Create & Deploy your private Blockchain on Multichain

After completing this module, you might have understood how a MultiChain platform works. Also, you would have learnt to create & deploy your private blockchain using MultiChain.

Q: Let's now create a private blockchain with some modifications in parameters in params.dat file
To modify the params.dat file use the following command:

nano ~/.multichain/<blockchain name> /params.dat

Following are the parameters we need to change:

Basic Chain parameters
Chain description = "you can provide description to your blockchain"

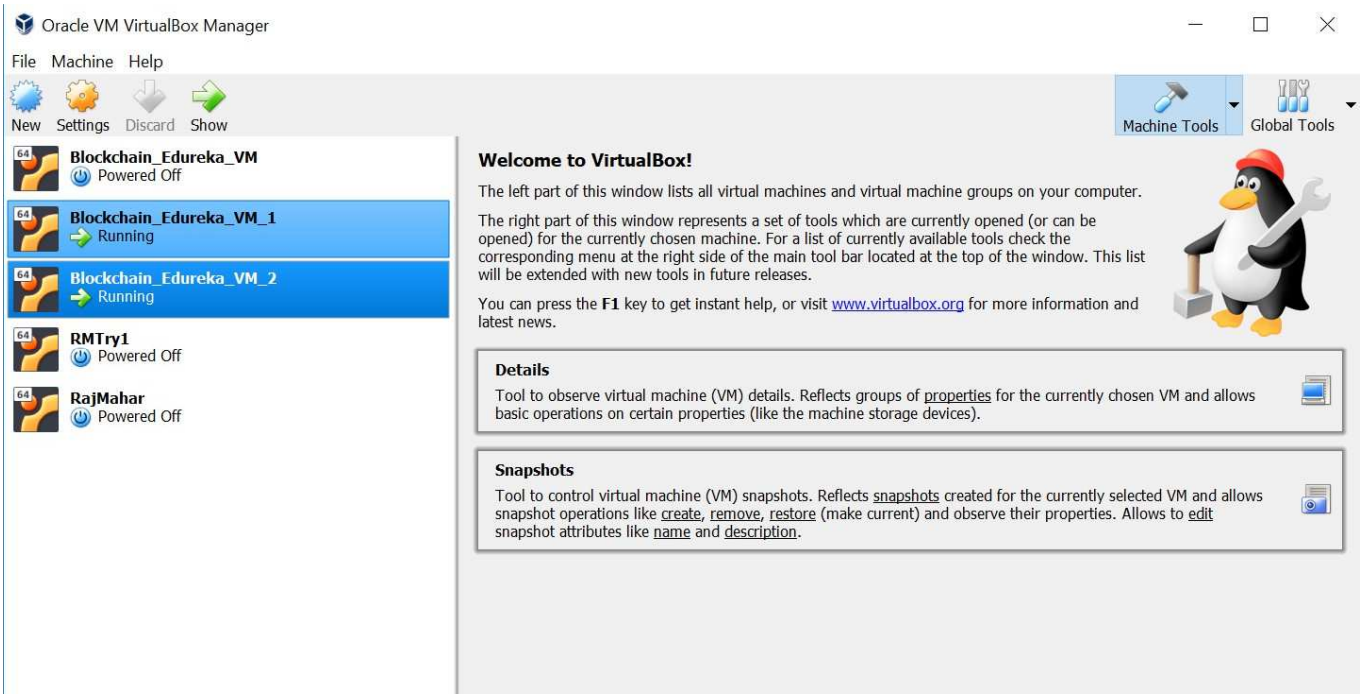
Global permissions
Anyone-can-connect= true

Native Blockchain currency
Initial-block-reward = 2562400
Minimum relay fee = 130

After modifying these parameters, deploy your MultiChain and send the screenshot.

Installing Private Multi Blockchain

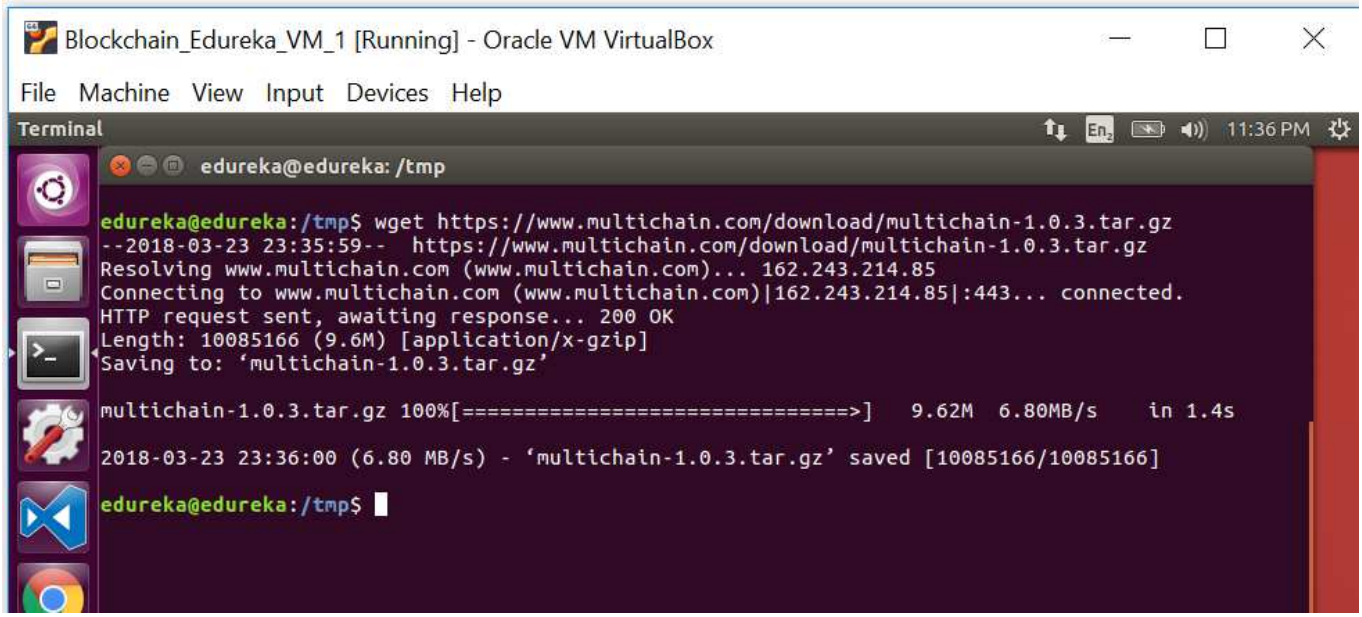
For this assignment, I am using two virtual Ubuntu 16.04 LTS machines and keep running. On both Ubuntu machines, I am going to install Multichain.



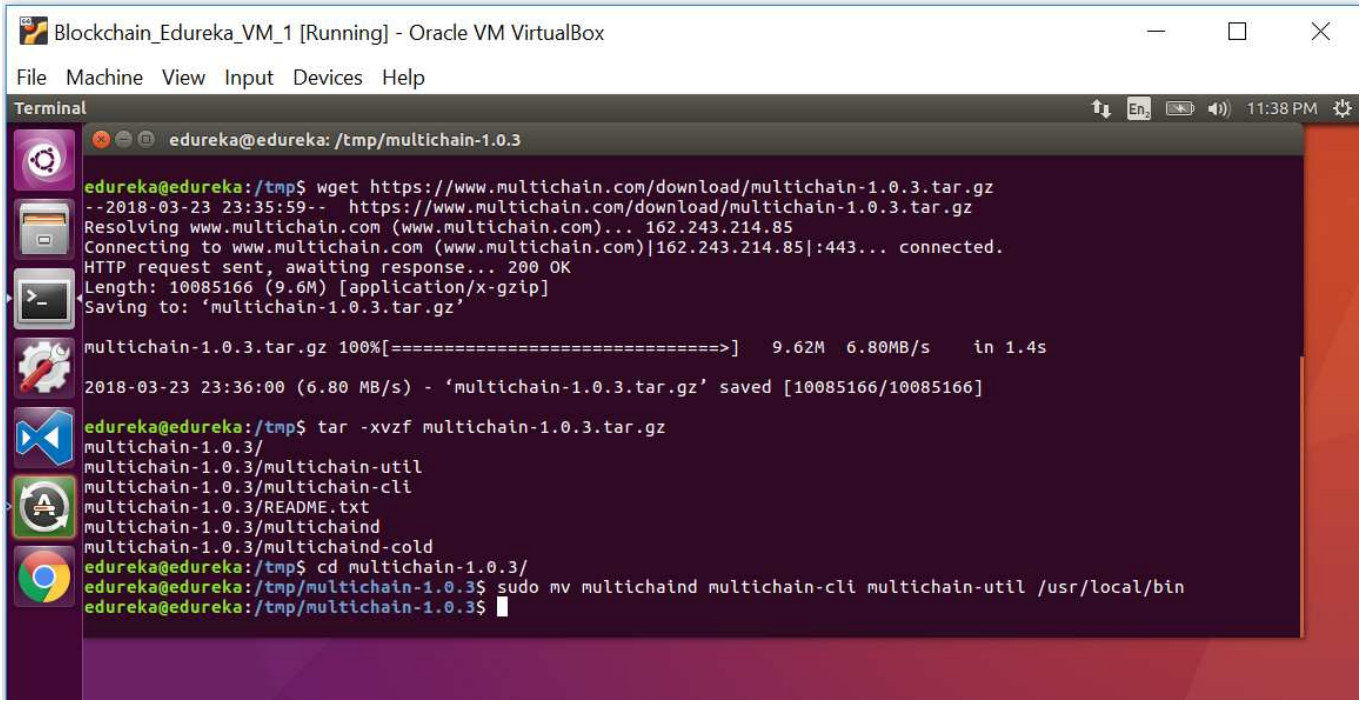
Installing Multichain

STEP 1: Run following commands on both machines

```
cd /tmp
wget https://www.multichain.com/download/multichain-1.0.3.tar.gz
```

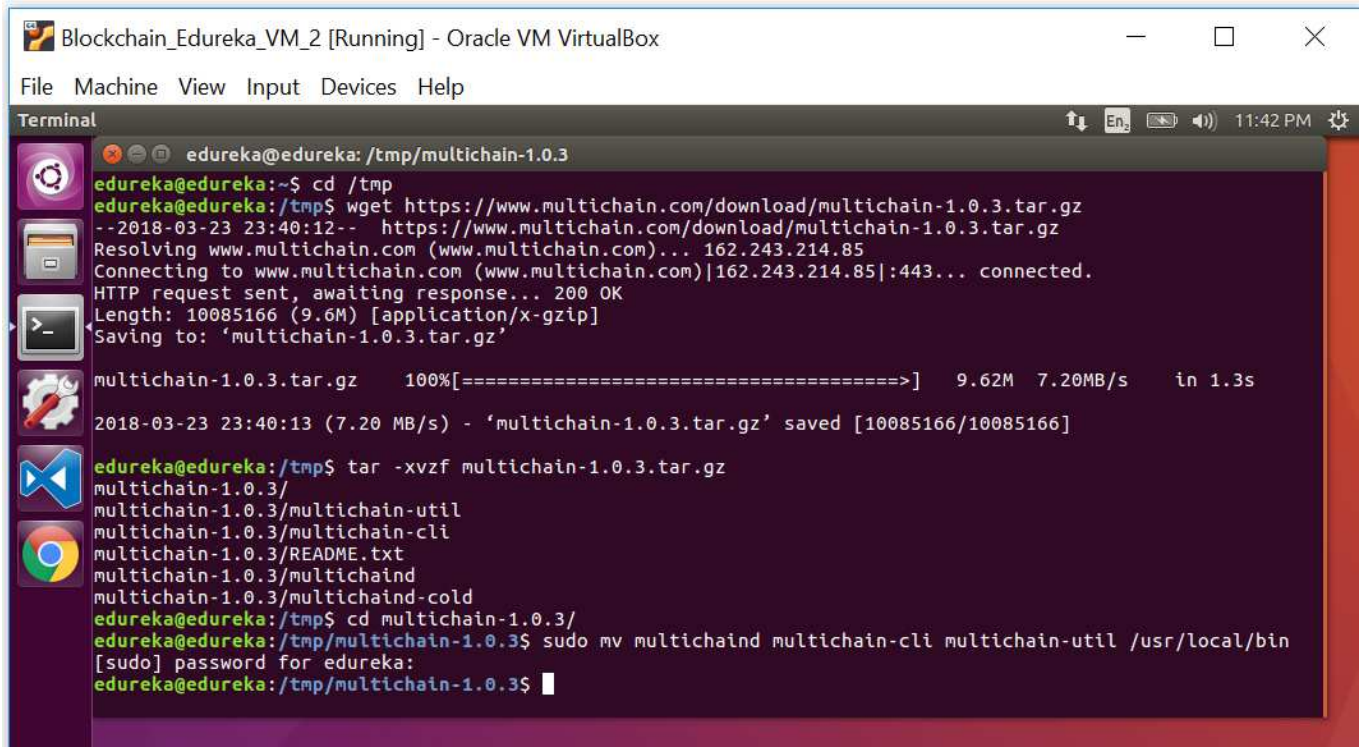



```
tar -xvzf multichain-1.0.3.tar.gz
cd multichain-1.0.3
sudo mv multichaind multichain-cli multichain-util /usr/local/bin
```



Let's select the first one (VM1) as First Server.

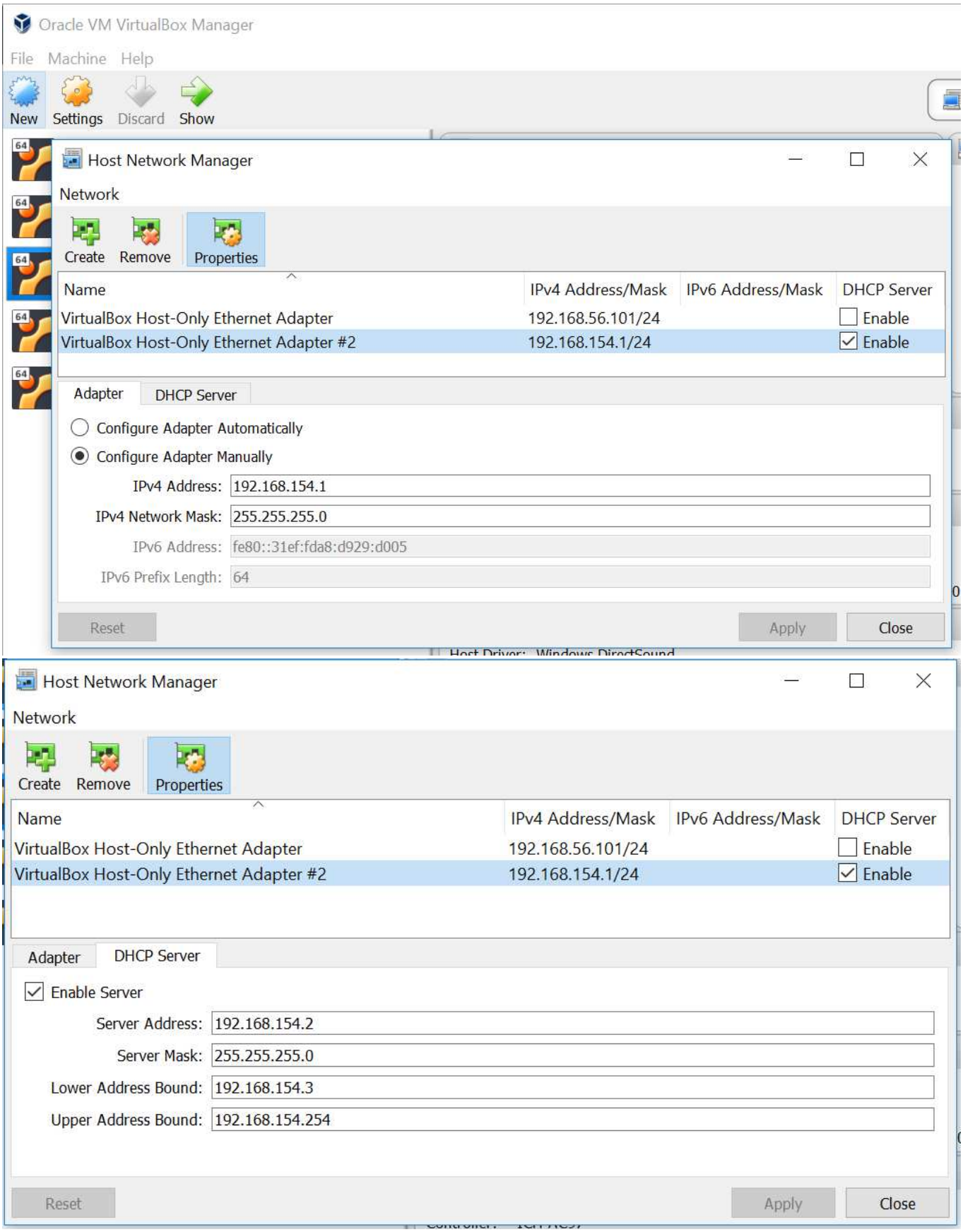
Now, let's repeat above commands to VM2



STEP 2: IP address configuration of each nodes

Please make sure that each VM node has different IP addresses. If we clone same VM, it will have same local host IPs. In that situation, we need to change the Mac address of all or some nodes so that it gives different IP addresses.

1. Go the Oracle VM VirtualBox Manager > File Menu > Host Network Manager (Ctrl + W). Click create new Host Network Manager and we are going to use this new Host Network.
   Please note the Adapter IPs and DHCP IP addresses.

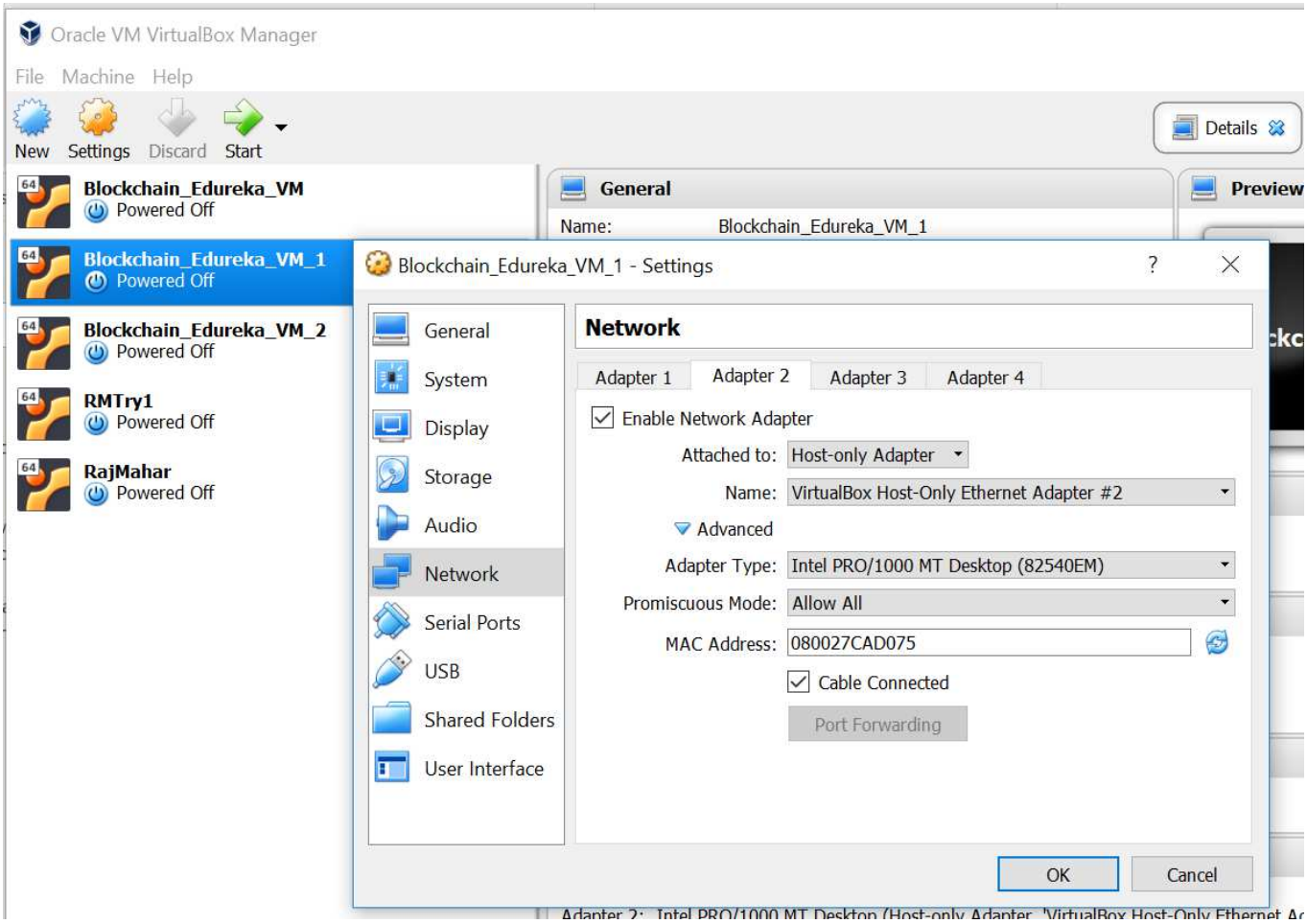

Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Show
Host Network Manager
Network
Create Remove Properties
Name IPv4 Address/Mask IPv6 Address/Mask DHCP Server
VirtualBox Host-Only Ethernet Adapter 192.168.56.101/24 Enable
VirtualBox Host-Only Ethernet Adapter #2 192.168.154.1/24 Enable
Adapter DHCP Server
Configure Adapter Automatically
Configure Adapter Manually
IPv4 Address: 192.168.154.1
IPv4 Network Mask: 255.255.255.0
IPv6 Address: fe80::31ef:fda8:d929:d005
IPv6 Prefix Length: 64
Reset Apply Close
Host Network Manager
Network
Create Remove Properties
Name IPv4 Address/Mask IPv6 Address/Mask DHCP Server
VirtualBox Host-Only Ethernet Adapter 192.168.56.101/24 Enable
VirtualBox Host-Only Ethernet Adapter #2 192.168.154.1/24 Enable
Adapter DHCP Server
Enable Server
Server Address: 192.168.154.2
Server Mask: 255.255.255.0
Lower Address Bound: 192.168.154.3
Upper Address Bound: 192.168.154.254
Reset Apply Close

2. Shutdown all VMs. Let's go again Oracle VM Virtual Manager window and change Mac address of the network card (which will change the new IP addresses in all VMs according new Network adapter configuration)

   To change this, go to Oracle VM Virtual Manager > Right click on VM1 > Settings > Network > Adapter2

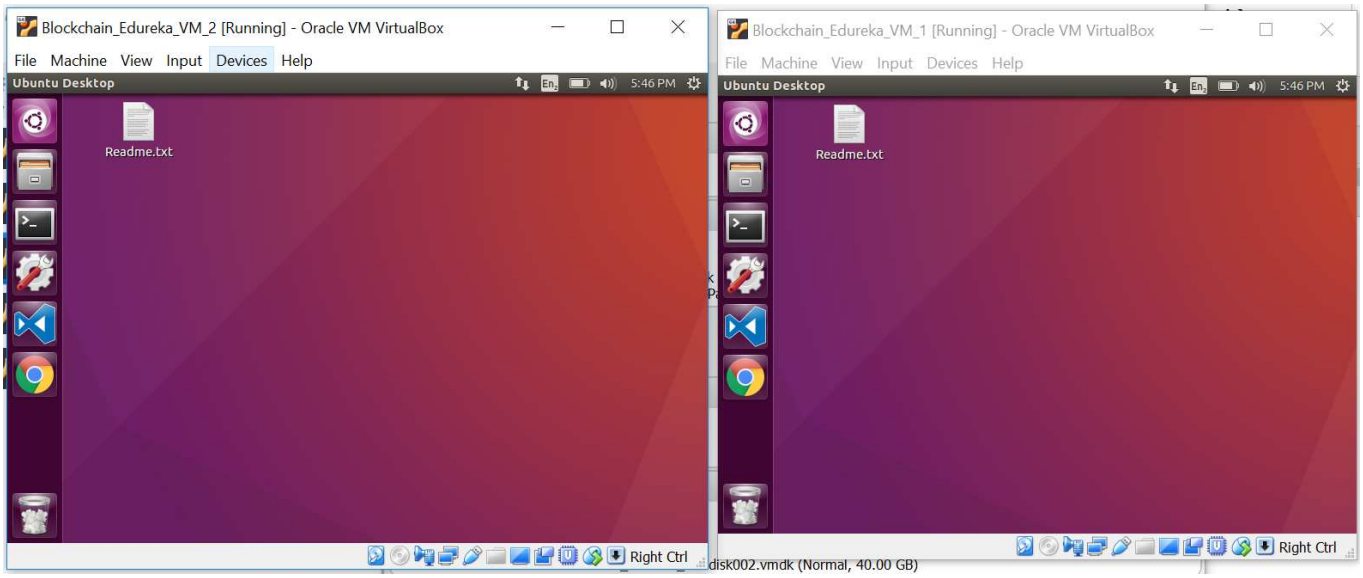
   And

   Make sure some fields are same as below
   Attached to: Host-only Adapter
   Name: VirtualBox Host-Only Ethernet Adapter #2
   Mac Address: Newly generated Mac Address (Click small Refresh icon to change Mac Address)



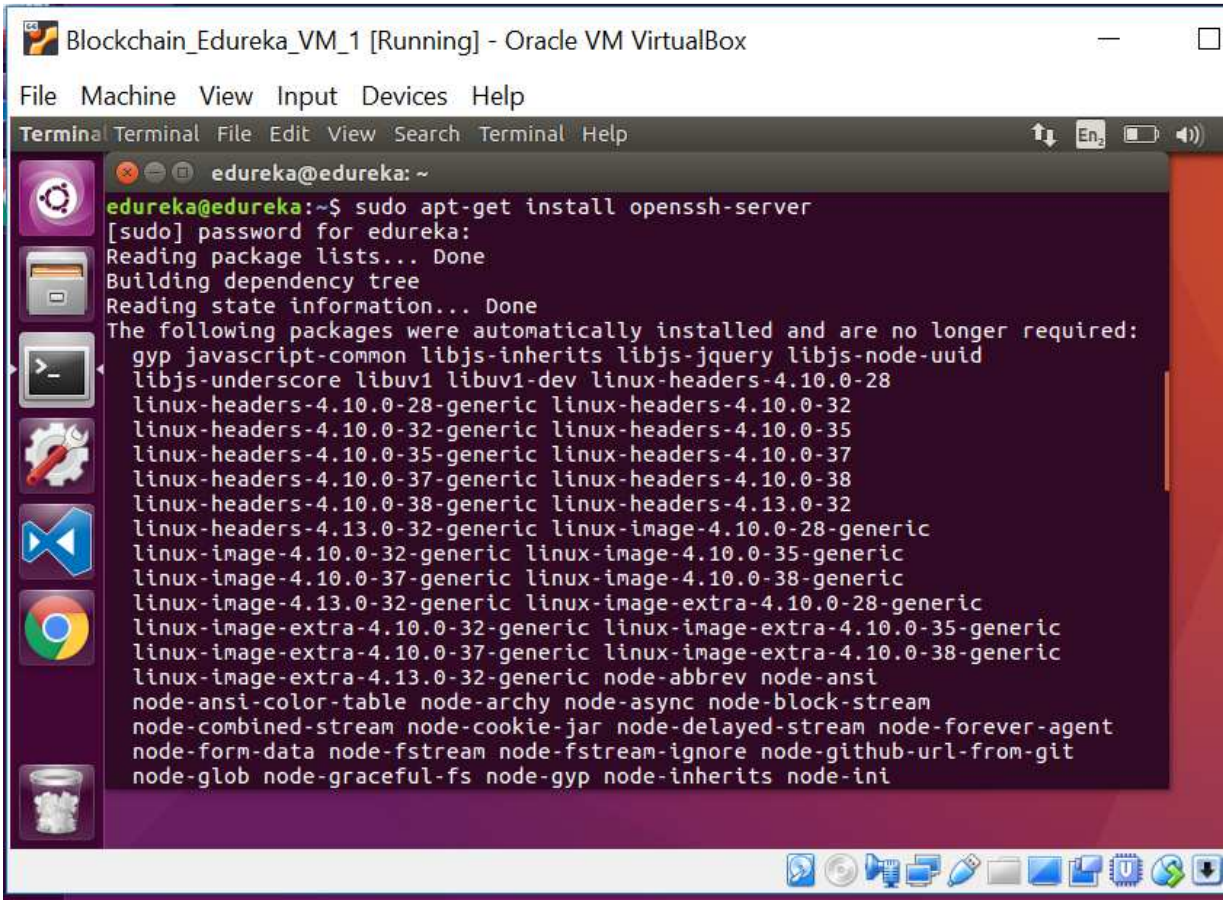
   If you want to change IP in VM2 and other nodes, you could repeat same steps.

3. Start VMs
   In Edureka VMs, you will some errors during after start VM that is because of same IP because of all VMs having same Mac address. Since we change the Mac address in our one of VM, now we don't see any error.



4. After you have configured the settings, check if ssh is installed in both the VMs. If ssh is not installed, run the following command:

```
sudo apt-get install openssh-server
```



5. After that, SSH service will be enabled in the system, we can check its status by running command:

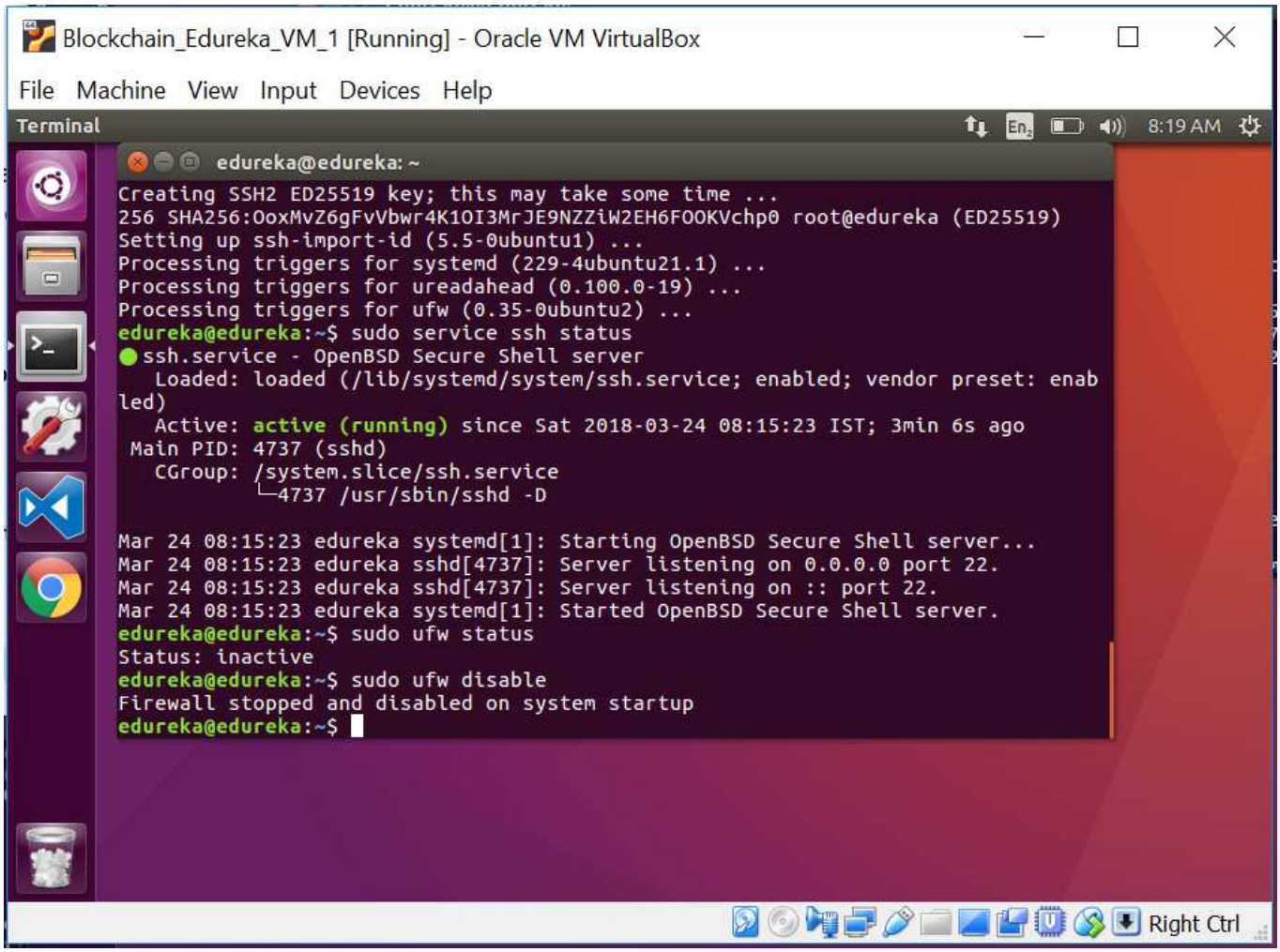
```
sudo service ssh status
```

6. Also, check for the status of firewall. Make the firewall inactive of both the VMs if the status is active. Run the following command to check the status of the firewall:

```
sudo ufw status
```

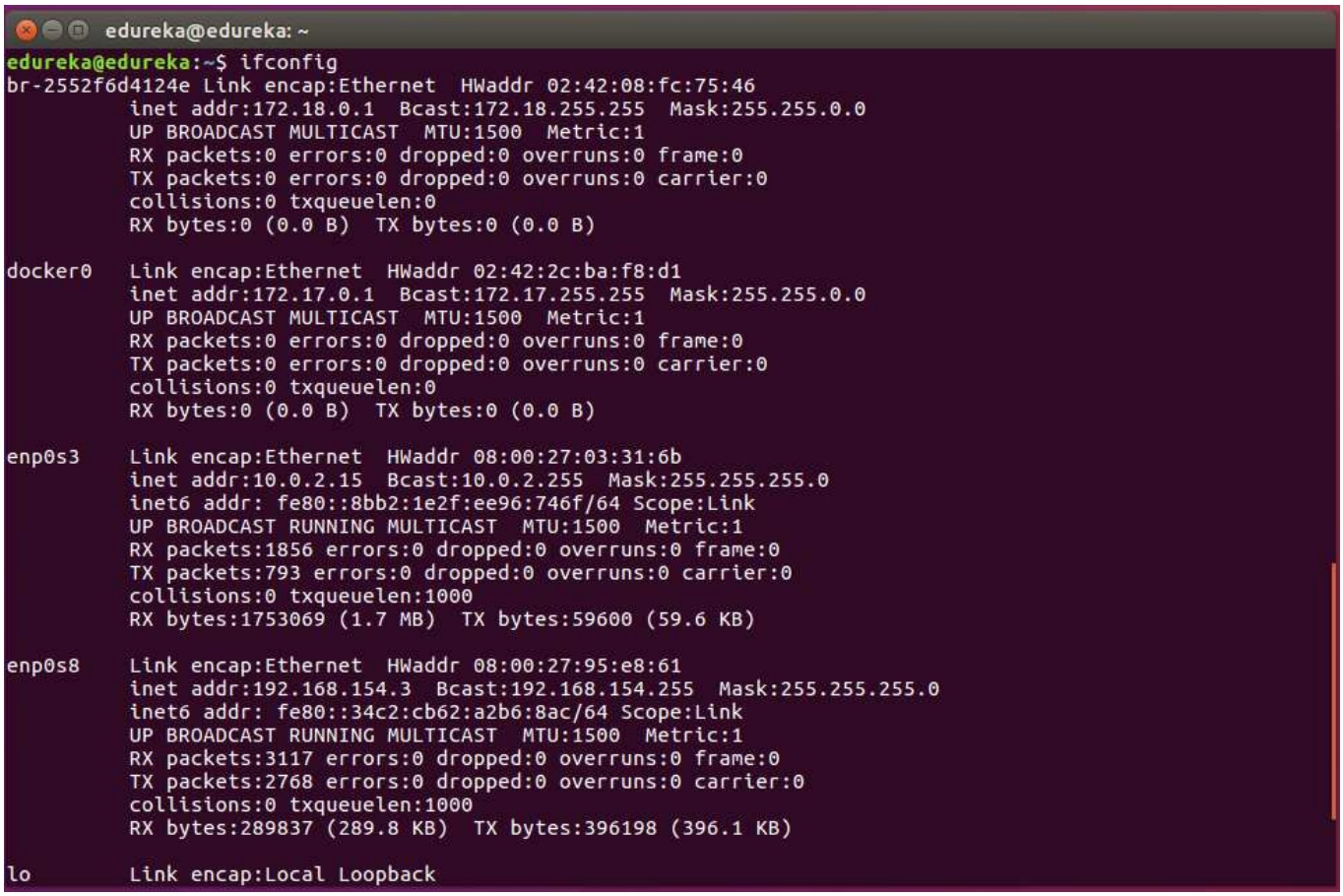
7. To disable the firewall run the following command:

```
sudo ufw disable
```

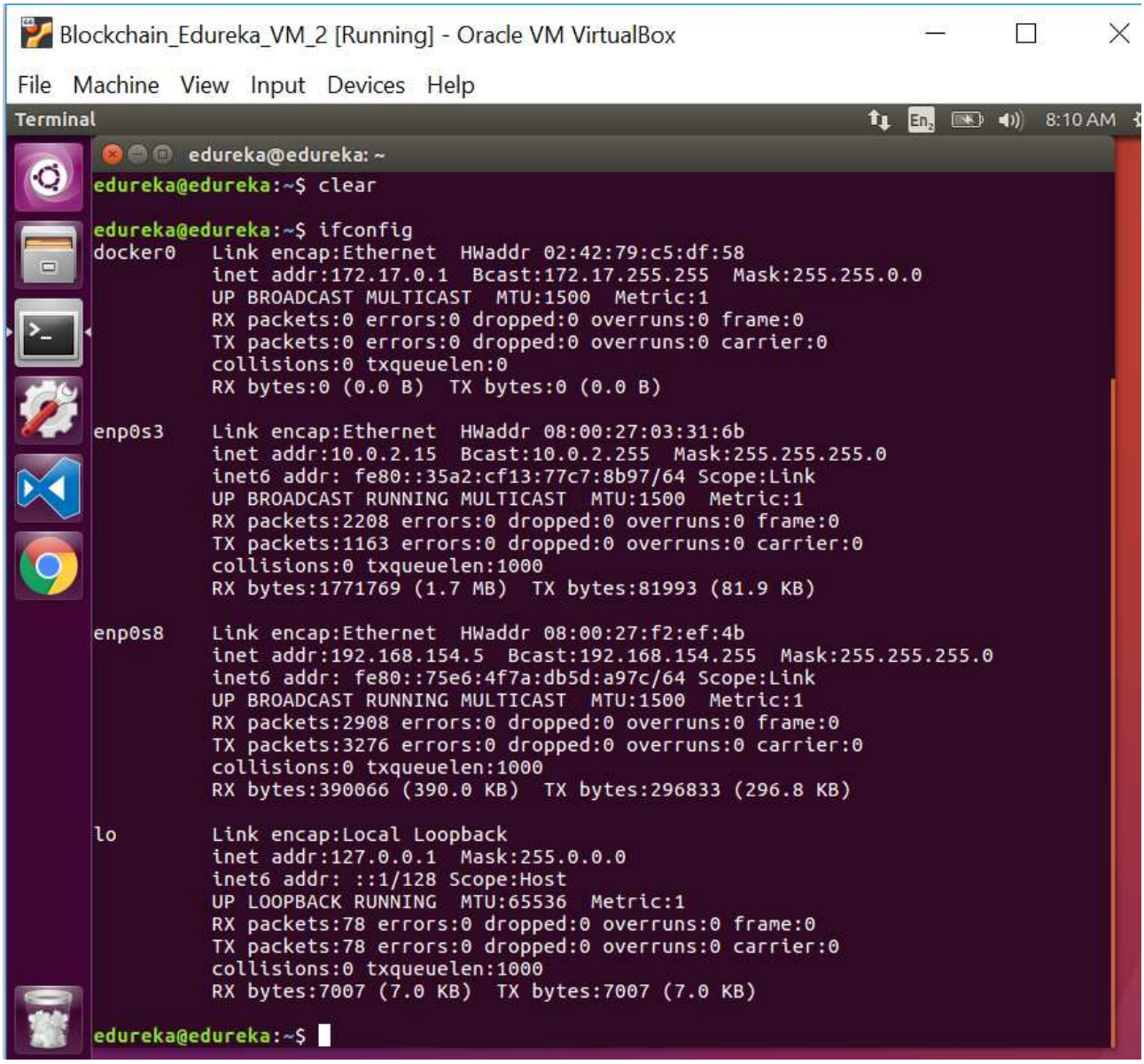


8. Check the IP address of VM1. The host only driver is enp0s8 and its IP is 192.168.154.3

```
ifconfig
```

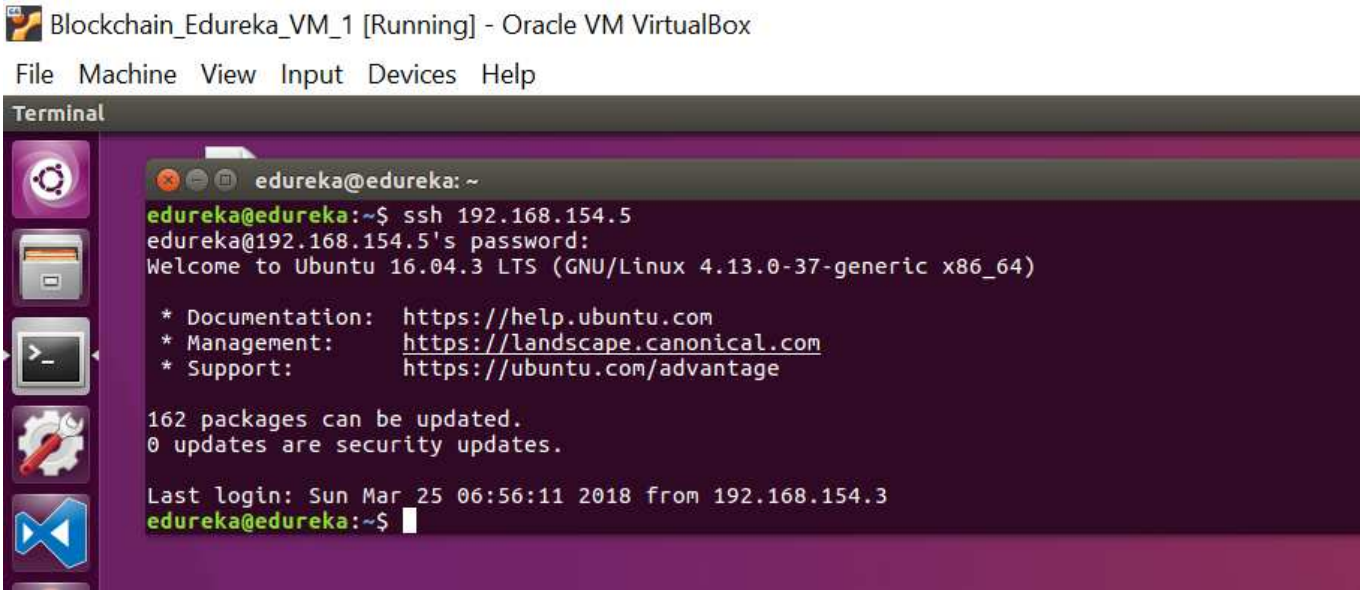


9. Check IP of VM2. Here we found enp0s8 and IP is 192.168.154.5 that is different IP than VM1



10. Execute the following command in the terminal of the first machine (VM1):

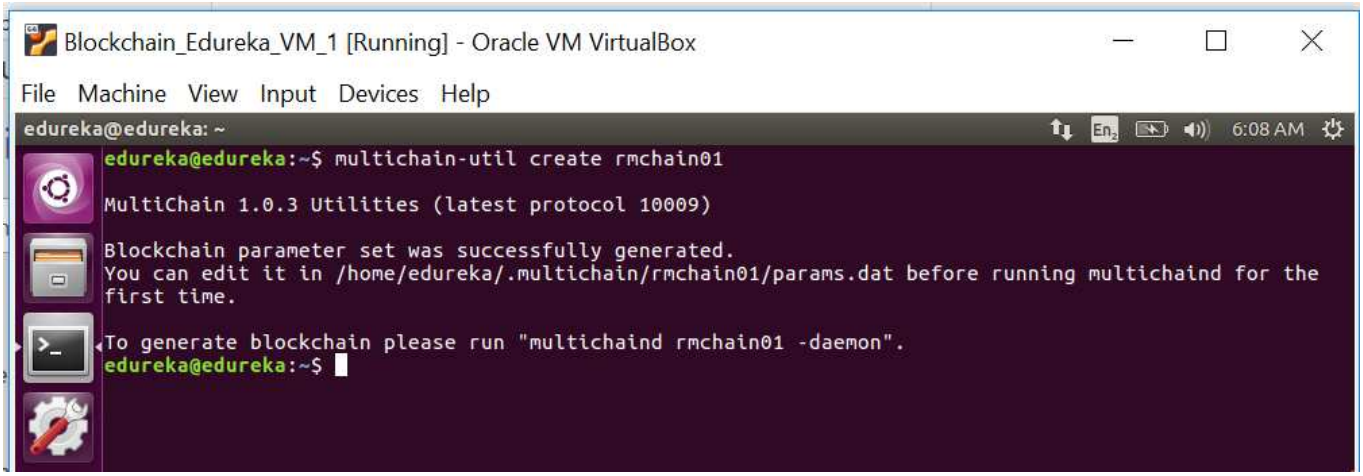
    `ssh 192.168.136.5` (This address is VM2 IP address)



STEP 2: Creating a blockchain on the first server

First, we will create a new blockchain named **rmchain01**. On first server, run below command

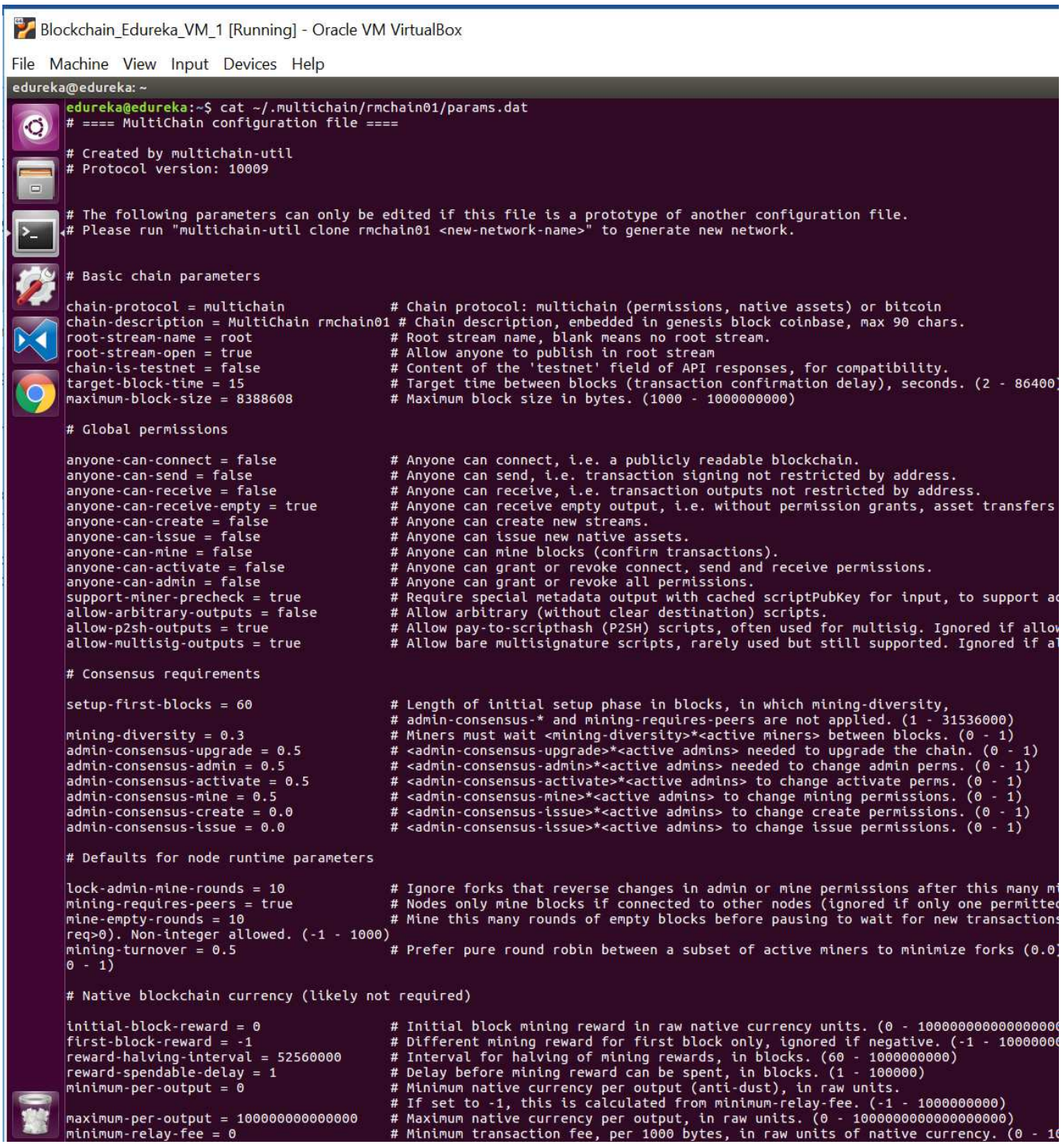
```
multichain-util create rmchain01
```



To view the blockchain's default settings (these can also be modified but we recommend using the defaults for now):

```
cat ~/.multichain/rmchain01/params.dat
```

rmchain01 Parameters



Some interesting parameters

target-block-time = 15 : Target time between blocks (transaction confirmation delay), seconds. (5 - 86400).

This is far faster than Bitcoin's 10-minute delay, which should make it far more useful for real-time transaction processing.

maximum-block-size = 1000000 **:** Maximum block size in bytes. (1000 - 1000000000)

This is the infamous "block size" parameter that is causing a crisis on the Bitcoin blockchain (June, 2016). Bitcoin is almost at its limit of transactions per block. With this value, and the 15-second delay above, this blockchain should be able to handle 40 times as many transactions per day as Bitcoin, if the transaction size is the same as Bitcoin.

skip-pow-check = false **:** Skip checking whether block hashes demonstrate proof of work.

This blockchain will force miners to work hard and prove it. This seems like a critical adjustment to me. Requiring proof-of-work makes it more expensive for legitimate miners, but it also requires attackers to work harder to corrupt the blockchain.

pow-minimum-bits = 16 **:** Initial and minimum proof of work difficulty, in leading zero bits. (1 - 32)

This means that each miner will have to calculate 2^16 hashes on average, hunting for a valid one to sign a block with. For comparison, the current bitcoin difficulty is 69.53 bits. So, it seems like a default Multichain is far easier to mine than Bitcoin, at least at first. And it also seems like the real difficulty will rapidly rise above 32 bits, if there's a free market of competing miners, and the asset value is comparable to a Bitcoin.

target-adjust-freq = 86400 : Interval between proof of work difficulty adjustments, in seconds. (3600 - 4294967295)

Bitcoin adjusts its difficulty every 14 days. This blockchain will adjust its difficulty every day. That seems sensible; roughly in accordance with the 15 second transaction delay.

Network Ports

default-network-port = 9713 : Default TCP/IP port for peer-to-peer connection with other nodes.

default-rpc-port = 9712 : Default TCP/IP port for incoming JSON-RPC API requests.

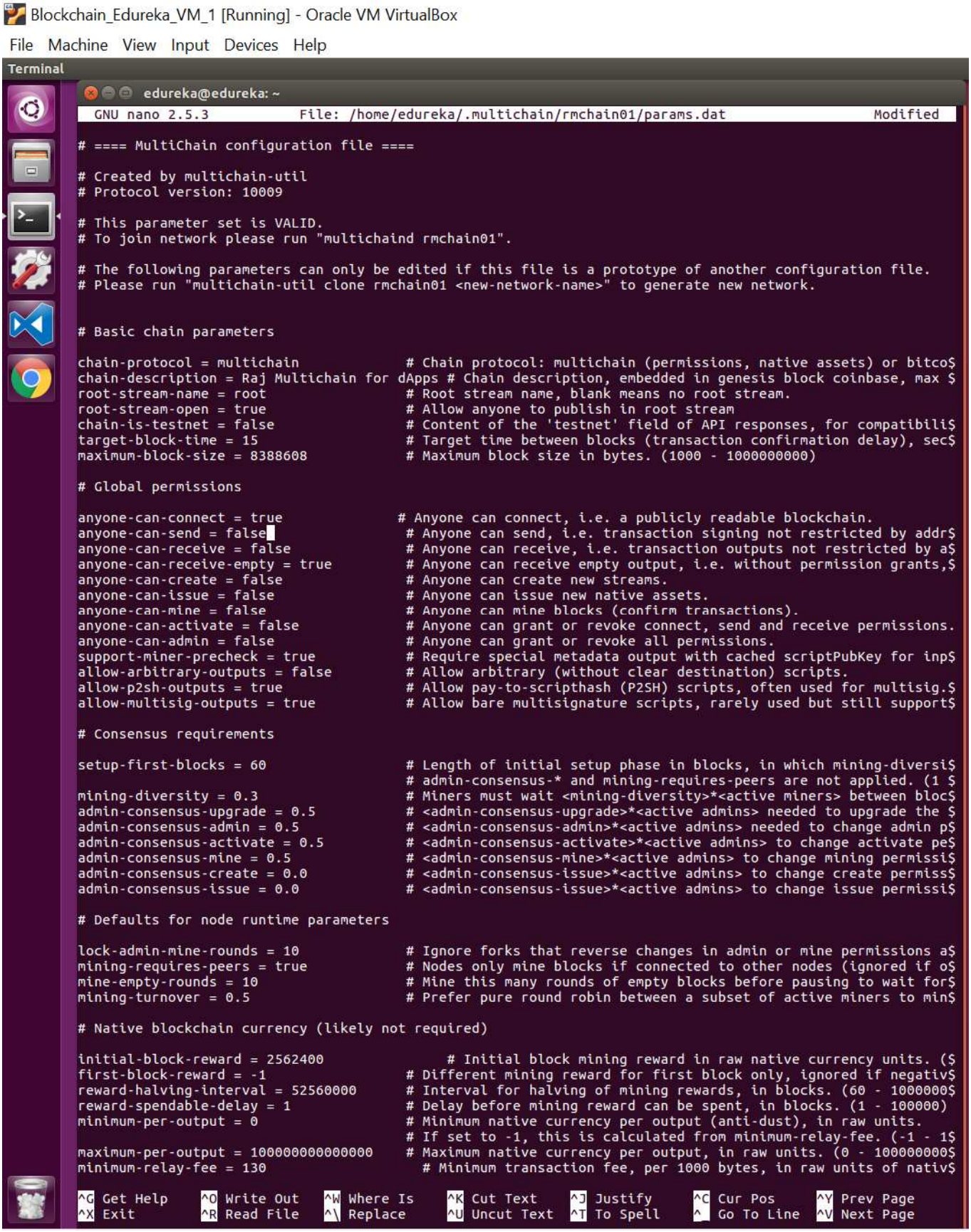
MultiCoin warns users not to change these values. It's useful to know what ports are used, so we can examine traffic with netstat, tcpdump, and wireshark later.

To make change in params.dat, run the following command

```
nano ~/.multichain/rmchain01/params.dat
```

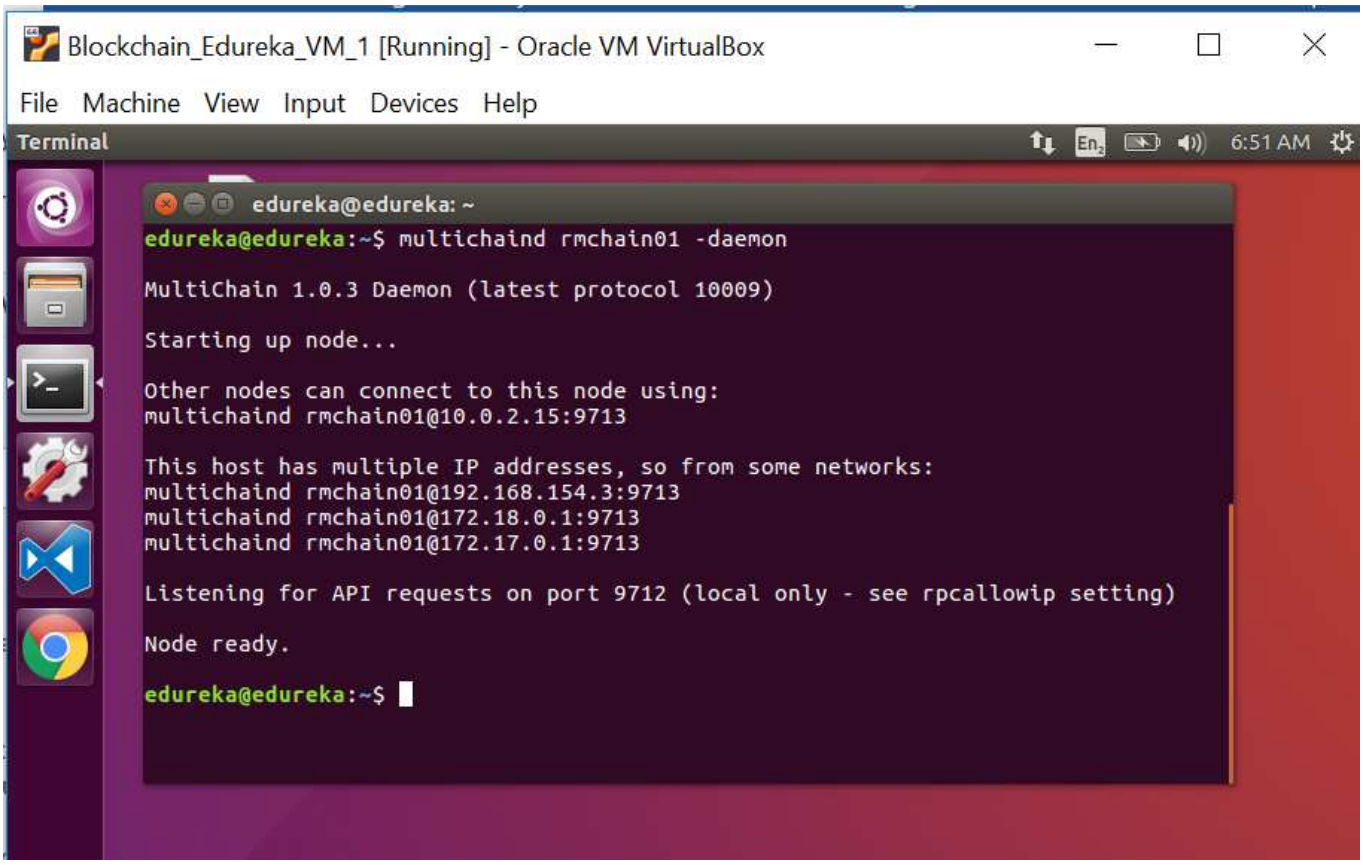
Now let's change some parameters those asked to change in this assignment

```
Chain description = Raj multichain for dApps
Anyone-can-connect = true
Initial-block-reward = 2562400
Minimum relay fee = 130
```


Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
edureka@edureka: ~
GNU nano 2.5.3 File: /home/edureka/.multichain/rmchain01/params.dat Modified
==== MultiChain configuration file ====
Created by multichain-util
Protocol version: 10009
This parameter set is VALID.
To join network please run "multichaind rmchain01".
The following parameters can only be edited if this file is a prototype of another configuration file.
Please run "multichain-util clone rmchain01 <new-network-name>" to generate new network.
Basic chain parameters
chain-protocol = multichain # Chain protocol: multichain (permissions, native assets) or bitco$
chain-description = Raj Multichain for dApps # Chain description, embedded in genesis block coinbase, max $
root-stream-name = root # Root stream name, blank means no root stream.
root-stream-open = true # Allow anyone to publish in root stream
chain-is-testnet = false # Content of the 'testnet' field of API responses, for compatibili$
target-block-time = 15 # Target time between blocks (transaction confirmation delay), sec$
maximum-block-size = 8388608 # Maximum block size in bytes. (1000 - 1000000000)
Global permissions
anyone-can-connect = true # Anyone can connect, i.e. a publicly readable blockchain.
anyone-can-send = false # Anyone can send, i.e. transaction signing not restricted by addr$
anyone-can-receive = false # Anyone can receive, i.e. transaction outputs not restricted by a$
anyone-can-receive-empty = true # Anyone can receive empty output, i.e. without permission grants,$
anyone-can-create = false # Anyone can create new streams.
anyone-can-issue = false # Anyone can issue new native assets.
anyone-can-mine = false # Anyone can mine blocks (confirm transactions).
anyone-can-activate = false # Anyone can grant or revoke connect, send and receive permissions.
anyone-can-admin = false # Anyone can grant or revoke all permissions.
support-miner-precheck = true # Require special metadata output with cached scriptPubKey for inp$
allow-arbitrary-outputs = false # Allow arbitrary (without clear destination) scripts.
allow-p2sh-outputs = true # Allow pay-to-scripthash (P2SH) scripts, often used for multisig.$
allow-multisig-outputs = true # Allow bare multisignature scripts, rarely used but still support$
Consensus requirements
setup-first-blocks = 60 # Length of initial setup phase in blocks, in which mining-diversi$
admin-consensus-* and mining-requires-peers are not applied. (1 $
mining-diversity = 0.3 # Miners must wait <mining-diversity>*<active miners> between bloc$
admin-consensus-upgrade = 0.5 # <admin-consensus-upgrade>*<active admins> needed to upgrade the $
admin-consensus-admin = 0.5 # <admin-consensus-admin>*<active admins> needed to change admin p$
admin-consensus-activate = 0.5 # <admin-consensus-activate>*<active admins> to change activate pe$
admin-consensus-mine = 0.5 # <admin-consensus-mine>*<active admins> to change mining permissi$
admin-consensus-create = 0.0 # <admin-consensus-issue>*<active admins> to change create permiss$
admin-consensus-issue = 0.0 # <admin-consensus-issue>*<active admins> to change issue permissi$
Defaults for node runtime parameters
lock-admin-mine-rounds = 10 # Ignore forks that reverse changes in admin or mine permissions a$
mining-requires-peers = true # Nodes only mine blocks if connected to other nodes (ignored if o$
mine-empty-rounds = 10 # Mine this many rounds of empty blocks before pausing to wait for$
mining-turnover = 0.5 # Prefer pure round robin between a subset of active miners to min$
Native blockchain currency (likely not required)
initial-block-reward = 2562400 # Initial block mining reward in raw native currency units. ($
first-block-reward = -1 # Different mining reward for first block only, ignored if negativ$
reward-halving-interval = 52560000 # Interval for halving of mining rewards, in blocks. (60 - 1000000$
reward-spendable-delay = 1 # Delay before mining reward can be spent, in blocks. (1 - 100000)
minimum-per-output = 0 # Minimum native currency per output (anti-dust), in raw units.
If set to -1, this is calculated from minimum-relay-fee. (-1 - 1$
maximum-per-output = 100000000000000 # Maximum native currency per output, in raw units. (0 - 100000000$
minimum-relay-fee = 130 # Minimum transaction fee, per 1000 bytes, in raw units of nativ$
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page

Initialize the blockchain, including mining the genesis block:

```
multichaind rmchain01 -daemon
```



Above command starts the server and the genesis block will be found. It gives now the node address that others can use to connect to this chain.

Make a note of this node address. In the picture above, it is

rmchain01@10.0.2.15:9713

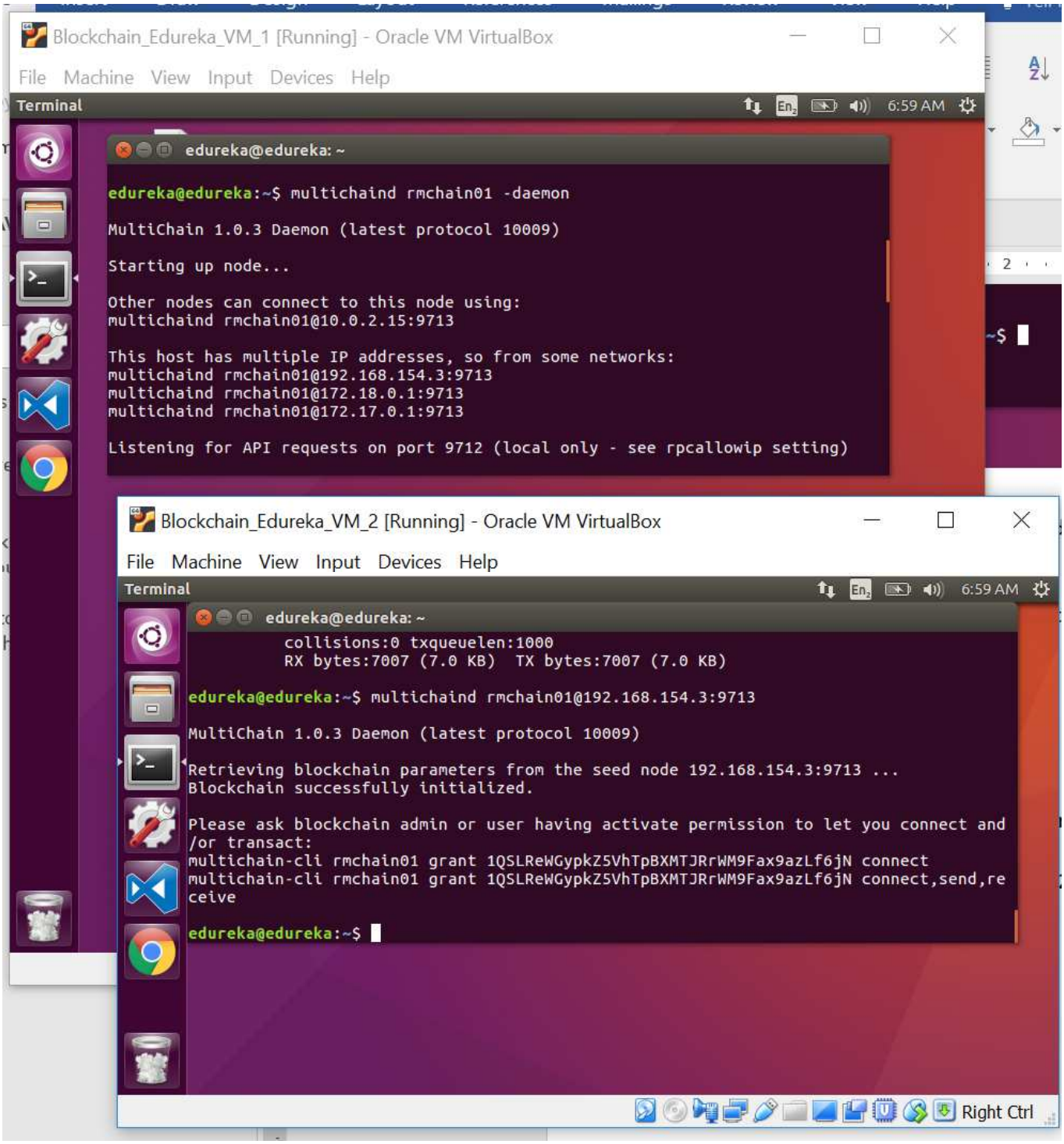
or

rmchain01@192.168.154.3:9713

STEP 3: Connecting to a blockchain

Now let's connect to this blockchain from second server VM2 using following command.

`multichaind` **rmchain01@192.168.154.3:9713**



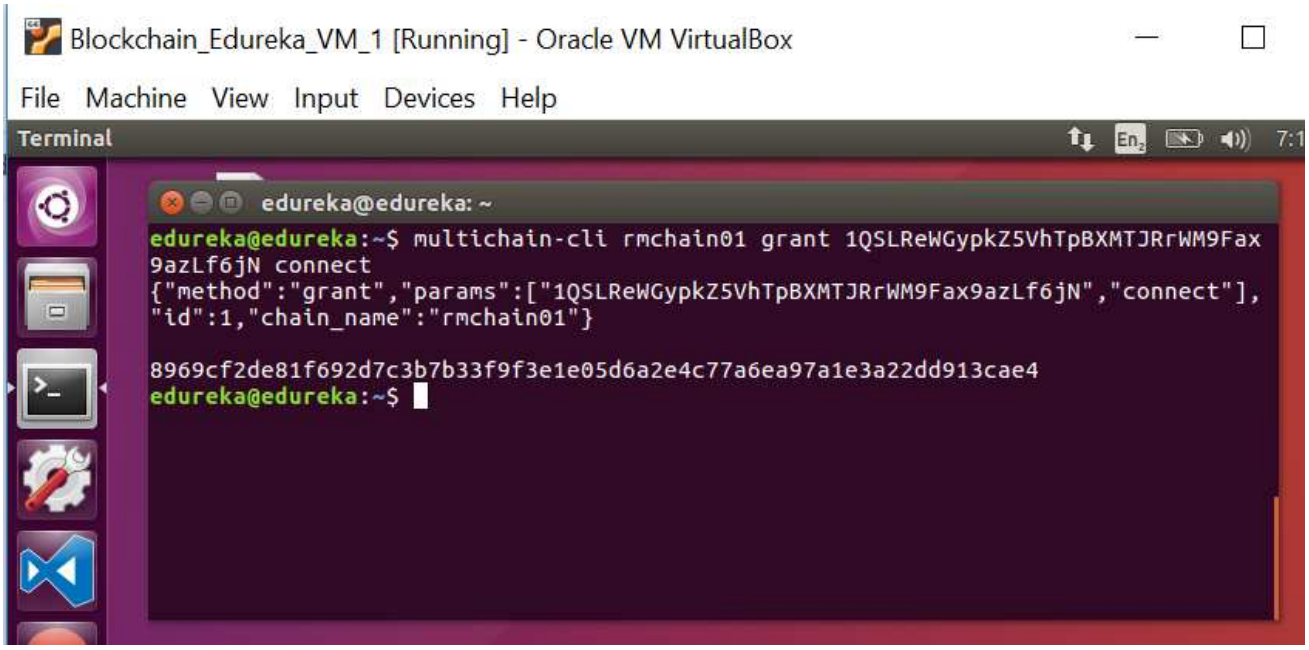
If there is an error, please try to connect the VM1 server from VM2 running following command

```
ssh VM1 IP address (192.168.154.3)
```

It has said Blockchain successfully initialized and it asked to grant access to specific address to be connected.

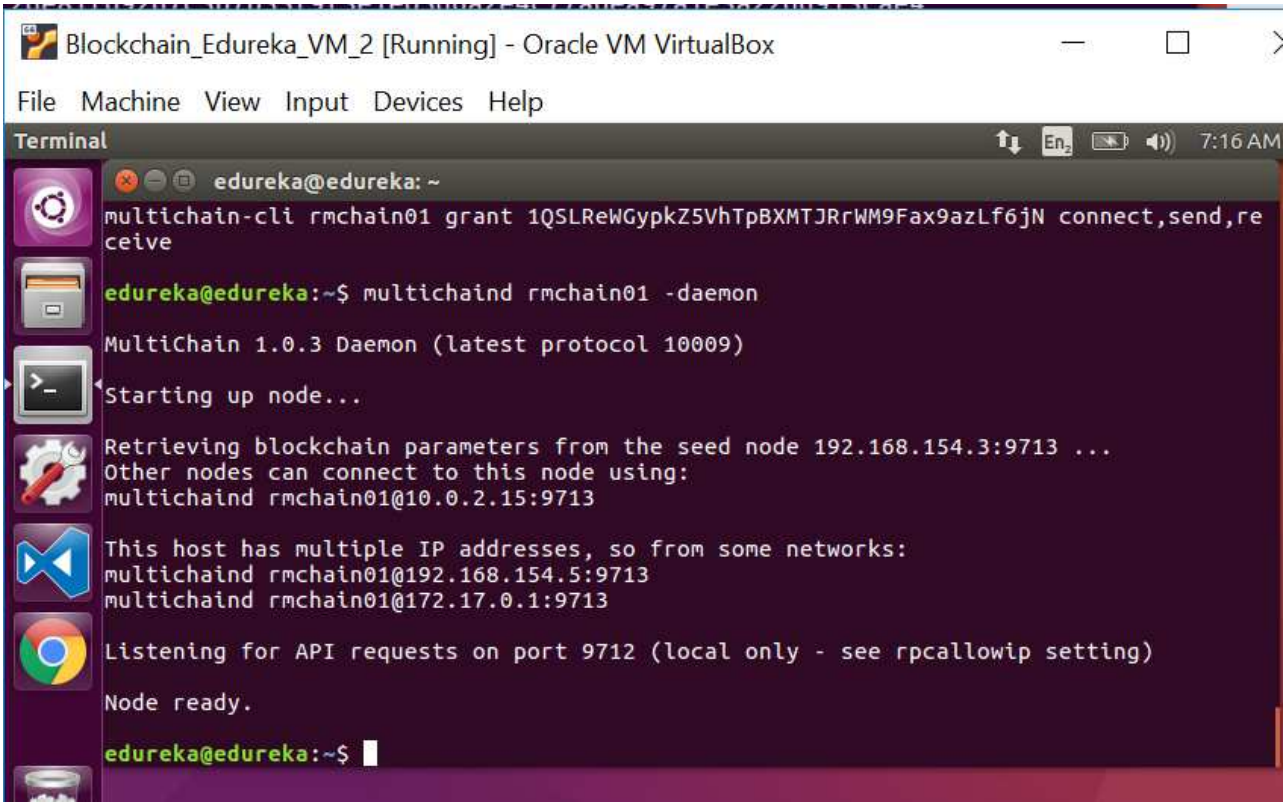
Back on the first server VM1, add connection permissions for this address:

```
multichain-cli rmchain01 grant 1QSLReWGypkZ5VhTpBXMTJRrWM9Fax9azLf6jN
connect
```



Now try reconnecting again from the second server:

```
multichaind rmchain01 -daemon
```

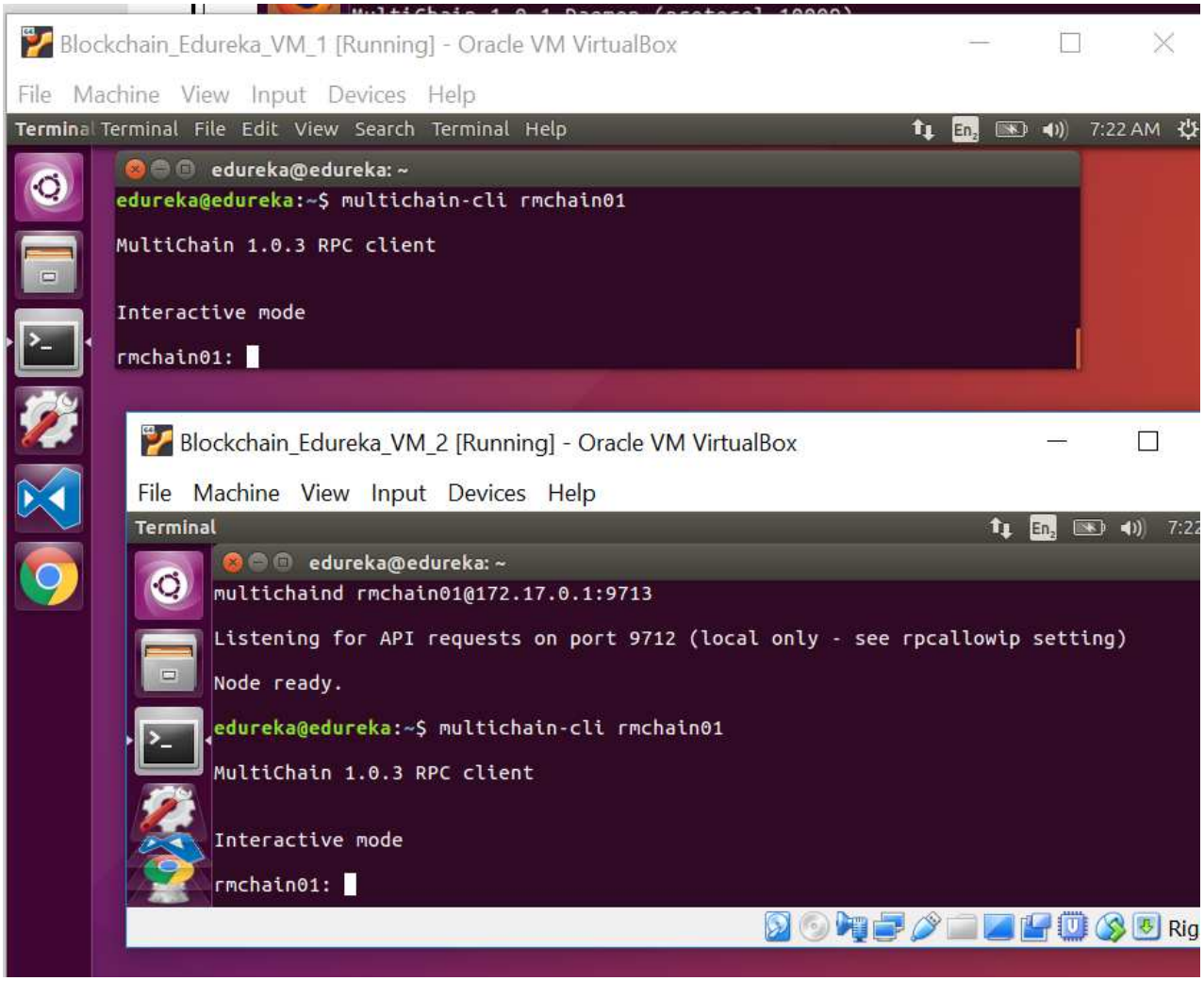


This shows now second node VM2 also started.

MultiChain interactive mode

Before we proceed, let's enter interactive mode so we can issue commands without typing multichain-cli rmchain01 every time. On both servers:

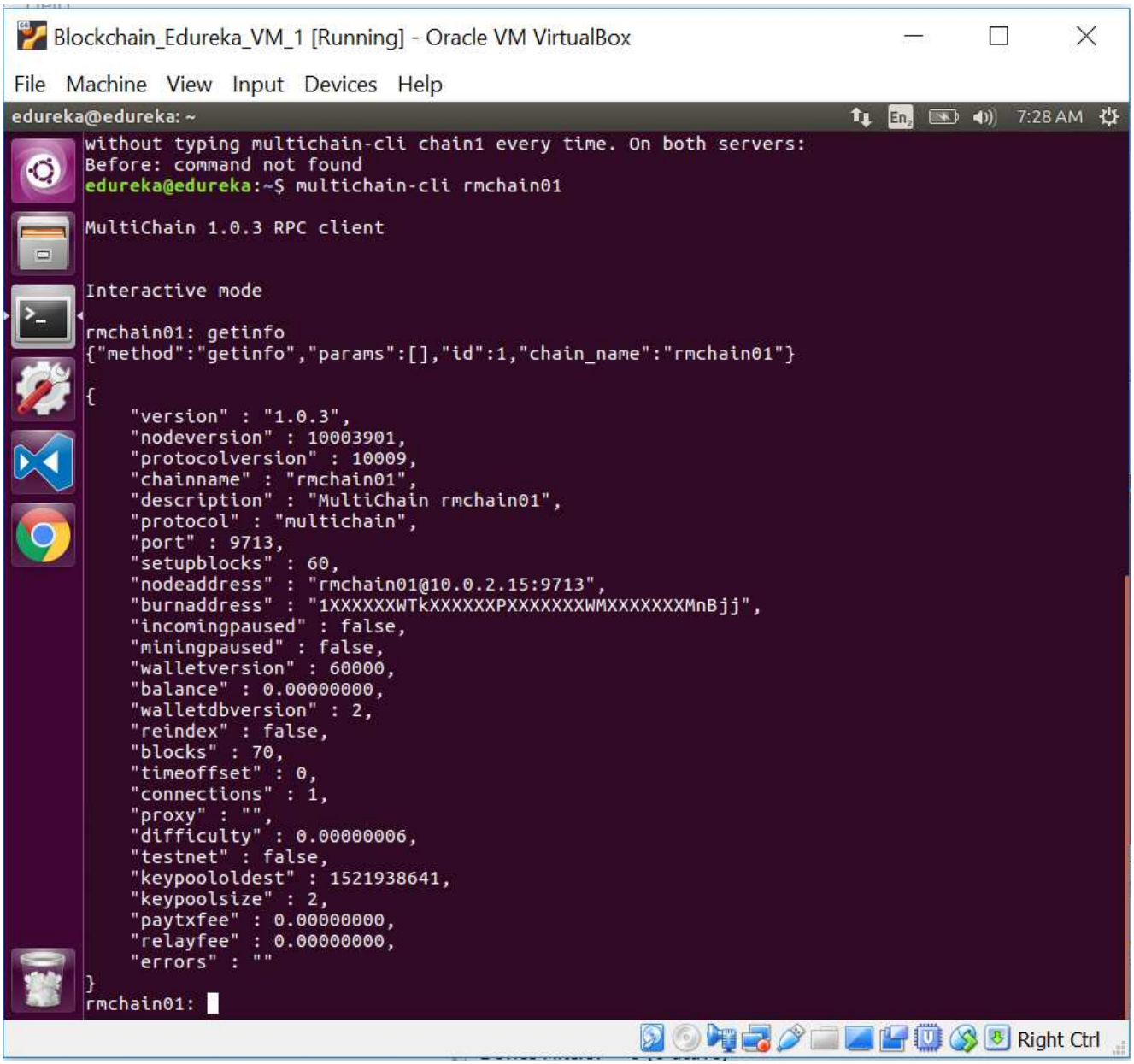
```
multichain-cli rmchain01
```



Now that the blockchain is working on two nodes, we can run the commands in this section on either or both.

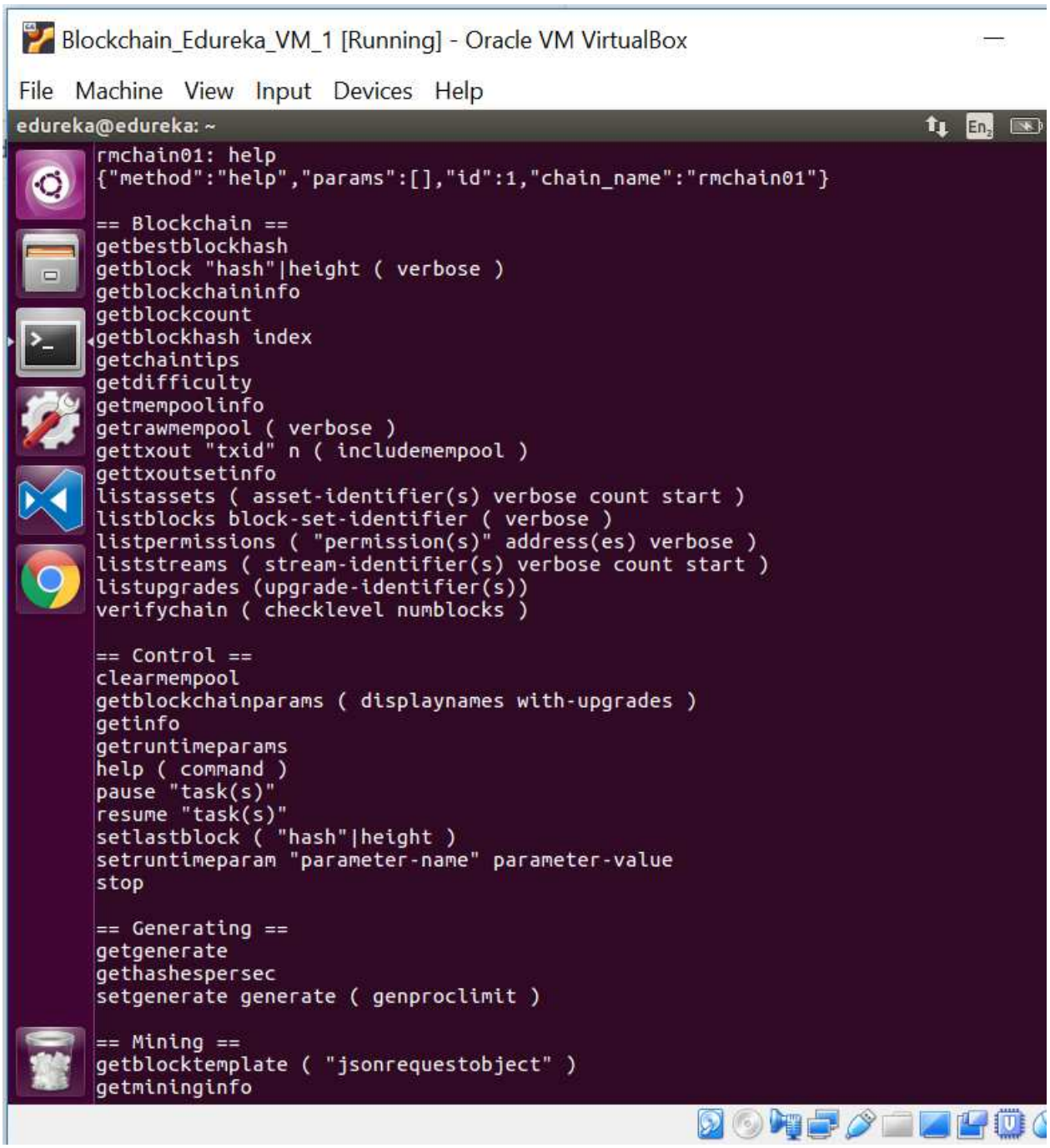
To get general information use following command

```
getinfo
```



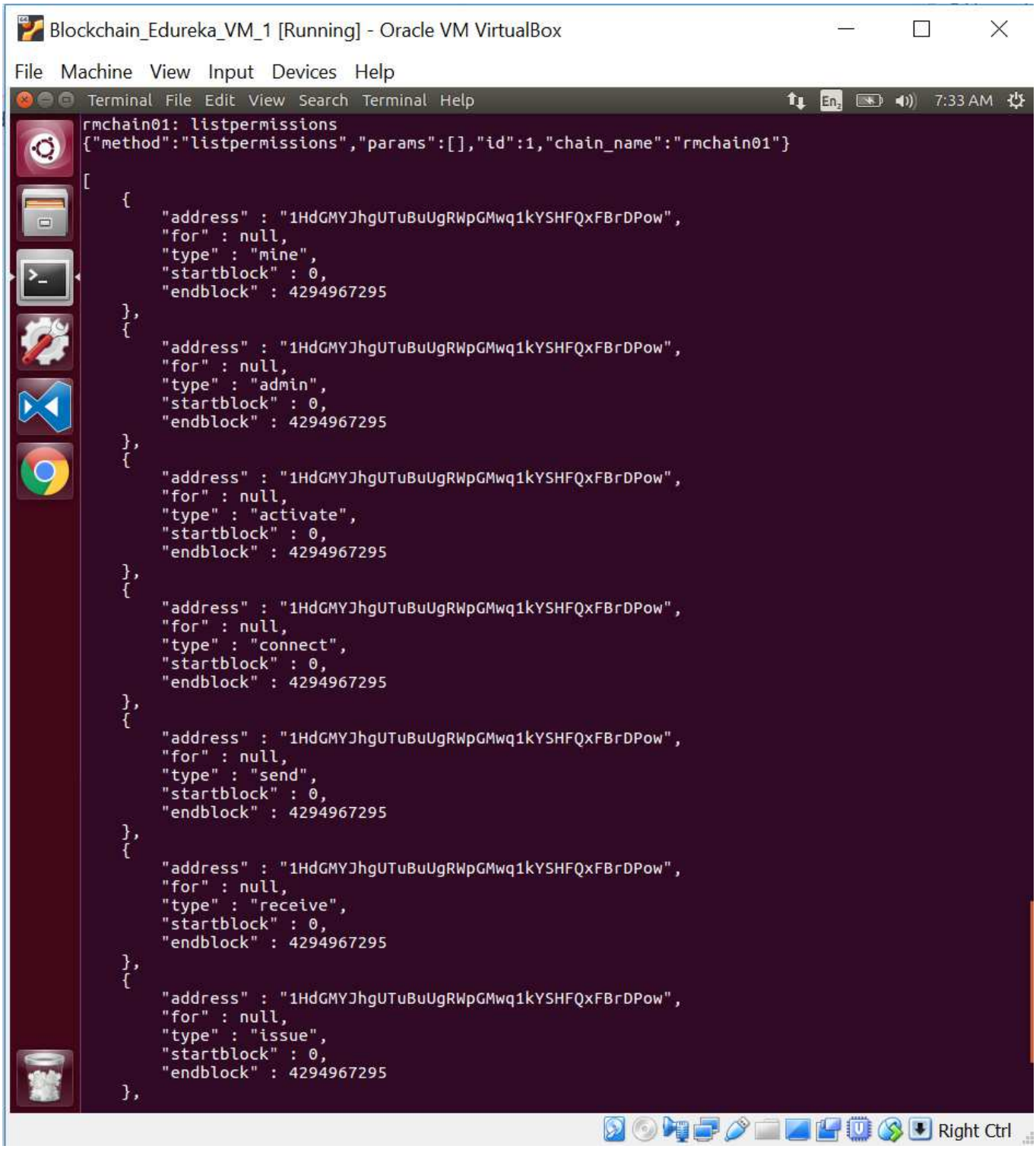
See a list of all available commands:

```
help
```


Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
edureka@edureka: ~
rmchain01: help
{"method":"help","params":[],"id":1,"chain_name":"rmchain01"}
== Blockchain ==
getbestblockhash
getblock "hash"|height (verbose)
getblockchaininfo
getblockcount
getblockhash index
getchaintips
getdifficulty
getmempoolinfo
getrawmempool (verbose)
gettxout "txid" n (includemempool)
gettxoutsetinfo
listassets (asset-identifier(s) verbose count start)
listblocks block-set-identifier (verbose)
listpermissions ("permission(s)" address(es) verbose)
liststreams (stream-identifier(s) verbose count start)
listupgrades (upgrade-identifier(s))
verifychain (checklevel numblocks)
== Control ==
clearmempool
getblockchainparams (displaynames with-upgrades)
getinfo
getruntimeparams
help (command)
pause "task(s)"
resume "task(s)"
setlastblock ("hash"|height)
setruntimeparam "parameter-name" parameter-value
stop
== Generating ==
getgenerate
gethashespersec
setgenerate generate (genproclimit)
== Mining ==
getblocktemplate ("jsonrequestobject")
getmininginfo

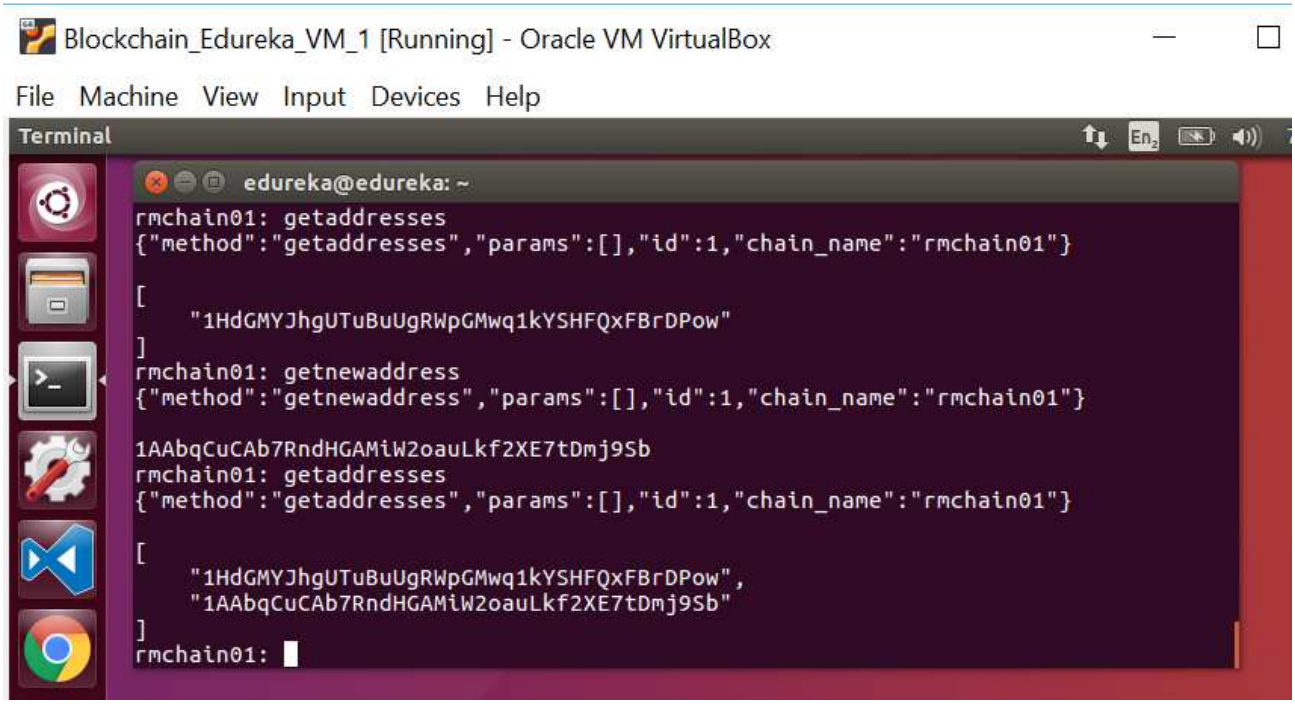
Show all permissions currently assigned:

```
listpermissions
```



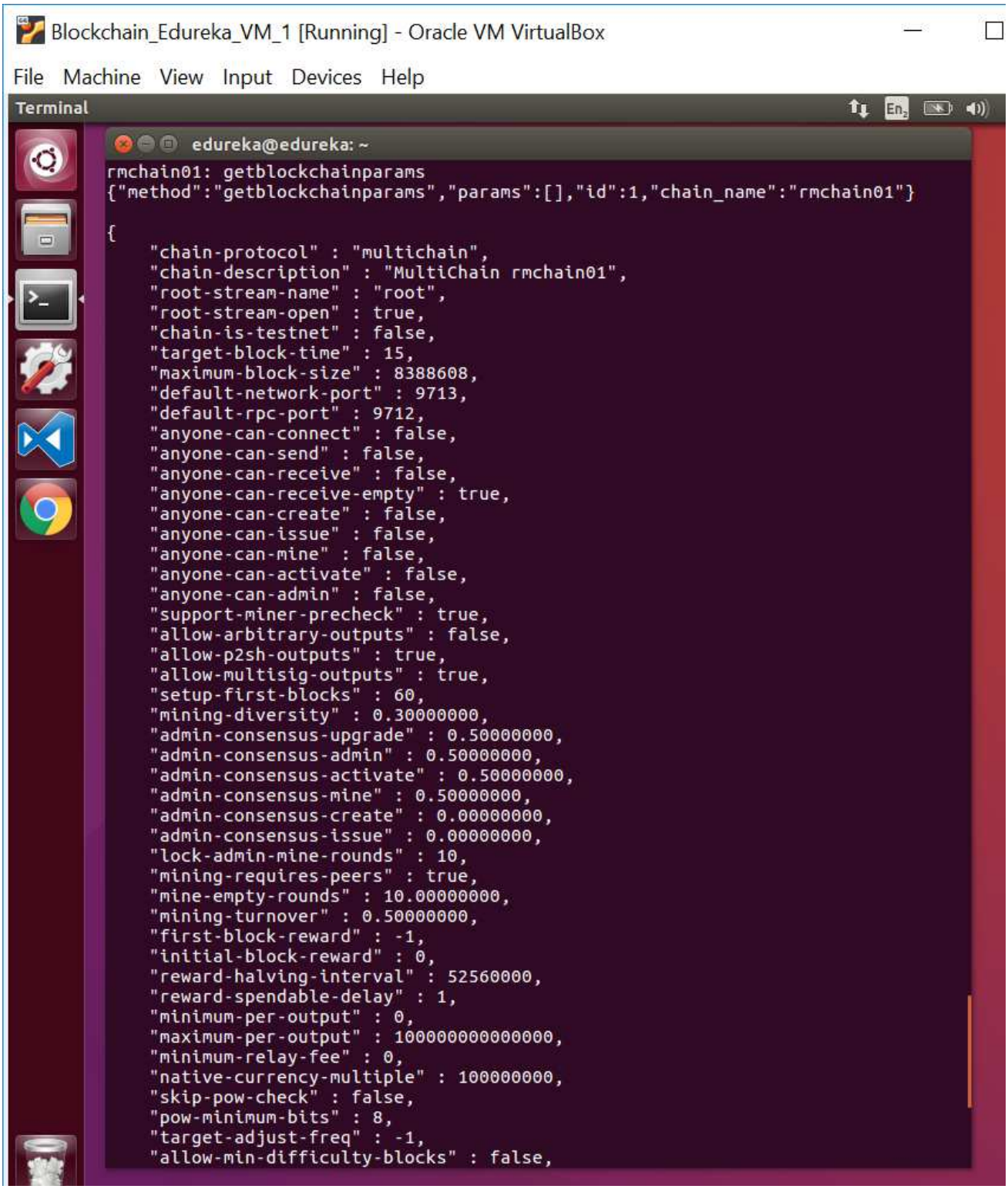
To get new address use command `getnewaddress`

And to list all addresses, use command `getaddresses`



Get the parameters of this blockchain (based on params.dat file):

`getblockchainparams`


Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
edureka@edureka: ~
rmchain01: getblockchainparams
{"method":"getblockchainparams","params":[],"id":1,"chain_name":"rmchain01"}
{
"chain-protocol" : "multichain",
"chain-description" : "MultiChain rmchain01",
"root-stream-name" : "root",
"root-stream-open" : true,
"chain-is-testnet" : false,
"target-block-time" : 15,
"maximum-block-size" : 8388608,
"default-network-port" : 9713,
"default-rpc-port" : 9712,
"anyone-can-connect" : false,
"anyone-can-send" : false,
"anyone-can-receive" : false,
"anyone-can-receive-empty" : true,
"anyone-can-create" : false,
"anyone-can-issue" : false,
"anyone-can-mine" : false,
"anyone-can-activate" : false,
"anyone-can-admin" : false,
"support-miner-precheck" : true,
"allow-arbitrary-outputs" : false,
"allow-p2sh-outputs" : true,
"allow-multisig-outputs" : true,
"setup-first-blocks" : 60,
"mining-diversity" : 0.30000000,
"admin-consensus-upgrade" : 0.50000000,
"admin-consensus-admin" : 0.50000000,
"admin-consensus-activate" : 0.50000000,
"admin-consensus-mine" : 0.50000000,
"admin-consensus-create" : 0.00000000,
"admin-consensus-issue" : 0.00000000,
"lock-admin-mine-rounds" : 10,
"mining-requires-peers" : true,
"mine-empty-rounds" : 10.00000000,
"mining-turnover" : 0.50000000,
"first-block-reward" : -1,
"initial-block-reward" : 0,
"reward-halving-interval" : 52560000,
"reward-spendable-delay" : 1,
"minimum-per-output" : 0,
"maximum-per-output" : 100000000000000,
"minimum-relay-fee" : 0,
"native-currency-multiple" : 100000000,
"skip-pow-check" : false,
"pow-minimum-bits" : 8,
"target-adjust-freq" : -1,
"allow-min-difficulty-blocks" : false,

For each node, get a list of connected peers:

```
getpeerinfo
```