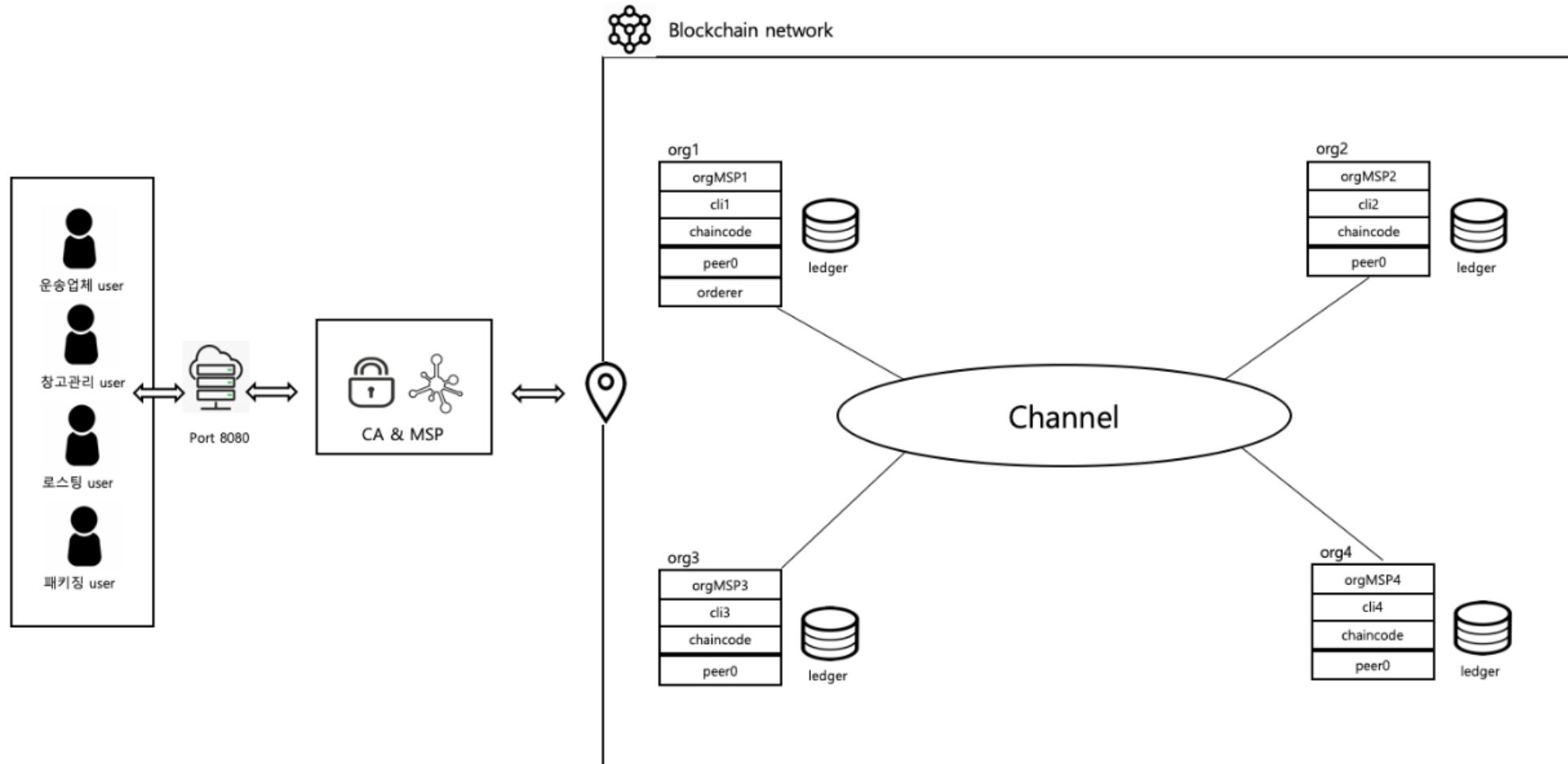
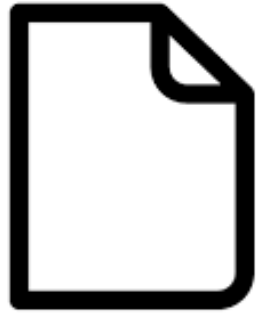


## 네트워크 구성도

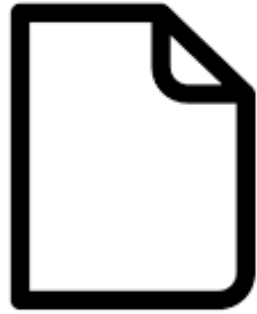
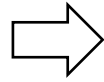


오픈소스 : <https://github.com/saarc>

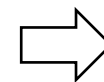
## 네트워크 구축 프로세스



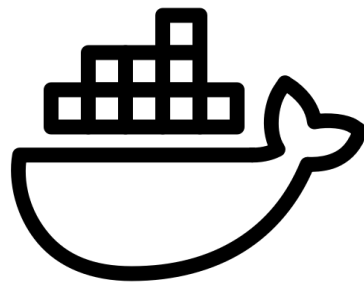
generaate.sh



Start.sh



cc,sh



docker-compose.yml

### 설명

generate.sh 실행을 통해 각 기관의 MSP 키와 채널과 제네시스 블록을 생성. 생성된 MSP 키 값을 docker-compose.yml의 기관별 keyFile 부분에 하드코딩 이후, start.sh 실행. Start.sh 실행을 통해 각 기관이 컨테이너를 통해 실행되고 채널에 가입 됨 (네트워크 합류). 그 이후 cc.sh 를 실행을 통해 채널에 체인코드를 배포하여 각 기관들에 체인코드가 배포됨.

## 네트워크 구축 프로세스 - 셸 스크립트

### 1 generate.sh

각 기관의 MSP Key, 채널, 제네시스 블록 생성

```
[(base) hongui-MacBook-Air:config hongjoonkim$ ls  
Org1MSPanchors.tx      Org2MSPanchors.tx      Org3MSPanchors.tx      Org4MSPanchors.tx      channel.tx      genesis.block
```

각 기관의 MSP key값

```
[(base) hongui-MacBook-Air:network hongjoonkim$ ls crypto-config/peerOrganizations/org1.example.com/ca/ | grep _sk  
46a67ada4d31a72ebeabdd27fdd732e3cd684537b5b22050a4897a6e2005ce8f_sk  
[(base) hongui-MacBook-Air:network hongjoonkim$ ls crypto-config/peerOrganizations/org2.example.com/ca/ | grep _sk  
8f9e35c3c3289b6bab87d1b0e622c005ee47a71663b4f35e50131a5ecddbbee31_sk  
[(base) hongui-MacBook-Air:network hongjoonkim$ ls crypto-config/peerOrganizations/org3.example.com/ca/ | grep _sk  
2ff3f56b047a26f105ec866a12c9b8e17283f34c0f8d5c68047f3128b59fdf57_sk  
[(base) hongui-MacBook-Air:network hongjoonkim$ ls crypto-config/peerOrganizations/org4.example.com/ca/ | grep _sk  
800c9b1dbaeb67ef2aeacae87d5f56e09b8cca24982aa4df9b1aeb1292a63428_sk
```

2 start.sh

Docker composer service 실행

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0789e10cafc3	hyperledger/fabric-tools	"/bin/bash"	4 seconds ago	Up Less than a second		cli3
a08335a83020	hyperledger/fabric-tools	"/bin/bash"	4 seconds ago	Up Less than a second		cli
6797bff12774	hyperledger/fabric-tools	"/bin/bash"	4 seconds ago	Up Less than a second		cli2
5ad4680eb0a2	hyperledger/fabric-tools	"/bin/bash"	6 seconds ago	Up 1 second		cli4
846dd5639ba1	hyperledger/fabric-peer	"peer node start"	9 seconds ago	Up 3 seconds	0.0.0.0:9051->7051/tcp, 0.0.0.0:9053->7053/tcp	peer0.org3.
example.com						
77bd7ea54a2e	hyperledger/fabric-peer	"peer node start"	9 seconds ago	Up 4 seconds	0.0.0.0:7051->7051/tcp, 0.0.0.0:7053->7053/tcp	peer0.org1.
example.com						
99d0daa58e0f	hyperledger/fabric-peer	"peer node start"	10 seconds ago	Up 5 seconds	0.0.0.0:6051->7051/tcp, 0.0.0.0:6053->7053/tcp	peer0.org4.
example.com						
fd595403dd39	hyperledger/fabric-peer	"peer node start"	10 seconds ago	Up 4 seconds	0.0.0.0:8051->7051/tcp, 0.0.0.0:8053->7053/tcp	peer0.org2.
example.com						
8659a630ad71	hyperledger/fabric-orderer	"orderer"	16 seconds ago	Up 9 seconds	0.0.0.0:7050->7050/tcp	orderer.exa
mple.com						
10cbb907ca53	hyperledger/fabric-couchdb	"tini -- /docker-ent..."	16 seconds ago	Up 9 seconds	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp	couchdb1
832aa43b22e0	hyperledger/fabric-ca	"sh -c 'fabric-ca-se..."	16 seconds ago	Up 9 seconds	0.0.0.0:9054->7054/tcp	ca3.example
.com						
f4c579be96f9	hyperledger/fabric-couchdb	"tini -- /docker-ent..."	16 seconds ago	Up 8 seconds	4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp	couchdb3
e4a98459bae4	hyperledger/fabric-ca	"sh -c 'fabric-ca-se..."	16 seconds ago	Up 10 seconds	0.0.0.0:8054->7054/tcp	ca2.example
.com						
7c94d30d6739	hyperledger/fabric-couchdb	"tini -- /docker-ent..."	16 seconds ago	Up 10 seconds	4369/tcp, 9100/tcp, 0.0.0.0:8984->5984/tcp	couchdb4
298344e320d4	hyperledger/fabric-couchdb	"tini -- /docker-ent..."	16 seconds ago	Up 11 seconds	4369/tcp, 9100/tcp, 0.0.0.0:6984->5984/tcp	couchdb2
85399c25ea47	hyperledger/fabric-ca	"sh -c 'fabric-ca-se..."	16 seconds ago	Up 11 seconds	0.0.0.0:6054->7054/tcp	ca4.example
.com						
150f625d08e3	hyperledger/fabric-ca	"sh -c 'fabric-ca-se..."	16 seconds ago	Up 12 seconds	0.0.0.0:7054->7054/tcp	ca1.example
.com						

그림1. 생성된 컨테이너들

## 네트워크 구축 프로세스 - 셸 스크립트

### 2 start.sh

생성된 container (각 기관들)를 생성된 채널에 가입

```
# Create the channel
docker exec cli peer channel create -o orderer.example.com:7050 -c mychannel -f /etc/hyperledger/configtx/channel.tx
2019-12-13 08:50:49.228 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2019-12-13 08:50:49.326 UTC [cli.common] readBlock -> INFO 002 Received block: 0
# Join peer0.org1.example.com to the channel.
docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example.com/msp" peer0.org1.example.com peer channel join -b /etc/hyperledger/configtx/mychannel.block
2019-12-13 08:50:49.770 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2019-12-13 08:50:50.273 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
sleep 5
# Join peer0.org2.example.com to the channel.
docker exec -e "CORE_PEER_LOCALMSPID=Org2MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org2.example.com/msp" peer0.org2.example.com peer channel join -b /etc/hyperledger/configtx/mychannel.block
2019-12-13 08:50:55.976 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2019-12-13 08:50:56.309 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
sleep 5
# Join peer0.org3.example.com to the channel.
docker exec -e "CORE_PEER_LOCALMSPID=Org3MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org3.example.com/msp" peer0.org3.example.com peer channel join -b /etc/hyperledger/configtx/mychannel.block
2019-12-13 08:51:01.760 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2019-12-13 08:51:02.104 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
sleep 5
# Join peer0.org4.example.com to the channel.
docker exec -e "CORE_PEER_LOCALMSPID=Org4MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org4.example.com/msp" peer0.org4.example.com peer channel join -b /etc/hyperledger/configtx/mychannel.block
2019-12-13 08:51:07.614 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2019-12-13 08:51:07.953 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
sleep 5
```

그림2. 각 컨테이너들의 채널 가입

3 cc.sh

각 기관에 체인코드 배포

```
(base) hongui-MacBook-Air:network hongjoonkim$ ./cc.sh
2019-12-13 08:59:34.896 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2019-12-13 08:59:34.896 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2019-12-13 08:59:36.955 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
2019-12-13 08:59:37.278 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2019-12-13 08:59:37.279 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2019-12-13 08:59:37.642 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
2019-12-13 08:59:37.990 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2019-12-13 08:59:37.990 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2019-12-13 08:59:38.334 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
2019-12-13 08:59:38.676 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2019-12-13 08:59:38.676 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2019-12-13 08:59:39.012 UTC [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
2019-12-13 08:59:39.360 UTC [chaincodeCmd] InitCmdFactory -> INFO 001 Retrieved channel (mychannel) orderer endpoint: orderer.example.com:7050
2019-12-13 08:59:39.363 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default escc
2019-12-13 08:59:39.363 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 003 Using default vscc
-----END-----
(base) hongui-MacBook-Air:network hongjoonkim$
```

그림3. 체인코드 배포 결과화면

## 네트워크 구축 프로세스 - 셸 스크립트

---

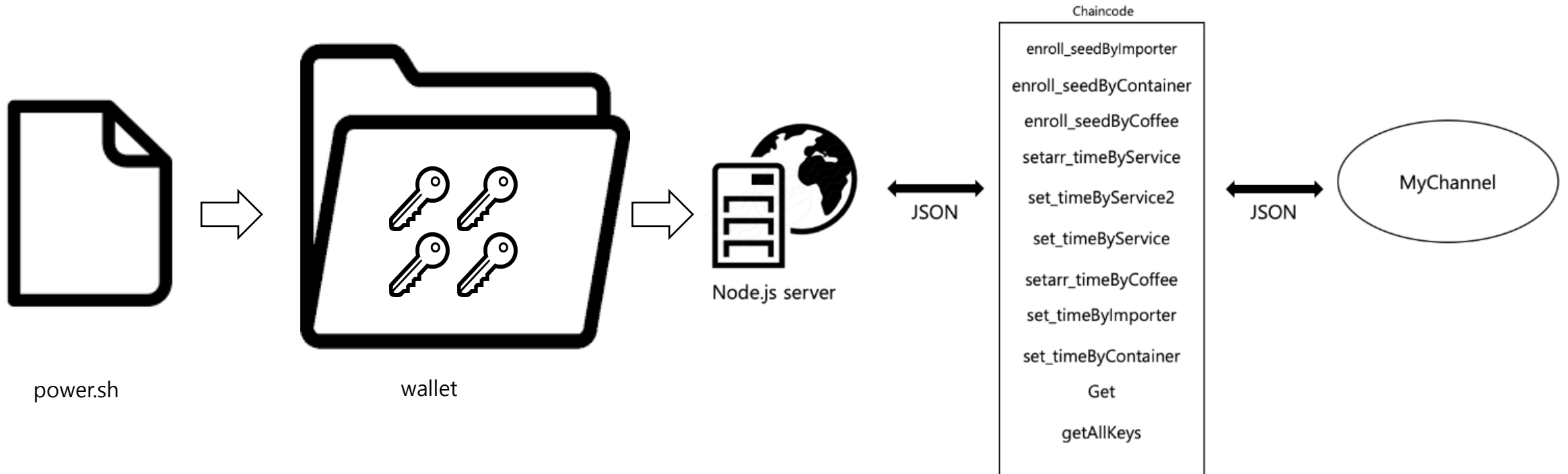
3

cc.sh

```
[root@c4863489d351:/# ls
bin boot dev etc home host lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@c4863489d351:/# cd var/
[root@c4863489d351:/var# ls
backups cache hyperledger lib local lock log mail opt run spool tmp
[root@c4863489d351:/var# cd hyperledger/
[root@c4863489d351:/var/hyperledger# ls
production
[root@c4863489d351:/var/hyperledger# cd production/
[root@c4863489d351:/var/hyperledger/production# ls
chaincodes ledgersData transientStore
[root@c4863489d351:/var/hyperledger/production# cd chaincodes/
[root@c4863489d351:/var/hyperledger/production/chaincodes# ls
sacc.1.7
[root@c4863489d351:/var/hyperledger/production/chaincodes#
```

그림4. Org4 컨테이너속에 체인코드 베포 확인

## 서버 프로세스





### 1 power.sh

```
[(base) hongui-MacBook-Air:enrollAdmin hongjoonkim$ ./power.sh
Wallet path: /Users/hongjoonkim/Desktop/Walton_final_project/application/wallet
Successfully enrolled admin user "admin" and imported it into the wallet
Wallet path: /Users/hongjoonkim/Desktop/Walton_final_project/application/wallet
Successfully enrolled admin user "admin" and imported it into the wallet
Wallet path: /Users/hongjoonkim/Desktop/Walton_final_project/application/wallet
Successfully enrolled admin user "admin" and imported it into the wallet
Wallet path: /Users/hongjoonkim/Desktop/Walton_final_project/application/wallet
Successfully enrolled admin user "admin4" and imported it into the wallet
(base) hongui-MacBook-Air:enrollAdmin hongjoonkim$ █
```

그림5. power.sh 실행 화면

#### 설명

각 기관의 MSP 키 값을 이용하여, 각 기관의 CA를 생성

### 1 power.sh

```
(base) hongui-MacBook-Air:application hongjoonkim$ cd wallet/  
(base) hongui-MacBook-Air:wallet hongjoonkim$ ls  
admin1 admin2 admin3 admin4
```

그림6. wallet 폴더에 생성된 기관별 CA

```
"=====  
" Netrw Directory Listing (netrw v165)  
" /Users/hongjoonkim/Desktop/Walton_final_project/application/wallet/admin1  
" Sorted by name  
" Sort sequence: [\v]$, \<core\%(\\.\\.d\\+\\)\>=\\>,\\.h$,\\.c$,\\.cpp$,\\.\\=\\*$,*,\\.o$,\\.obj$,\\.info$,\\.swp$,\\.bak$,\\.\\$  
" Quick Help: <F1>:help -:go up dir D:delete R:rename s:sort-by x:special  
"=====  
./  
./  
admin1  
fafe40427a39bcc930bae3c2bb92f3ef7f1c7be72d1bfac654e04af927265818-priv  
fafe40427a39bcc930bae3c2bb92f3ef7f1c7be72d1bfac654e04af927265818-pub  
~
```

그림7. 기관1의 CA 키 값