

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

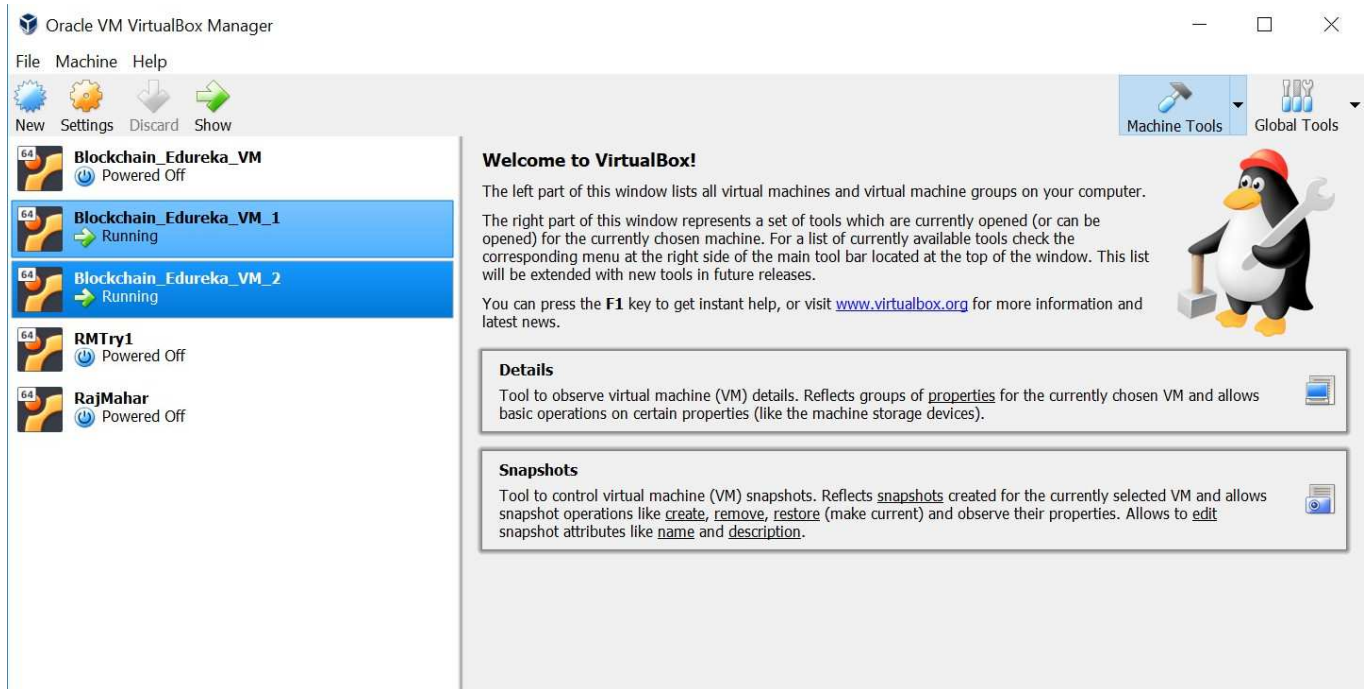
Initial-block-reward = 2562400

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
```

Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

```

edureka@edureka: /tmp
edureka@edureka: /tmp$ wget https://www.multichain.com/download/multichain-1.0.3.tar.gz
--2018-03-23 23:35:59-- https://www.multichain.com/download/multichain-1.0.3.tar.gz
Resolving www.multichain.com (www.multichain.com)... 162.243.214.85
Connecting to www.multichain.com (www.multichain.com)|162.243.214.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10085166 (9.6M) [application/x-gzip]
Saving to: 'multichain-1.0.3.tar.gz'

multichain-1.0.3.tar.gz 100%[=====] 9.62M 6.80MB/s in 1.4s
2018-03-23 23:36:00 (6.80 MB/s) - 'multichain-1.0.3.tar.gz' saved [10085166/10085166]

edureka@edureka: /tmp$

```

```
tar -xvzf multichain-1.0.3.tar.gz
```

```
cd multichain-1.0.3
```

```
sudo mv multichaind multichain-cli multichain-util /usr/local/bin
```

Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

```

edureka@edureka: /tmp/multichain-1.0.3
edureka@edureka: /tmp$ wget https://www.multichain.com/download/multichain-1.0.3.tar.gz
--2018-03-23 23:35:59-- https://www.multichain.com/download/multichain-1.0.3.tar.gz
Resolving www.multichain.com (www.multichain.com)... 162.243.214.85
Connecting to www.multichain.com (www.multichain.com)|162.243.214.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10085166 (9.6M) [application/x-gzip]
Saving to: 'multichain-1.0.3.tar.gz'

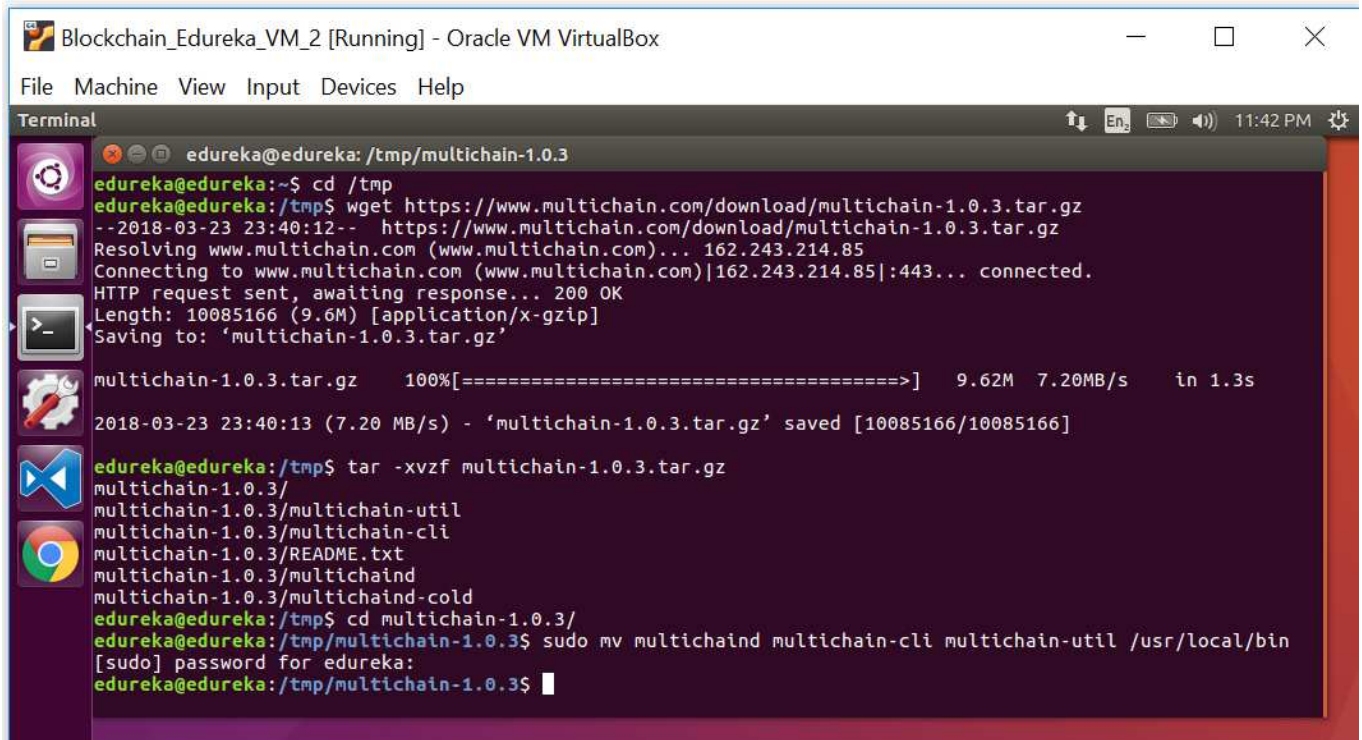
multichain-1.0.3.tar.gz 100%[=====] 9.62M 6.80MB/s in 1.4s
2018-03-23 23:36:00 (6.80 MB/s) - 'multichain-1.0.3.tar.gz' saved [10085166/10085166]

edureka@edureka: /tmp$ tar -xvzf multichain-1.0.3.tar.gz
multichain-1.0.3/
multichain-1.0.3/multichain-util
multichain-1.0.3/multichain-cli
multichain-1.0.3/README.txt
multichain-1.0.3/multichaind
multichain-1.0.3/multichaind-cold
edureka@edureka: /tmp$ cd multichain-1.0.3/
edureka@edureka: /tmp/multichain-1.0.3$ sudo mv multichaind multichain-cli multichain-util /usr/local/bin
edureka@edureka: /tmp/multichain-1.0.3$

```

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

Now, let's repeat above commands to VM2



The screenshot shows a terminal window titled "Blockchain_Edureka_VM_2 [Running] - Oracle VM VirtualBox". The terminal output shows the user 'edureka' at the prompt 'edureka@edureka: /tmp/multichain-1.0.3'. The user runs 'cd /tmp' and then 'wget https://www.multichain.com/download/multichain-1.0.3.tar.gz'. The terminal shows the download progress and completion. After the download, the user runs 'tar -xvzf multichain-1.0.3.tar.gz', which lists the contents of the archive: 'multichain-1.0.3/', 'multichain-1.0.3/multichain-util', 'multichain-1.0.3/multichain-cli', 'multichain-1.0.3/README.txt', 'multichain-1.0.3/multichaind', and 'multichain-1.0.3/multichaind-cold'. Finally, the user runs 'cd multichain-1.0.3/' and 'sudo mv multichaind multichain-cli multichain-util /usr/local/bin', with the terminal showing the password prompt and the successful execution of the command.

```
edureka@edureka: /tmp/multichain-1.0.3
edureka@edureka:~$ cd /tmp
edureka@edureka:/tmp$ wget https://www.multichain.com/download/multichain-1.0.3.tar.gz
--2018-03-23 23:40:12-- https://www.multichain.com/download/multichain-1.0.3.tar.gz
Resolving www.multichain.com (www.multichain.com)... 162.243.214.85
Connecting to www.multichain.com (www.multichain.com)|162.243.214.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10085166 (9.6M) [application/x-gzip]
Saving to: 'multichain-1.0.3.tar.gz'

multichain-1.0.3.tar.gz  100%[=====] 9.62M  7.20MB/s  in 1.3s

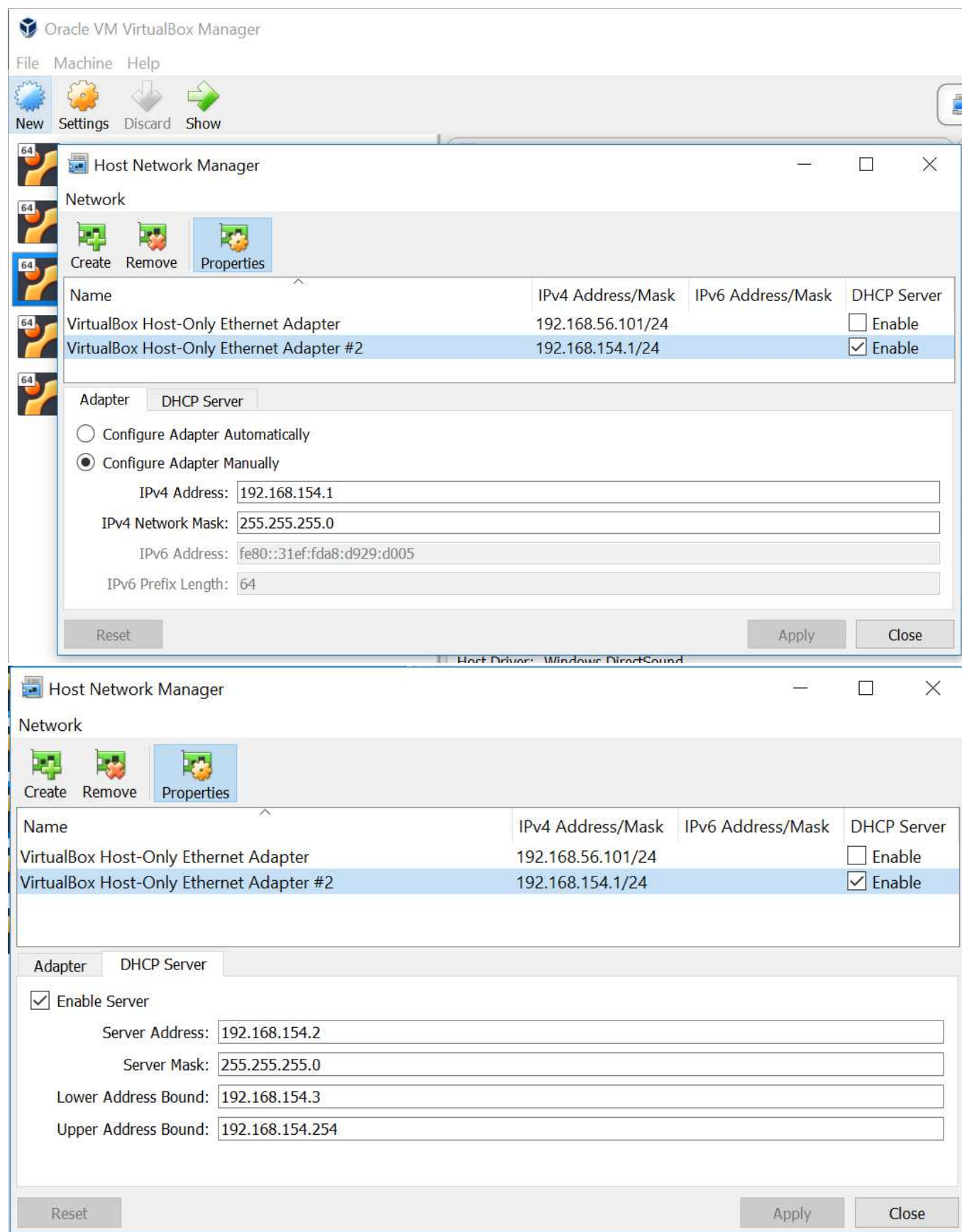
2018-03-23 23:40:13 (7.20 MB/s) - 'multichain-1.0.3.tar.gz' saved [10085166/10085166]

edureka@edureka:/tmp$ tar -xvzf multichain-1.0.3.tar.gz
multichain-1.0.3/
multichain-1.0.3/multichain-util
multichain-1.0.3/multichain-cli
multichain-1.0.3/README.txt
multichain-1.0.3/multichaind
multichain-1.0.3/multichaind-cold
edureka@edureka:/tmp$ cd multichain-1.0.3/
edureka@edureka:/tmp/multichain-1.0.3$ sudo mv multichaind multichain-cli multichain-util /usr/local/bin
[sudo] password for edureka:
edureka@edureka:/tmp/multichain-1.0.3$
```

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.



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

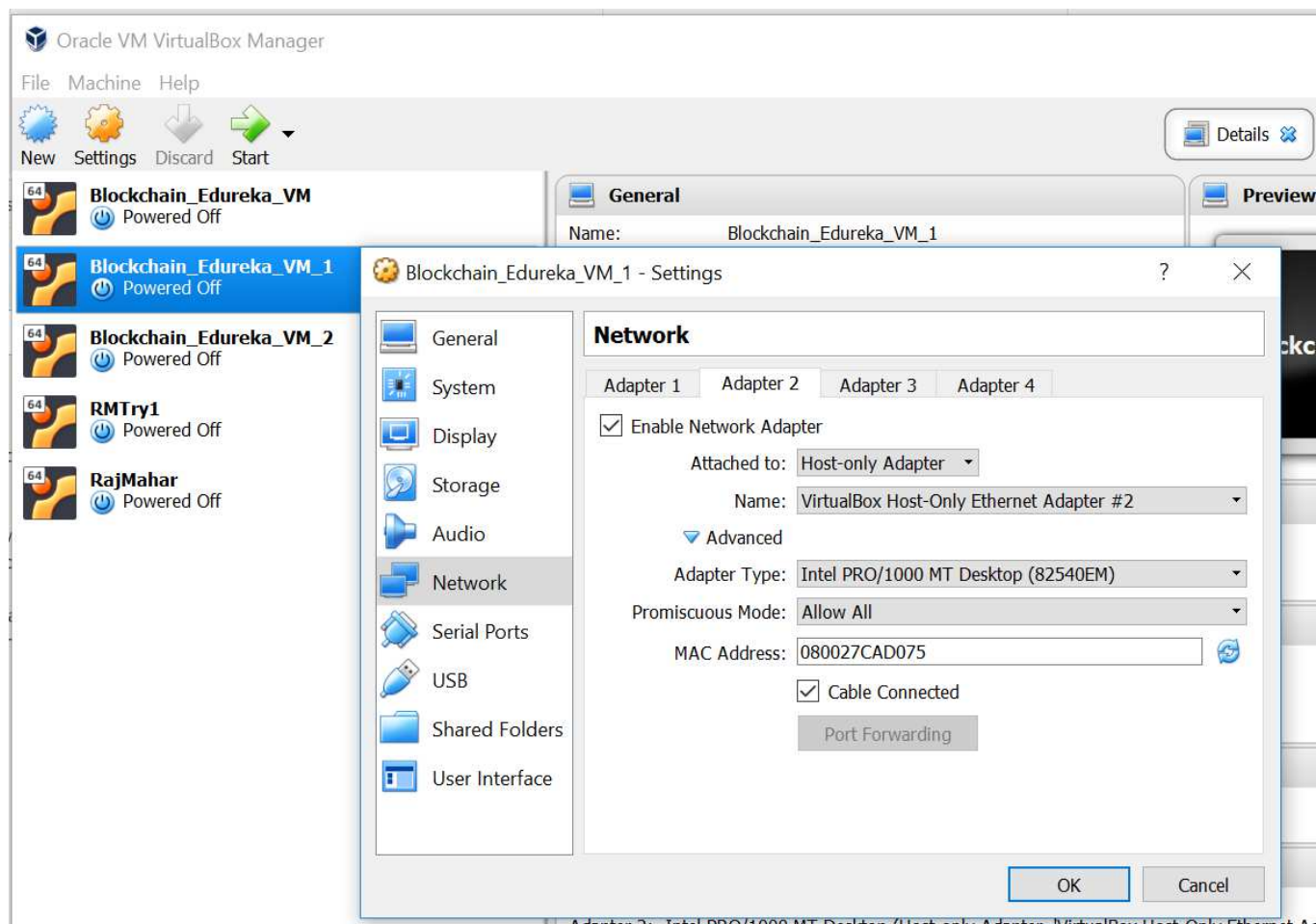
And

Make sure some fields are same as below

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter #2

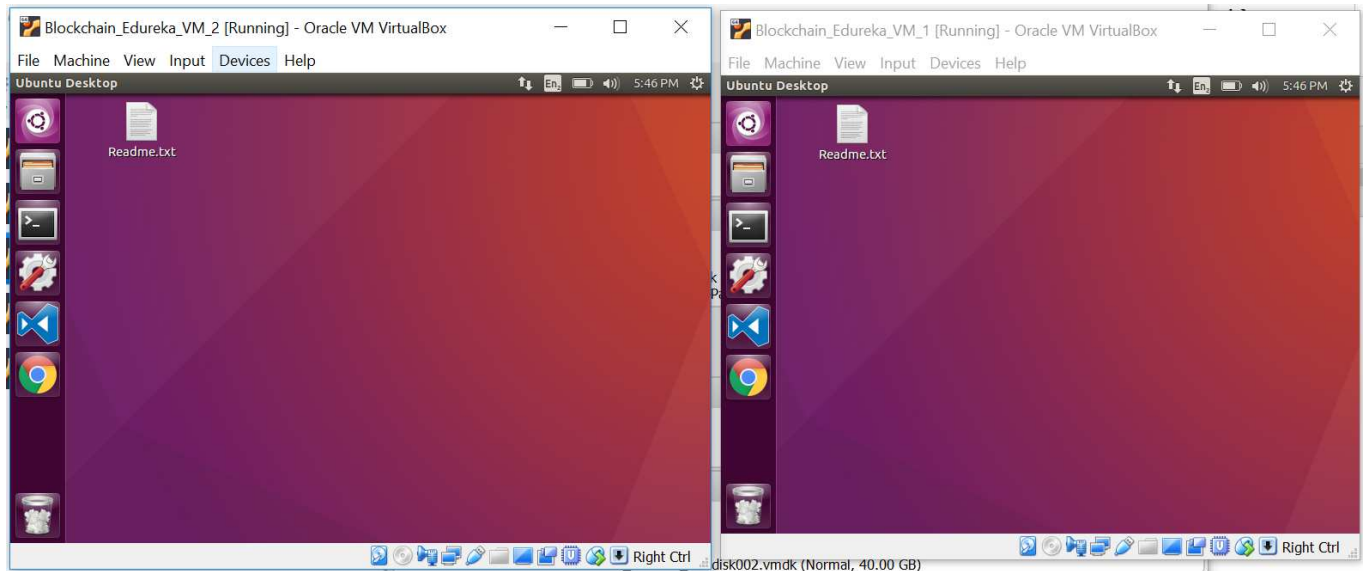
Mac Address: Newly generated Mac Address (Click small Refresh icon to change Mac Address)



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

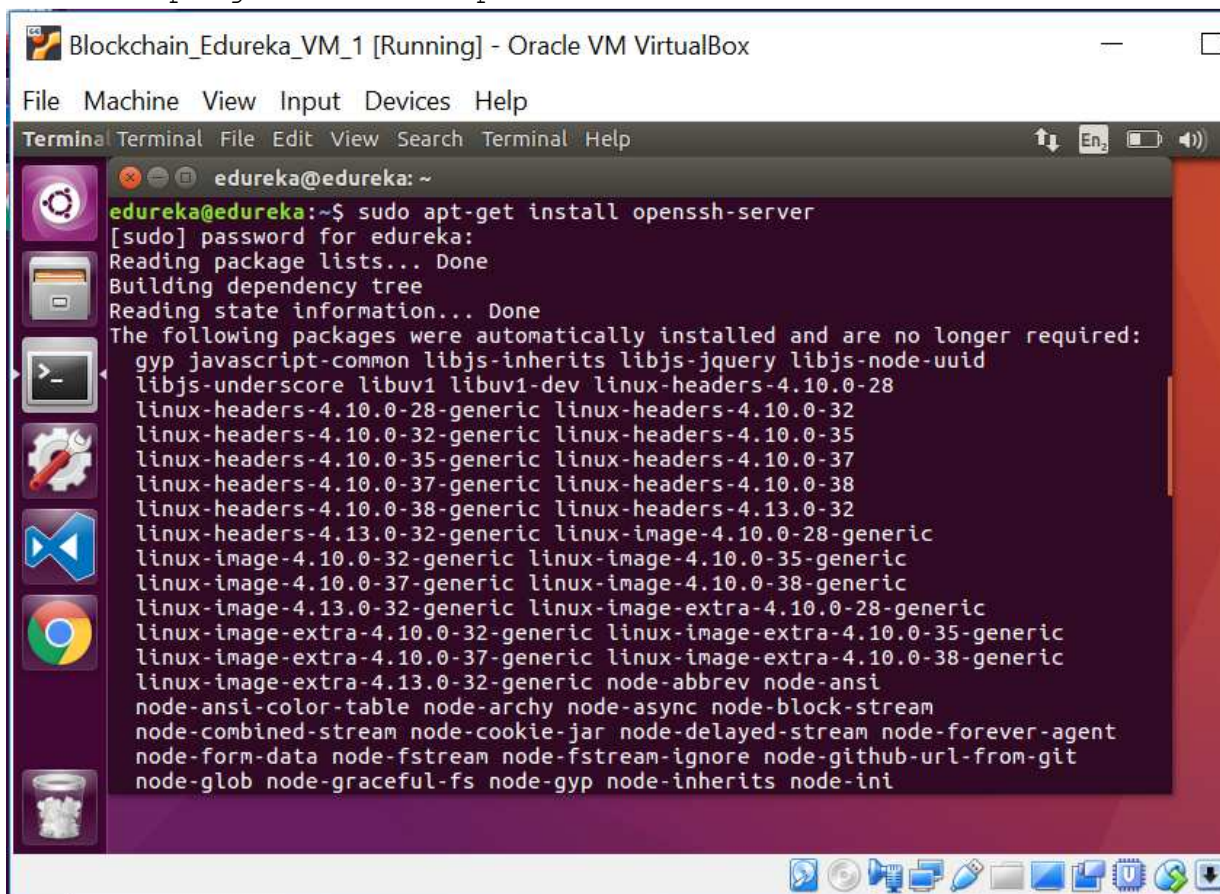
3. Start VMs

In Edureka VMs, you will see 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:

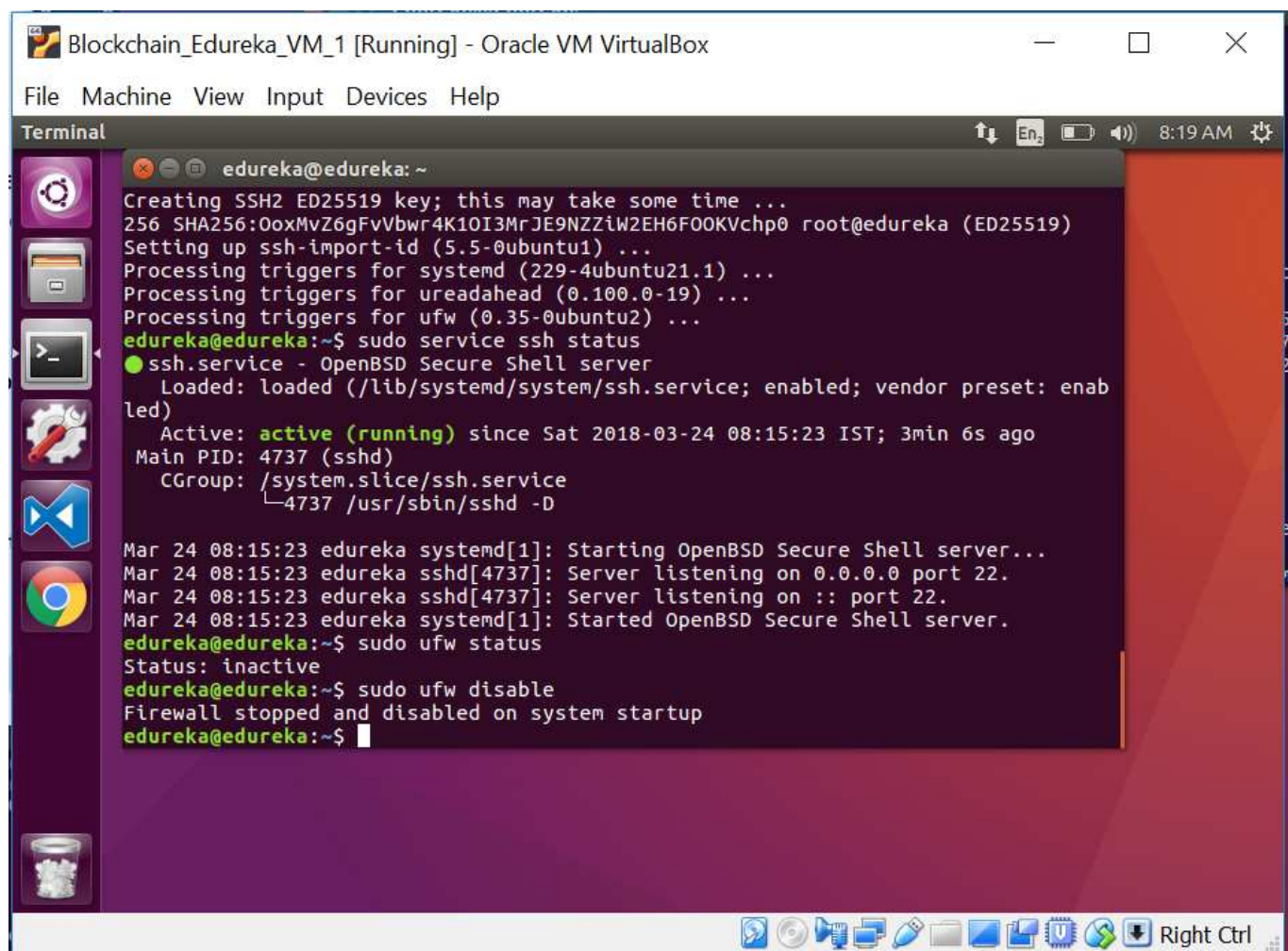
```
sudo service ssh status
```

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
```



The screenshot shows a terminal window titled "Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox". The terminal output shows the following commands and their results:

```
edureka@edureka: ~  
Creating SSH2 ED25519 key; this may take some time ...  
256 SHA256:0oxMvZ6gFvVbwr4K10I3MrJE9NZziW2EH6F00KVchp0 root@edureka (ED25519)  
Setting up ssh-import-id (5.5-0ubuntu1) ...  
Processing triggers for systemd (229-4ubuntu21.1) ...  
Processing triggers for ureadahead (0.100.0-19) ...  
Processing triggers for ufw (0.35-0ubuntu2) ...  
edureka@edureka:~$ sudo service ssh status  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sat 2018-03-24 08:15:23 IST; 3min 6s ago  
     Main PID: 4737 (sshd)  
       CGroup: /system.slice/ssh.service  
              └─4737 /usr/sbin/sshd -D  
  
Mar 24 08:15:23 edureka systemd[1]: Starting OpenBSD Secure Shell server...  
Mar 24 08:15:23 edureka sshd[4737]: Server listening on 0.0.0.0 port 22.  
Mar 24 08:15:23 edureka sshd[4737]: Server listening on :: port 22.  
Mar 24 08:15:23 edureka systemd[1]: Started OpenBSD Secure Shell server.  
edureka@edureka:~$ sudo ufw status  
Status: inactive  
edureka@edureka:~$ sudo ufw disable  
Firewall stopped and disabled on system startup  
edureka@edureka:~$
```

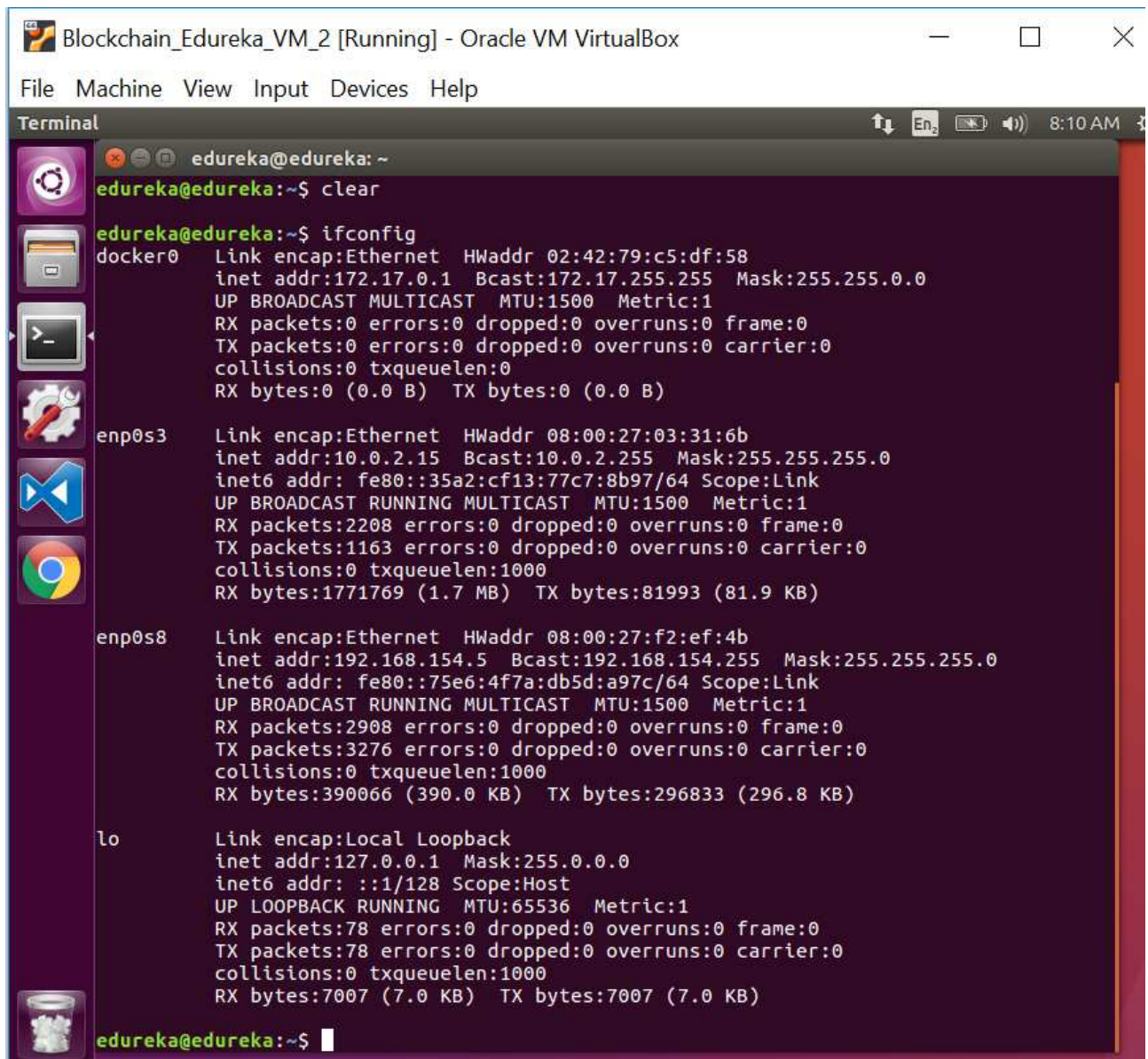
8. Check the IP address of VM1. The host only driver is enp0s8 and its IP is 192.168.154.3

```
ifconfig
```



```
edureka@edureka: ~  
edureka@edureka:~$ ifconfig  
br-2552f6d4124e Link encap:Ethernet HWaddr 02:42:08:fc:75:46  
    inet addr:172.18.0.1 Bcast:172.18.255.255 Mask:255.255.0.0  
    UP BROADCAST MULTICAST MTU:1500 Metric:1  
    RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:0  
    RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)  
  
docker0 Link encap:Ethernet HWaddr 02:42:2c:ba:f8:d1  
    inet addr:172.17.0.1 Bcast:172.17.255.255 Mask:255.255.0.0  
    UP BROADCAST MULTICAST MTU:1500 Metric:1  
    RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:0  
    RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)  
  
enp0s3 Link encap:Ethernet HWaddr 08:00:27:03:31:6b  
    inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0  
    inet6 addr: fe80::8bb2:1e2f:ee96:746f/64 Scope:Link  
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
    RX packets:1856 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:793 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:1000  
    RX bytes:1753069 (1.7 MB) TX bytes:59600 (59.6 KB)  
  
enp0s8 Link encap:Ethernet HWaddr 08:00:27:95:e8:61  
    inet addr:192.168.154.3 Bcast:192.168.154.255 Mask:255.255.255.0  
    inet6 addr: fe80::34c2:cb62:a2b6:8ac/64 Scope:Link  
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
    RX packets:3117 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:2768 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:1000  
    RX bytes:289837 (289.8 KB) TX bytes:396198 (396.1 KB)  
  
lo Link encap:Local Loopback
```

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



```

Blockchain_Edureka_VM_2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
edureka@edureka: ~
edureka@edureka:~$ clear
edureka@edureka:~$ ifconfig
docker0  Link encap:Ethernet  HWaddr 02:42:79:c5:df:58
         inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
         UP BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

enp0s3   Link encap:Ethernet  HWaddr 08:00:27:03:31:6b
         inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
         inet6 addr: fe80::35a2:cf13:77c7:8b97/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:2208 errors:0 dropped:0 overruns:0 frame:0
         TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:1771769 (1.7 MB)  TX bytes:81993 (81.9 KB)

enp0s8   Link encap:Ethernet  HWaddr 08:00:27:f2:ef:4b
         inet addr:192.168.154.5  Bcast:192.168.154.255  Mask:255.255.255.0
         inet6 addr: fe80::75e6:4f7a:db5d:a97c/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:2908 errors:0 dropped:0 overruns:0 frame:0
         TX packets:3276 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:390066 (390.0 KB)  TX bytes:296833 (296.8 KB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:78 errors:0 dropped:0 overruns:0 frame:0
         TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:7007 (7.0 KB)  TX bytes:7007 (7.0 KB)

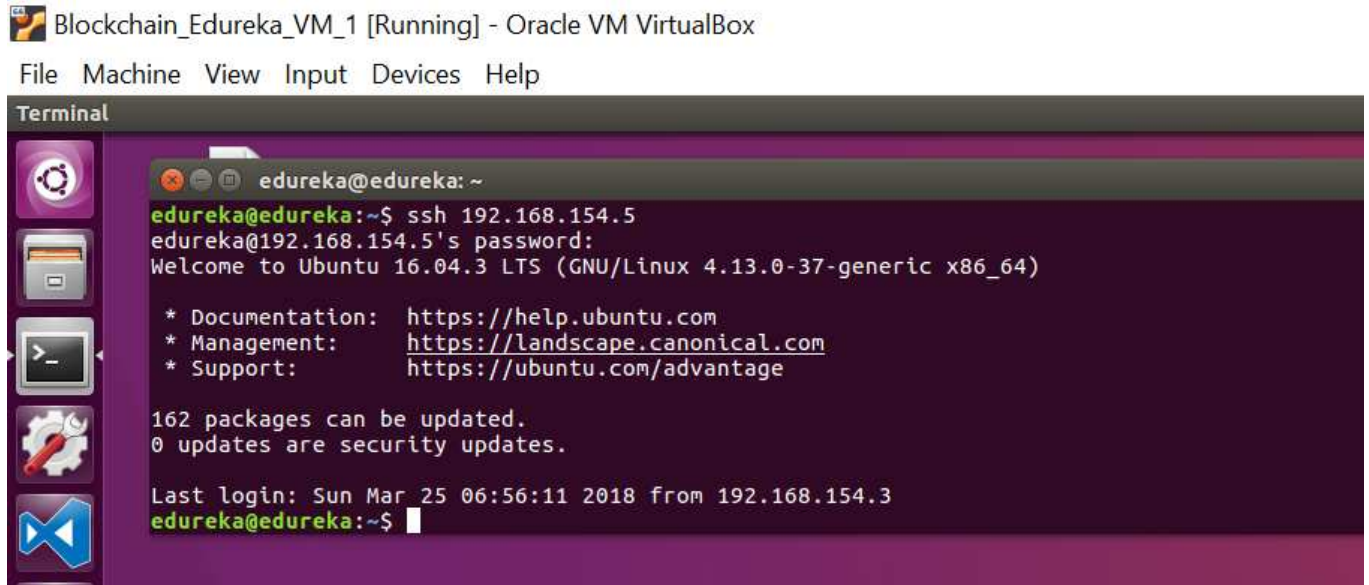
edureka@edureka:~$

```

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

```
ssh 192.168.136.5
```

(This address is VM2 IP address)



```
Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
edureka@edureka: ~
edureka@edureka:~$ ssh 192.168.154.5
edureka@192.168.154.5's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

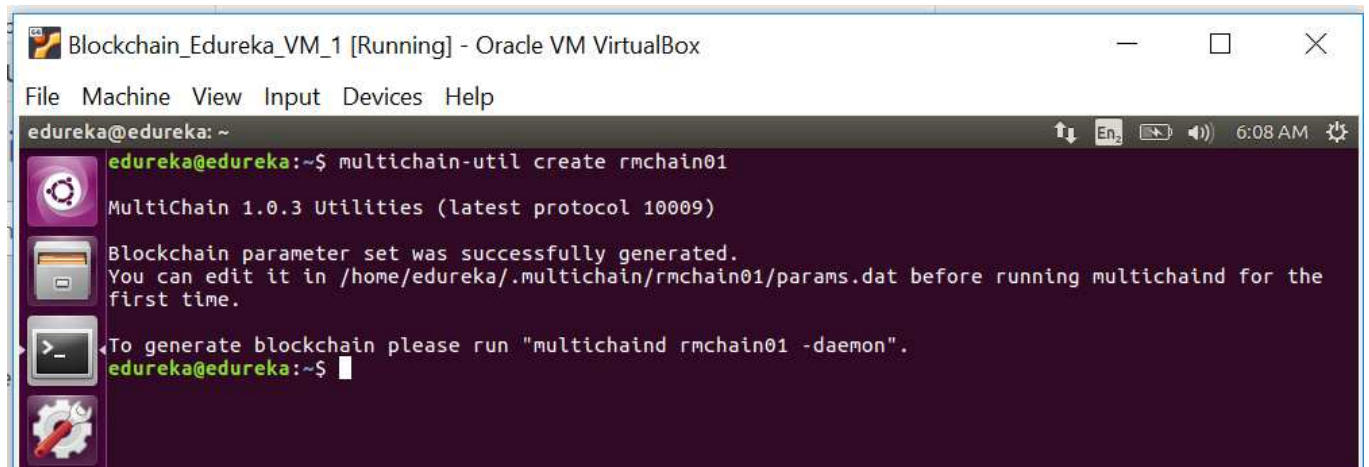
162 packages can be updated.
0 updates are security updates.

Last login: Sun Mar 25 06:56:11 2018 from 192.168.154.3
edureka@edureka:~$
```

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
```



```
Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
edureka@edureka: ~
edureka@edureka:~$ multichain-util create rmchain01
MultiChain 1.0.3 Utilities (latest protocol 10009)
Blockchain parameter set was successfully generated.
You can edit it in /home/edureka/.multichain/rmchain01/params.dat before running multichaind for the
first time.
To generate blockchain please run "multichaind rmchain01 -daemon".
edureka@edureka:~$
```

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

```

Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
edureka@edureka: ~
edureka@edureka:~$ cat ~/.multichain/rmchain01/params.dat
# ==== MultiChain configuration file ====

# Created by multichain-util
# Protocol version: 10009

# 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 bitcoin
chain-description = MultiChain rmchain01 # Chain description, embedded in genesis block coinbase, max 90 chars.
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 compatibility.
target-block-time = 15 # Target time between blocks (transaction confirmation delay), seconds. (2 - 86400)
maximum-block-size = 8388608 # Maximum block size in bytes. (1000 - 1000000000)

# Global permissions
anyone-can-connect = false # Anyone can connect, i.e. a publicly readable blockchain.
anyone-can-send = false # Anyone can send, i.e. transaction signing not restricted by address.
anyone-can-receive = false # Anyone can receive, i.e. transaction outputs not restricted by address.
anyone-can-receive-empty = true # Anyone can receive empty output, i.e. without permission grants, asset transfers
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-grant = false # Anyone can grant or revoke connect, send and receive permissions.
anyone-can-revoke = false # Anyone can grant or revoke all permissions.
support-miner-precheck = true # Require special metadata output with cached scriptPubKey for input, to support ad
allow-arbitrary-outputs = false # Allow arbitrary (without clear destination) scripts.
allow-p2sh-outputs = true # Allow pay-to-scripthash (P2SH) scripts, often used for multisig. Ignored if allow
allow-multisig-outputs = true # Allow bare multisignature scripts, rarely used but still supported. Ignored if al

# Consensus requirements
setup-first-blocks = 60 # Length of initial setup phase in blocks, in which mining-diversity,
# admin-consensus-* and mining-requires-peers are not applied. (1 - 31536000)
mining-diversity = 0.3 # Miners must wait <mining-diversity>*active miners> between blocks. (0 - 1)
admin-consensus-upgrade = 0.5 # <admin-consensus-upgrade>*active admins> needed to upgrade the chain. (0 - 1)
admin-consensus-admin = 0.5 # <admin-consensus-admin>*active admins> needed to change admin perms. (0 - 1)
admin-consensus-activate = 0.5 # <admin-consensus-activate>*active admins> to change activate perms. (0 - 1)
admin-consensus-mine = 0.5 # <admin-consensus-mine>*active admins> to change mining permissions. (0 - 1)
admin-consensus-create = 0.0 # <admin-consensus-issue>*active admins> to change create permissions. (0 - 1)
admin-consensus-issue = 0.0 # <admin-consensus-issue>*active admins> to change issue permissions. (0 - 1)

# Defaults for node runtime parameters
lock-admin-mine-rounds = 10 # Ignore forks that reverse changes in admin or mine permissions after this many m
mining-requires-peers = true # Nodes only mine blocks if connected to other nodes (ignored if only one permitted
mine-empty-rounds = 10 # Mine this many rounds of empty blocks before pausing to wait for new transactions
req>0). Non-integer allowed. (-1 - 1000)
mining-turnover = 0.5 # Prefer pure round robin between a subset of active miners to minimize forks (0.0
0 - 1)

# Native blockchain currency (likely not required)
initial-block-reward = 0 # Initial block mining reward in raw native currency units. (0 - 1000000000000000000)
first-block-reward = -1 # Different mining reward for first block only, ignored if negative. (-1 - 100000000)
reward-halving-interval = 52560000 # Interval for halving of mining rewards, in blocks. (60 - 1000000000)
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 - 10000000000)
maximum-per-output = 1000000000000000 # Maximum native currency per output, in raw units. (0 - 1000000000000000000)
minimum-relay-fee = 0 # Minimum transaction fee, per 1000 bytes, in raw units of native currency. (0 - 10

```

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$
mining-diversity = 0.3 # admin-consensus-* and mining-requires-peers are not applied. (1 $
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$

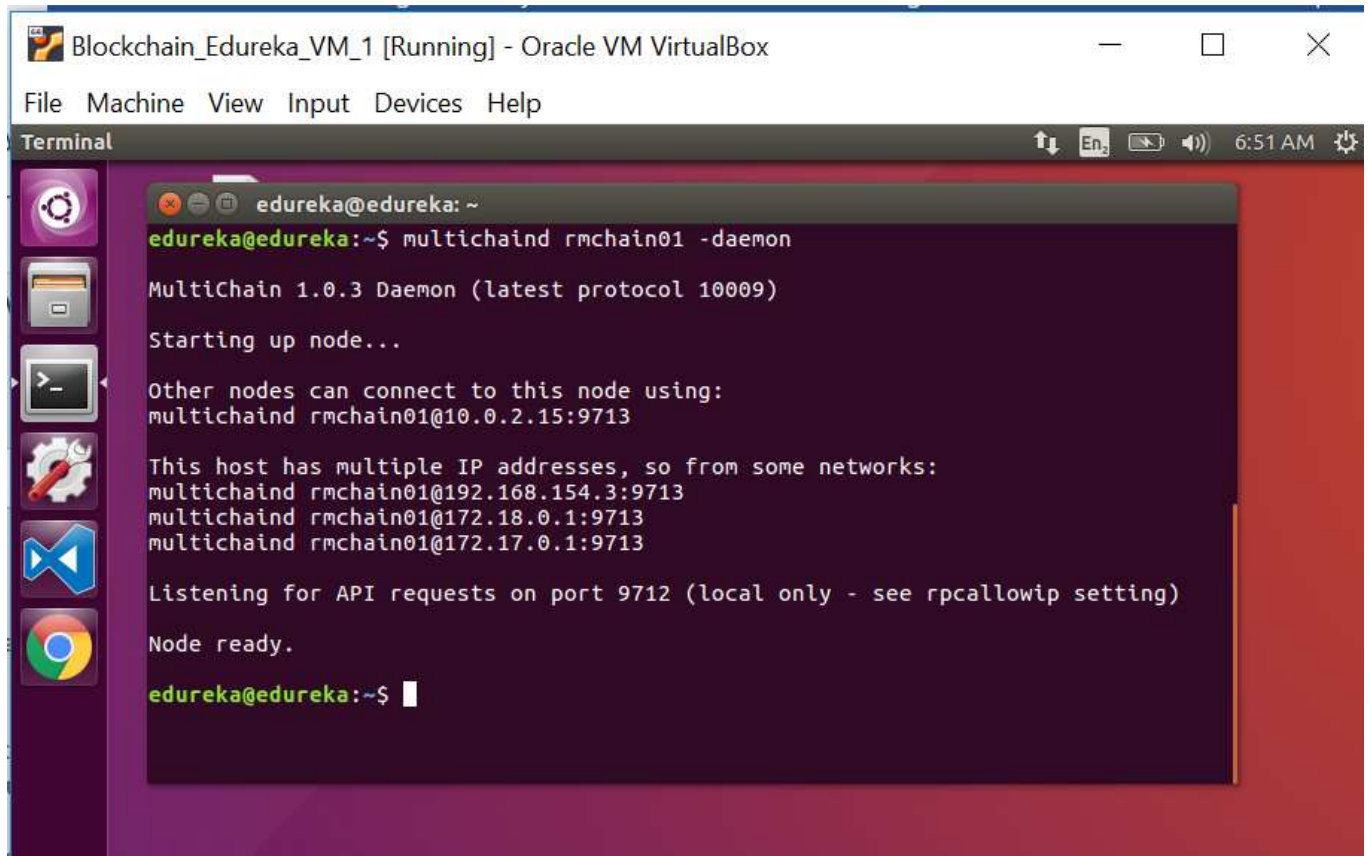
# Native blockchain currency (likely not required)
initial-block-reward = 2562400 # Initial block mining reward in raw native currency units. ($
first-block-reward = -1 # Different mining reward for first block only, ignored if negativ$
reward-halving-interval = 52560000 # Interval for halving of mining 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 - 1000000000$
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

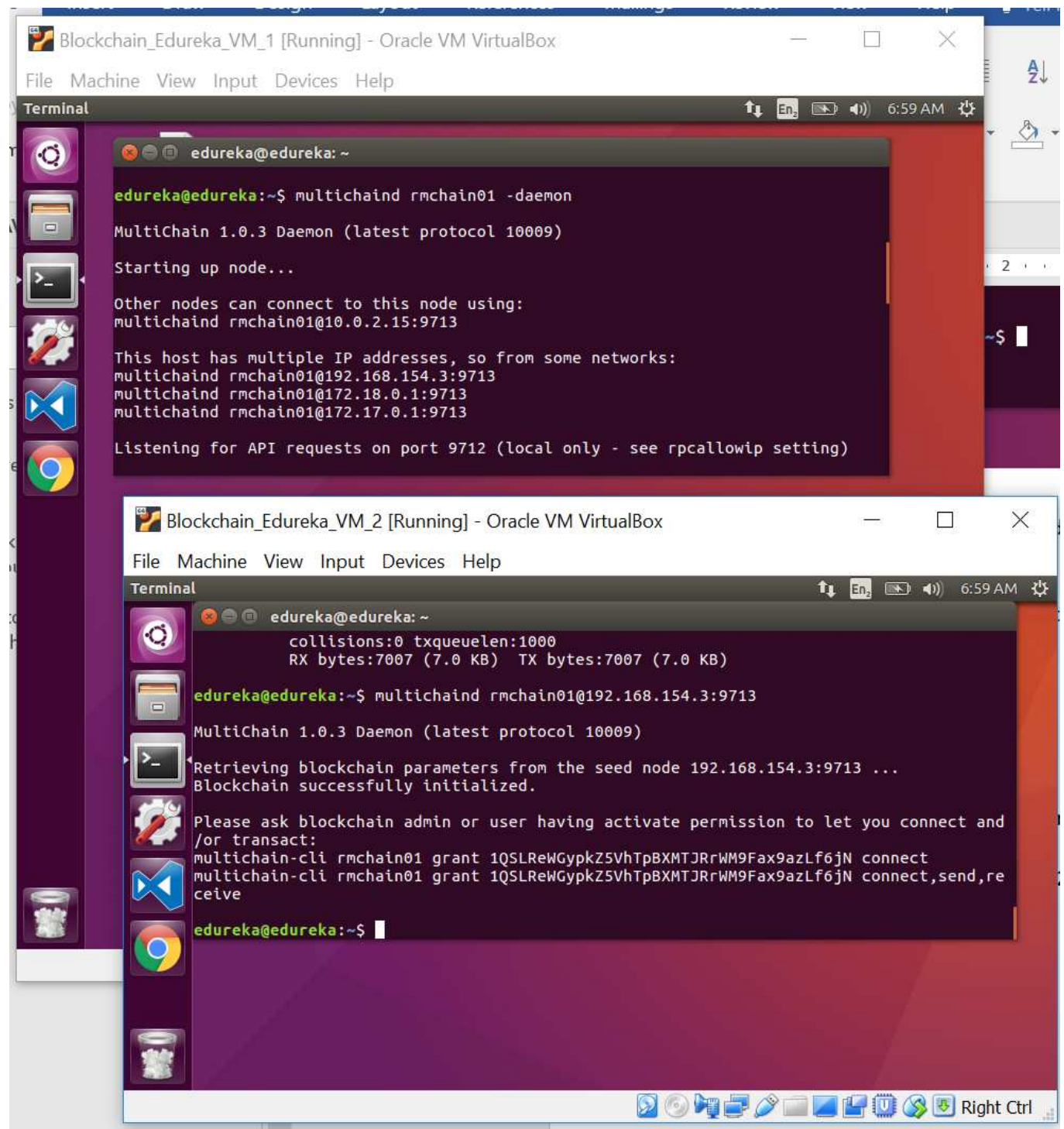
or

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
```



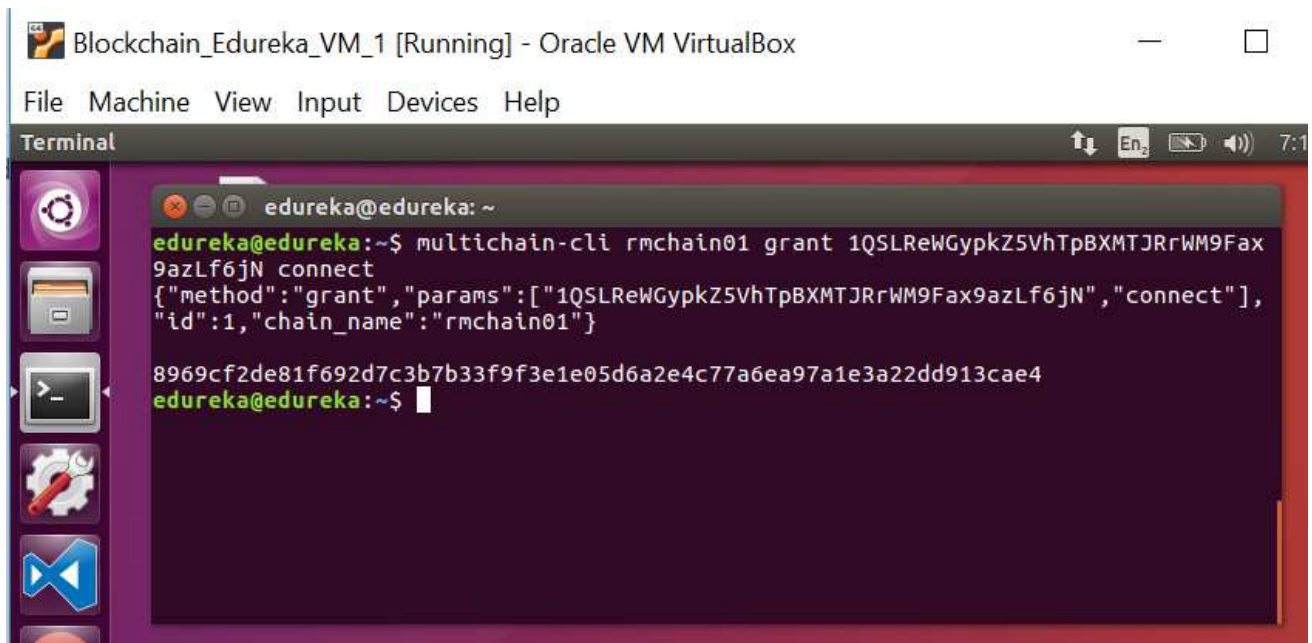
If there is an error, please try to connect the VM1 server from VM2 running following command

```
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.

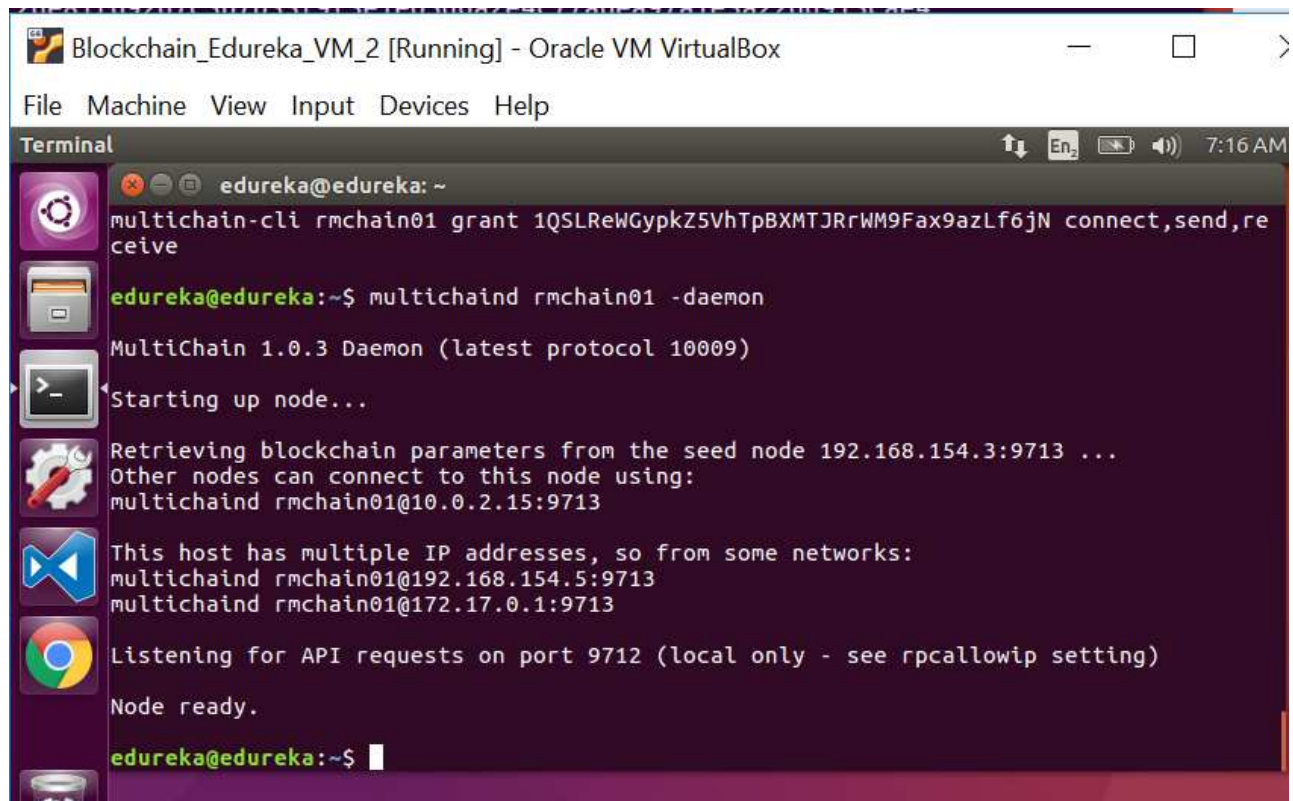
Back on the first server VM1, add connection permissions for this address:

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



Now try reconnecting again from the second server:

```
multichaind rmchain01 -daemon
```



The screenshot shows a VirtualBox window titled "Blockchain_Edureka_VM_2 [Running] - Oracle VM VirtualBox". Inside, a terminal window is open with the prompt "edureka@edureka: ~". The terminal output shows the following sequence of events:

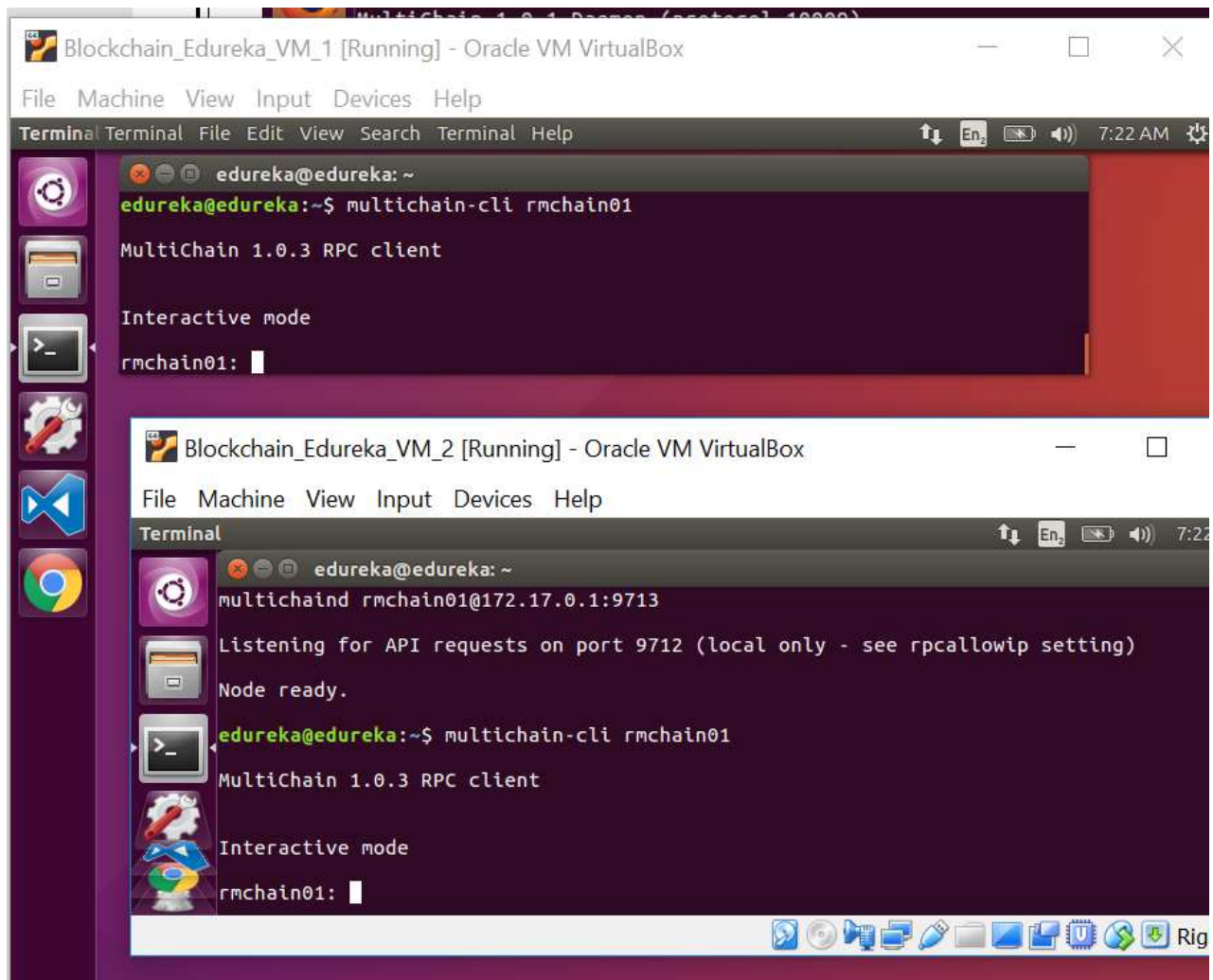
```
multichain-cli rmchain01 grant 1QSLReWGypkZ5VhTpBXMTJRrWM9Fax9azLf6jN connect,send,recv  
edureka@edureka:~$ multichaind rmchain01 -daemon  
MultiChain 1.0.3 Daemon (latest protocol 10009)  
Starting up node...  
Retrieving blockchain parameters from the seed node 192.168.154.3:9713 ...  
Other nodes can connect to this node using:  
multichaind rmchain01@10.0.2.15:9713  
This host has multiple IP addresses, so from some networks:  
multichaind rmchain01@192.168.154.5:9713  
multichaind rmchain01@172.17.0.1:9713  
Listening for API requests on port 9712 (local only - see rpcallowip setting)  
Node ready.  
edureka@edureka:~$
```

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:

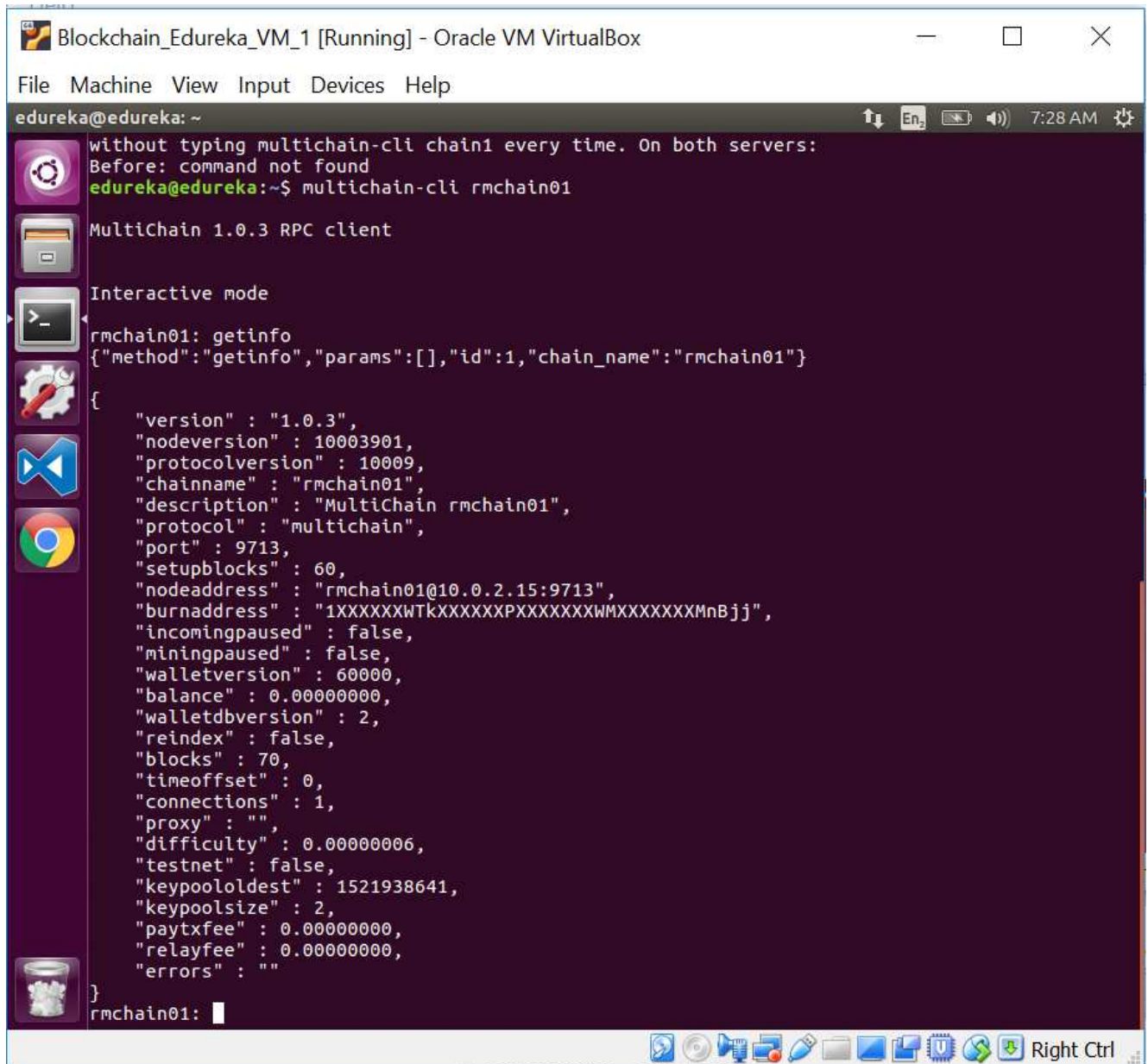
```
multichain-cli rmchain01
```



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



```
Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
edureka@edureka: ~
without typing multichain-cli chain1 every time. On both servers:
Before: command not found
edureka@edureka:~$ multichain-cli rmchain01

MultiChain 1.0.3 RPC client

Interactive mode
rmchain01: getinfo
{"method": "getinfo", "params": [], "id": 1, "chain_name": "rmchain01"}
{
  "version" : "1.0.3",
  "nodeversion" : 10003901,
  "protocolversion" : 10009,
  "chainname" : "rmchain01",
  "description" : "MultiChain rmchain01",
  "protocol" : "multichain",
  "port" : 9713,
  "setupblocks" : 60,
  "nodeaddress" : "rmchain01@10.0.2.15:9713",
  "burnaddress" : "1XXXXXXWTkXXXXXXPPXXXXXXWMXXXXXXMnBjj",
  "incomingpaused" : false,
  "miningpaused" : false,
  "walletversion" : 60000,
  "balance" : 0.00000000,
  "walletdbversion" : 2,
  "reindex" : false,
  "blocks" : 70,
  "timeoffset" : 0,
  "connections" : 1,
  "proxy" : "",
  "difficulty" : 0.00000006,
  "testnet" : false,
  "keypoololdest" : 1521938641,
  "keypoolsize" : 2,
  "paytxfee" : 0.00000000,
  "relayfee" : 0.00000000,
  "errors" : ""
}
rmchain01: 
```

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 )
listlocks 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

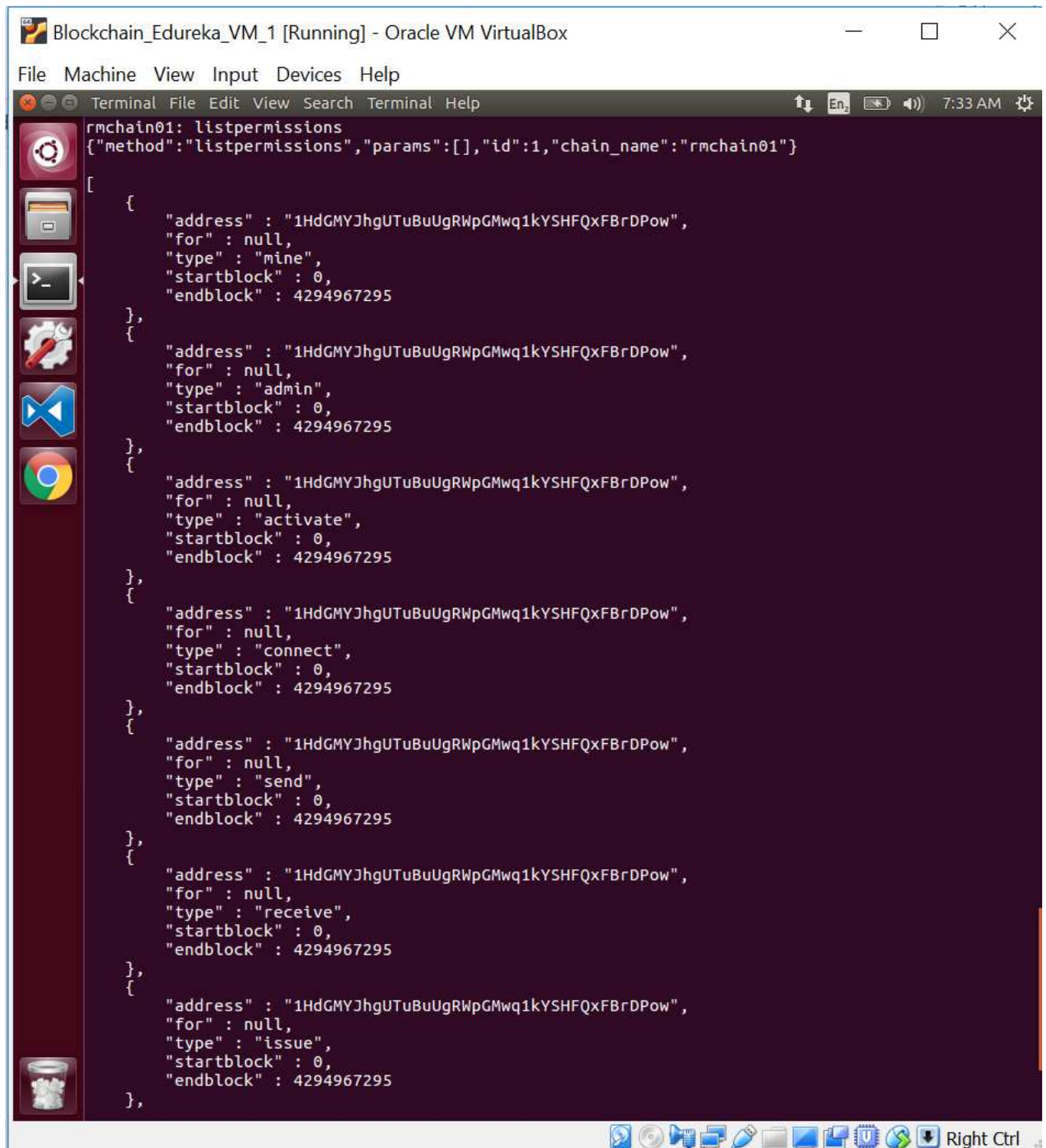
== Generating ==
getgenerate
gethashespersec
setgenerate generate ( genproclimit )

== Mining ==
getblocktemplate ( "jsonrequestobject" )
getmininginfo

```

Show all permissions currently assigned:

```
listpermissions
```

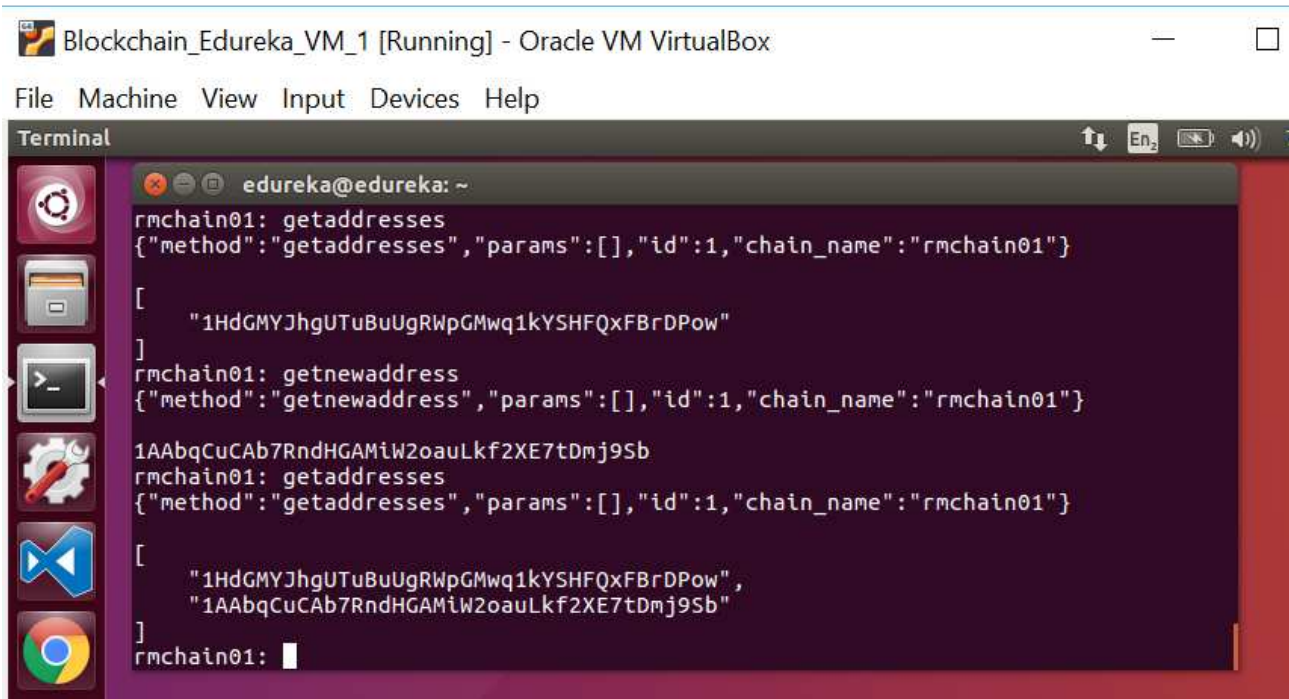


The screenshot shows a VirtualBox window titled "Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox". Inside the window is a terminal application with a dark background. The terminal shows the command `rmchain01: listpermissions` and its output, which is a JSON array of permission objects. Each object contains fields for `address`, `for`, `type`, `startblock`, and `endblock`. The `address` field for all permissions is `"1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow"`. The `for` field is `null` for all. The `type` field varies: `"mine"`, `"admin"`, `"activate"`, `"connect"`, `"send"`, `"receive"`, and `"issue"`. The `startblock` field is `0` for all, and the `endblock` field is `4294967295` for all. The terminal window has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The terminal application has a menu bar with "Terminal", "File", "Edit", "View", "Search", "Terminal", and "Help". The system tray at the bottom right shows a "Right Ctrl" button.

```
rmchain01: listpermissions
{"method":"listpermissions","params":[],"id":1,"chain_name":"rmchain01"}
[
  {
    "address" : "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
    "for" : null,
    "type" : "mine",
    "startblock" : 0,
    "endblock" : 4294967295
  },
  {
    "address" : "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
    "for" : null,
    "type" : "admin",
    "startblock" : 0,
    "endblock" : 4294967295
  },
  {
    "address" : "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
    "for" : null,
    "type" : "activate",
    "startblock" : 0,
    "endblock" : 4294967295
  },
  {
    "address" : "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
    "for" : null,
    "type" : "connect",
    "startblock" : 0,
    "endblock" : 4294967295
  },
  {
    "address" : "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
    "for" : null,
    "type" : "send",
    "startblock" : 0,
    "endblock" : 4294967295
  },
  {
    "address" : "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
    "for" : null,
    "type" : "receive",
    "startblock" : 0,
    "endblock" : 4294967295
  },
  {
    "address" : "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
    "for" : null,
    "type" : "issue",
    "startblock" : 0,
    "endblock" : 4294967295
  },
]
```

To get new address use command `getnewaddress`

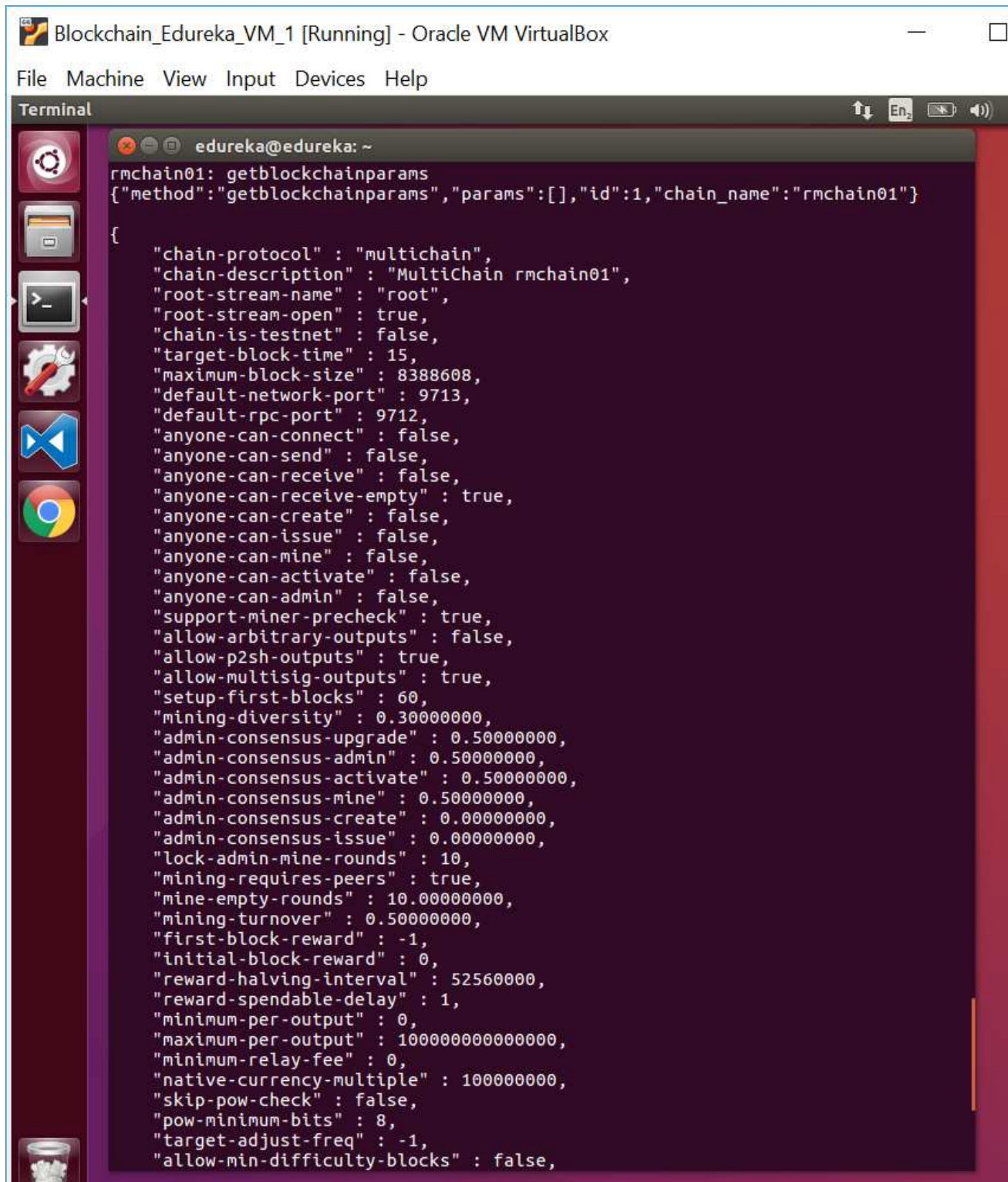
And to list all addresses, use command `getaddresses`



```
Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
edureka@edureka: ~
rmchain01: getaddresses
{"method": "getaddresses", "params": [], "id": 1, "chain_name": "rmchain01"}
[
  "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow"
]
rmchain01: getnewaddress
{"method": "getnewaddress", "params": [], "id": 1, "chain_name": "rmchain01"}
1AAbqCuCAB7RndHGAMiW2oauLkf2XE7tDmj9Sb
rmchain01: getaddresses
{"method": "getaddresses", "params": [], "id": 1, "chain_name": "rmchain01"}
[
  "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
  "1AAbqCuCAB7RndHGAMiW2oauLkf2XE7tDmj9Sb"
]
rmchain01: █
```

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

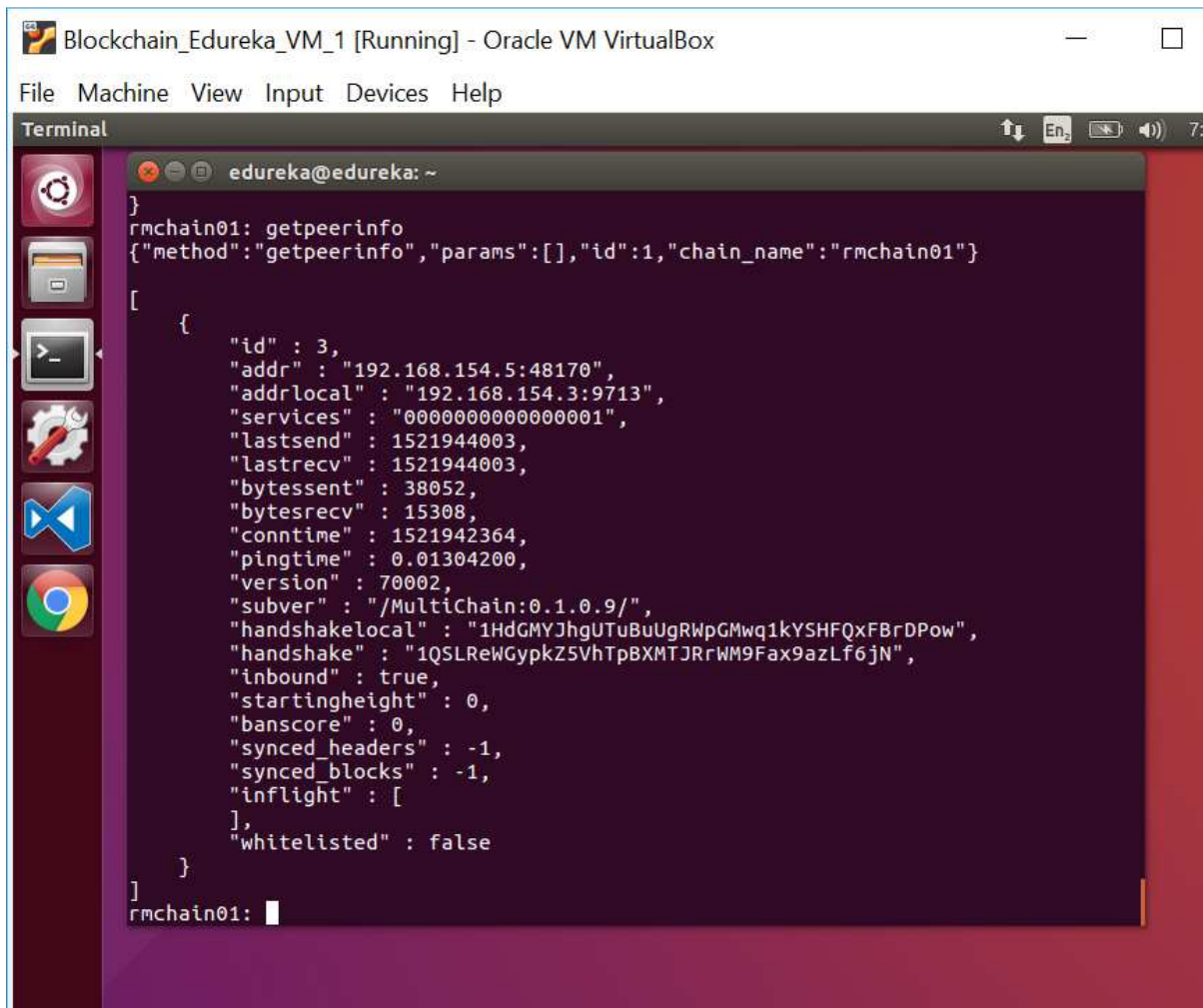
`getblockchainparams`



```
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" : 1000000000000000,
  "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



The screenshot shows a terminal window titled "Blockchain_Edureka_VM_1 [Running] - Oracle VM VirtualBox". The terminal is running a MultiChain node named "rmchain01". The command "getpeerinfo" has been executed, and the output is a JSON array containing one peer object. The peer object includes details such as ID, address, services, and connection status.

```
rmchain01: getpeerinfo
{"method": "getpeerinfo", "params": [], "id": 1, "chain_name": "rmchain01"}
[
  {
    "id" : 3,
    "addr" : "192.168.154.5:48170",
    "addrlocal" : "192.168.154.3:9713",
    "services" : "0000000000000001",
    "lastsend" : 1521944003,
    "lastrecv" : 1521944003,
    "bytessent" : 38052,
    "bytesrecv" : 15308,
    "conntime" : 1521942364,
    "pingtime" : 0.01304200,
    "version" : 70002,
    "subver" : "/MultiChain:0.1.0.9/",
    "handshake" : "1HdGMYJhgUTuBuUgRWpGMwq1kYSHFQxFBrDPow",
    "inbound" : true,
    "startingheight" : 0,
    "banscore" : 0,
    "synced_headers" : -1,
    "synced_blocks" : -1,
    "inflight" : [
    ],
    "whitelisted" : false
  }
]
rmchain01: 
```