Setting up Private Blockchain using geth:

1. You first need to create a new folder using the following command:

mkdir ethereumprivatenetwork

Then you need to move to this path using the command:

cd ethereumprivatenetwork/

After this, you need to open the folder which you have created in any text editor of your choice. Inside this folder create a file named Init.json​.Inside of this file, you will define the different parameters for genesis block of a private network.

1. Run this

geth --datadir ./datadir init . /Init.json

geth --datadir ./datadir/ --networkid 2019 console

Use similar commands to create another peer just give data dir and port.

geth --datadir ./datadir2 init ./genesis.json

geth --datadir ./datadir2 --networkid 2019 --port 30306 console

1. Use admin.addPeer with ethID to connect

Use admin.peers to check connected peers

> admin.peers

[{

caps: ["eth/63", "eth/64", "eth/65"],

enode: "enode://ee5f0fc9a0641882a9b0e7222105106654196e39e5fc1fb892d2eab19fc16577b11ea5758e357692827ef4f488f10d7193a74c41f6da060595f311bcffc10ff3@135.181.216.90:30306",

id: "e47ffd8a013d89525c88b98251bfb9ede0a77e69789db4e177db0049e719f5fd",

name: "Geth/v1.0.7-f1d92bef/linux-amd64/go1.15.5",

network: {

inbound: **false**,

localAddress: "192.168.29.132:49593",

remoteAddress: "135.181.216.90:30306",

static: **false**,

trusted: **false**

},

protocols: {

eth: "handshake"

}

}]

> INFO [06-10|16:51:37.110] Looking for peers peercount=0 tried=92 static=0

Setting up truffle:

dsolanki@dsolanki-a01 Submission % truffle init

Starting init...

================

> Copying project files to /Users/dsolanki/Desktop/PGDiploma/ethereum-1/Submission

Init successful, sweet!

dsolanki@dsolanki-a01 Submission %

dsolanki@dsolanki-a01 Submission % truffle compile

Compiling your contracts...

===========================

✔ Fetching solc version list from solc-bin. Attempt #1

✔ Downloading compiler. Attempt #1.

> Compiling ./contracts/KYC.sol

> Compiling ./contracts/Migrations.sol

> Artifacts written to /Users/dsolanki/Desktop/PGDiploma/ethereum-1/Submission/build/contracts

> Compiled successfully using:

- solc: 0.5.16+commit.9c3226ce.Emscripten.clang

dsolanki@dsolanki-a01 Submission %

Start the eth private network:

dsolanki@dsolanki-a01 ethereumprivatenetwork % geth --datadir ./datadir --networkid 2019 –rpc --rpcport 30303 --allow-insecure-unlock console

instance: Geth/v1.10.0-stable/darwin-amd64/go1.16

coinbase: 0xb36bd452684f49ca310b5b83957011d4c01e3871

at block: 8 (Thu Jun 10 2021 16:37:24 GMT+0530 (IST))

datadir: /Users/dsolanki/Desktop/PGDiploma/ethereumprivatenetwork/datadir

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

> ERROR[06-10|17:38:51.964] Snapshot extension registration failed peer=cb5b9771 err="peer connected on snap without compatible eth support"

INFO [06-10|17:38:56.637] New local node record seq=4 id=0affb349471f3499 ip=49.36.47.61 udp=30304 tcp=30303

INFO [06-10|17:39:00.404] Looking for peers peercount=1 tried=110 static=0

ERROR[06-10|17:39:09.693] Snapshot extension registration failed peer=9a9cbe96 err="peer connected on snap without compatible eth support"

INFO [06-10|17:39:10.415] Looking for peers peercount=0 tried=113 static=0

INFO [06-10|17:39:21.645] Looking for peers peercount=0 tried=33 static=0

INFO [06-10|17:39:31.945] Looking for peers peercount=0 tried=91 static=0

> eth.accounts

["0xb36bd452684f49ca310b5b83957011d4c01e3871", "0x4808870b2720703d966c94866985f7b544178990"]

> INFO [06-10|17:39:52.338] Looking for peers peercount=0 tried=83 static=0

> INFO [06-10|17:40:05.949] Looking for peers peercount=0 tried=60 static=0

ERROR[06-10|17:40:06.649] Snapshot extension registration failed peer=89027a8b err="peer connected on snap without compatible eth support"

> personal.unlockAccountINFO [06-10|17:40:16.346] Looking for peers peercount=0 tried=94 static=0

> personal.unlockAccount(INFO [06-10|17:40:26.550] Looking for peers peercount=0 tried=89 static=0

> personal.unlockAccount("0xb36bd452684f49ca310b5b83957011d4c01e3871",'deepika',0)INFO [06-10|17:40:37.577] Looking for peers peercount=0 tried=62 static=0

**true**

> INFO [06-10|17:40:47.577] Looking for peers

> miner.start()

Deploying smart contract over it:

Start the geth network

Then add accounts there

And unlock accounts so as to setup the smart contract

Do truffle migrate on the geth network

truffle migrate --network geth

dsolanki@dsolanki-a01 Submission % truffle migrate --network geth

Compiling your contracts...

===========================

> Everything is up to date, there is nothing to compile.

Starting migrations...

======================

> Network name: 'geth'

> Network id: 2019

> Block gas limit: 8000000 (0x7a1200)

1\_initial\_migration.js

======================

Deploying 'Migrations'

----------------------

> transaction hash: 0x33864973d52322922d973969d164919a20a45aba03d6662847375a0fe8f9b7c6

> Blocks: 0 Seconds: 0

> contract address: 0x261fA8e8cA9bBAbfB2D6f020884C872Ad3c8798A

> block number: 929

> block timestamp: 1623328001

> account: 0xB36bd452684f49Ca310B5B83957011d4C01e3871

> balance: 4650

> gas used: 221555 (0x36173)

> gas price: 20 gwei

> value sent: 0 ETH

> total cost: 0.0044311 ETH

> Saving migration to chain.

> Saving artifacts

-------------------------------------

> Total cost: 0.0044311 ETH

2\_KYC\_migration.js

==================

Deploying 'KYC'

---------------

> transaction hash: 0xa0cfff86a27292a06de1215253f35978b229b85662719814c4c0fcbb801109db

> Blocks: 0 Seconds: 0

> contract address: 0x51E2C2ebD92Fa92Eb44c1cAF530d235faBBB7043

> block number: 935

> block timestamp: 1623328007

> account: 0xB36bd452684f49Ca310B5B83957011d4C01e3871

> balance: 4675

> gas used: 3632598 (0x376dd6)

> gas price: 20 gwei

> value sent: 0 ETH

> total cost: 0.07265196 ETH

> Saving migration to chain.

> Saving artifacts

-------------------------------------

> Total cost: 0.07265196 ETH

Summary

=======

> Total deployments: 2

> Final cost: 0.07708306 ETH

dsolanki@dsolanki-a01 Submission % truffle console --network geth

truffle(geth)>

undefined

truffle(geth)> let kyc = await KYC.deployed()

undefined

truffle(geth)> kyc.

kyc.\_\_defineGetter\_\_ kyc.\_\_defineSetter\_\_ kyc.\_\_lookupGetter\_\_ kyc.\_\_lookupSetter\_\_

kyc.\_\_proto\_\_ kyc.hasOwnProperty kyc.isPrototypeOf kyc.propertyIsEnumerable

kyc.toLocaleString kyc.toString kyc.valueOf

kyc.abi kyc.addBank kyc.addCustomer kyc.addRequest

kyc.address kyc.allEvents kyc.banBank kyc.constructor

kyc.contract kyc.downvote kyc.getBankComplaints kyc.getPastEvents

kyc.methods kyc.modifyBank kyc.modifyCustomer kyc.removeBank

kyc.removeRequest kyc.reportBank kyc.send kyc.sendTransaction

kyc.transactionHash kyc.upVote kyc.viewBankDetails kyc.viewCustomer

truffle(geth)> kyc.addCustomer("abc","abc\_data")

{

tx: '0x59865846914e9fc12012092611ba7c9af3cc0b0acc353ca3110bef957aea251a',

receipt: {

blockHash: '0x47696c3a257aaa7b617bff1da7138b2f8cde50870bd45bd44c38d9f3be6befa0',

blockNumber: 1232,

contractAddress: **null**,

cumulativeGasUsed: 6721975,

from: '0xb36bd452684f49ca310b5b83957011d4c01e3871',

gasUsed: 6721975,

logs: [],

logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000',

root: '0x41c2df70bf1220dec8f20b6561b824d2c7940c1e713793fd0beae2619a8bbb22',

to: '0x51e2c2ebd92fa92eb44c1caf530d235fabbb7043',

transactionHash: '0x59865846914e9fc12012092611ba7c9af3cc0b0acc353ca3110bef957aea251a',

transactionIndex: 0,

type: '0x0',

rawLogs: []

},

logs: []

}

truffle(geth)>