Node1 (seednode)
model : Raspberry Pi3
CPU : ARM v7
geth : 1.8.11-stable [ARM v7 version]
Go : go1.10.3

Node2
model : Raspberry Pi3
CPU : ARM v7
geth : 1.8.11-stable [ARM v7 version]
Go : go1.10.3

Node3
model : Raspberry Pi3
CPU : ARM v7
geth : 1.8.11-stable [ARM v7 version]
Go : go1.10.3

Node4 (miner)
model : MSI NoteBook
CPU : Intel x64
geth : 1.8.11-stable [Linux x64 version]
Go : go1.10.3

1. 모든 노드에 Hdac이라는 디렉토리를 만든 후 cd 명령어를 통해 Hdac 디렉토리에 들어간다.

2.Hdac 디렉토리에 genesis.json 파일을 만들고 해당 파일을 모든 노드들이 갖도록 한다.

```
{
	"config":{
		"chainId" : 4816,
		"homesteadBlock" : 0,
		"eip155Block" : 0
	},
	"difficulty" : "400",
	"gasLimit" : "2100000",
	"alloc" : {
		"7b684d27167d208c66584ece7f09d8bc8f86ffff":{
			"balance": "10000000000000000000000000"
		}
	}
}
```

[genesis.json 내용]

3. 블록체인 데이터를 저장할 디렉토리인 data 디렉토리를 만든 후

4. 모든 노드에서 geth를 genesis.json으로 초기화하기 위한 `geth init genesis.json --datadir ./data` 명령어를 실행한다.

5. SeedNode로 사용될 node1번 라즈베리파이에서는 클라이언트 실행을 위해 `geth --datadir ./data --networkid 4816`를 실행한다.
[networkid 뒤에 나오는 숫자는 반드시 genesis.json의 chainId와 동일해야 한다.]

6. config에서 ChainID가 genesis.json에서 설정한 ChainID와 동일한지 확인하고 UDP listener up에서 enode 아이디 및 ipc를 확인한다.
[enode는 p2p network에서 노드를 식별하기 위한 주소이다]

7. 다음으로 Node1,2,3은 Blockchain Network에 참여하기 위해 `geth --datadir ./data --networkid [networkid] --rpc --rpcaddr "127.0.0.1" --rpccorsdomain "*" --rpcapi "admin,db,eth,net,web3,miner,personal" --bootnodes "enode://e7257beeb4e206aef5d64437680c9be748f542e6448ba509bcf33cb03f16eb0584e3de92896ffb87ab55e3e911b5c3280f3777ab79a099f02bb69705f7073124@[Seednode의 IP]:30303" console` 명령어를 실행한다.
[enode 부분은 이전에 실행시켰던 Seednode의 enode를 사용한다.]

8. raspberry pi에서 채굴은 out of memory에 의해 불가능 하므로 채굴을 위한 Node4에서 miner.start([thread]) 명령어를 통해 채굴을 실행한다.