

Blockchain and Cryptocurrency

Assignment 7: Introduction to Ethereum & geth

This assignment can help you understand Ethereum network. To do this, you should install and run the go-ethereum (**geth**). **geth** is an open-source software with full functionality of Ethereum, providing multiple interfaces. It enables interactive JavaScript consoles and JSON-RPC server features. Geth is an official golang implementation of the Ethereum protocol. You will perform this assignment in a similar way to the Bitcoin Core assignment.

Description of this assignment:

- This task consists of a series of steps, like a tutorial.
- You must submit a report containing the results obtained as you go through each step.
⇒ **Submit a report containing all results obtained from Steps 1 to 8.**

Minimum Requirements:

- For this assignment, you will be using a Linux Virtual Server provided by the department.
- Accessible Hard disk at a minimum read/write speed of 100 MB/s
- 4 gigabytes of memory (RAM).

<Step. 1> Install go-ethereum (geth)

There are several ways to install geth in your server. In this assignment, we introduce 2 ways to install geth.

1) Installation from PPA (Personal Package Archive).

```
sudo apt-get install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install geth
```

✂ After installing Go and **geth**, check the version and completion of installation.

2) Installation from github.

In this way, you should install Go language before installing geth.

Go to the following URL for installing Go ☞ <https://golang.org/dl/>

Or install with apt-get ☞ "sudo apt-get install golang-go"

Then, go to <https://github.com/ethereum/go-ethereum>, download the source code from git, and compile it on your server.

<Step. 2 Check out the command line and management APIs.

Search for useful commands and management APIs (See <https://geth.ethereum.org/> or use `geth --help`). Include them in your report.

<Step. 3 Building Private Network

1) Generate a genesis file.

To build a private network, you have to generate a json-type genesis file and name it something like genesis.json

2) Create a data directory to use in the private network.

3) Make genesis block by using a genesis file.

4) Run **geth** engine with console.

<Step. 4 Generate addresses / Mine Ether / Check the balance

1) Create two addresses which can be used for sender and receiver in transmission testing.

2) Look up your account list.

3) Set the etherbase address (address for mining profit). Before you transfer Ether, you must mine Ether to verify the transaction. So, you have to specify the etherbase to be rewarded after mining.

※ Basically, the address generated first is assigned as an etherbase address. You can check it first with a certain API.

4) Start mining.

※ To get some Ether to be used for transmission in the private network, you have to start to mine Ether.

5) After mining, check the balance of all accounts.

<Step. 5> Unlock the account / Send the transaction.

1) Check the status of sender's account.

2) Before sending Ether, you have to unlock the sender's account.

3) Send 20 Ethers to the receiver's address.

※ Keep in mind that the minimum gas for transfer is 21,000.

4) Check the pending transactions.

※ To validate the transaction, block mining is required.

5) Check the pending transactions again after mining.

6) Look up the balance of receiver's account to see if sender successfully transfer 20 Ethers.

<Step. 6> Search for blocks

Try to search for blocks contained to the private chain.

1) Search the total number of blocks connected to the main chain.

2) Height of block: 0

3) Height of block: 100

4) The block including the transaction which is used in transferring your Ether.

<Step. 7> Join other's private network

※ You can join other's private network by connecting a node. To connect a node, each node should have the same network id and genesis file. By using attached genesis file, network id and enode, join the private network which is already built before.

- 1) Connect the node from your PC.
- 2) Print out the information of connected peers.

<TA>

Jeongheon Kim <kjheon1118@postech.ac.kr>

Genesis file:

```
{
  "config": {
    "chainId": 8070,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0
  },
  "difficulty": "0x10000",
  "gasLimit": "0x10000000",
  "alloc": {
    "0xbBE9834bAe1F70e0D60C4230F3F5aD9f845d2b37": {"balance": "30000000000000000000"}
  },
  "nonce": "0x0000000000000000"
```

Network id: 8070

enode:

b34e6da5ffb93e8ef13c9280e4651fbb87d4f5b127dc60683d99984c9eabb4e590637d1c751853ca3a8a96b3c951e4972a1c4d25649df49ffb3a296956fe347a@141.223.197.246:30303