Private Network - What and Why?

Blockchain within your organization
Development environment as close to production(main net) env
No public access

Experiments

Geth Client

Command line interface for running full Ethereum node.
Implemented in Go language.

What you can do with it?

Mine real Ether
Transfer funds
Create and deploy contracts etc.

Geth Installation

Windows

Go to this, download setup file and install.

Linux

```
sudo apt-get install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install ethereum
```

Check the installation using command *geth version*

Running Geth

Types of Sync that Geth uses :

- **Fast Sync :** Download only the block headers not the content. It will not process each and every transaction again. Less secure.
- **Full Sync :** Download the block headers as well as the content of the blocks.It will process each and every transaction again to make sure that everything is securely baked into the block. More secure but lot of redundancy.
- **Light Sync :** Download the latest 1024 Headers only. Is very fast to occur.

Note:

1. Is we run command **geth** without any parameter for the first time, it will run in the **Fast Sync** mode. And If you interrupt the processing by pressing ctrl+c key, and re run the geth command again, it will now run in to **Full Sync** which will take much time.
2. To re start with Fast Sync mode, you need to remove / delete all the files that it has downloaded so far from the physical folder. I.e. datadir *C:\\Users\\Development\\AppData\\Roaming\\Ethereum\\geth\\chaindata*
3. If you use geth command without any parameter, then It will not open any **RPC** endpoint for you, rather by default it will open **IPC** endpoint only. Which means that if you opens a browser and try to connect to Go-Ethereum, you will not be able to do so, since no RPC endpoint is opened. I.e. you can not connect with HTTP. How ever you can connect with other the Node using Geth client.

Understanding genesis.json

```
{    "config": {           "chainId": 987,
"homesteadBlock": 0,        "eip155Block": 0,
"eip158Block": 0    },    "difficulty": "0x400",    "gasLimit":
"0x8000000",  //set this really high for testing    "alloc": {}}}
```

- **chainId** can’t be set to 1,2 and 3, which are reserved for Main Net. you can use anything else.
- 1 is reserved for Main Net. 2 and 3 are reserved for Test Net.
- **Homestead** : a latest ethereum release. Last one was Frontier.The value 0 means that you are using this release.
- **Eip** : stands for **Ethereum improvement protocol.** where developers propose ideas on how to improve Ethereum and contribute to this project**.** We need both the improvements start from block 0 in our network.
- **Difficulty :** lower it will be, faster the mining will be. If it is **higher, miners have to do to lot of computation to solve the problem or mine your blocks.** Now If you realise that minings taking too long, then setup the new ethereum private network and set it lower.
- **GasLimit :** the limit of gas cost per block . Amount of Gas limited on your blockchain to execute a smart contract. Current Main Net uses, gas limit 6 millions right now. - **Higher the Gas, more complex your contract can get.**
- **Alloc :** with this, you can pre-allocate ethers to accounts, if you want to.

Setting up private network

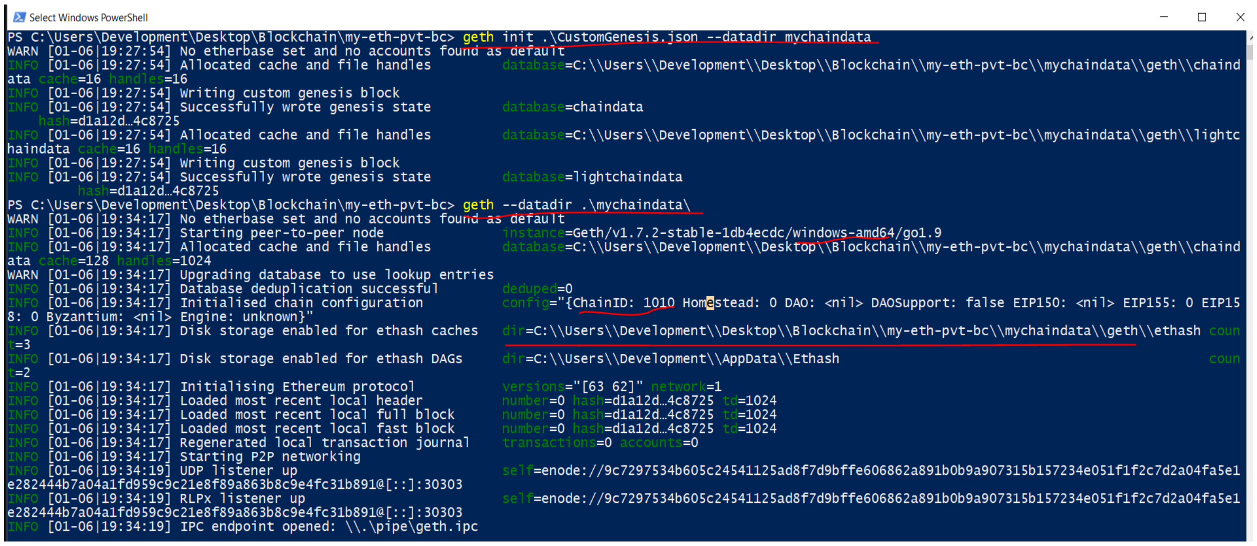
Setting up Node 1

- Create a new folder say, **my-eth-pvt-bc**
- Put **CustomGenesis.json** file into this folder.
- Open command line or powershell, and type command

  geth init .\CustomGenesis.json --datadir mychaindata

- It will initializes the genesis node and will create a data directory folder, *'mychaindata'*.
- Now again start our private geth with IPC

  geth --datadir .\mychaindata



```
PS C:\Users\Development\Desktop\Blockchain\my-eth-pvt-bc> geth init .\CustomGenesis.json --datadir mychaindata
WARN [01-06|19:27:54] No etherbase set and no accounts found as default
INFO [01-06|19:27:54] Allocated cache and file handles         database=C:\\Users\\Development\\Desktop\\Blockchain\\my-eth-pvt-bc\\mychaindata\\geth\\chaind
ata cache=16 handles=16
INFO [01-06|19:27:54] Writing custom genesis block
INFO [01-06|19:27:54] Successfully wrote genesis state         database=chaindata
    hash=d1a12d…4c8725
INFO [01-06|19:27:54] Allocated cache and file handles         database=C:\\Users\\Development\\Desktop\\Blockchain\\my-eth-pvt-bc\\mychaindata\\geth\\lightc
haindata cache=16 handles=16
INFO [01-06|19:27:54] Writing custom genesis block
INFO [01-06|19:27:54] Successfully wrote genesis state         database=lightchaindata
         hash=d1a12d…4c8725
PS C:\Users\Development\Desktop\Blockchain\my-eth-pvt-bc> geth --datadir .\mychaindata\
WARN [01-06|19:34:17] No etherbase set and no accounts found as default
INFO [01-06|19:34:17] Starting peer-to-peer node               instance=Geth/v1.7.2-stable-1db4ecdc/windows-amd64/go1.9
INFO [01-06|19:34:17] Allocated cache and file handles         database=C:\\Users\\Development\\Desktop\\Blockchain\\my-eth-pvt-bc\\mychaindata\\geth\\chaind
ata cache=128 handles=1024
WARN [01-06|19:34:17] Upgrading database to use lookup entries
INFO [01-06|19:34:17] Database deduplication successful        deduped=0
INFO [01-06|19:34:17] Initialised chain configuration          config="{ChainID: 1010 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: <nil> EIP155: 0 EIP15
8: 0 Byzantium: <nil> Engine: unknown}"
INFO [01-06|19:34:17] Disk storage enabled for ethash caches   dir=C:\\Users\\Development\\Desktop\\Blockchain\\my-eth-pvt-bc\\mychaindata\\geth\\ethash coun
t=3
INFO [01-06|19:34:17] Disk storage enabled for ethash DAGs     dir=C:\\Users\\Development\\AppData\\Ethash                                                   coun
t=2
INFO [01-06|19:34:17] Initialising Ethereum protocol           versions="[63 62]" network=1
INFO [01-06|19:34:17] Loaded most recent local header          number=0 hash=d1a12d…4c8725 td=1024
INFO [01-06|19:34:17] Loaded most recent local full block      number=0 hash=d1a12d…4c8725 td=1024
INFO [01-06|19:34:17] Loaded most recent local fast block      number=0 hash=d1a12d…4c8725 td=1024
INFO [01-06|19:34:17] Regenerated local transaction journal    transactions=0 accounts=0
INFO [01-06|19:34:17] Starting P2P networking
INFO [01-06|19:34:19] UDP listener up                          self=enode://9c7297534b605c24541125ad8f7d9bffe606862a891b0b9a907315b157234e051f1f2c7d2a04fa5e1
e282444b7a04a1fd959c9c21e8f89a863b8c9e4fc31b891@[::]:30303
INFO [01-06|19:34:19] RLPx listener up                         self=enode://9c7297534b605c24541125ad8f7d9bffe606862a891b0b9a907315b157234e051f1f2c7d2a04fa5e1
e282444b7a04a1fd959c9c21e8f89a863b8c9e4fc31b891@[::]:30303
INFO [01-06|19:34:19] IPC endpoint opened: \\.\pipe\geth.ipc
```

- Notice that the Geth does not start with the Block Synchronization, because you are on your own network. But sometimes it may start syncing with the Node on another network with the same chain id , i.e 1010 in our case. It happens because Geth is in **discovery mode**.
- Hence we need to start Geth with *--nodiscover* flag.

  geth --datadir .\mychaindata\ --nodiscover

Keep this node running as it is.

Setting up Node 2

- Create a new folder node02 and change directory
- Copy the same *CustomGenesis.Json* file and paste it in here.
- Now initialize the new node by providing new datadirectory
  - `geth init .\CustomGenesis.json --datadir datadir02`
- Start the geth using following command as we did for node 1
  - `geth --datadir .\datadir02\ --nodiscover`
- But this time error will come, saying '*Fatal: Error starting protocol stack: listen udp :30303: bind: address already in use*' It happened because by default geth is runs on port number 30303 and since we started node 01 already, so that port is occupied and hence we need to start nodes using different ports.
- To provide port while starting Geth, you can use *--port* flag. Stop the geth in node 01 by pressing ctrl+c and restart it using following command
  - `geth --datadir data01/ --nodiscover --port 30303`
- Then start node 02 using following command
  - *`geth --datadir data02/ --nodiscover --port 30304`*

Connecting nodes to the blockchain

As of now both the nodes are running on their own, they are not connected with to one blockchain, so they have not formed a network.

This time we will use --networkid flag . Provide network id same as chaind id you provided in genesis.

- Stop the first node again and restart it using following command
  - `geth --datadir data01/ --nodiscover --port 30303 --networkid 1010`
- And then node 02 using command
  - `geth --datadir data02/ --nodiscover --port 30304 --networkid 1010`

JSON RPC

https://github.com/ethereum/wiki/wiki/JSON-RPC

To talk to an ethereum node from inside a JavaScript application use the web3.js library, which gives a convenient interface for the RPC methods. JSON-RPC is a stateless, light-weight remote procedure call (RPC) protocol.

- You can start the HTTP JSON-RPC with the `--rpc` flag `geth --rpc`

- Change the default port (8545) and listing address (localhost) with:

  `geth --rpc --rpcaddr <ip> --rpcport <portnumber>`

- If accessing the RPC from a browser, CORS will need to be enabled with the appropriate domain set. Otherwise, JavaScript calls are limit by the same-origin policy and requests will fail: `geth --rpc --rpccorsdomain 'http://127.0.0.1:3000'`

Let's use the `--rpc` flag to run our nodes now

- Stop the first node again and restart it using following command
  - `geth --datadir data01/ --nodiscover --port 30303 --networkid 1010 --rpc --rpcaddr 127.0.0.1 --rpcport 8545 --rpccorsdomain`
- And then node 02 using command
  - `geth --datadir data02/ --nodiscover --port 30304 --networkid 1010 --rpc --rpcaddr 127.0.0.1 --rpcport 8546 --rpccorsdomain`

Attaching to Geth console

Once you have geth client running, you can attach a new console to the client using command

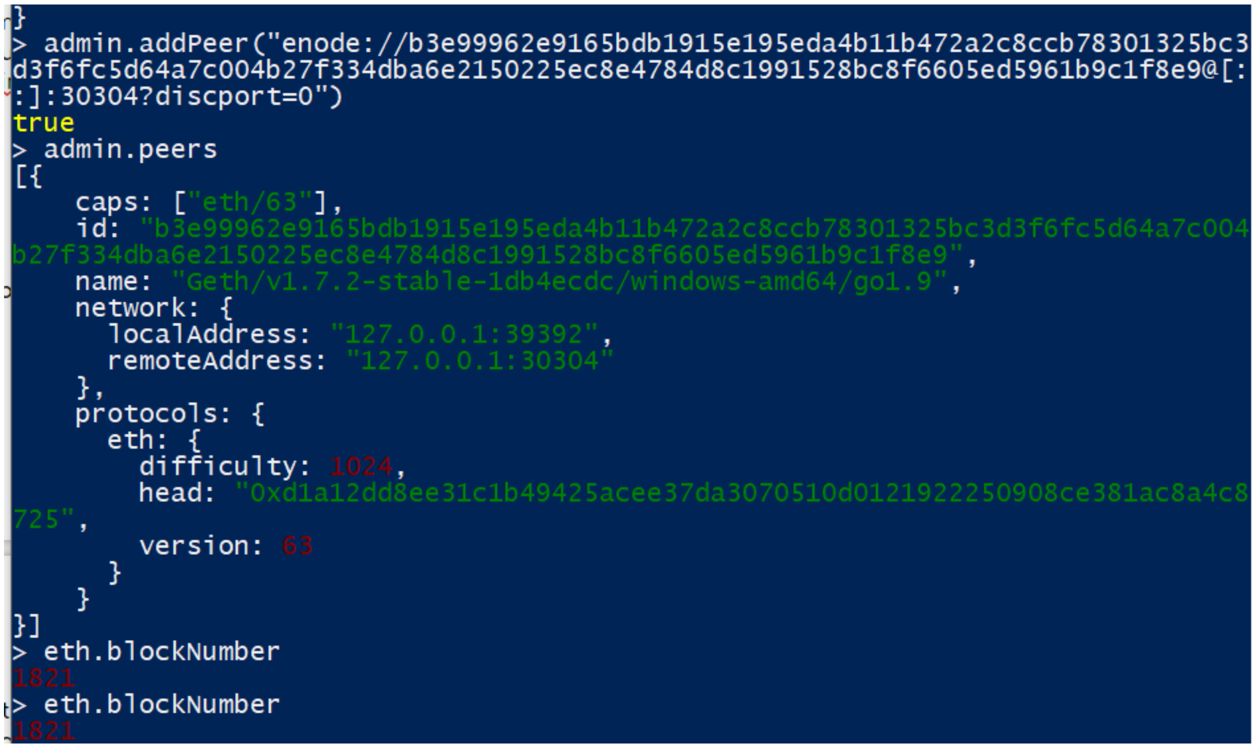
`geth attach` : for window user
geth attach <ipc file path> : for linux user

Ipc file path is present in the datadir - only for linux user

Connecting both nodes as peers

- Now that you have 2 nodes running into same network, you need to make them peers.
- To make them peers, copy the **enode value** from the second node (after attaching a new console to it) and go to the console of first node and use **admin.addPeer(enode value of 2nd node)** command to add them peer. See the screenshots below.
- Go to node 02 console to copy the enode value: and type *admin.nodeInfo* to get the enode value.
  - "enode://a2d43ebf1dcb2b1484b96d04a07e4eb744b849a9947e97b63e2f2152c8e61f004d7c98f9529f52ecd735210322e20431a04665316d1af3156b32c82fb9cc01a6@[::]:30304?discport=0"
- Now go to node 01 console and use admin.addPeer(enode value of 2nd node) to add as peer

The returned address is public address,not the private address. The private address is stored in Keystore.

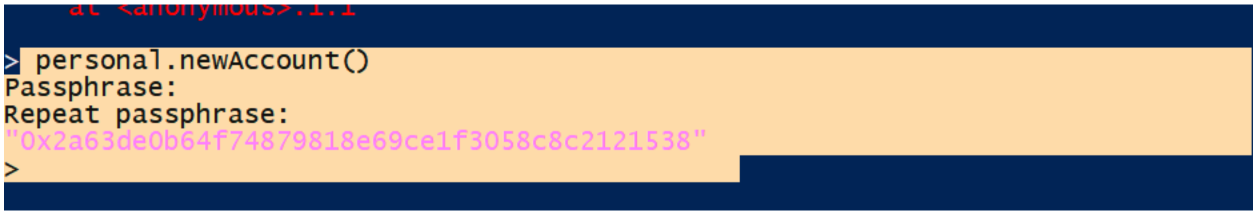


```
}
> admin.addPeer("enode://b3e99962e9165bdb1915e195eda4b11b472a2c8ccb78301325bc3
d3f6fc5d64a7c004b27f334dba6e2150225ec8e4784d8c1991528bc8f6605ed5961b9c1f8e9@[:
:]:30304?discport=0")
true
> admin.peers
[{
    caps: ["eth/63"],
    id: "b3e99962e9165bdb1915e195eda4b11b472a2c8ccb78301325bc3d3f6fc5d64a7c004
b27f334dba6e2150225ec8e4784d8c1991528bc8f6605ed5961b9c1f8e9",
    name: "Geth/v1.7.2-stable-1db4ecdc/windows-amd64/go1.9",
    network: {
      localAddress: "127.0.0.1:39392",
      remoteAddress: "127.0.0.1:30304"
    },
    protocols: {
      eth: {
        difficulty: 1024,
        head: "0xd1a12dd8ee31c1b49425acee37da3070510d0121922250908ce381ac8a4c8
725",
        version: 63
      }
    }
}]
> eth.blockNumber
1821
> eth.blockNumber
1821
```

- You can check the peers by running command : admin.peers

Creating Accounts

- Attach geth console to one of the node where you wanted to create accounts
- Type command personal.newAccount() and then provide a password.
-



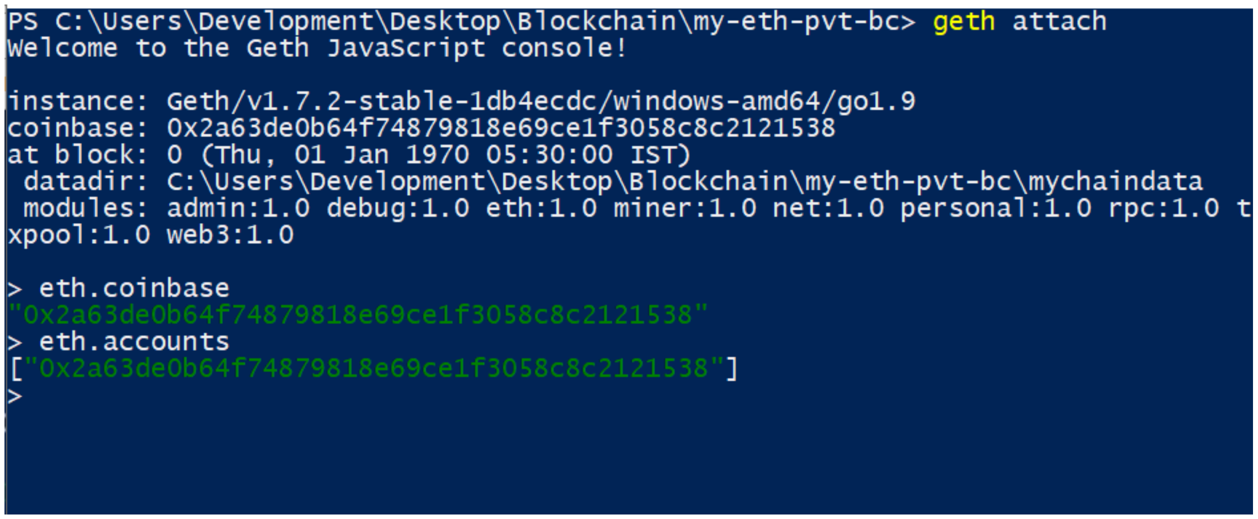
```
    at <anonymous>:1:1

> personal.newAccount()
Passphrase:
Repeat passphrase:
"0x2a63de0b64f74879818e69ce1f3058c8c2121538"
>
```

- The returned address is public address,not the private address.
- The private address is stored in Keystore folder present in the data directory
- Check accounts using command.
  - eth.accounts
- To check the balance
  - eth.getBalance(eth.accounts[0])

Mining Ether in private network

For mining , you have to have coinbase account set. That means, miners will credit ether to this account.

- Check the coinbase accounts is set or not
-

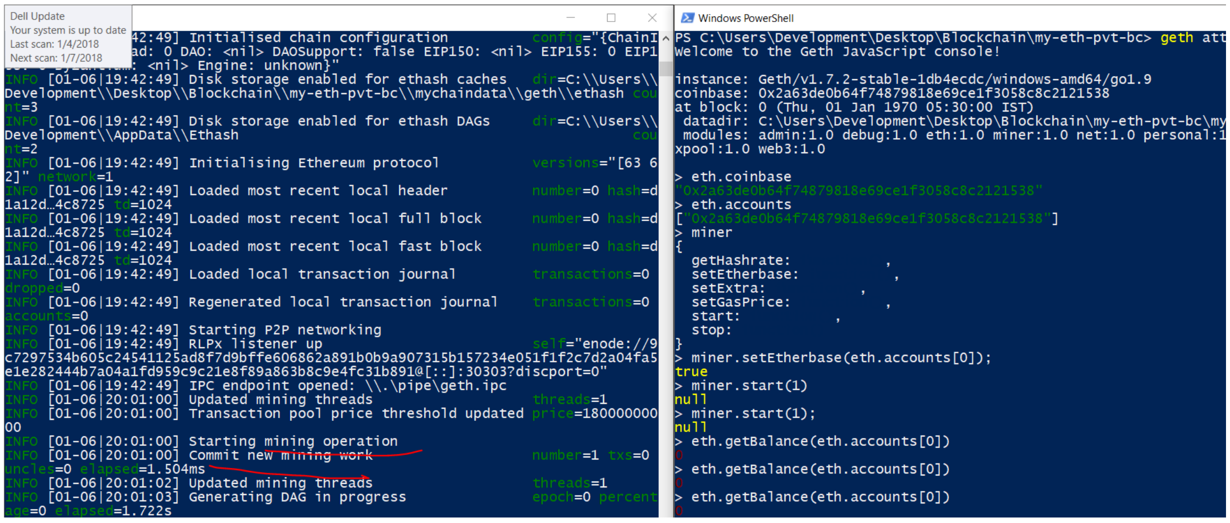


```
PS C:\Users\Development\Desktop\Blockchain\my-eth-pvt-bc> geth attach
Welcome to the Geth JavaScript console!

instance: Geth/v1.7.2-stable-1db4ecdc/windows-amd64/go1.9
coinbase: 0x2a63de0b64f74879818e69ce1f3058c8c2121538
at block: 0 (Thu, 01 Jan 1970 05:30:00 IST)
 datadir: C:\Users\Development\Desktop\Blockchain\my-eth-pvt-bc\mychaindata
 modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 t
xpool:1.0 web3:1.0

> eth.coinbase
"0x2a63de0b64f74879818e69ce1f3058c8c2121538"
> eth.accounts
["0x2a63de0b64f74879818e69ce1f3058c8c2121538"]
>
```

- Here it is set, but if not then, use below command to set
  - miner.setEtherbase(eth.accounts[0]);
  - It should return true.

To start mining : miner.start(1)
1 - is the amount of thread it run on.

It will return null on the client. But in the main window (geth instance) you will see that mining operation is running. See the screenshots below.



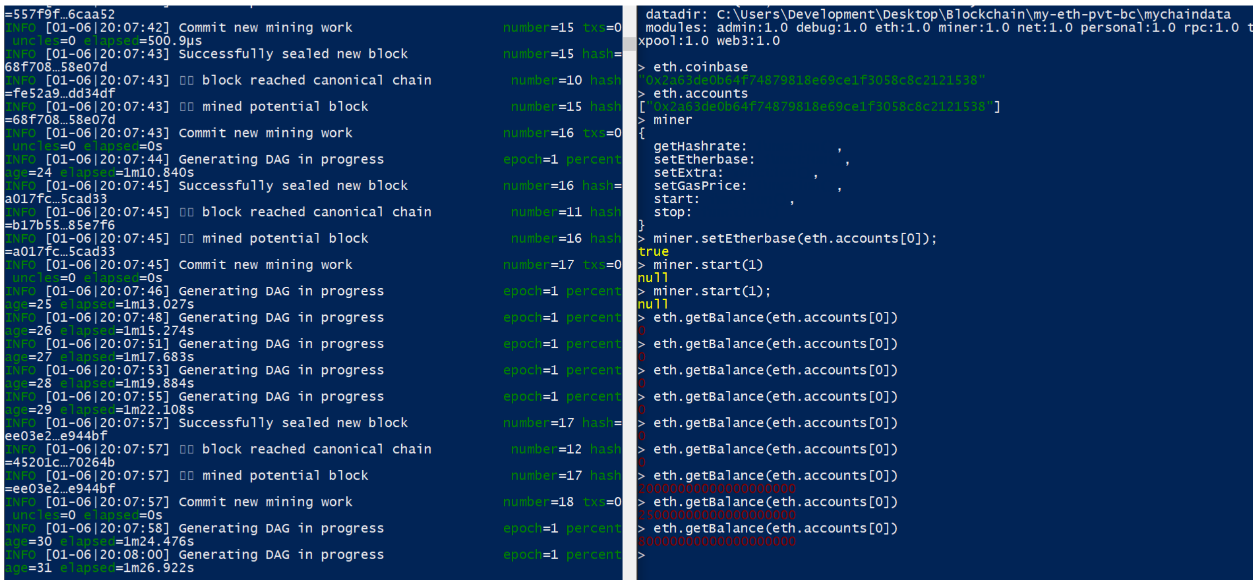
```
Dell Update
Your system is up to date
Last scan: 1/4/2018
Next scan: 1/7/2018
```

```
42:49] Initialised chain configuration             config="{ChainI
ad: 0 DAO: <nil> DAOSupport: false EIP150: <nil> EIP155: 0 EIP1
: <nil> Engine: unknown}"
INFO [01-06|19:42:49] Disk storage enabled for ethash caches   dir=C:\\Users\\
Development\\Desktop\\Blockchain\\my-eth-pvt-bc\\mychaindata\\geth\\ethash cou
nt=3
INFO [01-06|19:42:49] Disk storage enabled for ethash DAGs     dir=C:\\Users\\
Development\\AppData\\Ethash                                              cou
nt=2
INFO [01-06|19:42:49] Initialising Ethereum protocol           versions="[63 6
2]" network=1
INFO [01-06|19:42:49] Loaded most recent local header          number=0 hash=d
1a12d…4c8725 td=1024
INFO [01-06|19:42:49] Loaded most recent local full block      number=0 hash=d
1a12d…4c8725 td=1024
INFO [01-06|19:42:49] Loaded most recent local fast block      number=0 hash=d
1a12d…4c8725 td=1024
INFO [01-06|19:42:49] Loaded local transaction journal         transactions=0
dropped=0
INFO [01-06|19:42:49] Regenerated local transaction journal    transactions=0
accounts=0
INFO [01-06|19:42:49] Starting P2P networking
INFO [01-06|19:42:49] RLPx listener up                         self="enode://9
c7297534b605c24541125ad8f7d9bffe606862a891b0b9a907315b157234e051f1f2c7d2a04fa5
e1e282444b7a04a1fd959c9c21e8f89a863b8c9e4fc31b891@[::]:30303?discport=0"
INFO [01-06|19:42:49] IPC endpoint opened: \\.\pipe\geth.ipc
INFO [01-06|20:01:00] Updated mining threads                   threads=1
INFO [01-06|20:01:00] Transaction pool price threshold updated price=180000000
00
INFO [01-06|20:01:00] Starting mining operation
INFO [01-06|20:01:00] Commit new mining work                   number=1 txs=0
uncles=0 elapsed=1.504ms
INFO [01-06|20:01:02] Updated mining threads                   threads=1
INFO [01-06|20:01:03] Generating DAG in progress               epoch=0 percent
age=0 elapsed=1.722s
```

Windows PowerShell

```
PS C:\Users\Development\Desktop\Blockchain\my-eth-pvt-bc> geth att
Welcome to the Geth JavaScript console!

instance: Geth/v1.7.2-stable-1db4ecdc/windows-amd64/go1.9
coinbase: 0x2a63de0b64f74879818e69ce1f3058c8c2121538
at block: 0 (Thu, 01 Jan 1970 05:30:00 IST)
 datadir: C:\Users\Development\Desktop\Blockchain\my-eth-pvt-bc\my
 modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1
xpool:1.0 web3:1.0

> eth.coinbase
"0x2a63de0b64f74879818e69ce1f3058c8c2121538"
> eth.accounts
["0x2a63de0b64f74879818e69ce1f3058c8c2121538"]
> miner
{
  getHashrate:            ,
  setEtherbase:            ,
  setExtra:            ,
  setGasPrice:            ,
  start:            ,
  stop:
}
> miner.setEtherbase(eth.accounts[0]);
true
> miner.start(1)
null
> miner.start(1);
null
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
```

After some time check the balance



```
=557f9f…6caa52
INFO [01-06|20:07:42] Commit new mining work                   number=15 txs=0
 uncles=0 elapsed=500.9µs
INFO [01-06|20:07:43] Successfully sealed new block            number=15 hash=
68f708…58e07d
INFO [01-06|20:07:43] □□ block reached canonical chain          number=10 hash
=fe52a9…dd34df
INFO [01-06|20:07:43] □□ mined potential block                  number=15 hash
=68f708…58e07d
INFO [01-06|20:07:43] Commit new mining work                   number=16 txs=0
 uncles=0 elapsed=0s
INFO [01-06|20:07:44] Generating DAG in progress               epoch=1 percent
age=24 elapsed=1m10.840s
INFO [01-06|20:07:45] Successfully sealed new block            number=16 hash=
a017fc…5cad33
INFO [01-06|20:07:45] □□ block reached canonical chain          number=11 hash
=b17b55…85e7f6
INFO [01-06|20:07:45] □□ mined potential block                  number=16 hash
=a017fc…5cad33
INFO [01-06|20:07:45] Commit new mining work                   number=17 txs=0
 uncles=0 elapsed=0s
INFO [01-06|20:07:46] Generating DAG in progress               epoch=1 percent
age=25 elapsed=1m13.027s
INFO [01-06|20:07:48] Generating DAG in progress               epoch=1 percent
age=26 elapsed=1m15.274s
INFO [01-06|20:07:51] Generating DAG in progress               epoch=1 percent
age=27 elapsed=1m17.683s
INFO [01-06|20:07:53] Generating DAG in progress               epoch=1 percent
age=28 elapsed=1m19.884s
INFO [01-06|20:07:55] Generating DAG in progress               epoch=1 percent
age=29 elapsed=1m22.108s
INFO [01-06|20:07:57] Successfully sealed new block            number=17 hash=
ee03e2…e944bf
INFO [01-06|20:07:57] □□ block reached canonical chain          number=12 hash
=45201c…70264b
INFO [01-06|20:07:57] □□ mined potential block                  number=17 hash
=ee03e2…e944bf
INFO [01-06|20:07:57] Commit new mining work                   number=18 txs=0
 uncles=0 elapsed=0s
INFO [01-06|20:07:58] Generating DAG in progress               epoch=1 percent
age=30 elapsed=1m24.476s
INFO [01-06|20:08:00] Generating DAG in progress               epoch=1 percent
age=31 elapsed=1m26.922s
```

```
 datadir: C:\Users\Development\Desktop\Blockchain\my-eth-pvt-bc\mychaindata
 modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 t
xpool:1.0 web3:1.0

> eth.coinbase
"0x2a63de0b64f74879818e69ce1f3058c8c2121538"
> eth.accounts
["0x2a63de0b64f74879818e69ce1f3058c8c2121538"]
> miner
{
  getHashrate:            ,
  setEtherbase:            ,
  setExtra:            ,
  setGasPrice:            ,
  start:            ,
  stop:
}
> miner.setEtherbase(eth.accounts[0]);
true
> miner.start(1)
null
> miner.start(1);
null
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
20000000000000000000
> eth.getBalance(eth.accounts[0])
25000000000000000000
> eth.getBalance(eth.accounts[0])
80000000000000000000
>
```

Connecting wallet to private network

Summary

------- Step 001 : On Node 01 : Running node 01 -------------

```
geth --datadir data01/ --nodiscover --port 30303 --networkid 1010 --rpc --rpcaddr 127.0.0.1 --
rpcport 8545 --rpccorsdomain
```

------- Step 002 : On Node 02 : Running node 02 -------------

```
geth --datadir data02/ --nodiscover --port 30304 --networkid 1010 --rpc --rpcaddr 127.0.0.1 --
rpcport 8546 --rpccorsdomain
```

------- Step 003 : On Node 02 : Copy enode value -------------

```
admin.nodeInfo
```

copy the enode value:

```
enode://
b3e99962e9165bdb1915e195eda4b11b472a2c8ccb78301325bc3d3f6fc5d64a7c004b27f334db
a6e2150225ec8e4784d8c1991528bc8f6605ed5961b9c1f8e9@[::]:30304?discport=0
```

------- Step 004 : On Node 01 : connect to the peer -------------

```
admin.addPeer("enode://
b3e99962e9165bdb1915e195eda4b11b472a2c8ccb78301325bc3d3f6fc5d64a7c004b27f334db
a6e2150225ec8e4784d8c1991528bc8f6605ed5961b9c1f8e9@[::]:30304?discport=0")
```

------- Step 005 : On Node any : check peers -------------

```
admin.peers
```

Geth Flags

- --datadir
- --nodiscover
- --port
- --networkid
- --rpc
- --rpcaddr

- --rpcport
- --rpccorsdomain
- --

References

https://github.com/ethereum/go-ethereum

Geth download

Geth Installation

Management APIs

Json RPC

Command Line Options