08.573 KST [cli/common] readBlock -> INFO 00a Received block: 0
```

merchantchannel 을 실행하여 생성된 merchantchannel.block 을 확인

```
# ls
```



```
allchannel.block   create-channel.sh      merchantchannel.tx
allchannel.tx      crypto-config          orderer.yaml
core.yaml          merchantchannel.block  runPeer0.sh
```

※ 채널로 연결할 admin 에게 생성한 merchantchannel.block 을 보낸다. merchantchannel.block 을 이용하는 merchantchannel 이름의 채널은 SalesOrg 조직과 MerchantOrg 조직이 공유해야 하기에 MerchatOrg 조직의 admin 인 peer2 에게 merchantchannel.block 을 보낸다.

```
# scp merchantchannel.block root@peer2:
```

3. peer0 노드에서 customer 채널 실행

```
# mv /root/basic.tx /root/testnet/
# cd /root/testnet
```

채널에서 채널을 실행할 쉘스크립트 create-channel.sh 을 생성

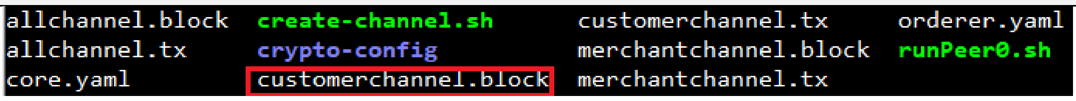
```
# vim create-channel.sh
 export CORE_PEER_LOCALMSPID="SalesOrgMSP"
 export CORE_PEER_TLS_ROOTCERT_FILE=/root/testnet/crypto-
 config/peerOrganizations/salesorg/peers/peer0.salesorg/tls/ca.crt

 export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
 config/peerOrganizations/salesorg/users/Admin@salesorg/msp

 export CORE_PEER_ADDRESS=peer0:7051
peer channel create -o orderer0:7050 -c customerchannel -f customerchannel.tx
# chmod 777 create-channel.sh
# ./create-channel.sh
```

```
2020-03-10 08:39:32.903 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 08:39:33.173 KST [cli/common] readBlock -> INFO 002 Got status: &{NOT_FOUND}
2020-03-10 08:39:33.179 KST [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
```

```
2020-03-10 08:39:33.766 KST [cli/common] readBlock -> INFO 004 Got status: &{SERVICE_UNAVAILABLE}
2020-03-10 08:39:33.768 KST [channelCmd] InitCmdFactory -> INFO 005 Endorser and orderer connections initialized
2020-03-10 08:39:33.969 KST [cli/common] readBlock -> INFO 006 Got status: &{SERVICE_UNAVAILABLE}
2020-03-10 08:39:33.970 KST [channelCmd] InitCmdFactory -> INFO 007 Endorser and orderer connections initialized
2020-03-10 08:39:34.293 KST [cli/common] readBlock -> INFO 008 Received block: 0
```

customerchannel 을 실행하여 생성된 customerchannel.block 을 확인

```
# ls
```



```
allchannel.block   create-channel.sh     customerchannel.tx     orderer.yaml
allchannel.tx      crypto-config         merchantchannel.block  runPeer0.sh
core.yaml          customerchannel.block merchantchannel.tx
```

※ 채널로 연결할 admin 에게 생성한 customerchannel.block 을 보낸다. customerchannel.block 을 이용하는 customerchannel 이름의 채널은 SalesOrg 조직과 CustomercOrg 조직이 공유해야 하기에 CustomerOrg 조직의 admin 인 peer4 에게 customerchannel.block 을 보낸다.

```
# scp customerchannel.block root@peer4:
```

Peer 의 채널 참여

1. peer0 에서 실행(SalesOrg 의 admin)

1.1 allchannel 에 가입

```
# cd /root/testnet
```

allchannel 에 peer0 을 가입하기 위한 peer0-join.sh 쉘스크립트 생성

```
# vim peer0-join.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer channel join -b allchannel.block
```

peer0-join.sh 쉘스크립트 실행

```
# chmod 777
# ./peer0-join.sh
```

```
2020-03-10 09:19:35.483 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:19:35.634 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

allchannel 에 peer1 을 가입하기 위한 peer1-join.sh 쉘스크립트 생성

```
# vim peer1-join.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer1:7051
peer channel join -b allchannel.block
```

peer1-join.sh 쉘스크립트 실행

```
# chmod 777 peer1-join.sh
# ./peer1-join.sh
```

```
2020-03-10 09:20:03.538 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:20:03.627 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

1.2 merchantchannel 에 가입

```
# cd /root/testnet
```

merchantchannel 에 peer0 을 가입하기 위한 peer0-join.sh 쉘스크립트 생성

```
# vim peer0-join.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer channel join -b merchantchannel.block
```

peer0-join.sh 쉘스크립트 실행

```
# chmod 777
# ./peer0-join.sh
2020-03-10 09:21:43.582 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:21:43.623 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

merchantchannel 에 peer1 을 가입하기 위한 peer1-join.sh 쉘스크립트 생성

```
# vim peer1-join.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer1:7051
peer channel join -b merchantchannel.block
```

peer1-join.sh 쉘스크립트 실행

```
# chmod 777 peer1-join.sh
# ./peer1-join.sh
2020-03-10 09:22:23.327 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:22:23.392 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

1.3 customerchannel 에 가입

customerchannel 에 peer0 을 가입하기 위한 peer0-join.sh 쉘스크립트 생성

```
# vim peer0-join.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer channel join -b customerchannel.block
```

peer0-join.sh 쉘스크립트 실행

```
# chmod 777
# ./peer0-join.sh
2020-03-10 09:24:13.119 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:24:13.278 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

customerchannel 에 peer1 을 가입하기 위한 peer1-join.sh 쉘스크립트 생성

```
# vim peer1-join.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer1:7051
peer channel join -b customerchannel.block
```

peer1-join.sh 쉘스크립트 실행

```
# chmod 777 peer1-join.sh
# ./peer1-join.sh
```

```
2020-03-10 09:24:38.459 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:24:38.610 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

2. peer2 에서 실행(MerchantOrg 의 admin)

2.1 allchannel 에 가입

```
# cd /root/testnet
```

allchannel 에 peer2 을 가입하기 위한 peer2-join.sh 쉘스크립트 생성

```
# vim peer2-join.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer2:7051
peer channel join -b allchannel.block
```

peer2-join.sh 쉘스크립트 실행

```
# chmod 777 peer2-join.sh
# ./peer2-join.sh
```

```
2020-03-10 09:28:49.422 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:28:49.577 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

allchannel 에 peer3 을 가입하기 위한 peer3-join.sh 쉘스크립트 생성

```
# vim peer3-join.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer3:7051
peer channel join -b allchannel.block
```

peer3-join.sh 쉘스크립트 실행

```
# chmod 777 peer3-join.sh
# ./peer3-join.sh
```

```
2020-03-10 09:29:07.942 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:29:08.119 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

2.2 merchantchannel 에 가입

merchantchannel 에 peer2 을 가입하기 위한 peer2-join.sh 쉘스크립트 생성

```
# vim peer2-join.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer2:7051
peer channel join -b merchantchannel.block
```

peer2-join.sh 쉘스크립트 실행

```
# chmod 777 peer2-join.sh
# ./peer2-join.sh
```

```
2020-03-10 09:30:58.808 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:30:58.837 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

merchantchannel 에 peer3 을 가입하기 위한 peer3-join.sh 쉘스크립트 생성

```
# vim peer3-join.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer3:7051
peer channel join -b merchantchannel.block
```

peer3-join.sh 쉘스크립트 실행

```
# chmod 777 peer3-join.sh
# ./peer3-join.sh
```

```
2020-03-10 09:31:20.449 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:31:20.506 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

3. peer4 에서 실행(CustomerOrg 의 admin)

3.1 allchannel 에 가입

```
# cd /root/testnet
```

allchannel 에 peer4 을 가입하기 위한 peer4-join.sh 쉘스크립트 생성

```
# vim peer4-join.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer4:7051
peer channel join -b allchannel.block
# chmod 777 peer4-join.sh
```

peer4-join.sh 쉘스크립트 실행

```
# chmod 777 peer4-join.sh
# ./peer4-join.sh
2020-03-10 09:35:34.698 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:35:34.769 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

allchannel 에 peer5 을 가입하기 위한 peer5-join.sh 쉘스크립트 생성

```
# vim peer5-join.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer5:7051
peer channel join -b allchannel.block
```

peer5-join.sh 쉘스크립트 실행

```
# chmod 777 peer5-join.sh
# ./peer5-join.sh
2020-03-10 09:35:55.146 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:35:55.272 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

3.3 customerchannel 에 가입

customerchannel 에 peer4 을 가입하기 위한 peer4-join.sh 쉘스크립트 생성

```
# vim peer4-join.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer4:7051
peer channel join -b customerchannel.block
```

peer4-join.sh 쉘스크립트 실행

```
# chmod 777 peer4-join.sh
# ./peer4-join.sh
2020-03-10 09:37:38.681 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:37:38.711 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

customerchannel 에 peer5 을 가입하기 위한 peer5-join.sh 쉘스크립트 생성

```
# vim peer5-join.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer5:7051
peer channel join -b customerchannel.block
```

peer5-join.sh 쉘스크립트 실행

```
# chmod 777 peer5-join.sh
# ./peer5-join.sh
2020-03-10 09:37:54.696 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 09:37:54.716 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
```

Anchor peer 업데이트

1. allchannel 채널에 Anchor peer 등록

1.1 admin-orderer 노드에서 SalesOrgMSP 의 anchor peer 등록

```
# cd /root/testnet
# configtxgen -profile BasicChannel -outputAnchorPeersUpdate SalesOrgMSPanchors.tx -channelID allchannel -asOrg SalesOrgMSP
```

```
2020-03-10 09:56:50.010 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 09:56:50.032 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor peer update
2020-03-10 09:56:50.046 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer update
```

admin-orderer 노드에서 MerchantOrgMSP 의 anchor peer 등록

```
# configtxgen -profile BasicChannel -outputAnchorPeersUpdate MerchantOrgMSPanchors.tx -channelID allchannel -asOrg MerchantOrgMSP
```

```
2020-03-10 09:57:16.953 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 09:57:16.982 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor peer update
2020-03-10 09:57:16.982 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer update
```

admin-orderer 노드에서 CustomerOrgMSP 의 anchor peer 등록

```
# configtxgen -profile BasicChannel -outputAnchorPeersUpdate CustomerOrgMSPanchors.tx -channelID allchannel -asOrg CustomerOrgMSP
```

```
2020-03-10 09:58:18.803 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 09:58:18.828 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor peer update
2020-03-10 09:58:18.840 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer update
```

anchor peer 가 만든 트랜잭션을 각각의 admin 에 전송

```
# scp SalesOrgMSPanchors.tx root@peer0:
# scp MerchantOrgMSPanchors.tx root@peer2:
# scp CustomerOrgMSPanchors.tx root@peer4:
```

1.2 peer0 노드에서 anchor peer 의 트랜잭션 실행

```
# cd /root/testnet
```

SaleOrg 의 anchor 을 등록하기 위한 SalesOrg-anchor.sh 의 쉘스크립트 생성

```
# vim SalesOrg-anchor.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer channel create -o orderer0:7050 -c allchannel -f SalesOrgMSPanchors.tx
```

SalesOrg-anchor.sh 실행

```
# chmod 777 SalesOrg-anchor.sh
# ./SalesOrg-anchor.sh
2020-03-10 11:42:05.022 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 11:42:05.117 KST [cli/common] readBlock -> INFO 002 Received block: 0
```

1.3 peer2 노드에서 실행

```
# cd /root/testnet
```

MerchantOrg 의 anchor 을 등록하기 위한 MerchantOrg-anchor.sh 의 쉘스크립트 생성

```
# vim MerchantOrg-anchor.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer2:7051
peer channel create -o orderer0:7050 -c allchannel -f MerchantOrgMSPanchors.tx
```

MerchantOrg-anchor.sh 실행

```
# chmod 777 MerchantOrg-anchor.sh
# ./MerchantOrg-anchor.sh
2020-03-10 11:48:17.459 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 11:48:17.758 KST [cli/common] readBlock -> INFO 002 Received block: 0
```

1.4 peer4 노드에서 실행

```
# cd /root/testnet
```

CustomerOrg 의 anchor 을 등록하기 위한 CustomerOrg-anchor.sh 의 쉘스크립트 생성

```
# vim CustomerOrg-anchor.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer4:7051
peer channel create -o orderer0:7050 -c allchannel -f ConsumerOrgMSPanchors.tx
```

CustomerOrg-anchor.sh 실행

```
# chmod 777 ConsumerOrg-anchor.sh
# ./ConsumerOrg-anchor.sh
2020-03-10 11:56:40.900 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 11:56:41.001 KST [cli/common] readBlock -> INFO 002 Received block: 0
```

2. merchantchannel 채널에 Anchor peer 등록

2.1 admin-orderer 노드에서 SalesOrgMSP 의 anchor peer 등록

```
# configtxgen -profile MerchantChannel -outputAnchorPeersUpdate SalesMerchantOrgMSPanchors.tx -channelID
merchantchannel -asOrg SalesOrgMSP
```

```
2020-03-10 13:23:04.594 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 13:23:04.622 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor
peer update
2020-03-10 13:23:04.623 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer
update
```

2.2 admin-orderer 노드에서 MerchantOrgMSP 의 anchor peer 등록

```
# configtxgen -profile MerchantChannel -outputAnchorPeersUpdate MerchantSalesOrgMSPanchors.tx -channelID
merchantchannel -asOrg MerchantOrgMSP
2020-03-10 13:23:45.473 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 13:23:45.484 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor
peer update
2020-03-10 13:23:45.485 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer
update
```

anchor peer 가 만든 트랜잭션을 각각의 admin 에 전송

```
# scp SalesMerchantOrgMSPanchors.tx linux@peer0:
# scp MerchantSalesOrgMSPanchors.tx linux@peer2:
```

2.3 peer0 노드에서 실행

```
# cd /root/testnet
```

SaleOrg 의 anchor 을 등록하기 위한 SalesOrg-anchor.sh 의 쉘스크립트 생성

```
# vim SalesOrg-anchor.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer channel create -o orderer0:7050 -c allchannel -f SalesMerchantOrgMSPanchors.tx
```

SalesOrg-anchor.sh 실행

```
# chmod 777 SalesOrg-anchor.sh
# ./SalesOrg-anchor.sh
2020-03-10 13:38:26.533 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 13:38:26.699 KST [cli/common] readBlock -> INFO 002 Received block: 0
```

2.4 peer2 노드에서 실행

```
# cd /root/testnet
```

MerchantOrg 의 anchor 을 등록하기 위한 MerchantOrg-anchor.sh 의 쉘스크립트 생성

```
# vim MerchantOrg-anchor.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer2:7051
peer channel create -o orderer0:7050 -c merchantchannel -f MerchantSalesOrgMSPanchors.tx
```

MerchantOrg-anchor.sh 실행

```
# chmod 777 MerchantOrg-anchor.sh
# ./MerchantOrg-anchor.sh
2020-03-10 13:41:28.218 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 13:41:28.341 KST [cli/common] readBlock -> INFO 002 Received block: 0
```

3. customerchannel 채널에 Anchor peer 등록

3.1 admin-orderer 노드에서 SalesOrgMSP 의 anchor peer 등록

```
# configtxgen -profile CustomerChannel -outputAnchorPeersUpdate SalesCustomerOrgMSPanchors.tx -channelID
customerchannel -asOrg SalesOrgMSP
2020-03-10 13:44:58.448 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 13:44:58.475 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor
peer update
2020-03-10 13:44:58.476 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer
update
```

3.2 admin-orderer 노드에서 CustomerOrgMSP 의 anchor peer 등록

```
# configtxgen -profile CustomerChannel -outputAnchorPeersUpdate CustomerSalesOrgMSPanchors.tx -channelID
customerchannel -asOrg CustomerOrgMSP
2020-03-10 13:46:26.720 KST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2020-03-10 13:46:26.741 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor
peer update
2020-03-10 13:46:26.741 KST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer
update
```

anchor peer 가 만든 트랜잭션을 각각의 admin 에 전송

```
# scp SalesCustomerOrgMSPanchors.tx root@peer0:
# scp CustomerSalesOrgMSPanchors.tx root@peer4:
```

3.3 peer0 노드에서 실행

```
# cd /root/testnet
```

SaleOrg 의 anchor 을 등록하기 위한 SalesOrg-anchor.sh 의 쉘스크립트 생성

```
# vim SalesOrg-anchor.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer channel create -o orderer0:7050 -c customerchannel-f SalesCustomerOrgMSPanchors.tx
```

SalesOrg-anchor.sh 실행

```
# chmod 777 SalesOrg-anchor.sh
# ./SalesOrg-anchor.sh
2020-03-10 13:51:15.245 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 13:51:15.476 KST [cli/common] readBlock -> INFO 002 Received block: 0
```

3.4 peer4 노드에서 실행

```
# cd /root/testnet
```

CustomerOrg 의 anchor 을 등록하기 위한 CustomerOrg-anchor.sh 의 쉘스크립트 생성

```
# vim CustomerOrg-anchor.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer4:7051
peer channel create -o orderer0:7050 -c customerchannel -f CustomerSalesOrgMSPanchors.tx
```

CustomerOrg-anchor.sh 실행

```
# chmod 777 ConsumerOrg-anchor.sh
# ./ConsumerOrg-anchor.sh
```

```
2020-03-10 13:52:55.184 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-03-10 13:52:55.325 KST [cli/common] readBlock -> INFO 002 Received block: 0
```

Chaincode 설치

peer0 에서 peer0-1 그리고 peer2 에서 peer2-3 그리고 peer4 에서 peer4-5 에 각 각 Chaincode 설치하기

1. peer0 에서 peer0 와 peer1 의 체인코드 설치

peer0 에서 peer0 의 체인코드 설치할 installCCpeer0.sh 쉘스크립트 파일 생성

```
# vim installCCpeer0.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer chaincode install -n allchannelCC -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd
```

installCCpeer0 실행

```
# chmod 777 installCCpeer0.sh
# ./installCCpeer0.sh
```

```
2020-03-10 15:23:21.261 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-03-10 15:23:21.261 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-03-10 15:23:29.819 KST [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
```

peer0 에서 peer1 의 체인코드 설치할 installCCpeer1.sh 쉘스크립트 파일 생성

```
# vim installCCpeer1.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer1:7051
peer chaincode install -n allchannelCC -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd
```

installCCpeer0 실행

```
# chmod 777 installCCpeer1.sh
# ./installCCpeer1.sh
```

```
2020-03-10 15:25:39.374 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-03-10 15:25:39.375 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-03-10 15:25:43.788 KST [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
```

peer0 에서 체인코드가 잘 설치되었는지 확인하는 installedCClist.sh 쉘스크립트 생성

```
# vim installedCClist.sh                                    //chaincode 설치 확인하기
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer chaincode list -C allchannel --installed
```

installedCClist.sh 실행

```
# chmod 777 installedCClist.sh
# ./installedCClist.sh
```

```
Name: allchannelCC, Version: 1.0, Path: github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd, Id:
84fd7365a62c52c90c9fc9c6f99068ffe63ebb4a2f80afb895b10c7983f3a7fd
```

2. peer2 에 체인코드 설치

peer2 에서 peer2 의 체인코드 설치할 installCCpeer2.sh 쉘스크립트 파일 생성

```
# vim installCCpeer2.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer2:7051
peer chaincode install -n allchannelCC -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd
```

installCCpeer2.sh 실행

```
# chmod 777 installCCpeer2.sh
# ./installCCpeer2.sh
```

```
2020-03-10 15:43:53.428 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-03-10 15:43:53.428 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-03-10 15:44:04.587 KST [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
```

peer2 에서 peer3 의 체인코드 설치할 installCCpeer3.sh 쉘스크립트 파일 생성

```
# vim installCCpeer3.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer3:7051
peer chaincode install -n allchannelCC -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd
```

installCCpeer3.sh 실행

```
# chmod 777 installCCpeer3.sh
# ./installCCpeer3.sh
```

```
2020-03-10 15:45:17.215 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-03-10 15:45:17.216 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-03-10 15:45:21.591 KST [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
```

peer2 에서 체인코드가 잘 설치되었는지 확인하는 installedCClist.sh 쉘스크립트 생성

```
# vim installedCClist.sh                                  //chaincode 설치 확인하기
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer2:7051
peer chaincode list -C allchannel --installed
```

installedCClist.sh 실행

```
# chmod 777 installedCClist.sh
# ./installedCClist.sh
```

```
Get installed chaincodes on peer:
Name: allchannelCC, Version: 1.0, Path: github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd, Id:
84fd7365a62c52c90c9fc9c6f99068ffe63ebb4a2f80afb895b10c7983f3a7fd
```

3. peer4 에 체인코드 설치

peer4 에서 peer4 의 체인코드 설치할 installCCpeer4.sh 쉘스크립트 파일 생성

```
# vim installCCpeer4.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer4:7051
peer chaincode install -n allchannelCC -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd
```

installCCpeer4.sh 실행

```
# chmod 777 installCCpeer4.sh
# ./installCCpeer4.sh
```

```
2020-03-10 15:49:47.469 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-03-10 15:49:47.475 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-03-10 15:49:56.040 KST [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
```

peer5 에서 peer5 의 체인코드 설치할 installCCpeer5.sh 쉘스크립트 파일 생성

```
# vim installCCpeer5.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer5:7051
peer chaincode install -n allchannelCC -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd
```

installCCpeer5.sh 실행

```
# chmod 777 installCCpeer5.sh
# ./installCCpeer5.sh
```

```
2020-03-10 15:51:04.311 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-03-10 15:51:04.344 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-03-10 15:51:08.018 KST [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
```

peer4 에서 체인코드가 잘 설치되었는지 확인하는 installedCClist.sh 쉘스크립트 생성

```
# vim installedCClist.sh                                        //chaincode 설치 확인하기
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer4:7051
peer chaincode list -C allchannel --installed
```

installedCClist.sh 실행

```
# chmod 777 installedCClist.sh
# ./installedCClist.sh
```

```
Get installed chaincodes on peer:
Name: allchannelCC, Version: 1.0, Path: github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd, Id:
84fd7365a62c52c90c9fc9c6f99068ffe63ebb4a2f80afb895b10c7983f3a7fd
```

Chaincode 인스턴스 생성

각 조직의 운영자 노드 중 한 곳에서만 체인코드 인스턴스를 실행한다.

1. peer0 에서 실행 인스턴스 생성을 위한 instantiateCC.sh 쉘스크립트 생성

```
# vim instantiateCC.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer chaincode instantiate -o orderer0:7050 -C allchannel -n allchannelCC -v 1.0 -c '{"Args":["init","a", "1000", "b","2000"]}'
P "OR ('SalesOrgMSP.member','MerchantOrgMSP.member','CustomerOrgMSP.member')"
```

instantiateCC.sh 실행

```
# chmod 777 instantiateCC.sh
# ./instantiateCC.sh
```

```
2020-03-10 16:05:09.456 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-03-10 16:05:09.476 KST [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
```

2. peer0 에서 생성된 인스턴스 확인을 위한 instantiatedCClist.sh 쉘스크립트 생성

```
# vim instantiatedCClist.sh                              //chaincode 인스턴스 생성 확인
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer chaincode list -C allchannel --instantiated
```

instantiatedCClist.sh 실행

```
# chmod 777 instantiatedCClist.sh
# ./instantiatedCClist.sh
```

```
Get instantiated chaincodes on channel allchannel:
Name: allchannelCC, Version: 1.0, Path: github.com/hyperledger/fabric/examples/chaincode/go/example02/cmd, Escc: escc,
Vscc: vscc
```

분산원장의 데이터 읽기

1. peer0 노드에서 체인코드 실행을 위한 query.sh 쉘스크립트 생성

```
# vim query.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer chaincode query -C allchannel -n allchannelCC -c '{"Args":["query","b"]}'
```

query.sh 실행

```
# chmod 777 query.sh
# ./query.sh
```

```
2000
```

2. peer2 노드에서 체인코드 실행을 위한 query.sh 쉘스크립트 생성

```
# vim query.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer2:7051
peer chaincode query -C allchannel -n allchannelCC -c '{"Args":["query","a"]}'
```

query.sh 실행

```
# chmod 777 query.sh
# ./query.sh
```

```
1000
```

3. peer4 노드에서 체인코드 실행 위한 query.sh 쉘스크립트 생성

```
# vim query.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer4:7051
peer chaincode query -C allchannel -n allchannelCC -c '{"Args":["query","b"]}'
```

query.sh 실행

```
# chmod 777 query.sh
# ./query.sh
```

```
2000
```

분산원장에 데이터 기록하기

1. peer0 노드에서 데이터 기록를 위한 invoke.sh 쉘스크립트 생성

```
# vim invoke.sh
export CORE_PEER_LOCALMSPID="SalesOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-config/peerOrganizations/salesorg/users/Admin@salesorg/msp
export CORE_PEER_ADDRESS=peer0:7051
peer chaincode invoke -o orderer0:7050 -C allchannel -n allchannelCC -c '{"Args":["invoke","a","b","50"]}'
```

invoke.sh 실행

```
# chmod 777 invoke.sh
# ./invoke.sh
2020-03-10 16:31:27.676 KST [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
```

query.sh 실행

```
# ./query.sh
2050
```

2. peer2 노드에서 데이터 기록을 위한 invoke.sh 쉘스크립트 생성

```
# vim invoke.sh
export CORE_PEER_LOCALMSPID="MerchantOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/merchantorg/users/Admin@merchantorg/msp
export CORE_PEER_ADDRESS=peer2:7051
peer chaincode invoke -o orderer0:7050 -C allchannel -n allchannelCC -c '{"Args":["invoke","a","b","150"]}'
```

invoke.sh 실행

```
# chmod 777 invoke.sh
# ./invoke.sh
2020-03-10 16:34:39.579 KST [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
```

query.sh 실행

```
# ./query.sh
800
```

3. peer4 노드에서 데이터 기록 위한 invoke.sh 쉘스크립트 생성

```
# vim invoke.sh
export CORE_PEER_LOCALMSPID="CustomerOrgMSP"
export CORE_PEER_MSPCONFIGPATH=/root/testnet/crypto-
config/peerOrganizations/customerorg/users/Admin@customerorg/msp
export CORE_PEER_ADDRESS=peer4:7051
peer chaincode invoke -o orderer0:7050 -C allchannel -n allchannelCC -c '{"Args":["invoke","b","a","2000"]}'
```

invoke.sh 실행

```
# chmod 777 invoke.sh
# ./invoke.sh
```

```
2020-03-10 16:37:14.438 KST [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result:
status:200
```

query.sh 실행

```
# ./query.sh
```

```
2000
```