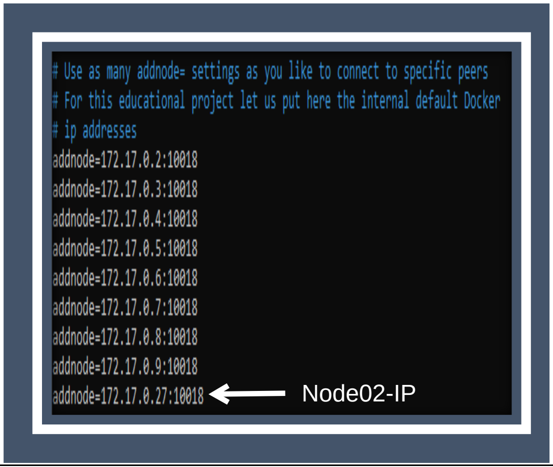
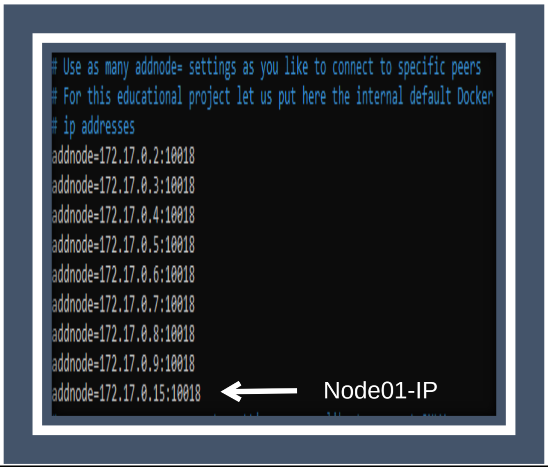
Name: Arnab Rahman Chowdhury **Date: 31 May 2019**
Title: Report on Own Cryptocurrency

Name of the coin:	Barccoin I have named my coin according a football club Barcelona.
Denomination of coin:	lamasia masia The denominators are named after the youth academy of Barcelona Football Club.
Magic Number:	To identify “Barccoin”, we have to use 4 identifying bytes which is known as Magic Number. These bytes identify clients in your network as belonging to a protocol. As an example, Litecoin uses 0xfb, 0xc0, 0xb6, 0xdb; while Bitcoin uses0xf9, 0xbe, 0xb4, and 0xd9 to identify itself in network messages. I have used 0x62, 0x61, 0x72, 0x63. Here, 0x62 -> b 0x61 -> a 0x72 -> r 0x63 -> c Which represent barc as Barccoin.

Base58Prefix	Base58 encoded prefixes are used as prefixes for public, private, and stealth addresses (addresses which can receive but not spend).
	base58Prefixes[PUBKEY_ADDRESS] is chosen in such a way that the public address will always start with “b”.
	I have used the base58prefix as following:
	base58Prefixes[PUBKEY_ADDRESS] = 85 -> b (leading symbol)
	base58Prefixes[SCRIPT_ADDRESS] = 35 -> F (leading symbol)
	base58Prefixes[PUBKEY_ADDRESS2] = 28 -> C (leading symbol)
	base58Prefixes[SECRET_KEY] = 25 -> B (leading symbol)
	
	This value is chosen according to the name of the coin. The name of the coin starts with the letter b, so the public address and the secret key are also start with b or B.
	
	base58Prefixes[EXT_PUBLIC_KEY] = {0x67, 0x6f, 0x61, 0x74} -> g,o,a,t
	base58Prefixes[EXT_SECRET_KEY] = {0x47, 0x4f, 0x41, 0x54} -> G,O,A,T

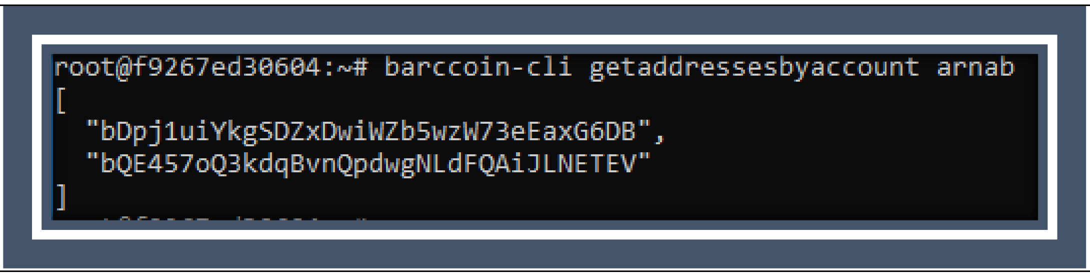
Genesis Block and Markel root	Genesis block is the first block of blockchain network. I have used GenesisH0 script to generate the genesis block. I have used the following parameter to generate genesis block: News text: The Guardian 1 May 2019 Best Male Footballer This news text verifies that my chain did not pre-exist before a given date. This is important as users will want to rest assured that your chain was not pre-mined for coins at low difficulty when the chain was started. As public key, I have used the public key of Litecoin. Timestamp: I have used the unix timestamp, when I was generating the genesis block.

	I have generated three nonce for three different network such as, Main Network, TestNet Network and RegTest Network. For generating three nonce, I have used the same news text but different unix timestamp.
Ports	Nodes communicate with each other over P2P using a designated port. To reduce unwanted traffic to our nodes it is advised to use a unique value. I have changed the default port as 10018.
Working Procedure:	

Run script and creating docker container	1. I have run RunMe1st.sh script to generate two docker images such as coin01 and coin02.
	2. In order to have a working P2P network for your cryptocurrency, you will need to deploy a minimum of 2 nodes. I have created two nodes running the following commands:
	`# docker run -P -dit --name node01 -v "$PWD/data:/root/.barccoin/" -w /root coin02 barccoind`
	Here node01 container is created and mount the container in local machine and start my node as a daemon which will run in the background and attempt to restart itself if needed.
	`# docker run -P -it --name node02 -w /root coin02`
	Here node02 container is created.
	Check the configuration file of both the nodes and add the IP address of the each other nodes in the configuration file.
	`# cat .barccoin/barccoin.conf`
	
	`# Use as many addnode= settings as you like to connect to specific peers`
	`# For this educational project let us put here the internal default Docker`
	`# ip addresses`
	`addnode=172.17.0.2:10018`
	`addnode=172.17.0.3:10018`
	`addnode=172.17.0.4:10018`
	`addnode=172.17.0.5:10018`
	`addnode=172.17.0.6:10018`
	`addnode=172.17.0.7:10018`
	`addnode=172.17.0.8:10018`
	`addnode=172.17.0.9:10018`
	`addnode=172.17.0.27:10018`
	
	
	`# Use as many addnode= settings as you like to connect to specific peers`
	`# For this educational project let us put here the internal default Docker`
	`# ip addresses`
	`addnode=172.17.0.2:10018`
	`addnode=172.17.0.3:10018`
	`addnode=172.17.0.4:10018`
	`addnode=172.17.0.5:10018`
	`addnode=172.17.0.6:10018`
	`addnode=172.17.0.7:10018`
	`addnode=172.17.0.8:10018`
	`addnode=172.17.0.9:10018`
	`addnode=172.17.0.15:10018`

Generating public address	The following command generate the public address of users. We can run it multiple times to generate different public address for the same user or different user. All the public addresses are bound to the same node and share wallet information. Each node has its own wallet which is shared by all the users. `# barccoin-cli getnewaddress “User_Name”`

<table>
<tr>
<td></td>
<td>



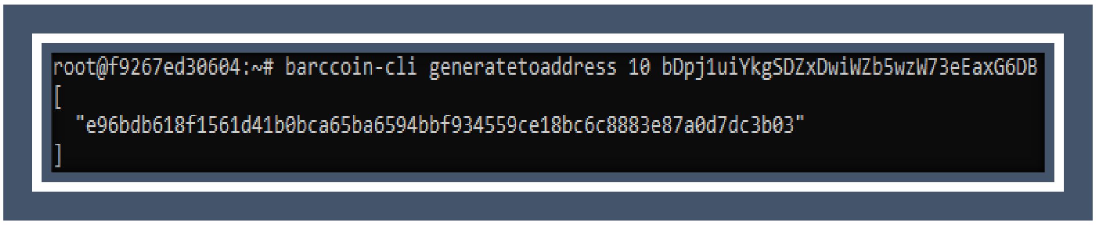
```
root@f9267ed30604:~# barccoin-cli getaddressesbyaccount arnab
[
  "bDpj1uiYkgSDZxDwiWZb5wzW73eEaxG6DB",
  "bQE457oQ3kdqBvnQpdwgNLdFQAiJLNETEV"
]
```

</td>
</tr>
<tr>
<td>Mining coins</td>
<td>

To mine coins to a particular address I have run the following command:
# barccoin-cli generatetoaddress 10 PUBLIC_ADDRESS

Here, formula of generatetoaddress command:
# coinname-cli generatetoaddress N PUBLIC_ADDESS I

N = How many blocks are generated immediately
PUBLIC_ADDRESS = The address to send the newly generated bitcoin to
I = How many iterations to try (default = 1000000)

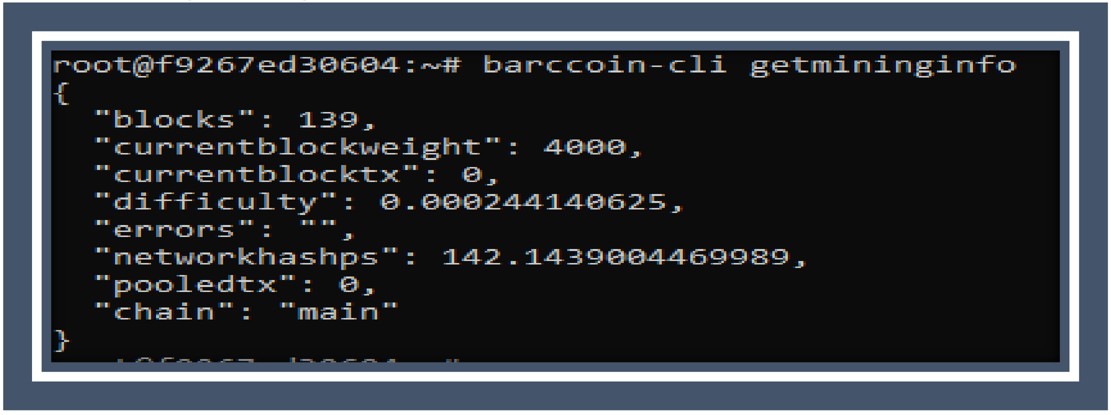


```
root@f9267ed30604:~# barccoin-cli generatetoaddress 10 bDpj1uiYkgSDZxDwiWZb5wzW73eEaxG6DB
[
  "e96bdb618f1561d41b0bca65ba6594bbf934559ce18bc6c8883e87a0d7dc3b03"
]
```

Each block mining will reward 50 barccoins but the coins are immature to transfer. To make the barccoins mature at least 100 blocks need to be mined. To clarify the issue, if we have 120 blocks then 20*50 = 1000 coins are matured and ready for transfer.

</td>
</tr>
<tr>
<td>Mining Information</td>
<td>

We can get mining information by the following commands:
# barccoin-cli getmininginfo



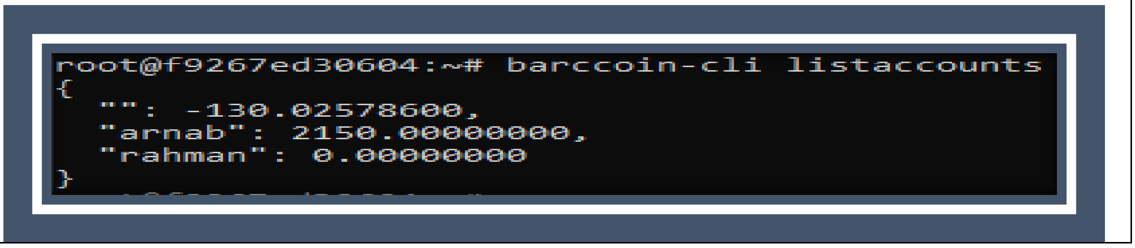
```
root@f9267ed30604:~# barccoin-cli getmininginfo
{
  "blocks": 139,
  "currentblockweight": 4000,
  "currentblocktx": 0,
  "difficulty": 0.000244140625,
  "errors": "",
  "networkhashps": 142.1439004469989,
  "pooledtx": 0,
  "chain": "main"
}
```

This is the state of one of the nodes wallets.

</td>
</tr>
<tr>
<td>Transaction</td>
<td>

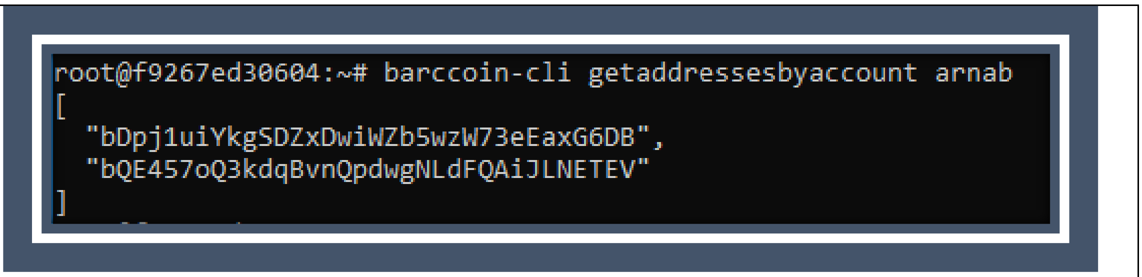
I have created one node (node01) which has an account named “arnab”. I have also created another node (node02) which has an account named “zabir”. The public addresses of both the account is given below:
# barccoin-cli getaddressesbyaccount account_name

Account: arnab



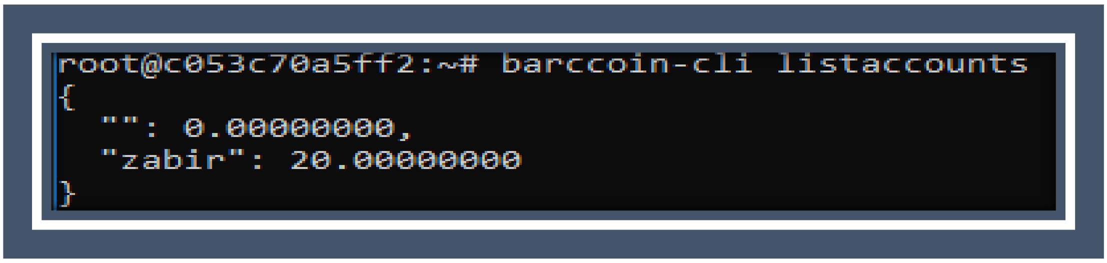
```
root@f9267ed30604:~# barccoin-cli listaccounts
{
  "": -130.02578600,
  "arnab": 2150.00000000,
  "rahman": 0.00000000
}
```

</td>
</tr>
</table>

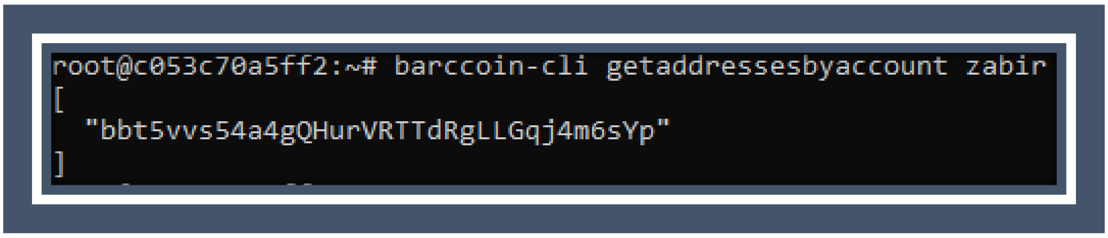


```
root@f9267ed30604:~# barccoin-cli getaddressesbyaccount arnab
[
  "bDpj1uiYkgSDZxDwiWZb5wzW73eEaxG6DB",
  "bQE457oQ3kdqBvnQpdwgNLdFQAiJLNETEV"
]
```

Account: zabir

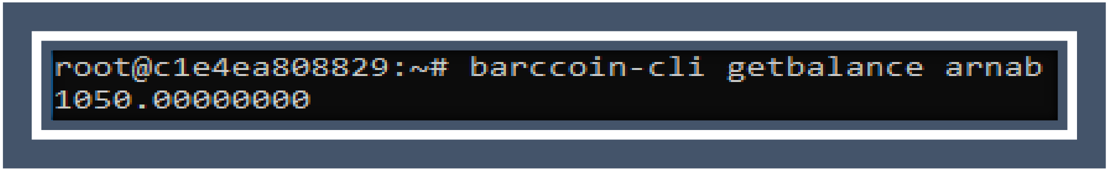


```
root@c053c70a5ff2:~# barccoin-cli listaccounts
{
  "": 0.00000000,
  "zabir": 20.00000000
}
```



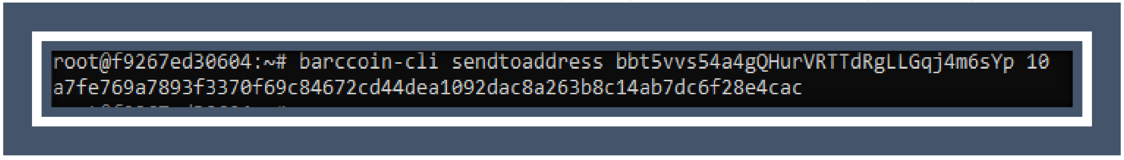
```
root@c053c70a5ff2:~# barccoin-cli getaddressesbyaccount zabir
[
  "bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp"
]
```

The balance of the account "arnab" is given below:
# barccoin-cli getbalance arnab



```
root@c1e4ea808829:~# barccoin-cli getbalance arnab
1050.00000000
```

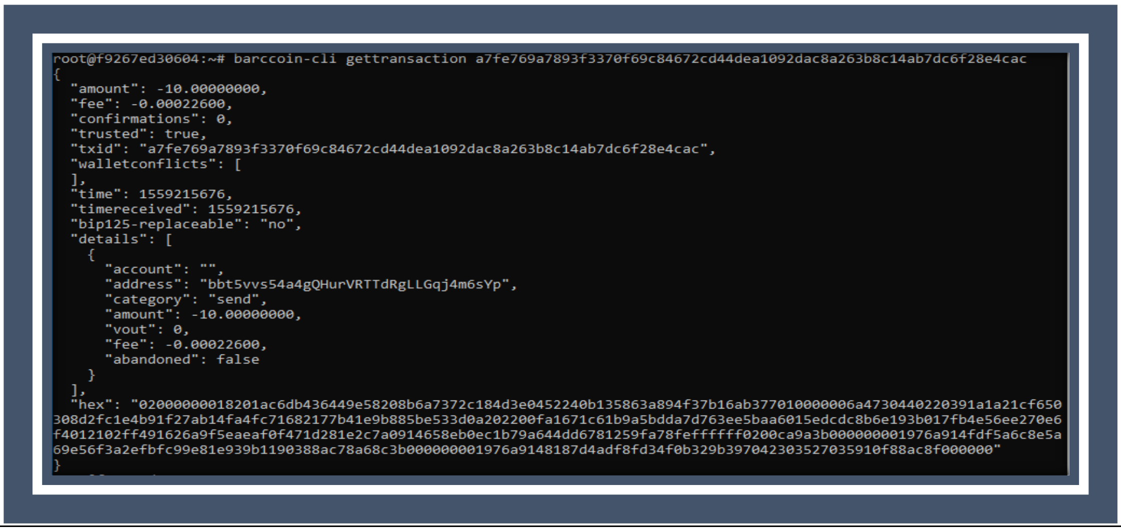
Now, I will transfer 10 coins to an account (zabir) in another node (node02).



```
root@f9267ed30604:~# barccoin-cli sendtoaddress bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp 10
a7fe769a7893f3370f69c84672cd44dea1092dac8a263b8c14ab7dc6f28e4cac
```

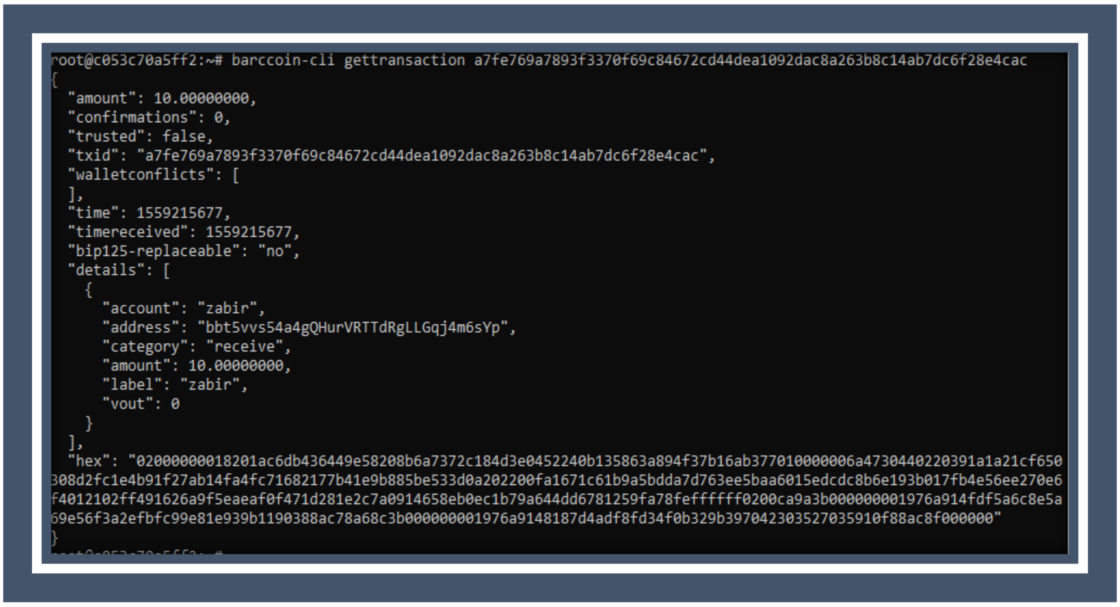
The transaction status is given below:

node01:



```
root@f9267ed30604:~# barccoin-cli gettransaction a7fe769a7893f3370f69c84672cd44dea1092dac8a263b8c14ab7dc6f28e4cac
{
  "amount": -10.00000000,
  "fee": -0.00022600,
  "confirmations": 0,
  "trusted": true,
  "txid": "a7fe769a7893f3370f69c84672cd44dea1092dac8a263b8c14ab7dc6f28e4cac",
  "walletconflicts": [
  ],
  "time": 1559215676,
  "timereceived": 1559215676,
  "bip125-replaceable": "no",
  "details": [
    {
      "account": "",
      "address": "bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp",
      "category": "send",
      "amount": -10.00000000,
      "vout": 0,
      "fee": -0.00022600,
      "abandoned": false
    }
  ],
  "hex": "02000000018201ac6db436449e58208b6a7372c184d3e0452240b135863a894f37b16ab377010000006a4730440220391a1a21cf650
308d2fc1e4b91f27ab14fa4fc71682177b41e9b885be533d0a202200fa1671c61b9a5bdda7d763ee5baa6015edcdc8b6e193b017fb4e56ee270e6
f4012102ff491626a9f5eaeaf0f471d281e2c7a0914658eb0ec1b79a644dd6781259fa78feffffff0200ca9a3b000000001976a914fdf5a6c8e5a
69e56f3a2efbfc99e81e939b1190388ac78a68c3b000000001976a9148187d4adf8fd34f0b329b3970423035270355910f88ac8f000000"
}
```

node02:

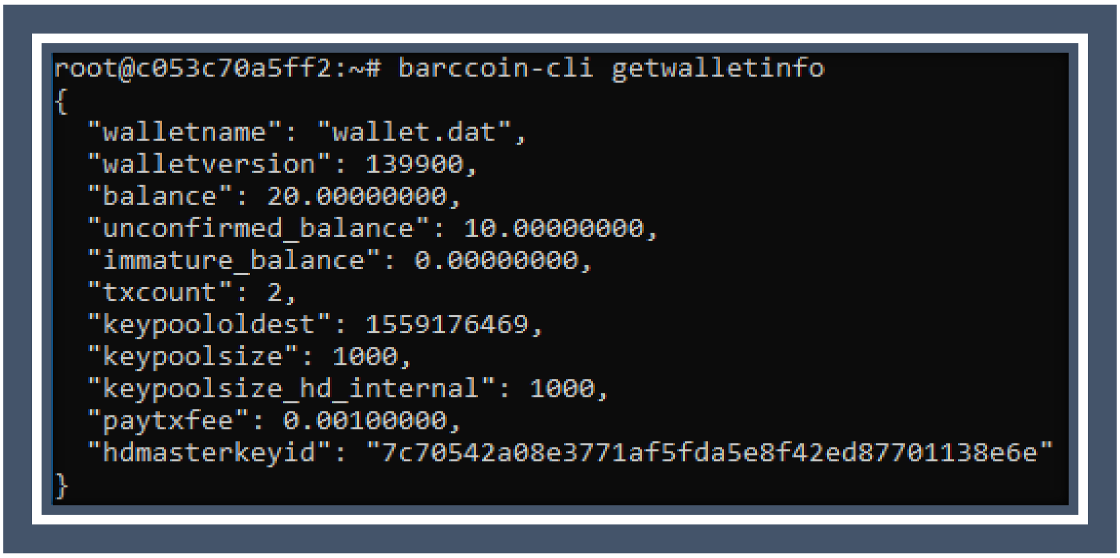


```
root@c053c70a5ff2:~# barccoin-cli gettransaction a7fe769a7893f3370f69c84672cd44dea1092dac8a263b8c14ab7dc6f28e4cac
{
  "amount": 10.00000000,
  "confirmations": 0,
  "trusted": false,
  "txid": "a7fe769a7893f3370f69c84672cd44dea1092dac8a263b8c14ab7dc6f28e4cac",
  "walletconflicts": [
  ],
  "time": 1559215677,
  "timereceived": 1559215677,
  "bip125-replaceable": "no",
  "details": [
    {
      "account": "zabir",
      "address": "bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp",
      "category": "receive",
      "amount": 10.00000000,
      "label": "zabir",
      "vout": 0
    }
  ],
  "hex": "02000000018201ac6db436449e58208b6a7372c184d3e0452240b135863a894f37b16ab377010000006a4730440220391a1a21cf650
308d2fc1e4b91f27ab14fa4fc71682177b41e9b885be533d0a202200fa1671c61b9a5bdda7d763ee5baa6015edcdc8b6e193b017fb4e56ee270e6
f4012102ff491626a9f5eaeaf0f471d281e2c7a0914658eb0ec1b79a644dd6781259fa78feffffff0200ca9a3b000000001976a914fdf5a6c8e5a
69e56f3a2efbfc99e81e939b1190388ac78a68c3b000000001976a9148187d4adf8fd34f0b329b397042303527035910f88ac8f000000"
}
```

Another command to send coins:
# barccoin-cli sendfrom <sender_account_name> <recipient_address> <amount> <minimum_confirmation> <comment>

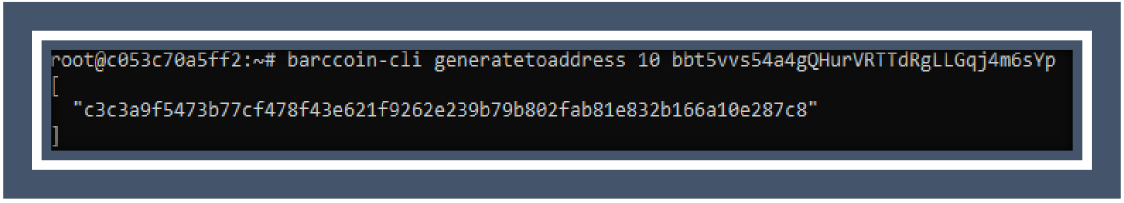
After transferring some coins, the coins are listed as unconfirmed balance in the recipient’s account. After successfully mining one block, the coins are added as balance in recipient’s wallet.

Unconfirmed amount in node02:



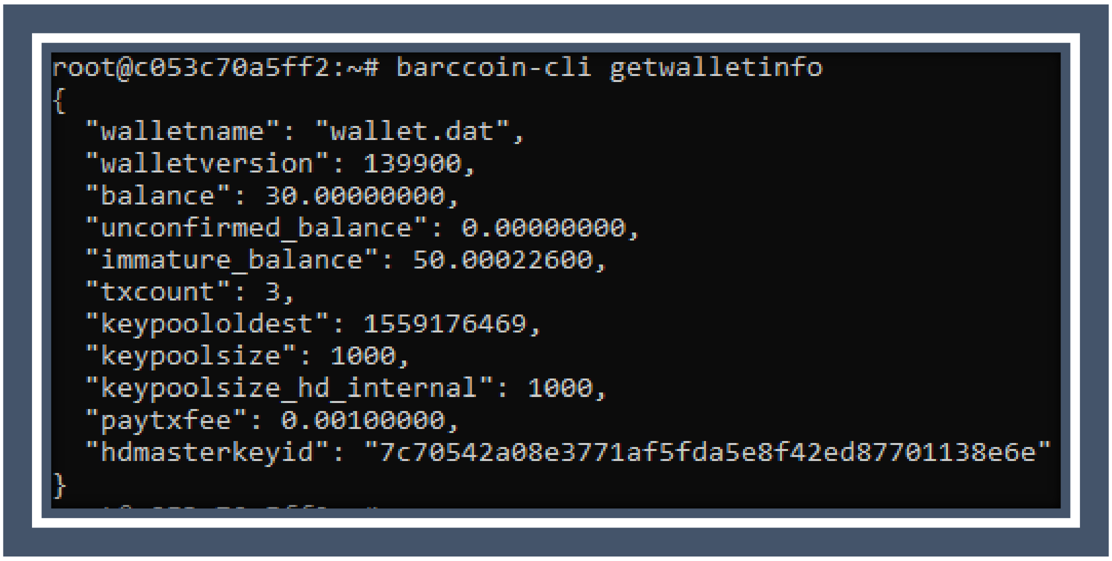
```
root@c053c70a5ff2:~# barccoin-cli getwalletinfo
{
  "walletname": "wallet.dat",
  "walletversion": 139900,
  "balance": 20.00000000,
  "unconfirmed_balance": 10.00000000,
  "immature_balance": 0.00000000,
  "txcount": 2,
  "keypoololdest": 1559176469,
  "keypoolsize": 1000,
  "keypoolsize_hd_internal": 1000,
  "paytxfee": 0.00100000,
  "hdmasterkeyid": "7c70542a08e3771af5fda5e8f42ed87701138e6e"
}
```

Mining block:



```
root@c053c70a5ff2:~# barccoin-cli generatetoaddress 10 bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp
[
  "c3c3a9f5473b77cf478f43e621f9262e239b79b802fab81e832b166a10e287c8"
]
```

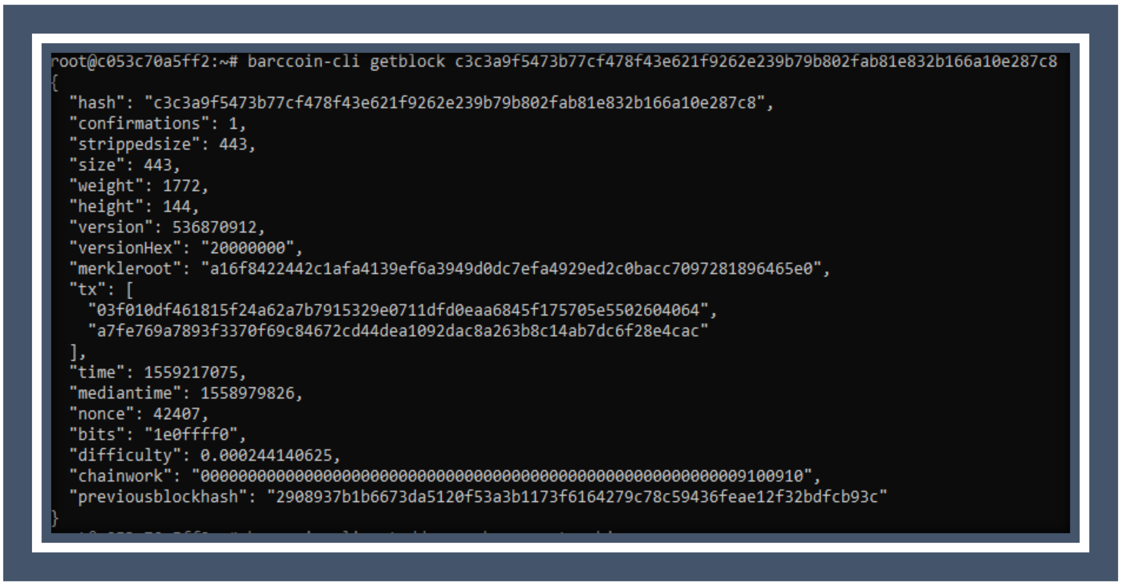
Current state of wallet in node02:



```
root@c053c70a5ff2:~# barccoin-cli getwalletinfo
{
  "walletname": "wallet.dat",
  "walletversion": 139900,
  "balance": 30.00000000,
  "unconfirmed_balance": 0.00000000,
  "immature_balance": 50.00022600,
  "txcount": 3,
  "keypoololdest": 1559176469,
  "keypoolsize": 1000,
  "keypoolsize_hd_internal": 1000,
  "paytxfee": 0.00100000,
  "hdmasterkeyid": "7c70542a08e3771af5fda5e8f42ed87701138e6e"
}
```

Balance has been added to the wallet.

Mined block information:



```
root@c053c70a5ff2:~# barccoin-cli getblock c3c3a9f5473b77cf478f43e621f9262e239b79b802fab81e832b166a10e287c8
{
  "hash": "c3c3a9f5473b77cf478f43e621f9262e239b79b802fab81e832b166a10e287c8",
  "confirmations": 1,
  "strippedsize": 443,
  "size": 443,
  "weight": 1772,
  "height": 144,
  "version": 536870912,
  "versionHex": "20000000",
  "merkleroot": "a16f8422442c1afa4139ef6a3949d0dc7efa4929ed2c0bacc7097281896465e0",
  "tx": [
    "03f010df461815f24a62a7b7915329e0711dfd0eaa6845f175705e5502604064",
    "a7fe769a7893f3370f69c84672cd44dea1092dac8a263b8c14ab7dc6f28e4cac"
  ],
  "time": 1559217075,
  "mediantime": 1558979826,
  "nonce": 42407,
  "bits": "1e0ffff0",
  "difficulty": 0.000244140625,
  "chainwork": "0000000000000000000000000000000000000000000000000000000009100910",
  "previousblockhash": "2908937b1b6673da5120f53a3b1173f6164279c78c59436feae12f32bdfcb93c"
}
```

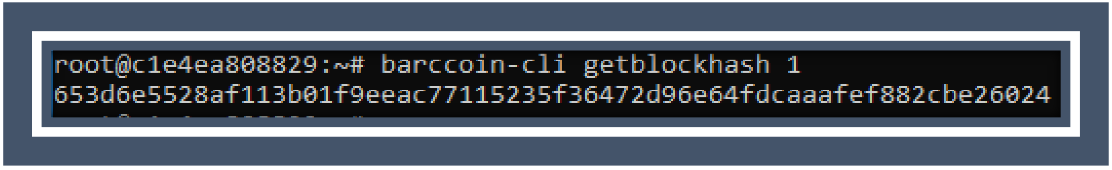
The transaction id of the previous transaction is added in the block.

Block information

Here I have listed some command to get the information about block.

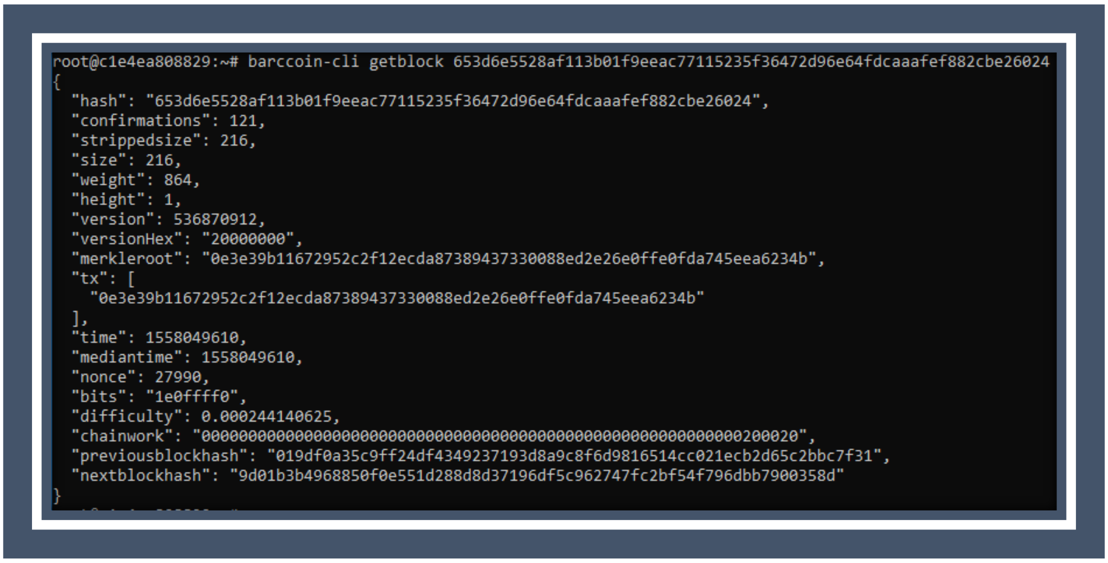
1. Get block hash:

barccoin-cli getblockhash <index>



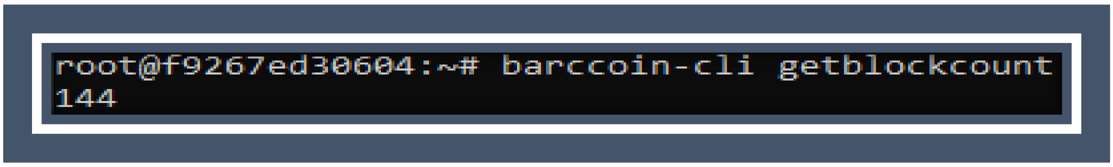
```
root@c1e4ea808829:~# barccoin-cli getblockhash 1
653d6e5528af113b01f9eeac77115235f36472d96e64fdcaaafef882cbe26024
```

2. Block information



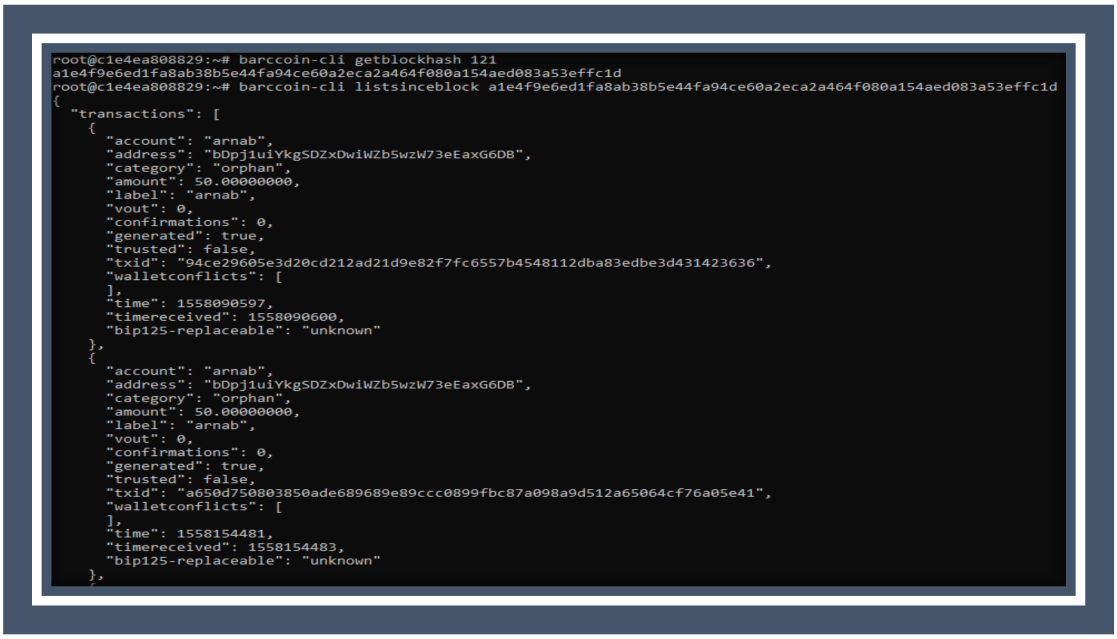
```
root@c1e4ea808829:~# barccoin-cli getblock 653d6e5528af113b01f9eeac77115235f36472d96e64fdcaaafef882cbe26024
{
  "hash": "653d6e5528af113b01f9eeac77115235f36472d96e64fdcaaafef882cbe26024",
  "confirmations": 121,
  "strippedsize": 216,
  "size": 216,
  "weight": 864,
  "height": 1,
  "version": 536870912,
  "versionHex": "20000000",
  "merkleroot": "0e3e39b11672952c2f12ecda87389437330088ed2e26e0ffe0fda745eea6234b",
  "tx": [
    "0e3e39b11672952c2f12ecda87389437330088ed2e26e0ffe0fda745eea6234b"
  ],
  "time": 1558049610,
  "mediantime": 1558049610,
  "nonce": 27990,
  "bits": "1e0ffff0",
  "difficulty": 0.000244140625,
  "chainwork": "0000000000000000000000000000000000000000000000000000000000200020",
  "previousblockhash": "019df0a35c9ff24df4349237193d8a9c8f6d9816514cc021ecb2d65c2bbc7f31",
  "nextblockhash": "9d01b3b4968850f0e551d288d8d37196df5c962747fc2bf54f796dbb7900358d"
}
```

3. block count:



```
root@f9267ed30604:~# barccoin-cli getblockcount
144
```

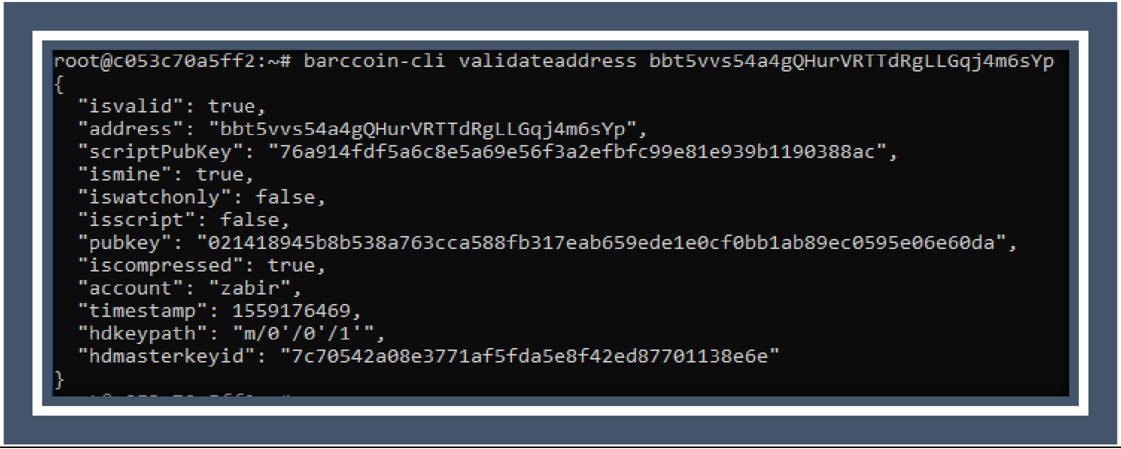
4. Get all transactions in blocks since block
# barccoin-cli listsinceblock blockhash



```
root@c1e4ea808829:~# barccoin-cli getblockhash 121
a1e4f9e6ed1fa8ab38b5e44fa94ce60a2eca2a464f080a154aed083a53effc1d
root@c1e4ea808829:~# barccoin-cli listsinceblock a1e4f9e6ed1fa8ab38b5e44fa94ce60a2eca2a464f080a154aed083a53effc1d
{
  "transactions": [
    {
      "account": "arnab",
      "address": "bDpj1uiYkgSDZxDwiWZb5wzW73eEaxG6DB",
      "category": "orphan",
      "amount": 50.00000000,
      "label": "arnab",
      "vout": 0,
      "confirmations": 0,
      "generated": true,
      "trusted": false,
      "txid": "94ce29605e3d20cd212ad21d9e82f7fc6557b4548112dba83edbe3d431423636",
      "walletconflicts": [
      ],
      "time": 1558090597,
      "timereceived": 1558090600,
      "bip125-replaceable": "unknown"
    },
    {
      "account": "arnab",
      "address": "bDpj1uiYkgSDZxDwiWZb5wzW73eEaxG6DB",
      "category": "orphan",
      "amount": 50.00000000,
      "label": "arnab",
      "vout": 0,
      "confirmations": 0,
      "generated": true,
      "trusted": false,
      "txid": "a650d750803850ade689689e89ccc0899fbc87a098a9d512a65064cf76a05e41",
      "walletconflicts": [
      ],
      "time": 1558154481,
      "timereceived": 1558154483,
      "bip125-replaceable": "unknown"
    },
```

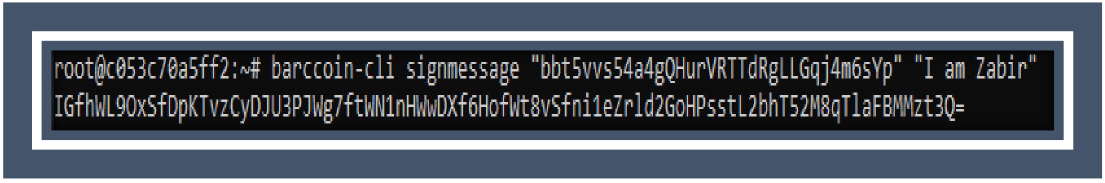
Additional commands

1. Validating address:



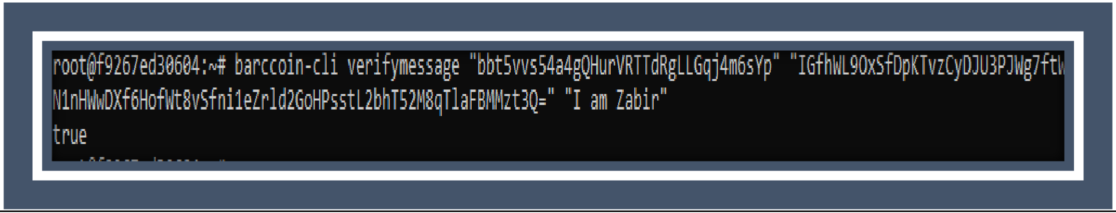
```
root@c053c70a5ff2:~# barccoin-cli validateaddress bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp
{
  "isvalid": true,
  "address": "bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp",
  "scriptPubKey": "76a914fdf5a6c8e5a69e56f3a2efbfc99e81e939b1190388ac",
  "ismine": true,
  "iswatchonly": false,
  "isscript": false,
  "pubkey": "021418945b8b538a763cca588fb317eab659ede1e0cf0bb1ab89ec0595e06e60da",
  "iscompressed": true,
  "account": "zabir",
  "timestamp": 1559176469,
  "hdkeypath": "m/0'/0'/1'",
  "hdmasterkeyid": "7c70542a08e3771af5fda5e8f42ed87701138e6e"
}
```

2. Sign message:



```
root@c053c70a5ff2:~# barccoin-cli signmessage "bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp" "I am Zabir"
IGfhWL9OxSfDpKTvzCyDJU3PJWg7ftWN1nHWwDXf6HofWt8vSfni1eZrld2GoHPsstL2bhT52M8qTlaFBMMzt3Q=
```

3. Verify message:

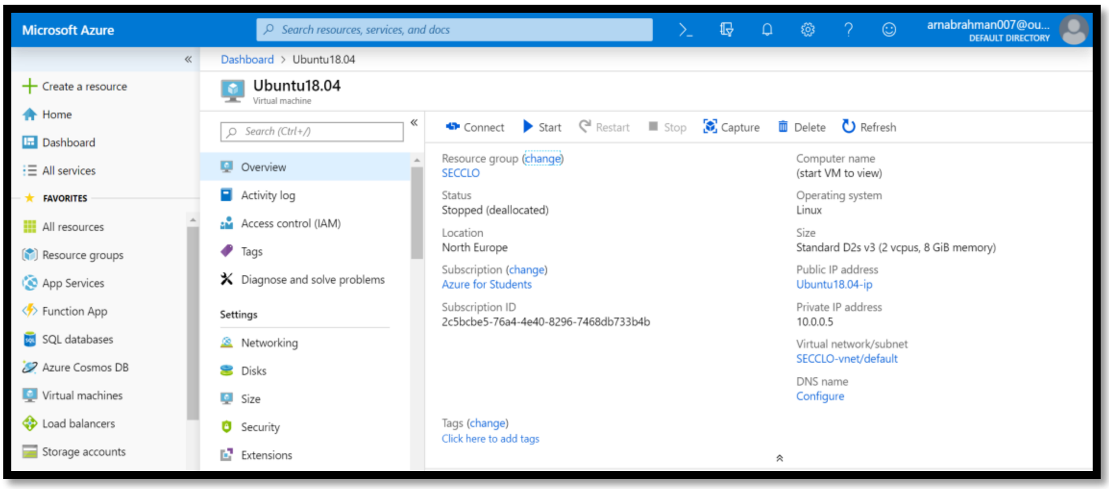


```
root@f9267ed30604:~# barccoin-cli verifymessage "bbt5vvs54a4gQHurVRTTdRgLLGqj4m6sYp" "IGfhWL9OxSfDpKTvzCyDJU3PJWg7ftW
N1nHWwDXf6HofWt8vSfni1eZrld2GoHPsstL2bhT52M8qTlaFBMMzt3Q=" "I am Zabir"
true
```

Local altcoin implementation

I have used Microsoft Azure to create by barccoin. I have created virtual machine using the following requirement:

- Size: Standard D2s v3 (2 vcpus, 8 GiB memory)
- OS: Linux



Ubuntu18.04
Virtual machine

Resource group (change)
SECCLO

Status
Stopped (deallocated)

Location
North Europe

Subscription (change)
Azure for Students

Subscription ID
2c5bcbe5-76a4-4e40-8296-7468db733b4b

Tags (change)
Click here to add tags

Computer name
(start VM to view)

Operating system
Linux

Size
Standard D2s v3 (2 vcpus, 8 GiB memory)

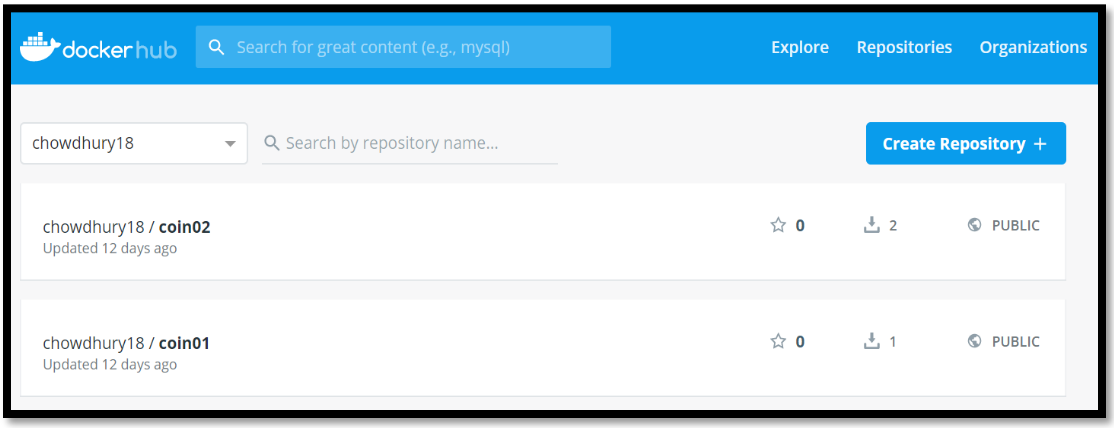
Public IP address
Ubuntu18.04-ip

Private IP address
10.0.0.5

Virtual network/subnet
SECCLO-vnet/default

DNS name
Configure

I have created two nodes to run the peer-to-peer network and transactions.
To keep backup of my coins, I have uploaded the coins in my docker hub account. So, when ever I need to build a container to any machine, I can pull the images from my docker repository.



chowdhury18 / **coin02**
Updated 12 days ago
☆ 0 2 PUBLIC

chowdhury18 / **coin01**
Updated 12 days ago
☆ 0 1 PUBLIC

I have also backed up my data folder where I have mounted my barccoin.
mylocalmachin# scp -r arnab@40.112.92.5:/home/arnab/data/* $PWD/data

- 40.112.92.5 is the IP address of the Azure machine.

I have also save the image as tar file according to the instruction.
# docker tag coin02 coin02:Arnab
# docker image save -o ./coin02-arnab.tar coin02:Arnab

	To import the image from tar, I have used docker load command. # docker load - -input coin02-arnab.tar I have tried with docker import command but after running this command, the image is created but the repository and tag is not assigned to my image.
Reference	
	1. https://www.hackster.io/pjdecarlo/how-to-make-a-cryptocurrency-using-litecoin-v0-15-source-fb5e82 2. https://en.bitcoin.it/wiki/Original_Bitcoin_client/API_calls_list 3. https://medium.com/@karthikmargabandu7/public-keys-private-keys-and-bitcoin-address-bf26125addf7 4. https://www.mycryptopedia.com/genesis-block-explained/ 5. https://lifehacker.com/how-to-create-your-own-cryptocurrency-1825337462 6. https://masterthecrypto.com/verifying-cryptocurrency-transactions/ 7. https://cryptocurrencyfacts.com/what-is-a-cryptocurrency-wallet/ 8. https://coinguides.org/immature-confirmed-cleared/ 9. https://www.quora.com/How-many-transactions-are-included-in-a-block-chain 10. https://bitcoin.stackexchange.com/questions/43189/what-is-the-magic-number-used-in-the-block-structure