Gsym 사용자 가이드

gsym 실행 및 private network 구축

INDEX

1. 개요

개요

1. Symverse Network 에 접속하기 위해서는 OS에 맞는 GSYM 바이너리 파일을 다운 받아야 합니다.
    - 다운로드: Symverse 다운로드 페이지
2. 현재 테스트용으로 Test-net을 운영 중이며 Test-net 접속 방법은 2장을 참조하시기 바랍니다. 단, Test-net은 예고없이 S/W 업그레이드나, 노드 shut-down이 진행될 수 있습니다.
3. Main-net은 현재 운영 중이 아니며 추후 공개 예정입니다.
4. GSYM을 이용하여 Private network를 구성할 수 있으며, 실행 방법은 3장을 참조하시기 바랍니다.

2. Gsym 실행 방법

Gsym 실행 방법

기타 option에 대한 사항은 Ethereum 문서를 참조 (https://github.com/ethereum/go-ethereum/wiki/Command-Line-Options)

✓ Gsym 실행

```
gsym --testnet --rpc --rpcaddr "0.0.0.0" --syncmode "full"
     --rpcport 8545 --rpcapi "admin,sym,debug,net,personal,web3,pon,citizen" --verbosity 3
```

- Main-net과 연결을 하기 위해서는 networkid 옵션을 주지 않고 실행
- Test-net에 접근하기 위해서는 --networkid 옵션을 2으로 주거나 --testnet 옵션을 넣어 실행
- Options
  - **datadir** - gsym 데이터 저장 폴더를 의미 (keystore, 블록체인 데이터)
  - **networkid** - private network 연결시 지정 (1: main-net, 2: test-net, 3 이상: private-net)
  - **port** - Network listening p2p port (default: 30303)
  - **rpcapi** - 제공할 http-rpc interface API
  - **rpcaddr** - HTTP-RPC server listening interface (default: "localhost").
  - **rpcport** - HTTP-RPC server listening port (default: " 8545")
  - **rpc** - HTTP-RPC 사용
  - **verbosity** - Logging 레벨 0=silent, 1=error, 2=warn, 3=info, 4=debug, 5=detail (default: 3)
  - **testnet** - testnet 에 연결 (networkid 2와 동일)

✓ 실행 예제

```
➢ Gsym                          // Main-net 연결 실행
➢ Gsym –testnet                 // Test-net 연결 실행
➢ Gsym --networkid 9999 --rpc --rpcaddr "0.0.0.0" --rpcport 8000  //본인이 구축한 Private-net 연결 실행
```

Gsym 접속 방법

✓ **Gsym console 접속** (COMMAND - attach)

gsym attach http://localhost:30303

- 실행중인 Gsym 실행시 rpc 관련 Option 들을 참고 하여 해당 명령어를 실행

```
Welcome to the Gsym JavaScript console!

instance: Gsym/v0.0.6-Develope-e0369e15/linux-amd64/go1.10.4
blockcreator: 0x00021000000000010002
at mainblock: 33528 (Fri, 26 Apr 2019 10:13:20 UTC)
at citizenblock: 2
at warrantblock: 33528
 datadir: /nodes/node1
 modules: admin:1.0 citizen:1.0 debug:1.0 net:1.0 personal:1.0 pon:1.0 rpc:1.0 sct:1.0 sym:1.0 warrant:1.0 web3:1.0

>
```

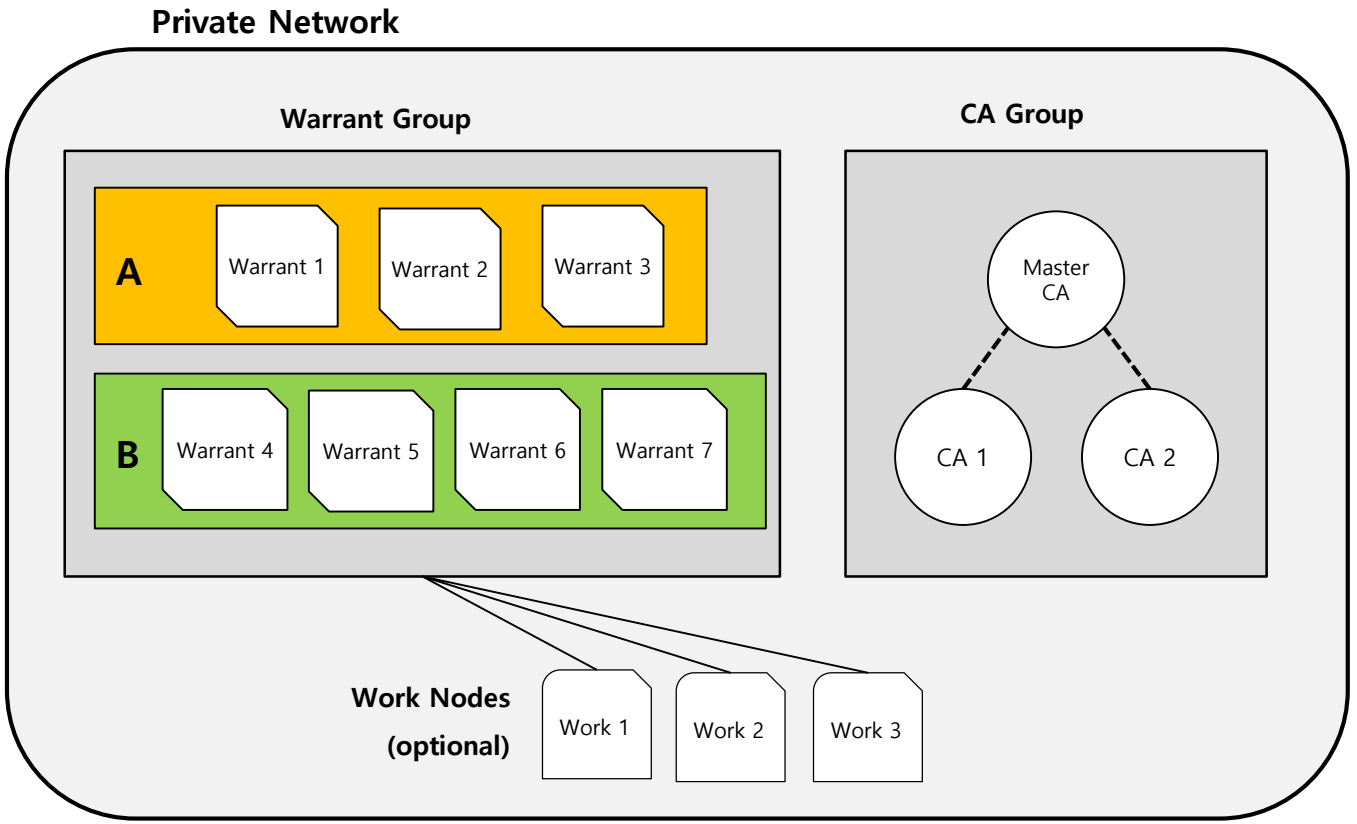
실행하고 나면 위와 같은 console 화면이 나타납니다.

- Gsym Http-RPC Method 는 https://github.com/symverse-lab/Document 참조
- 그 외 기본 Method는 geth base이므로 https://github.com/ethereum/wiki/wiki/JSON-RPC 참조
  (**eth_** 로 되어 있는 부분은 **sym_** 으로 변경 후 사용)

3. Private Network 구성 방법

Private Network 구성 방법

✓ Private Network 구성도



※ 필수 운용 노드 **(최소 9개의 노드 필요)**

- CA 기관 등록을 담당하는 **1개의 Master CA 노드**
- Citizen 등록을 담당하는 **최소 1개 이상의 CA 노드**
- 2/3 블록 합의를 위해 **최소 7개의 보증 노드 (A그룹 3, B그룹 4)**

Private Network 구성 방법

✓ **Private Network 구성 순서**

1. 구성 준비

1) 필수 노드(Warrant, Master CA, CA)의 계정(keystore)을 생성
2) 생성된 각 노드의 keystore에서 public key 정보를 추출
3) 추출된 public key 정보를 이용하여 genesis.json 파일을 작성
4) 부트 노드 생성 및 실행

2. 필수 노드(Warrant, Master CA, CA) 노드 생성 및 실행

1) genesis.json을 이용해 Warrant, Master CA, CA 노드를 생성
2) Warrant 노드 실행
3) Master CA, CA 노드 실행

3. Work 노드 생성, citizen 등록 및 실행

1) Work 노드의 계정(keystore)을 생성
2) 생성된 노드의 keystore에서 public key 정보를 추출
3) genesis.json을 이용해 Work 노드를 생성
4) CA 노드에서 추출된 Work 노드의 public key 정보를 넣어 citizen 등록을 진행
5) Work node 실행

Private Network 구성 방법

✓ 구성 준비

1. 필수 노드(Warrant, Master CA, CA)의 계정(keystore)을 생성

gsym --datadir {dir} **account new** "{symid}" "{passphrase}"

결과 :
```
05-03 11:09:33.216 INFO  Maximum peer count                    SYM=25 LES=0 total=25
05-03 11:09:33.216 INFO  accountCreate                         len(ctx.Args())=0
Address: {0000000000000000001}
```

2. 생성된 각 노드의 keystore에서 public key 정보를 추출

gsym --datadir {dir} **account info** "{symid}" "{passphrase}"

결과 :
```
05-03 10:32:50.187 INFO  Unlocked account                      address=0x00020000000000010002
05-03 10:32:50.187 INFO  Account information                   address=0x00020000000000010002 pubkeyhash=0xc9FEd8367c4B22e3572ecF883739C848e6079256
```

3. 추출된 public key 정보 중 필수 노드의 정보를 이용하여 genesis.json 파일을 작성

- page 12 참조

4. 부트 노드 생성 및 실행

- page 13 참조

Private Network 구성 방법

✓ 구성 준비

※ genesis.json 구성 요소 설명

- **필수 구성 요소**
  - **chainId** – private network chinid를 설정
  - **caList** – 초기 생성할 CA 노드들의 정보
    (CA 노드의 symid 와 그에 맞는 pubkey hash를 기록)
  - **mcaddr** – 마스터 CA 노드의 SymID
  - **mcapubkey** – 마스터 CA 노드의 pubkeyhash
  - **beginTerm** – Warrant 노드 활동 시작 블록 번호
  - **endTerm** – Warrant 노드 활동 종료 블록 번호
  - **warrantList** – Warrant 노드 리스트
    1. Key값 – Warrant 노드의 SymID
    2. Pubkeyhash – Warrant 노드 SymID에 해당하는 pubkeyhash
    3. From – Warrant 노드의 citizen 생성을 담당할 CA의 SymID
    4. Group – A / B 그룹 (0 – A그룹, 1 – B그룹)
- **기타 genesis 구성 요소**
  - 나머지 구성요소(coinbase, extraData, gasLimit, nonce, mixhash, parentHash, timestamp, alloc 등)는 Ethereum genesis rule을 따름.

Genesis.json 예제 파일

```
{
  "config":{
    "chainId":7777,
    "linitialBlock":1
  },
 "caList":{
  "0x0002000000000000010002":{
    "pubkeyhash":"0xc9FEd8367c4B22e3572ecF883739C848e6079256"
  },"0x0002000000000000020002":{
    "pubkeyhash":"0xc017C0C5C71B74eF719B6cfc79B715085fBC736D"
  }
},
  "mcaaddr" : "0x0001000000000000010002",
  "mcapubkey" : "0xc6391BA254A31c80fEA6F68cCe4C233B21F2E8C5",
  "beginTerm" : "1",
  "endTerm" : "10",
  "warrantList":{
    "0x0002100000000000010002":{
      "pubkeyhash":"0xD1d6fb417A0F6a86228FAfc2Ade8CA3EfCF0e952",
      "from":  "0x0002000000000000010002",
      "group":"0x0"
    },
    "0x0002100000000000020002":{
      "pubkeyhash":"0x6e6276E23110ae558d74d0aCF377a22e37aC09FD",
      "from":  "0x0002000000000000010002",
      "group":"0x0"
    },
    "0x0002100000000000030002":{
      "pubkeyhash":"0x31b83aFd41C64704647d23065b29B61a396987A8",
      "from":  "0x0002000000000000010002",
      "group":"0x0"
    },
    "0x0002100000000000040002":{
      "pubkeyhash":"0xE9f5425E7F907b457253E444F0203bAA53189e92",
      "from":  "0x0002000000000000010002",
      "group":"0x1"
    },
    "0x0002100000000000050002":{
      "pubkeyhash":"0x1333919Fe11Bf0B9f9f693197c27C618BE08E139",
      "from":  "0x0002000000000000010002",
      "group":"0x1"
    },
    "0x0002100000000000060002":{
      "pubkeyhash":"0x1078283F06f01C995BCa20952Db7c7033044C9b3",
      "from":  "0x0002000000000000010002",
      "group":"0x1"
    },
    "0x0002100000000000070002":{
      "pubkeyhash":"0xbAd83768c589c47766e466D619680f1a47C58361",
      "from":  "0x0002000000000000010002",
      "group":"0x1"
    }
  }
}
```

Private Network 구성 방법

✓ 구성 준비

※ Bootnode 생성 및 실행

1) Bootnode key 생성

bootnode -genkey {filename}

- Bootnode --genkey 명령어를 통해 bootnode 의 Key 를 생성

2) Bootnode 실행

bootnode -nodekey {filename}

- Bootnode를 생성한 Key로 실행
- **Option**
  - nodekey – gsym 데이터 저장 폴더를 의미 (keystore, 블록체인 데이터)
  - addr – bootnode 의 endpoint ***(default ":30301")***
- **실행 예제**

```
[root@e231e79ede56:/# bootnode -nodekey bootkey                                                                    ]
05-07 07:35:37.352 INFO  UDP listener up                          self=enode://3ff23b4305efb443b0273ce3f29881c03e526d8d50394e6471b0cc83d4f66bbbf79b92d104a18987a9a896379e54f23298f333c4a5e7a
070c2332c5bcd331c73@[::]:30301
```

- Bootnode 실행 시 나오는 주소를 gsym 실행시 옵션값으로 추가
- Bootnode 주소 형식 – enode://{주소}@{포트}

Private Network 구성 방법

✓ 필수 노드(Warrant, Master CA, CA) 생성 및 실행

1. genesis.json을 이용해 Warrant, Master CA, CA 노드를 생성

```
gsym --datadir {dir} init genesis.json
```

2. Warrant 노드 실행

- Primary 노드(최초 실행 시 primary 지정) - 최초 gsym실행 시 warrant 노드 중 하나는 primary로 실행하거나 warrant 노드 시작 후 pon.primary(true) API command로 블록 생성을 시작할 수 있음.

```
gsym --networkid 7770 --addr 172.30.1.6 --port 3101 --datadir ./node1
     --primary --warrant --symbase_password "1234" --n2nport 7001 --n2nboot "0x000210000000000010002@172.30.1.6:7001"
     --rpc --rpcaddr "0.0.0.0" --rpccorsdomain "*" --rpcport 8001 --rpcapi "admin,sym,debug,net,personal,web3,pon,warrant,citizen"
     --bootnodes "enode://afd6fcbf8cfa30ac1bffff6eebf7442c9caa5126eaf3175ee6b90fec58161a166c24d10c16327358ae7fdaa55b823381b4
       cc3b4941faa27603983eb66bce6e84@54.180.27.161:30301" --verbosity 3 console
```

- 나머지 Warrant 노드

```
gsym --networkid 7770 --addr 172.30.1.6 --port 3102 --datadir ./node2
     --warrant --symbase_password "1234" --n2nport 7002 --n2nboot "0x000210000000000010002@172.30.1.6:7001"
     --rpc --rpcaddr "0.0.0.0" --rpccorsdomain "*" --rpcport 8002 --rpcapi "admin,sym,debug,net,personal,web3,pon,warrant,citizen"
     --bootnodes "enode://afd6fcbf8cfa30ac1bffff6eebf7442c9caa5126eaf3175ee6b90fec58161a166c24d10c16327358ae7fdaa55b823381b4
       cc3b4941faa27603983eb66bce6e84@54.180.27.161:30301" --verbosity 3 console
```

※ Options

- **networkid** – private network 구분자
- **addr** – node ip address
- **port** – node p2p port
- **datadir** – node data path
- **n2nport** – n2n 프로토콜 사용할 포트. warrant 노드 간 합의를 위한 port 정보.
- **n2nboot** – n2n boot node 지정. Warrant 노드 중 1개를 n2n boot 노드로 지정. (symid@ip:port)
- **primary** – 최초 실행시 warrant 노드 중에서 최초 블록 생성을 담당할 primary를 지정. 옵션 미지정시 노드 실행 후 임의의 노드에서 pon.primary(true) API command를 사용하여 블록 생성을 시작할 수 있음.
- **warrant** – warrant 노드로 실행합니다
- **symbase_password** – 블록 생성시 서명을 위한 warrant node keystore의 passphrase 입니다.
- **bootnodes** – bootnode의 주소 정보입니다. (private network에서 worknode 운용시 반드시 필요)
- **console** – gsym console 실행

Private Network 구성 절차

✓ 필수 노드(Warrant, Master CA, CA) 생성 및 실행

3. Master CA, CA 노드 실행

```
gsym --networkid 7770 --addr 172.30.1.6 --port 3111 --datadir ./node11
     --rpc --rpcaddr "0.0.0.0" --rpccorsdomain "*" --rpcport 8011 --rpcapi "admin,sym,debug,net,personal,web3,pon,warrant,citizen"
     --bootnodes "enode://afd6fcbf8cfa30ac1bffff6eebf7442c9caa5126eaf3175ee6b90fec58161a166c24d10c16327358ae7fdaa55b823381b4
       cc3b4941faa27603983eb66bce6e84@54.180.27.161:30301" --verbosity 3 console
```

※ Options

- Warrant 노드 실행 옵션 참조
- CA 노드는 Work 노드와 동일하게 동작함

Private Network 구성 방법

✓ Work 노드 생성, citizen 등록 및 실행

1. Work 노드로 실행할 노드의 계정(keystore)을 생성

- 필수 노드의 계정(keystore) 생성 참조

2. 생성된 각 노드의 keystore에서 public key 정보를 추출

- 필수 노드의 public key 정보 추출 참조

3. CA 노드에서 추출된 work 노드의 public key 정보를 넣어 citizen 등록을 진행

- 추가로 CA 노드를 등록할 경우는 Master CA 노드에서 citizen 등록을 진행

```
gsym attach http://127.0.0.1:8545                                          //CA console 접속
> personal.unlockAccount("00020000000000020002", "1234")                   //CA 계정 unlock
> citizen.sendCitizen({"from":"00020000000000020002","to":"0x0000000000000000001","symid":"0x0002A000000000010002",
                       "pubkeyhash":"0xe04E79dE463b31516994257A7F4fcA50C1414dB1","status":"0x1","role":"0x1","vflag": "0x1",
                       "refcode": "0x1", "country":"0x0", "credit":"0x1"})  //citizen 등록
```

※ sendCitizen() parameter는 Symverse RPC-API 참조

4. Work node 실행

- Master CA, CA 노드 실행 참조