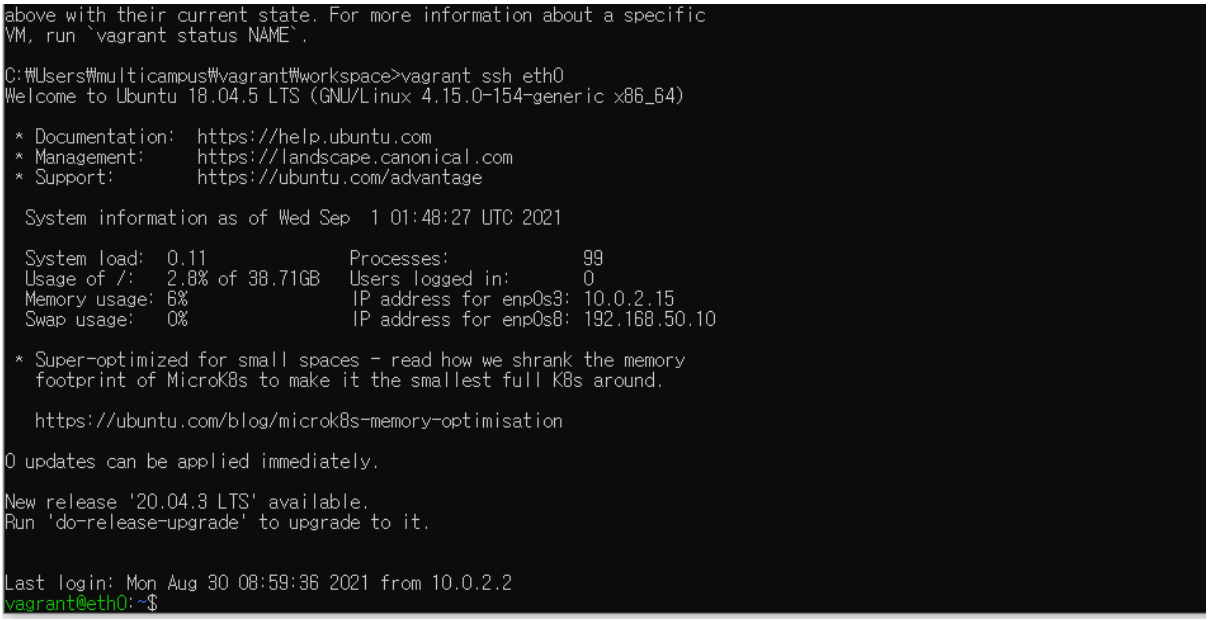
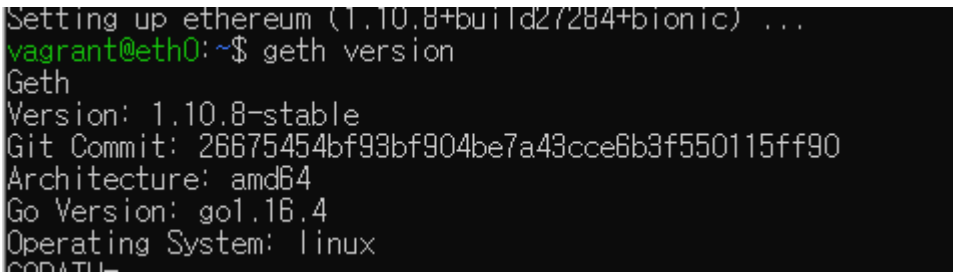
블록체인 명세서

- REQ 1-1) 가상 머신 구성

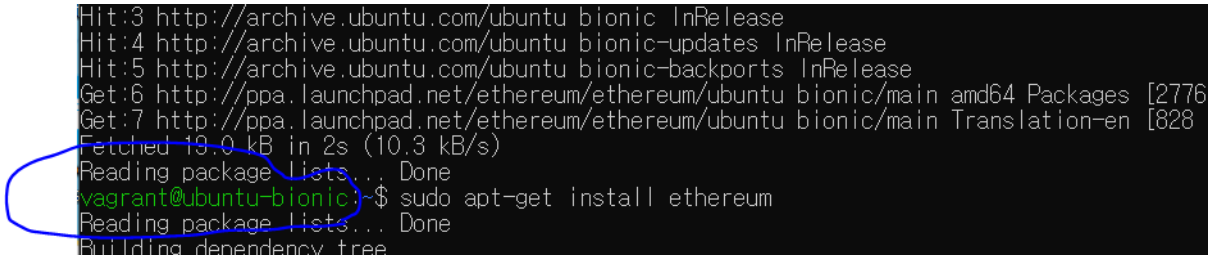


- REQ 1-2) 이더리움 eth0 노드 구성geth는 가상 머신 상에서 동작하도록 구축

REQ 1-3) 이더리움 eth1 노드 구성

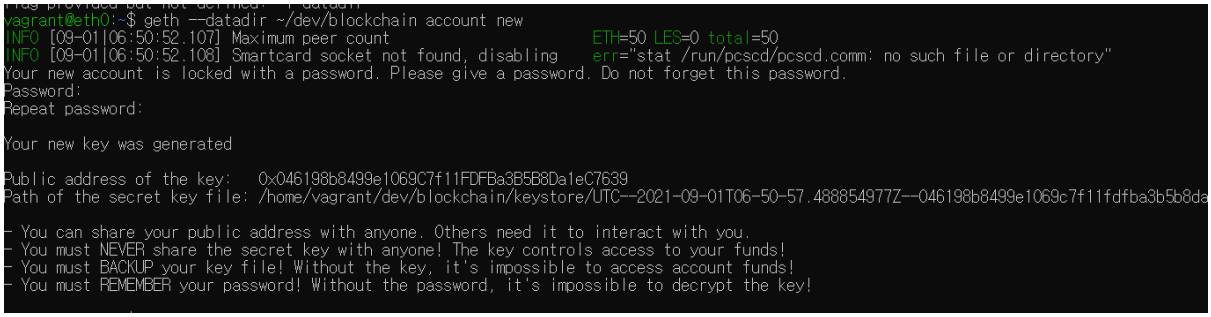


geth는 가상 머신 상에서 동작하도록 구축



- REQ 2-1) 계정 생성

--datadir 에 계정 생성 : geth 서버경로에 새 계정을 생성한다
geth –1-datadir ~/dev/blockchain account new

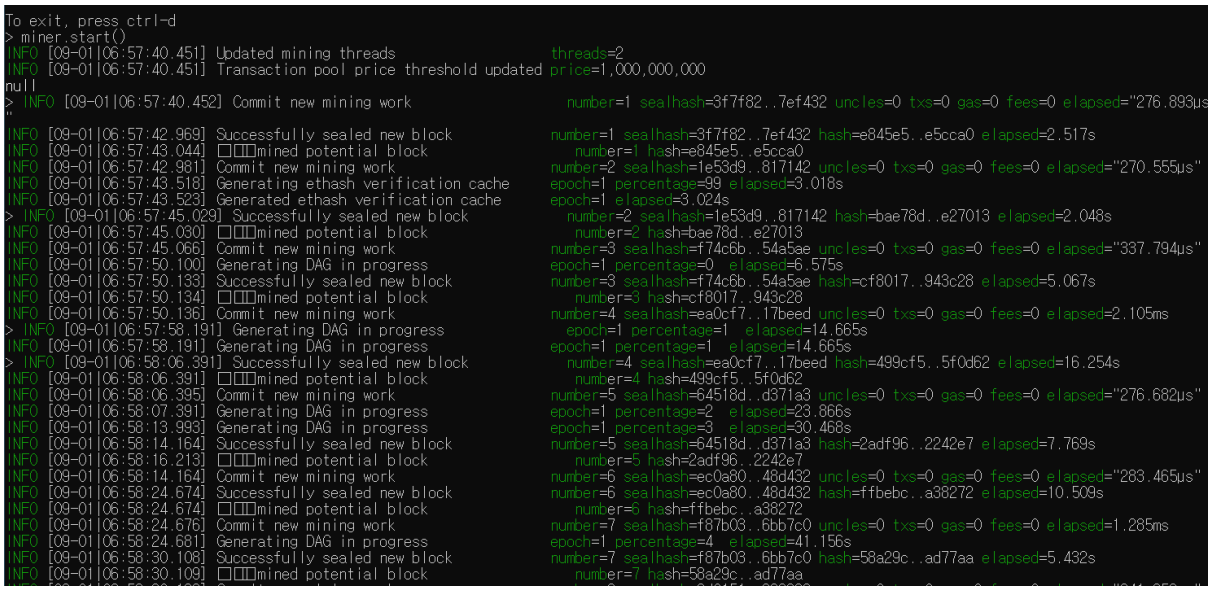


- REQ 2-2) genesis.json 파일로 설정해준다

블록 헤더 항목	설명
config	제네시스 블록의 설정값을 정의합니다.
chaind	블록체인을 식별하는 정수값을 입력합니다 "값이 비어있으면 안됩니다."
homesteadBlock	홈스테드를 적용하는 하드 포크 블록 번호를 뜻한다. 제네시스 블록은 0을 설정합니다.
eip155Block	이더리움 개선 제안(EIPs)의 155번 논의(한 번 이루어진 거래 정보를 이용해 재사용하는 리플레이 공격을 방지)를 적용한 하드 포크 블록 번호입니다.
eip158Block	이더리움 개선 제안(EIPs)의 158번 논의 (빈 계정 empty accounts) 를 어떻게 다룰 것인지에 관한 프로토콜 변경)를 적용한 하드 포크 블록 번호입니다 제네시스 블록은 0을 설정
nonce	mixHash와 함께 해당 블록에 충분한 양의 작업 증명 연산을 실행했음을 증명하는 값이다.
timestamp	블록체인에 저장한 시간입니다., 유닉스의 타임 스탬프 형식입니다.
parent Hash	부모 블록 (이전 블록) 헤더의 해시값입니다.
gasLimit	해당 블록에서 사용할 수 있는 가스의 최대 크기입니다.
difficulty	해당 생성 난이도를 뜻합니다. 이전 블록의 생성 난이도와 타임스탬프, 블록 번호를 이용해 계산합니다.
mix Hash	nonce와 함께 해당 블록에 충분한 양의 작업 증명 연산을 실행했음을 증명합니다. 증명하는 시간을 단축하려고 일종의 중간값인 Pre-validation을 함께 넣어뒀습니다. 가벼운 검증(Lightweight) 검증 때 활용합니다.
coinbase	블록을 생성하려고 채굴 했을 때 보상을 받는 계정 주소입니다.
alloc	제네시스 블록을 생성할 때 특정 계정에 미리 정해진 액수의 디어를 지급해 블록을 만들 수 있습니다.

파일안에 들어갈 각 요소들에 대한 설명

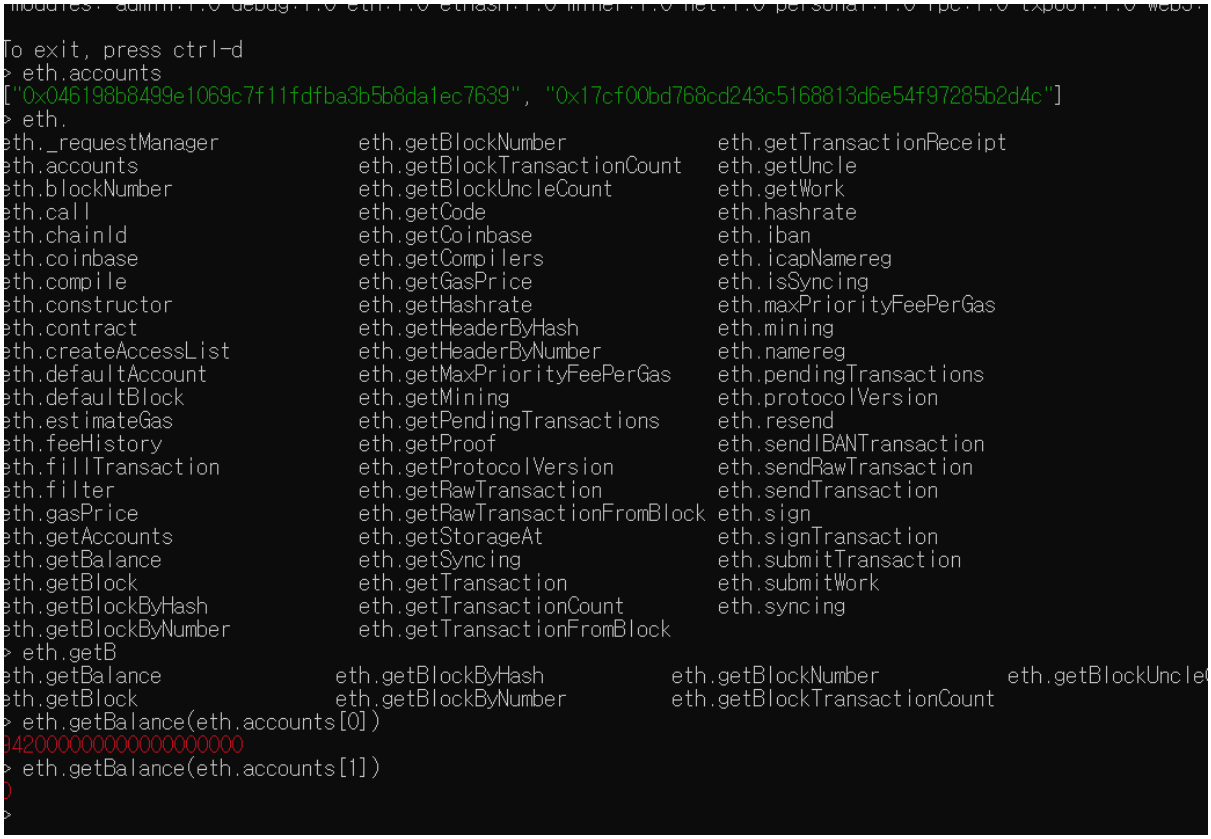
- REQ2-3) 채굴



채굴시작: miner.start()

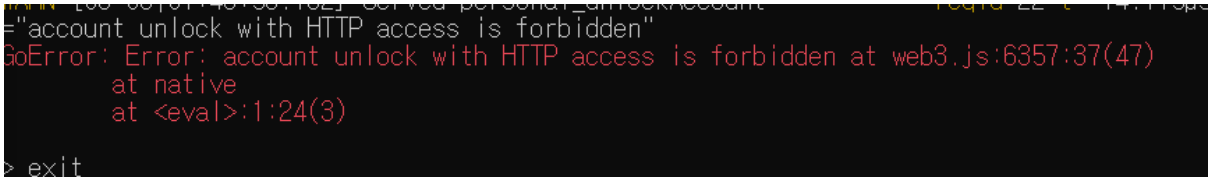
채굴종료: miner.stop()

명령어 : geth --networkid 921 --datadir ~/dev/blockchain --nodiscover --port 30303 --rpc --rpcport "8545" --maxpeers 2 --rpcaddr "0.0.0.0" --rpccorsdomain "*" --rpcapi "eth, net, web3, miner, debug, personal, rpc" console



- REQ 3-1) 트랜잭션 생성

personal.unlockAccount(from) 에러시



—allow-insecure-unlock 을 이용해 강제적으로 unlock시켜줌

send

```
true
> tx_hash = eth.sendTransaction(tx)
INFO [09-03|01:56:53.425] Setting new local account                address=0x046198b8499e1069
C7f11FDFBa3B5B8Da1eC7639
INFO [09-03|01:56:53.426] Submitted transaction                    hash=0xf615fe99bc423a7f12b
158218fe922a1e8fbddad83c3af36c10d9f7b015d3339 from=0x046198b8499e1069C7f11FDFBa3B5B8Da1eC7639
 nonce=0 recipient=0x17cF00bd768Cd243C5168813D6e54f97285B2D4c value=0
"0xf615fe99bc423a7f12b158218fe922a1e8fbddad83c3af36c10d9f7b015d3339"
>
```

- REQ 3-2) 트랜잭션 결과

트랜잭션 서명


```
{
 raw: "0xf86d018504a817c80083015f909417cf00bd768cd243c5168813d6e54f97285b2d4c88016345785d8a0
000808041a0c97af22f6551b65b66900c1bb381a5ccfedd1e0b24c8a99f6c97d7b06c6065eca062e369506e9b434709
4f952a4f26db2428a8fba48772e988b25ba03fe02a0e6",
 tx: {
   gas: "0x15f90",
   gasPrice: "0x4a817c800",
   hash: "0xe5feb1e45fbedc19c8952276b5649bd2fc2e5eee3a94eb11898783197bfe03c1",
   input: "0x",
   maxFeePerGas: null,
   maxPriorityFeePerGas: null,
   nonce: "0x1",
   r: "0xc97af22f6551b65b66900c1bb381a5ccfedd1e0b24c8a99f6c97d7b06c6065ec",
   s: "0x62e369506e9b43470914f952a4f26db2428a8fba48772e988b25ba03fe02a0e6",
   to: "0x17cf00bd768cd243c5168813d6e54f97285b2d4c",
   type: "0x0",
   v: "0x41",
   value: "0x16345785d8a0000"
 }
}
```


트랜잭션 결과


```
> eth.getTransaction(tx_hash)
{
  blockHash: null,
  blockNumber: null,
  from: "0x046198b8499e1069c7f11fdfba3b5b8da1ec7639",
  gas: 90000,
  gasPrice: 20000000000,
  hash: "0xe5feb1e45fbedc19c8952276b5649bd2fc2e5eee3a94eb11898783197bfe03c1",
  input: "0x",
  nonce: 1,
  r: "0xc97af22f6551b65b66900c1bb381a5ccfedd1e0b24c8a99f6c97d7b06c6065ec",
  s: "0x62e369506e9b43470914f952a4f26db2428a8fba48772e988b25ba03fe02a0e6",
  to: "0x17cf00bd768cd243c5168813d6e54f97285b2d4c",
  transactionIndex: null,
  type: "0x0",
  v: "0x41",
  value: 100000000000000000
}
```