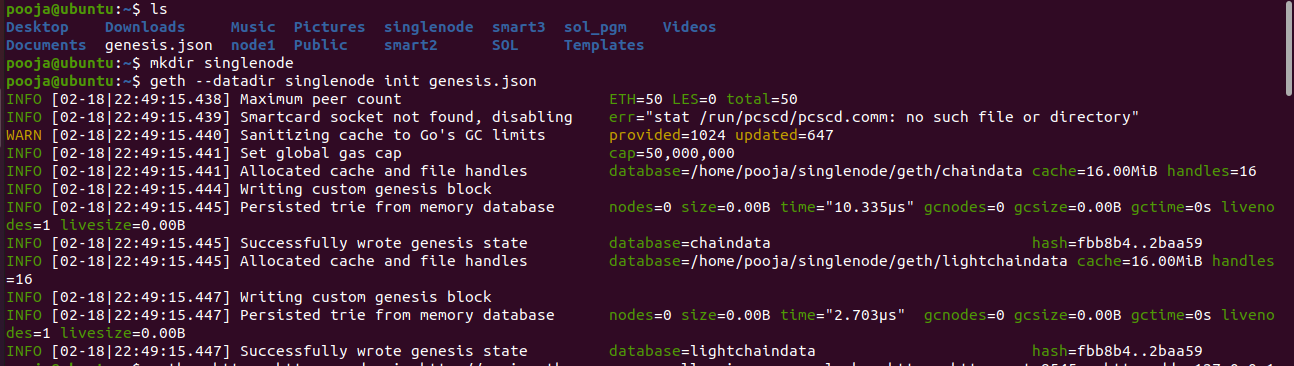
**Deploying Smart Contract using geth**

1. Create directory by name singlenode

$cd singlenode

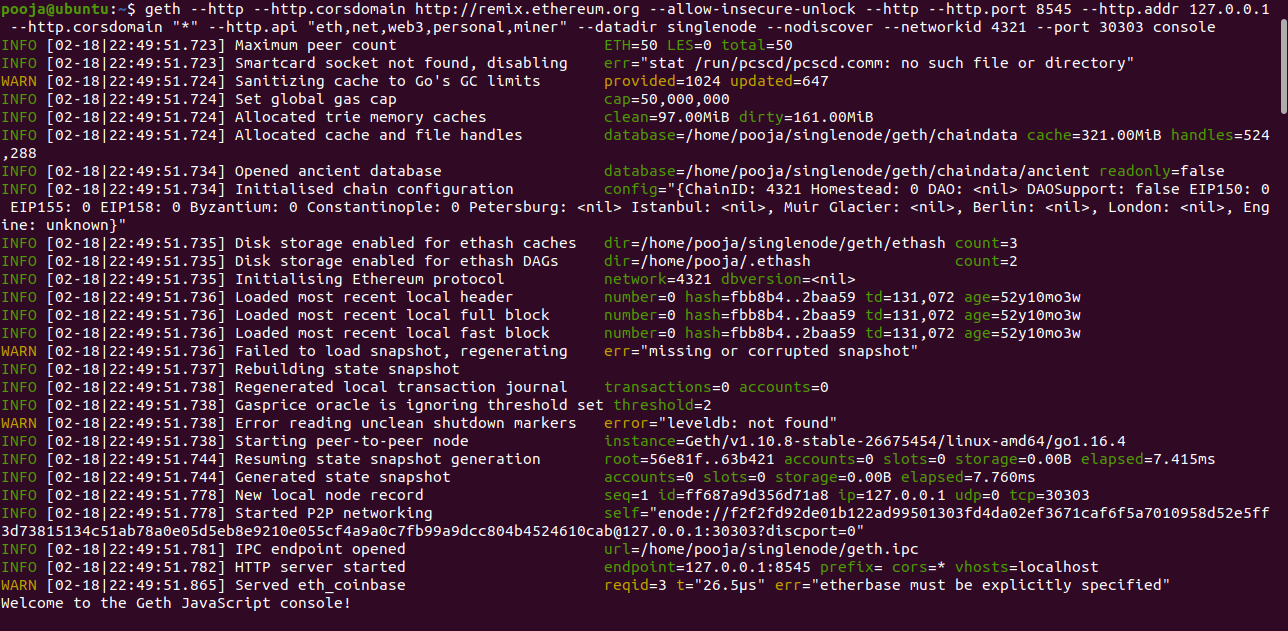
1. Copy the genesis file in the current folder and run the following command

$ geth --datadir singlenode init genesis.json



1. Attach the Geth using following command

geth --http --http.corsdomain http://remix.ethereum.org --allow-insecure-unlock --http --http.port 8545 --http.addr 127.0.0.1 --http.corsdomain "\*" --http.api "eth,net,web3,personal,miner" --datadir node1 --nodiscover --networkid 4321 --port 30303 console

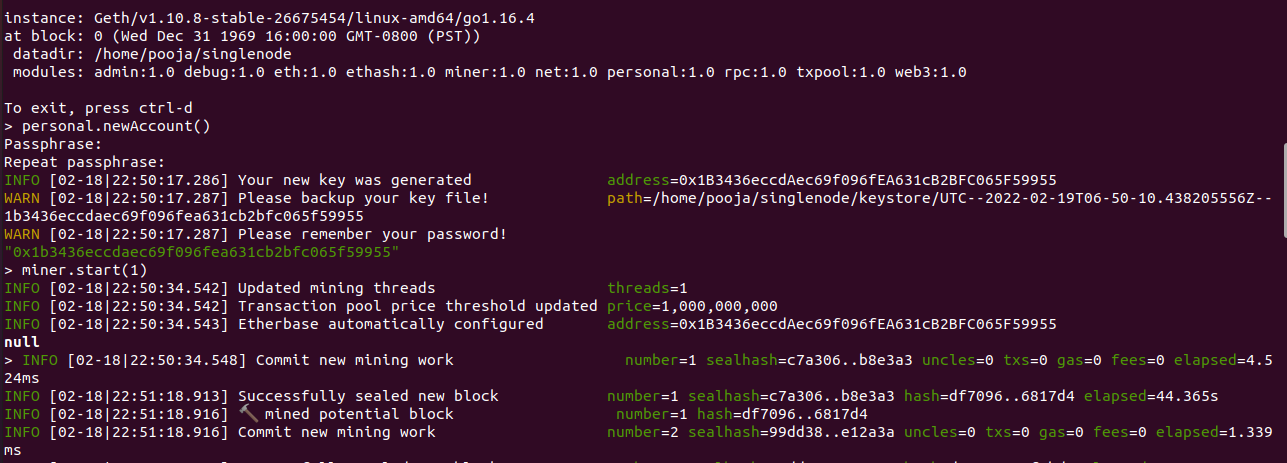


1. Create an account

$ personal.newAccount()

$ miner.setEtherbase(eth.accounts[0])

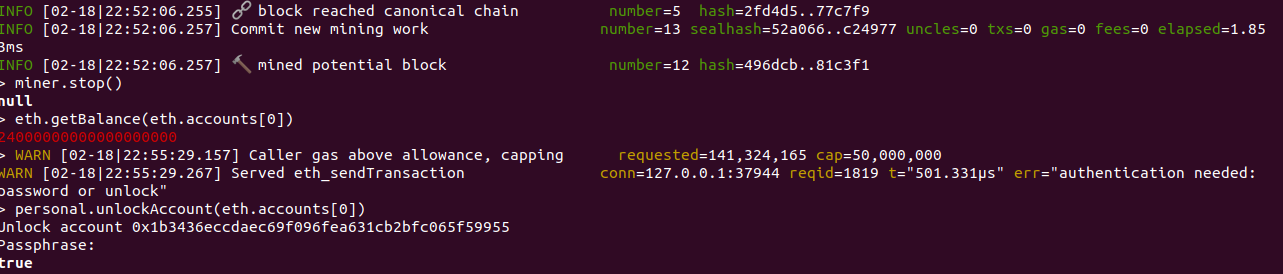
$ miner.start(1)



$ miner.stop()// it should be done only after you get potential block symbol

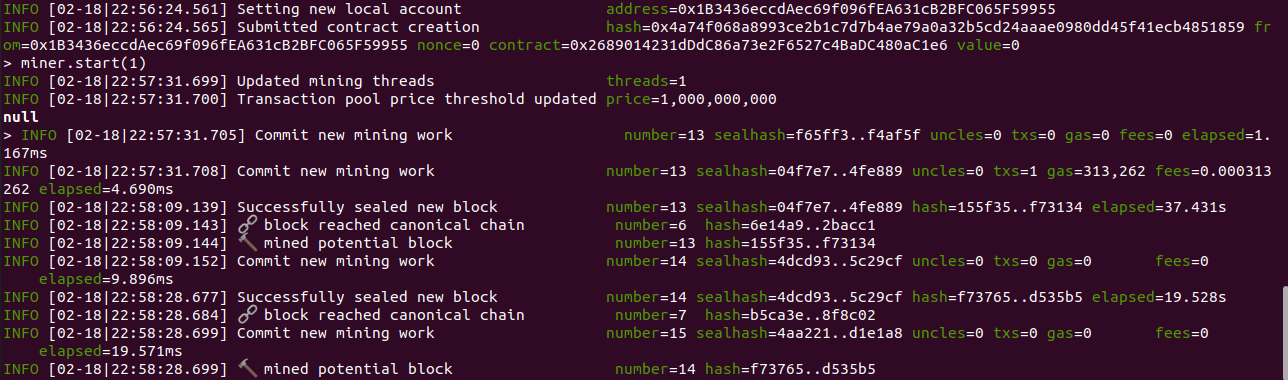
$ eth.getBalance(eth.accounts[0])

$ personal.unlockAccount(eth.accounts[0])

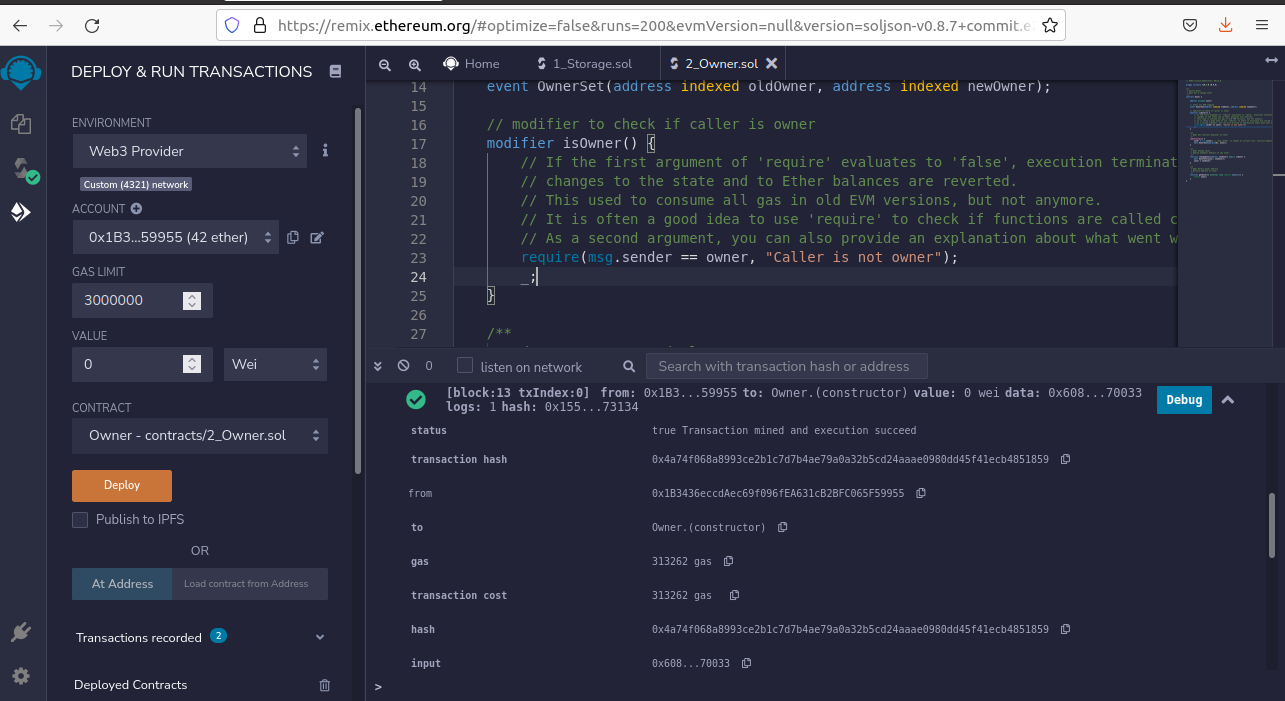


1. Compile the smart contract in remix, and deploy.( Using web3 provider environment in the web3provider give the RPC address of geth along with port number (here http://127.0.0.1:8545)).
2. After deploying the contract will be submitted for transaction, and copy the contract address from geth and paste it in js file.
3. then start the mining to make the transaction.

$ miner.start(1) // 2 to 3 blocks need to be generated or wait for green tick mark in the remix.



$ miner.stop()



1. After mining we can see the transaction receipt in remix IDE.
2. Copy following information in js file
3. Account address (obtained from geth)
4. ABI (obtained from remix)
5. Contract address (from geth)
6. Copy the function names need to be written appropriately in js file using send or call method

In order to run node js files web3 must be installed, to install run following command

$ npm install web3

1. To invoke the smart contracts methods. We use node js.

$ node new.js

