Deploying Smart Contract using geth

1. Create directory by name singlenode
   $cd singlenode
2. Copy the genesis file in the current folder and run the following command
   $ geth --datadir singlenode init genesis.json

```
pooja@ubuntu:~$ ls
Desktop    Downloads    Music  Pictures  singlenode  smart3  sol_pgm   Videos
Documents  genesis.json  node1  Public    smart2      SOL     Templates
pooja@ubuntu:~$ mkdir singlenode
pooja@ubuntu:~$ geth --datadir singlenode init genesis.json
INFO [02-18|22:49:15.438] Maximum peer count                       ETH=50 LES=0 total=50
INFO [02-18|22:49:15.439] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
WARN [02-18|22:49:15.440] Sanitizing cache to Go's GC limits       provided=1024 updated=647
INFO [02-18|22:49:15.441] Set global gas cap                       cap=50,000,000
INFO [02-18|22:49:15.441] Allocated cache and file handles         database=/home/pooja/singlenode/geth/chaindata cache=16.00MiB handles=16
INFO [02-18|22:49:15.444] Writing custom genesis block
INFO [02-18|22:49:15.445] Persisted trie from memory database      nodes=0 size=0.00B time="10.335µs" gcnodes=0 gcsize=0.00B gctime=0s liveno
des=1 livesize=0.00B
INFO [02-18|22:49:15.445] Successfully wrote genesis state         database=chaindata                          hash=fbb8b4..2baa59
INFO [02-18|22:49:15.445] Allocated cache and file handles         database=/home/pooja/singlenode/geth/lightchaindata cache=16.00MiB handles
=16
INFO [02-18|22:49:15.447] Writing custom genesis block
INFO [02-18|22:49:15.447] Persisted trie from memory database      nodes=0 size=0.00B time="2.703µs"  gcnodes=0 gcsize=0.00B gctime=0s liveno
des=1 livesize=0.00B
INFO [02-18|22:49:15.447] Successfully wrote genesis state         database=lightchaindata                     hash=fbb8b4..2baa59
```

3. 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
```

```
pooja@ubuntu:~$ 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 singlenode --nodiscover --networkid 4321 --port 30303 console
INFO [02-18|22:49:51.723] Maximum peer count                       ETH=50 LES=0 total=50
INFO [02-18|22:49:51.723] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
WARN [02-18|22:49:51.724] Sanitizing cache to Go's GC limits       provided=1024 updated=647
INFO [02-18|22:49:51.724] Set global gas cap                       cap=50,000,000
INFO [02-18|22:49:51.724] Allocated trie memory caches             clean=97.00MiB dirty=161.00MiB
INFO [02-18|22:49:51.724] Allocated cache and file handles         database=/home/pooja/singlenode/geth/chaindata cache=321.00MiB handles=524
,288
INFO [02-18|22:49:51.734] Opened ancient database                  database=/home/pooja/singlenode/geth/chaindata/ancient readonly=false
INFO [02-18|22:49:51.734] Initialised chain configuration          config="{ChainID: 4321 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: 0
 EIP155: 0 EIP158: 0 Byzantium: 0 Constantinople: 0 Petersburg: <nil> Istanbul: <nil>, Muir Glacier: <nil>, Berlin: <nil>, London: <nil>, Eng
ine: unknown}"
INFO [02-18|22:49:51.735] Disk storage enabled for ethash caches   dir=/home/pooja/singlenode/geth/ethash count=3
INFO [02-18|22:49:51.735] Disk storage enabled for ethash DAGs     dir=/home/pooja/.ethash                count=2
INFO [02-18|22:49:51.735] Initialising Ethereum protocol           network=4321 dbversion=<nil>
INFO [02-18|22:49:51.736] Loaded most recent local header          number=0 hash=fbb8b4..2baa59 td=131,072 age=52y10mo3w
INFO [02-18|22:49:51.736] Loaded most recent local full block      number=0 hash=fbb8b4..2baa59 td=131,072 age=52y10mo3w
INFO [02-18|22:49:51.736] Loaded most recent local fast block      number=0 hash=fbb8b4..2baa59 td=131,072 age=52y10mo3w
WARN [02-18|22:49:51.736] Failed to load snapshot, regenerating    err="missing or corrupted snapshot"
INFO [02-18|22:49:51.737] Rebuilding state snapshot
INFO [02-18|22:49:51.738] Regenerated local transaction journal    transactions=0 accounts=0
INFO [02-18|22:49:51.738] Gasprice oracle is ignoring threshold set threshold=2
WARN [02-18|22:49:51.738] Error reading unclean shutdown markers   error="leveldb: not found"
INFO [02-18|22:49:51.738] Starting peer-to-peer node               instance=Geth/v1.10.8-stable-26675454/linux-amd64/go1.16.4
INFO [02-18|22:49:51.744] Resuming state snapshot generation       root=56e81f..63b421 accounts=0 slots=0 storage=0.00B elapsed=7.415ms
INFO [02-18|22:49:51.744] Generated state snapshot                 accounts=0 slots=0 storage=0.00B elapsed=7.760ms
INFO [02-18|22:49:51.778] New local node record                    seq=1 id=ff687a9d356d71a8 ip=127.0.0.1 udp=0 tcp=30303
INFO [02-18|22:49:51.778] Started P2P networking                   self="enode://f2f2fd92de01b122ad99501303fd4da02ef3671caf6f5a7010958d52e5ff
3d73815134c51ab78a0e05d5eb8e9210e055cf4a9a0c7fb99a9dcc804b4524610cab@127.0.0.1:30303?discport=0"
INFO [02-18|22:49:51.781] IPC endpoint opened                      url=/home/pooja/singlenode/geth.ipc
INFO [02-18|22:49:51.782] HTTP server started                      endpoint=127.0.0.1:8545 prefix= cors=* vhosts=localhost
WARN [02-18|22:49:51.865] Served eth_coinbase                      reqid=3 t="26.5µs" err="etherbase must be explicitly specified"
Welcome to the Geth JavaScript console!
```

4. Create an account

```
$ personal.newAccount()
$ miner.setEtherbase(eth.accounts[0])
$ miner.start(1)
```

```
instance: Geth/v1.10.8-stable-26675454/linux-amd64/go1.16.4
at block: 0 (Wed Dec 31 1969 16:00:00 GMT-0800 (PST))
 datadir: /home/pooja/singlenode
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
> personal.newAccount()
Passphrase:
Repeat passphrase:
INFO [02-18|22:50:17.286] Your new key was generated               address=0x1B3436eccdAec69f096fEA631cB2BFC065F59955
WARN [02-18|22:50:17.287] Please backup your key file!             path=/home/pooja/singlenode/keystore/UTC--2022-02-19T06-50-10.438205556Z--
1b3436eccdaec69f096fea631cb2bfc065f59955
WARN [02-18|22:50:17.287] Please remember your password!
"0x1b3436eccdaec69f096fea631cb2bfc065f59955"
> miner.start(1)
INFO [02-18|22:50:34.542] Updated mining threads                   threads=1
INFO [02-18|22:50:34.542] Transaction pool price threshold updated price=1,000,000,000
INFO [02-18|22:50:34.543] Etherbase automatically configured       address=0x1B3436eccdAec69f096fEA631cB2BFC065F59955
null
> INFO [02-18|22:50:34.548] Commit new mining work                   number=1 sealhash=c7a306..b8e3a3 uncles=0 txs=0 gas=0 fees=0 elapsed=4.5
24ms
INFO [02-18|22:51:18.913] Successfully sealed new block            number=1 sealhash=c7a306..b8e3a3 hash=df7096..6817d4 elapsed=44.365s
INFO [02-18|22:51:18.916] 🔨 mined potential block                  number=1 hash=df7096..6817d4
INFO [02-18|22:51:18.916] Commit new mining work                   number=2 sealhash=99dd38..e12a3a uncles=0 txs=0 gas=0 fees=0 elapsed=1.339
ms
```

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

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

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

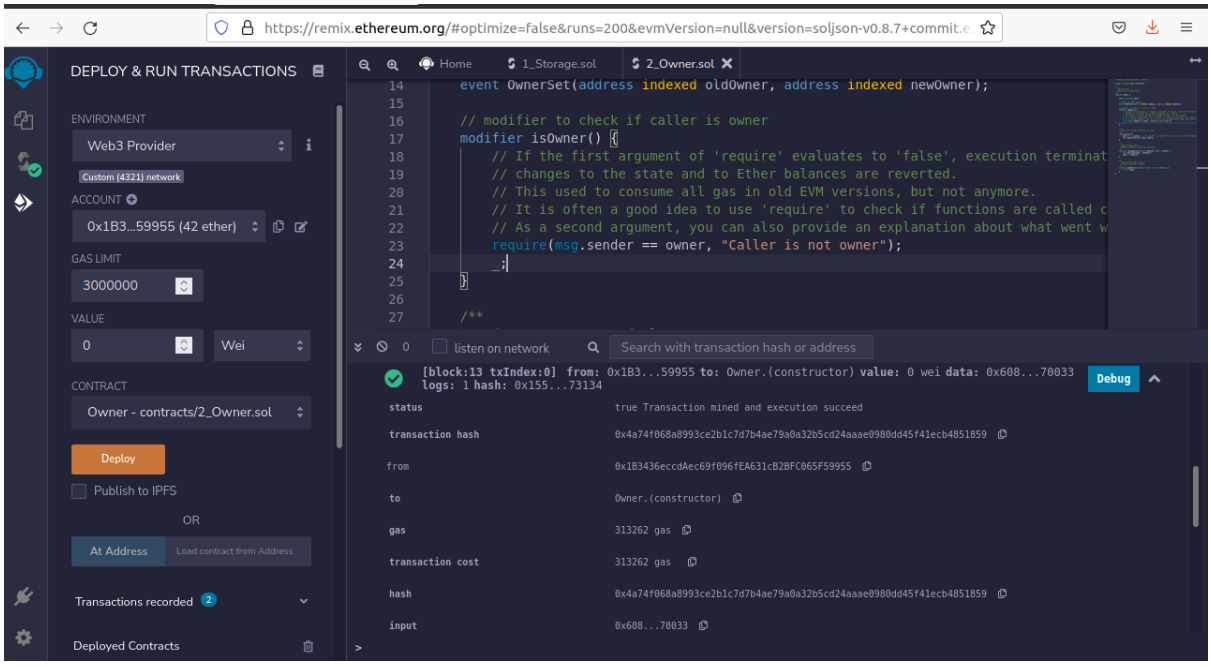
```
INFO [02-18|22:52:06.255] 🔗 block reached canonical chain          number=5  hash=2fd4d5..77c7f9
INFO [02-18|22:52:06.257] Commit new mining work                   number=13 sealhash=52a066..c24977 uncles=0 txs=0 gas=0 fees=0 elapsed=1.85
8ms
INFO [02-18|22:52:06.257] 🔨 mined potential block                  number=12 hash=496dcb..81c3f1
> miner.stop()
null
> eth.getBalance(eth.accounts[0])
400000000000000000
> WARN [02-18|22:55:29.157] Caller gas above allowance, capping      requested=141,324,165 cap=50,000,000
WARN [02-18|22:55:29.267] Served eth_sendTransaction               conn=127.0.0.1:37944 reqid=1819 t="501.331µs" err="authentication needed:
password or unlock"
> personal.unlockAccount(eth.accounts[0])
Unlock account 0x1b3436eccdaec69f096fea631cb2bfc065f59955
Passphrase:
true
```

5. 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)).
6. After deploying the contract will be submitted for transaction, and copy the contract address from geth and paste it in js file.
7. 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.

```
INFO [02-18|22:56:24.561] Setting new local account                address=0x1B3436eccdAec69f096fEA631cB2BFC065F59955
INFO [02-18|22:56:24.565] Submitted contract creation              hash=0x4a74f068a8993ce2b1c7d7b4ae79a0a32b5cd24aaae0980dd45f41ecb4851859 fr
om=0x1B3436eccdAec69f096fEA631cB2BFC065F59955 nonce=0 contract=0x2689014231dDdC86a73e2F6527c4BaDC480aC1e6 value=0
> miner.start(1)
INFO [02-18|22:57:31.699] Updated mining threads                   threads=1
INFO [02-18|22:57:31.700] Transaction pool price threshold updated price=1,000,000,000
null
> INFO [02-18|22:57:31.705] Commit new mining work                   number=13 sealhash=f65ff3..f4af5f uncles=0 txs=0 gas=0 fees=0 elapsed=1.
167ms
INFO [02-18|22:57:31.708] Commit new mining work                   number=13 sealhash=04f7e7..4fe889 uncles=0 txs=1 gas=313,262 fees=0.000313
262 elapsed=4.690ms
INFO [02-18|22:58:09.139] Successfully sealed new block            number=13 sealhash=04f7e7..4fe889 hash=155f35..f73134 elapsed=37.431s
INFO [02-18|22:58:09.143] 🔗 block reached canonical chain          number=6  hash=6e14a9..2bacc1
INFO [02-18|22:58:09.144] 🔨 mined potential block                  number=13 hash=155f35..f73134
INFO [02-18|22:58:09.152] Commit new mining work                   number=14 sealhash=4dcd93..5c29cf uncles=0 txs=0 gas=0       fees=0
    elapsed=9.896ms
INFO [02-18|22:58:28.677] Successfully sealed new block            number=14 sealhash=4dcd93..5c29cf hash=f73765..d535b5 elapsed=19.528s
INFO [02-18|22:58:28.684] 🔗 block reached canonical chain          number=7  hash=b5ca3e..8f8c02
INFO [02-18|22:58:28.699] Commit new mining work                   number=15 sealhash=4aa221..d1e1a8 uncles=0 txs=0 gas=0       fees=0
    elapsed=19.571ms
INFO [02-18|22:58:28.699] 🔨 mined potential block                  number=14 hash=f73765..d535b5
```

$ miner.stop()



8. After mining we can see the transaction receipt in remix IDE.

9. Copy following information in js file
   i. Account address (obtained from geth)
   ii. ABI (obtained from remix)
   iii. Contract address (from geth)
   iv. 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

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

$ node new.js

new.js *Untitled Document 1

```
const Web3 = require('web3');
var address = '0x1b3436eccdaec69f096fea631cb2bfc065f59955'    // change the account address
var web3 = new Web3('http://localhost:8545')
//console.log(web3);
//web3.eth.personal.unlockAccount(address,'123', 600)
// //.then(console.log('Account unlocked!'));
//  web3.eth.getBlockNumber()
//  .then(console.log);

var abi =[
        {
                "inputs": [],
                "stateMutability": "nonpayable",
                "type": "constructor"
        },
        {
                "anonymous": false,
                "inputs": [
                        {
                                "indexed": true,
                                "internalType": "address",
                                "name": "oldOwner",
                                "type": "address"
                        },
                        {
                                "indexed": true,
                                "internalType": "address",
                                "name": "newOwner",
                                "type": "address"
                        }
                ],
                "name": "OwnerSet",
                "type": "event"
        },
```

JavaScript Tab Width: 8 Ln 10, Col 1 INS

new.js × *Untitled Document 1 ×

```
                "type": "event"
        },
        {
                "inputs": [
                        {
                                "internalType": "address",
                                "name": "newOwner",
                                "type": "address"
                        }
                ],
                "name": "changeOwner",
                "outputs": [],
                "stateMutability": "nonpayable",
                "type": "function"
        },
        {
                "inputs": [],
                "name": "getOwner",
                "outputs": [
                        {
                                "internalType": "address",
                                "name": "",
                                "type": "address"
                        }
                ],
                "stateMutability": "view",
                "type": "function"
        }
]
var cont_addr = '0x2689014231dDdC86a73e2F6527c4BaDC480aC1e6'              // change the contract address
var MyContract = new web3.eth.Contract(abi, cont_addr);
//MyContract.methods.changeOwner("0xdb274ae93ed497a336781493153f974bdabeef0f").send({from: address}) //to send the data through contract
//.then(console.log)
MyContract.methods.getOwner().call({from: address}) //to get the data through contract(mining is not necessary)
.then(console.log)
```

JavaScript ▾ Tab Width: 8 ▾ Ln 16, Col 11 ▾ INS

```
pooja@ubuntu:~$ node new.js
{ blockHash:
   '0xaafbde60d0573c59ef43fb4b5ddd58d179e7a2eebd75b6dfa9f541fbfa8218b4',
  blockNumber: 21,
  contractAddress: null,
  cumulativeGasUsed: 30447,
  effectiveGasPrice: 1000000000,
  from: '0x1b3436eccdaec69f096fea631cb2bfc065f59955',
  gasUsed: 30447,
  logsBloom:
   '0x000000000000000000000000000000000000000104000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000880000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000004
000000000000000000000000000000000001000000000100000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000400000000000000000000000000000000000000000000000000000000002000000000000000000000000',
  status: true,
  to: '0x2689014231dddc86a73e2f6527c4badc480ac1e6',
  transactionHash:
   '0x41cd3d5e9717b15ba5f7db32050b7fc69db361ecb7a208817cff7e919b4780f4',
  transactionIndex: 0,
  type: '0x0',
  events:
   { OwnerSet:
      { address: '0x2689014231dDdC86a73e2F6527c4BaDC480aC1e6',
        blockNumber: 21,
        transactionHash:
         '0x41cd3d5e9717b15ba5f7db32050b7fc69db361ecb7a208817cff7e919b4780f4',
        transactionIndex: 0,
        blockHash:
         '0xaafbde60d0573c59ef43fb4b5ddd58d179e7a2eebd75b6dfa9f541fbfa8218b4',
        logIndex: 0,
        removed: false,
        id: 'log_6e37e9a9',
        returnValues: [Result],
        event: 'OwnerSet',
```

```
  events:
   { OwnerSet:
      { address: '0x2689014231dDdC86a73e2F6527c4BaDC480aC1e6',
        blockNumber: 21,
        transactionHash:
         '0x41cd3d5e9717b15ba5f7db32050b7fc69db361ecb7a208817cff7e919b4780f4',
        transactionIndex: 0,
        blockHash:
         '0xaafbde60d0573c59ef43fb4b5ddd58d179e7a2eebd75b6dfa9f541fbfa8218b4',
        logIndex: 0,
        removed: false,
        id: 'log_6e37e9a9',
        returnValues: [Result],
        event: 'OwnerSet',
        signature:
         '0x342827c97908e5e2f71151c08502a66d44b6f758e3ac2f1de95f02eb95f0a735',
        raw: [Object] } } }
```