Ethereum Private Network

권수호

suho.kwon@gmail.com, http://blog.naver.com/knix008

# Enterprise Blockchain

- **Blockchain for Business :**
  - The enterprise blockchain market is huge, and there are different solutions available to implement blockchain for enterprise -- **Hyperledger**, **Corda**, **Quorum**, and **Ethereum** are the well-known providers. You can set up enterprise blockchain in different ways, on a **cloud**, or **on-premise**.
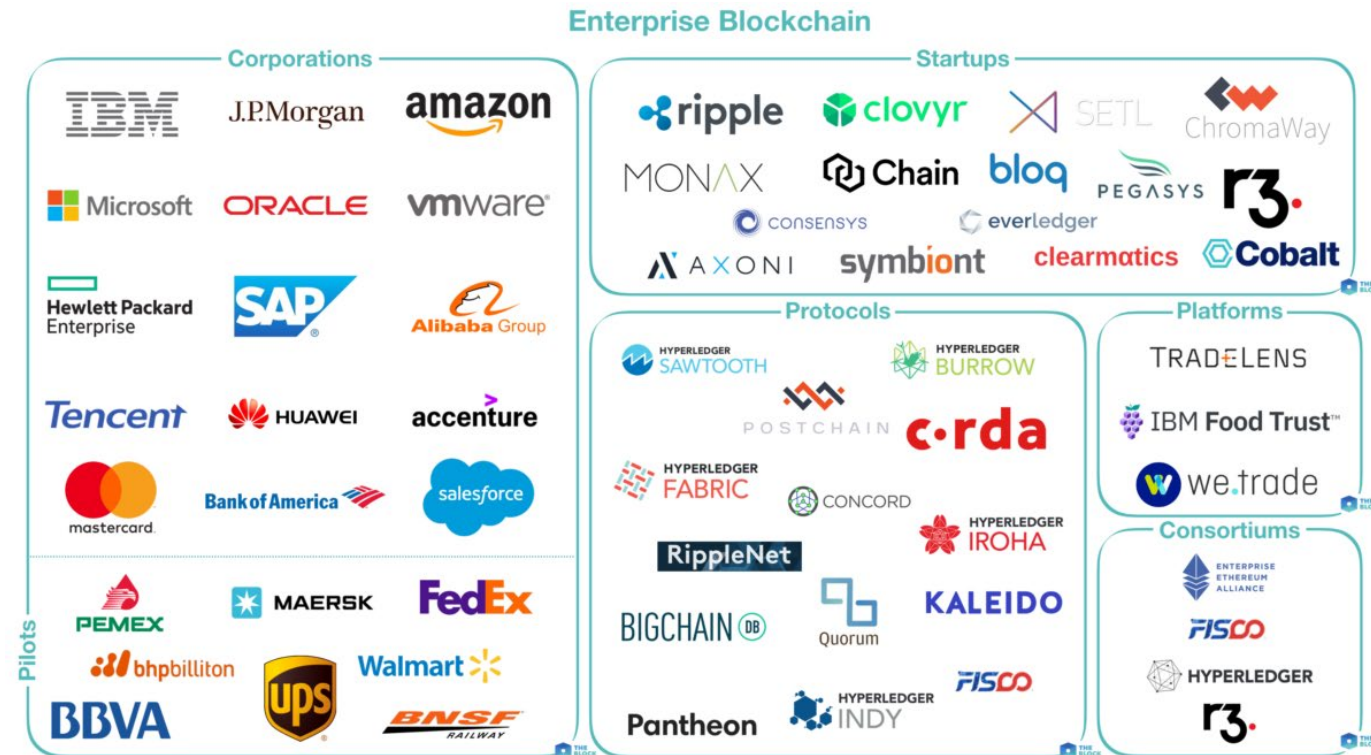
Validator Nodes

- **Proof of Authority :**
    - We will use **proof of authority (PoA)** consensus to setup the private network. **In PoA-based networks, transactions and blocks are validated by approved accounts**, known as **validators** or **sealer**. The process is automated and does not require validators to be constantly monitoring their computers. **PoA algorithms rely on a set of N trusted nodes called the authorities**.
    - Types of Nodes in **PoA** Network
        - **Validator/Authority** - Validate, confirm transactions, and add new blocks into blockchain.
        - **Full Node** - This node keeps a copy of the full blockchain.
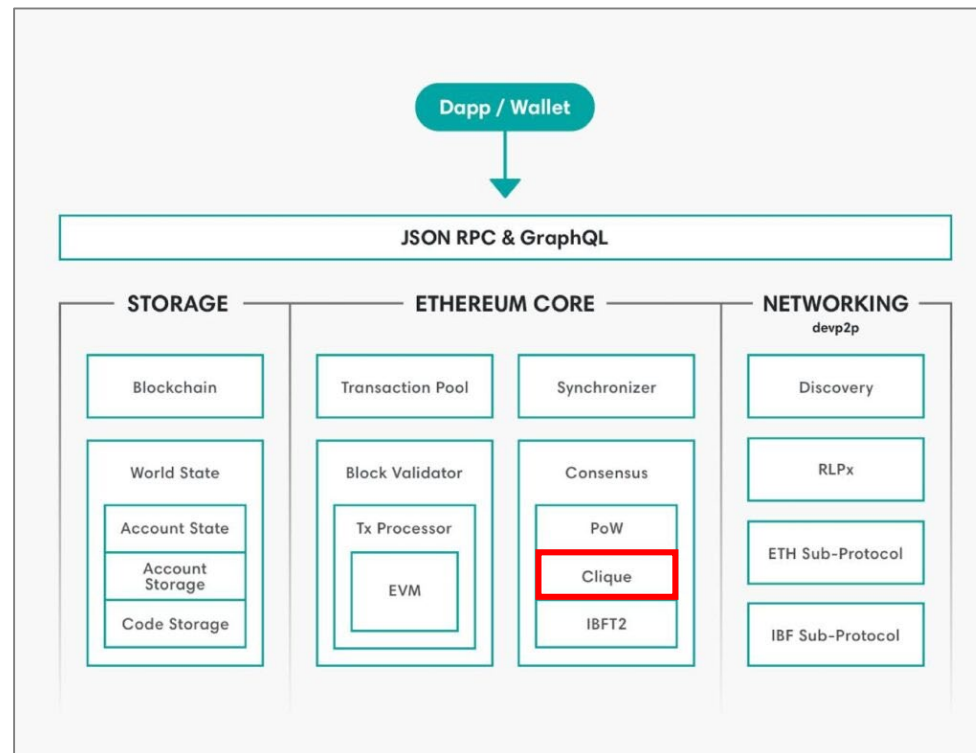
Private Network(1/12)

- **Make Your Own Private Ethereum Network :**
  - **Step 01**. Create folders like node1, node2, node3. Use "**mkdir xxx**" command to create each folders.

  - **Step 02**. Create validator accounts for each node. Use "**geth --datadir xxx accont new**" command for each folders and **save the accounts**. You also need to **remember all passwords** for each newly created accounts.

Private Network(2/12)

- **Make Your Own Private Ethereum Network :**
    - **Step 03**. Create "**genesis.json**" file. Use "**puppeth**" tool for generating genesis file. This tool will be installed with geth when you are choosing "**geth with tools**" option for download or build from the source code with "make all" command. Run "**puppeth**" command.
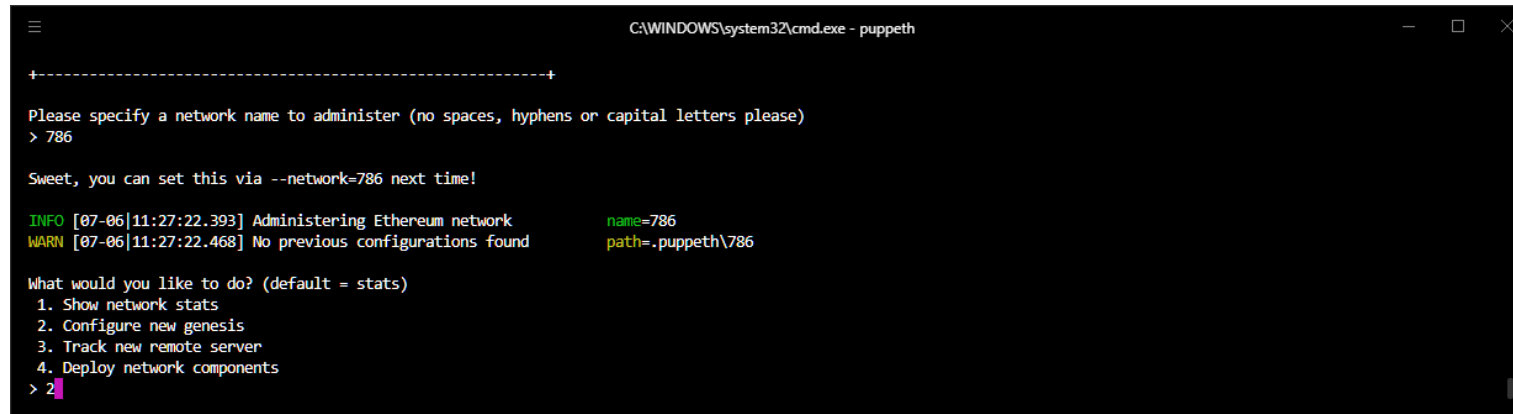   - **Step 04**. Specify the name of the genesis file.

```
                                          C:\WINDOWS\system32\cmd.exe - puppeth        —    □    ✕


c:\Home\Project\Smart-Contract\Private>puppeth
+-----------------------------------------------------------+
| Welcome to puppeth, your Ethereum private network manager |
|                                                           |
| This tool lets you create a new Ethereum network down to  |
| the genesis block, bootnodes, miners and ethstats servers |
| without the hassle that it would normally entail.         |
|                                                           |
| Puppeth uses SSH to dial in to remote servers, and builds |
| its network components out of Docker containers using the |
| docker-compose toolset.                                   |
+-----------------------------------------------------------+

Please specify a network name to administer (no spaces, hyphens or capital letters please)
> 786
```

    - **Step 05**. Generate new genesis file. Select 2 for "Configure new genesis" and select 1 for "Create new genesis from scratch".

```
                                          C:\WINDOWS\system32\cmd.exe - puppeth        —    □    ✕


+-----------------------------------------------------------+

Please specify a network name to administer (no spaces, hyphens or capital letters please)
> 786

Sweet, you can set this via --network=786 next time!

INFO [07-06|11:27:22.393] Administering Ethereum network           name=786
WARN [07-06|11:27:22.468] No previous configurations found         path=.puppeth\786

What would you like to do? (default = stats)
 1. Show network stats
 2. Configure new genesis
 3. Track new remote server
 4. Deploy network components
> 2
```

Private Network(3/12)

- **Make Your Own Private Ethereum Network :**
    - **Step 06**. Select **PoA**. Select 2 for "Clique - proof-of-authority".
    - **Step 07**. Enter **1** for block time. It means the block interval time in seconds.


```
                                      C:\WINDOWS\system32\cmd.exe - puppeth                          _  □  ×

  3. Track new remote server
  4. Deploy network components
 > 2

What would you like to do? (default = create)
  1. Create new genesis from scratch
  2. Import already existing genesis
 > 1

Which consensus engine to use? (default = clique)
  1. Ethash - proof-of-work
  2. Clique - proof-of-authority
 > 2

How many seconds should blocks take? (default = 15)
 > 1
```

    - **Step 08**. Provides accounts for authority you already created before. When you are done, just type "enter" key.


```
                                      C:\WINDOWS\system32\cmd.exe - puppeth                          _  □  ×

  2. Import already existing genesis
 > 1

Which consensus engine to use? (default = clique)
  1. Ethash - proof-of-work
  2. Clique - proof-of-authority
 > 2

How many seconds should blocks take? (default = 15)
 > 1

Which accounts are allowed to seal? (mandatory at least one)
 > 0x75E3692F01Dc714F82637c9AC99bF6145b39Be86
 > 0x47e0e9906B615Cb4238B72E398673933A590AE30
 > 0xb9749440CB4704b6b4487BF6499471C2A733BB2F
 > 0x
```

Private Network(4/12)

- **Make Your Own Private Ethereum Network :**
    - **Step 09**. Provide accounts to get some free ethers. You can use your previous accounts again.

    - **Step 10**. Specify your chain/network ID.

Private Network(5/12)

- **Make Your Own Private Ethereum Network :**
  - **Step 11**. Select 2 for "Manage existing genesis" and select 2 for "Export genesis configurations".

  - **Step 12**. Just type "enter" key to generate genesis file in the current directory. You can see that "**Aleth**" and "**Parity**" will not support the newly generated "**gensis.json**" file. "**Ctrl+C**" will let you leave the prompt.

Private Network(6/12)

- **Make Your Own Private Ethereum Network :**
  - **Step 13**. Check your genesis files. You can see 2 "**786-xxx.json**" files in your current folder.
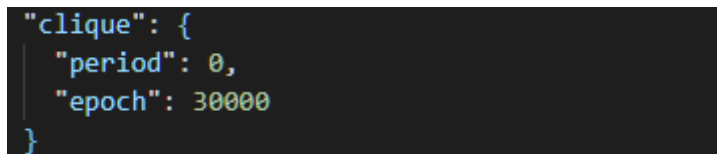
```
c:\Home\Project\Smart-Contract\Private 디렉터리

2021-07-06  오전 11:48    <DIR>          .
2021-07-06  오전 11:48    <DIR>          ..
2021-07-06  오전 11:45    <DIR>          .puppeth
2021-07-06  오전 11:48            22,034 786-harmony.json
2021-07-06  오전 11:48            22,034 786.json
2021-07-06  오전 10:18               130 accounts.txt
2021-07-06  오전 10:08    <DIR>          node1
2021-07-06  오전 10:11    <DIR>          node2
2021-07-06  오전 10:17    <DIR>          node3
               3개 파일              44,198 바이트
               6개 디렉터리  400,547,450,880 바이트 남음

c:\Home\Project\Smart-Contract\Private>
```

  - **Step 14**. Edit the created "**genesis.json(=786.json)**" file to set "**period**" field to 0 in "**clique(PoA)**". **This will automatically force "geth" to only create a new block whenever there is a transaction in the network.**

```
"clique": {
  "period": 0,
  "epoch": 30000
}
```

Private Network(7/12)

- **Make Your Own Private Ethereum Network :**
  - **Step 15**. Initialized the Nodes with the same "**genesis(genesis.json)**" file. Before you do this step, you can rename your genesis file to "**genesis.json**" with "**rename 768.json genesis.json**" command in Windows. In Linux, you can use "**mv 786.json genesis.json**" command. To initialize geth client, use "**geth --datadir directory_name init genesis.json**" command.

Private Network(8/12)

- **Run the Private Network :**
    - **Step 01**. Make your password file like "**password.txt**" in each folders.
    - **Step 02**. Run "**geth**" with the following option in each folders. **You need to use the account created for the folder**.

```
geth --nousb --datadir $pwd --nodiscover --ipcpath gethnode1.ipc --syncmode full --port 30310 --miner.gasprice 0 --
miner.gastarget 470000000000 --http --http.addr 127.0.0.1 --http.port 8545 --
http.api admin,eth,miner,net,txpool,personal,web3 --mine --allow-insecure-unlock --
unlock "0x75E3692F01Dc714F82637c9AC99bF6145b39Be86" --password password.txt
```

Note : Geth Command Option

- **Other Nodes :**
  - **Node2 -**

    geth --nousb --datadir $pwd --nodiscover --ipcpath **gethnode2.ipc** --syncmode full --port **30311** --miner.gasprice 0 --miner.gastarget 470000000000 --http --http.addr 127.0.0.1 --http.port **8546** --http.api admin,eth,miner,net,txpool,personal,web3 --mine --allow-insecure-unlock --unlock "**0x47e0e9906B615Cb4238B72E398673933A590AE30**" --password password.txt

  - **Node3 -**

    geth --nousb --datadir $pwd --nodiscover --ipcpath **gethnode3.ipc** --syncmode full --port **30312** --miner.gasprice 0 --miner.gastarget 470000000000 --http --http.addr 127.0.0.1 --http.port **8547** --http.api admin,eth,miner,net,txpool,personal,web3 --mine --allow-insecure-unlock --unlock "**0xb9749440CB4704b6b4487BF6499471C2A733BB2F**" --password password.txt

Note : Geth Command

- **Options :**
    - **--nousb :** Disables monitoring for and managing USB hardware wallets.
    - **--datadir=$pwd :** Data directory for the databases and keystore.
    - **--syncmode full** : Enables full chain sync (default: "fast").
    - **--port 3030** : assign port geth process.
    - **--miner.gasprice 0** : On private network we don't want to pay anything, so make it zero.
    - **--miner.gastarget 470000000000** : Use same as the genesis file. However, actual gaslimit is dynamic for new block and depends on the previous block.
    - **--http** : Enable the HTTP-RPC server.
    - **--http.addr 127.0.0.1** : HTTP-RPC server listening interface (default: "localhost").
    - **--http.port 8545** : HTTP-RPC server listening port (default: 8545).
    - **--http.api** : Bunch of APIs to be used over RPC calls.
    - **--mine** : Enable mining.
    - **--allow-insecure-unlock** : Allow insecure account unlocking when account-related RPCs are exposed by http.
    - **--unlock "0x251B7dC42D7472314fE4d6c3b9CAD2fb5927d7c8"** : Account to unlock, we'll use the address of node1.
    - **--password password.txt** : Password file to use for non-interactive password input.
- **Additional Options** :
    - **--http.corsdomain "*"** : Comma separated list of domains from which to accept cross origin requests (browser enforced).
    - **--http.vhosts="*"** : Comma separated list of virtual hostnames from which to accept requests (server enforced). Accepts '*' wildcard.
    - **--miner.threads value :** Number of CPU threads to use for mining (default: 0).

Private Network(9/12)

- **Run the Private Network :**
  - **Step 03**. Keep the "**enode**" information when you run the "**geth**" command in previous page.

  > enode://32d866aa8b9bddd4d0d2a6e076b41d921173c7fdb2b2b2b258b90a39c6bdf7f71603
  > 55dc4ea20b6b4d95192835e1e7a4c2ad3c13343a202172c2c003301f91d4@127.0.0.1:30310

  - **Step 04**. Run the "**geth**" command with **different port, http.port, account and password** options in each directory and get the "**enode**" information.
  - **Step 05**. Make "**static-nodes.json**" file with all the "**enode**" information you got from the previous "**geth**" run. Copy "**static-nodes.json**" file into each folders.

  > [
  >
  > "enode://32d866aa8b9bddd4d0d2a6e076b41d921173c7fdb2b2b2b258b90a39c6bdf7f7160355dc4ea20b6b4d951928
  > 35e1e7a4c2ad3c13343a202172c2c003301f91d4@127.0.0.1:30310",
  >
  > "enode://a19a5f05738ef9bf6175d9b0b2cf30ab6cae5b3d9e45c5e7f1b94d063cfdd66ea83d9bb33daa6f6502c126ea57
  > e575fe27e0dcb6cd0efd0d3d0ab8004a475d98@127.0.0.1:30311",
  >
  > "enode://ecb307906a28fd41d3a2969f7106b4f7300effdea8ba44f3bbce820d60cd512981f516c018c20710b5ac59f80a3
  > 4638ff930fc4d586f454f8a48f25759cb8828@127.0.0.1:30312"
  >]

Private Network(10/12)

- **Run the Private Network :**
  - **Step 06**. Run "**geth**" in different terminals for each folders. Here we will have 3 geth node.
  - **Step 07**. Run "**geth attach http://127.0.0.1/8545**" command.

Private Network(11/12)

- **Run the Private Network :**
  - **Step 08**. Check all the peer nodes in geth attached console with "**admin.peers**" command.

Private Network(12/12)

- **Peers :**
  - As you can the following screen shot, you have 2 peers for each geth nodes.

```
INFO [07-06|16:59:42.828] Etherbase automatically configured      address=0xb9749440CB4704b6b4487BF6499471C2A733BB2F
INFO [07-06|16:59:42.833] Commit new mining work                  number=1 sealhash=142621..ec8039 uncles=0 txs=0 gas=0 fees=0 elapsed="587.8µs"
INFO [07-06|16:59:42.833] Sealing paused, waiting for transactions
INFO [07-06|17:00:16.950] Looking for peers                       peercount=2 tried=2 static=3
```

  - You can also find out the number of discovered peers with "**net.peerCount**" command in geth attached console.

```
> net.peerCount
2
>
```

Note : Transmitting Transaction

- **Truffle :**
  - Use "**truffle migrate --network=develop**" command to send "**contract creation transaction**" to the private network.
  - Make sure that the "**truffle-config.js**" file has "**develop**" network configuration like the following.

```
develop: {
  host: "127.0.0.1",
  port: 8545,
  network_id: "*",
},
```

  - When you meet an error like "**block gaslimit**", then you need to change the "**genesis.json**" file. Increase the **block gaslimit value** in hexadecimal format.

```
"nonce": "0x0",
"timestamp": "0x60e3c044",
"extraData":
"0x0000000000000000000000000000000000000000000000000000000000047e0e9906b615cb4238b72e398673933a590ae3075e3692f0
1dc714f82637c9ac99bf6145b39be86b9749440cb4704b6b4487bf6499471c2a733bb2f0000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
"gasLimit": "0x7a1200",
"difficulty": "0x1",
"mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
"coinbase": "0x0000000000000000000000000000000000000000",
```

Note : Check Result

- **Screenshot :**
  - You can see that the transactions are mined and included in a block.