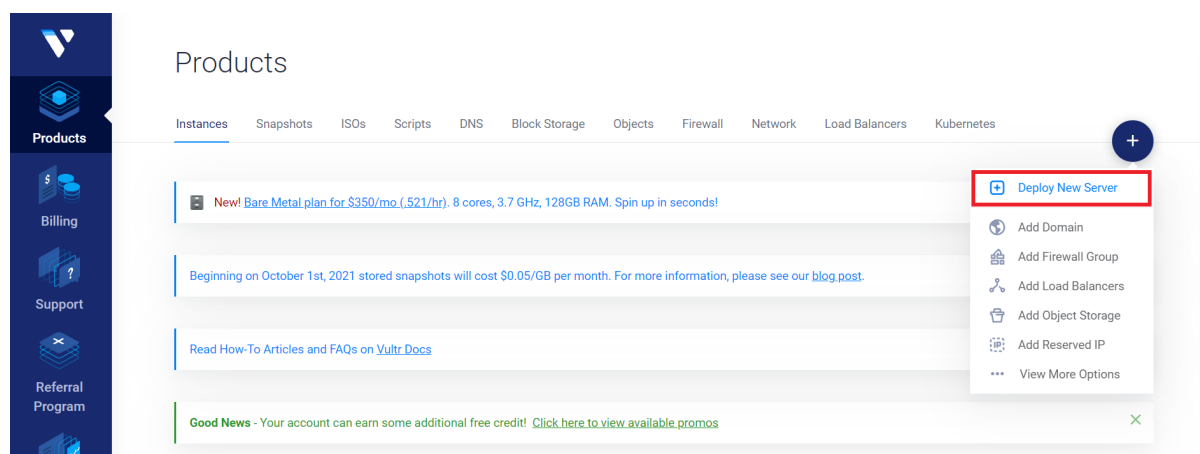


How to run multiple MNs on the same VPS

In this guide we explain how to configure multiple masternodes of the same coin in a single server, using IPv6.

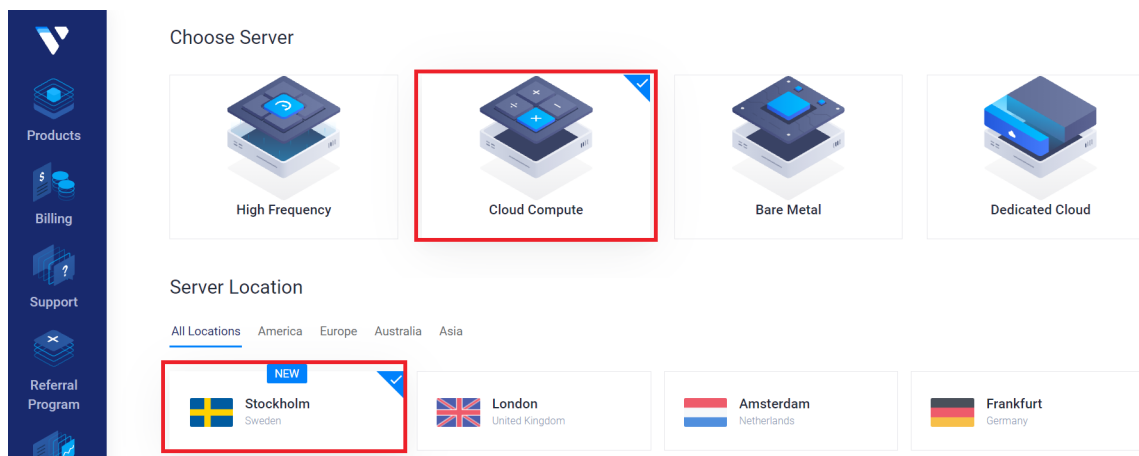
We buy a VPS on [Vultr.com](https://vultr.com)

Click on deploy new server



We choose

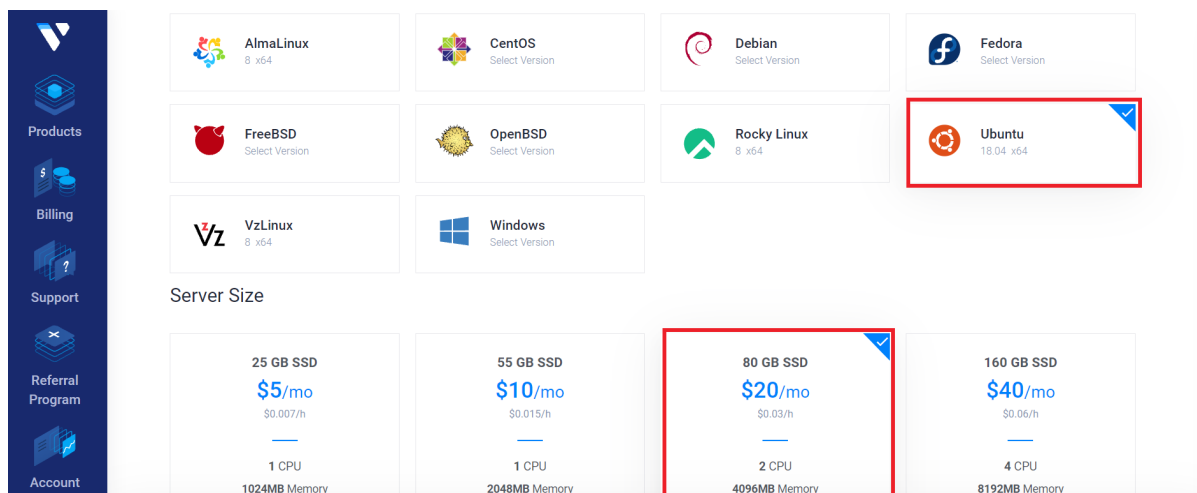
- Cloud Compute
- A location near us



The screenshot shows a sidebar with navigation links: Products, Billing, Support, Referral Program, and Account. The main content area is titled 'Choose Server' and features a grid of server options. Under the 'Server Location' section, the 'Stockholm' location (Sweden) is highlighted with a red border and a 'NEW' tag. Other locations shown include London (United Kingdom), Amsterdam (Netherlands), and Frankfurt (Germany).

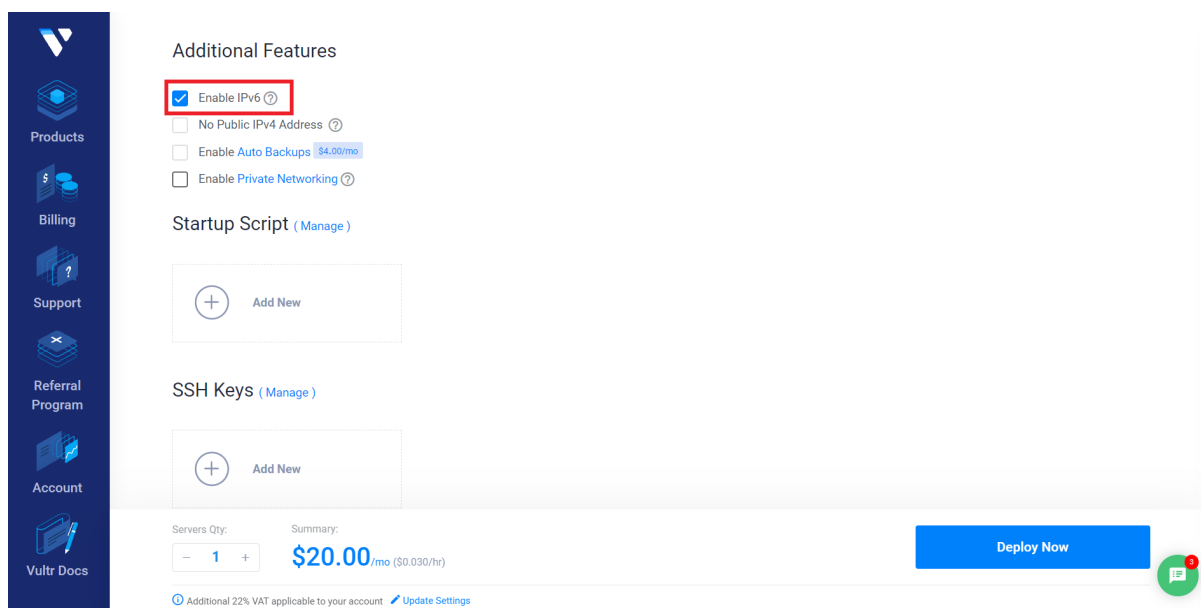
We choose

- Ubuntu 18 or 20
- Server size (we chose the \$ 20 server)



The screenshot shows the 'Server Size' section of the interface. It displays a grid of operating system options (AlmaLinux, CentOS, Debian, Fedora, FreeBSD, OpenBSD, Rocky Linux, Ubuntu, VzLinux, Windows) and a grid of server sizes. The 'Ubuntu' operating system is highlighted with a red border. The '\$20/mo' server size is also highlighted with a red border. This server size offers 80 GB SSD, 2 CPU, and 4096MB Memory.

So let's make sure we add IPv6 and then deploy the server.

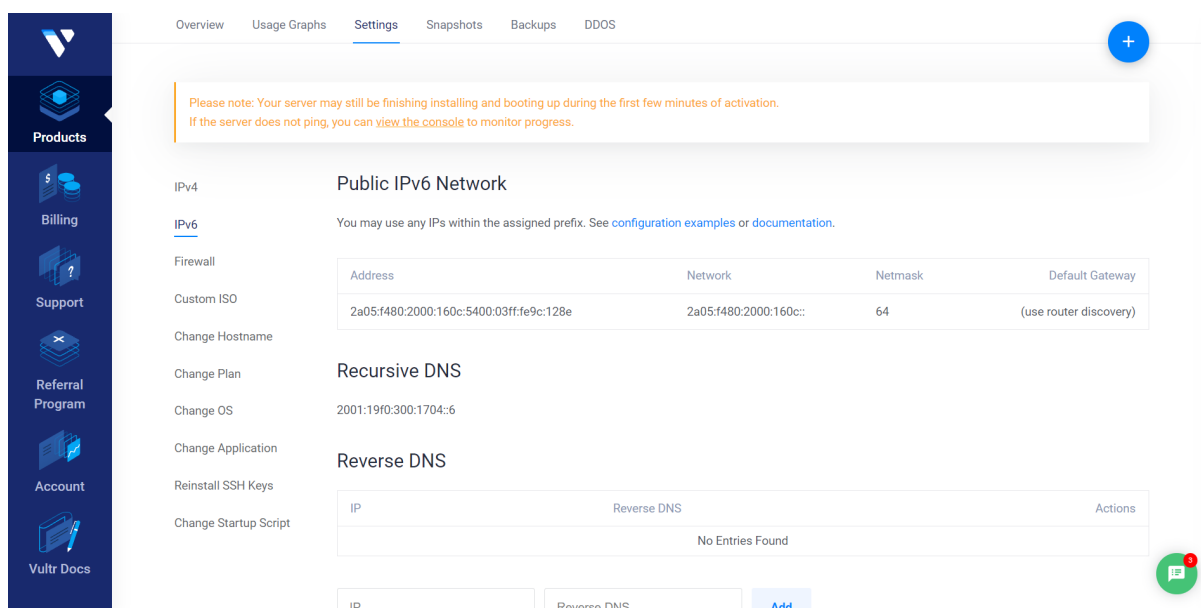


The screenshot shows the 'Additional Features' section of the Vultr deployment interface. A sidebar on the left contains navigation links: Products, Billing, Support, Referral Program, Account, and Vultr Docs. The main content area has a section titled 'Additional Features' with the following options:

- ☒ Enable IPv6 [?](#)
- ☐ No Public IPv4 Address [?](#)
- ☐ Enable Auto Backups [\\$4.00/mo](#)
- ☐ Enable Private Networking [?](#)

Below this is the 'Startup Script' section with an 'Add New' button. The 'SSH Keys' section also has an 'Add New' button. At the bottom, the 'Servers Qty' is set to 1, and the 'Summary' shows a price of \$20.00/mo (\$0.030/hr). A 'Deploy Now' button is on the right. A small green chat icon with a red notification bubble is in the bottom right corner.

We can then move to the settings-> IPv6 menu to see our configuration.



The screenshot shows the 'Settings' page for a server in the Vultr dashboard. The sidebar on the left is the same as in the previous image. The top navigation bar includes 'Overview', 'Usage Graphs', 'Settings' (active), 'Snapshots', 'Backups', and 'DDOS'. A blue '+' button is in the top right. A yellow warning box at the top states: 'Please note: Your server may still be finishing installing and booting up during the first few minutes of activation. If the server does not ping, you can view the console to monitor progress.'

The 'Settings' page is divided into sections:

- IPv4**
- IPv6** (selected): Titled 'Public IPv6 Network', it states 'You may use any IPs within the assigned prefix. See [configuration examples](#) or [documentation](#).' Below is a table:

Address	Network	Netmask	Default Gateway
2a05:f480:2000:160c:5400:03ff:fe9c:128e	2a05:f480:2000:160c::	64	(use router discovery)

- Firewall**
- Custom ISO**
- Change Hostname**
- Change Plan**
- Change OS**: 2001:19f0:300:1704::6
- Change Application**
- Reverse DNS**: A table with columns 'IP', 'Reverse DNS', and 'Actions'. It shows 'No Entries Found'.
- Reinstall SSH Keys**
- Change Startup Script**

At the bottom, there is a form with 'IP' and 'Reverse DNS' input fields and an 'Add' button. A green chat icon with a red notification bubble is in the bottom right corner.

With Putty let's connect to our server using the credentials provided in the overview section. Let's move to the **/etc/netplan** folder and edit the **10-ens3.yaml** configuration file With the following commands

- `cd / etc / netplan`
- `nano 10-ens3.yaml`

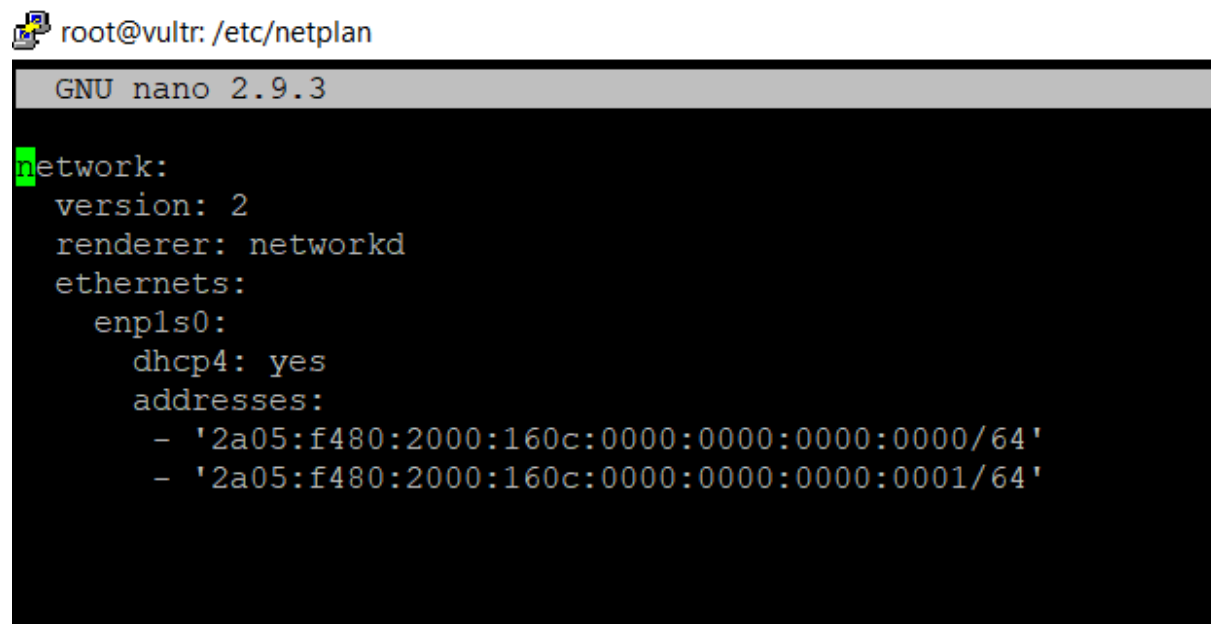
We add all the IPv6 addresses that we want to activate in the range provided by our hosting provider.

In this case we have access, in addition to the primary IPv6 address assigned to us, to the entire dynamic 64bit range of our primary IP (the last four groups of four digits).

For convenience we will start at 0000:0000:0000:0000 and increment the last digit by one number.

(We are using Vultr as an example, but that should work in a similar way with any other VPS/dedicated server provider, from Ubuntu 17.04 or above).

As shown in the picture



```

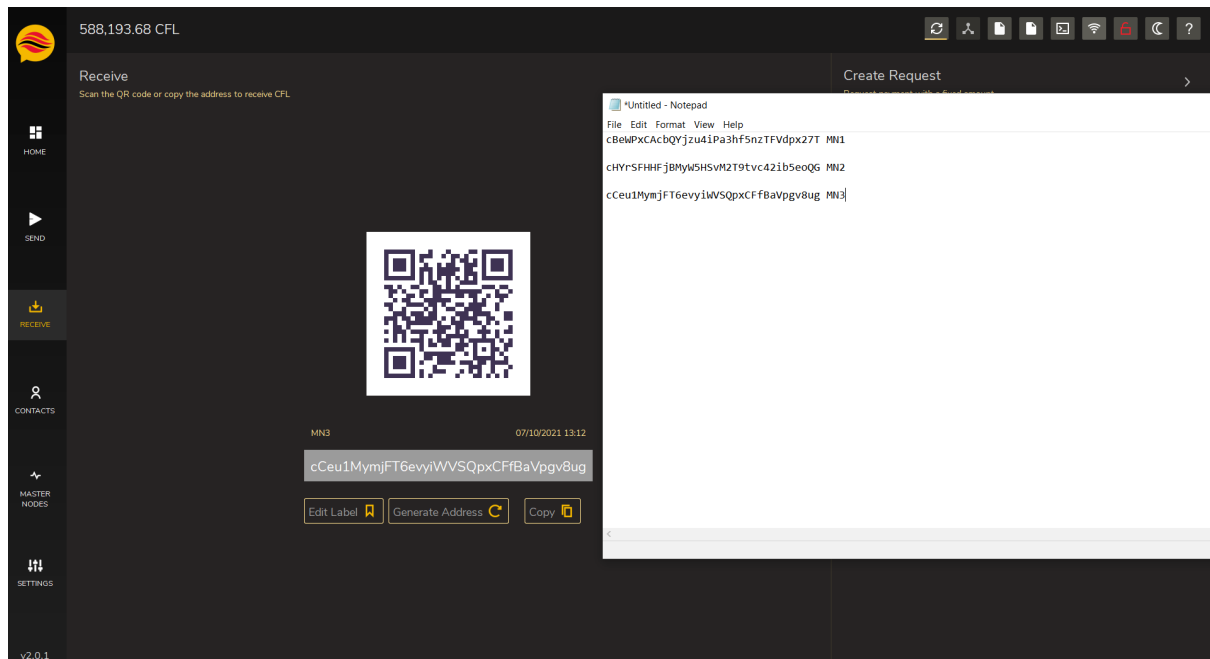
root@vultr: /etc/netplan
GNU nano 2.9.3
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      dhcp4: yes
      addresses:
        - '2a05:f480:2000:160c:0000:0000:0000:0000/64'
        - '2a05:f480:2000:160c:0000:0000:0000:0001/64'
  
```

We press `ctrl + x`, `y`, `enter` to exit and save the changes.

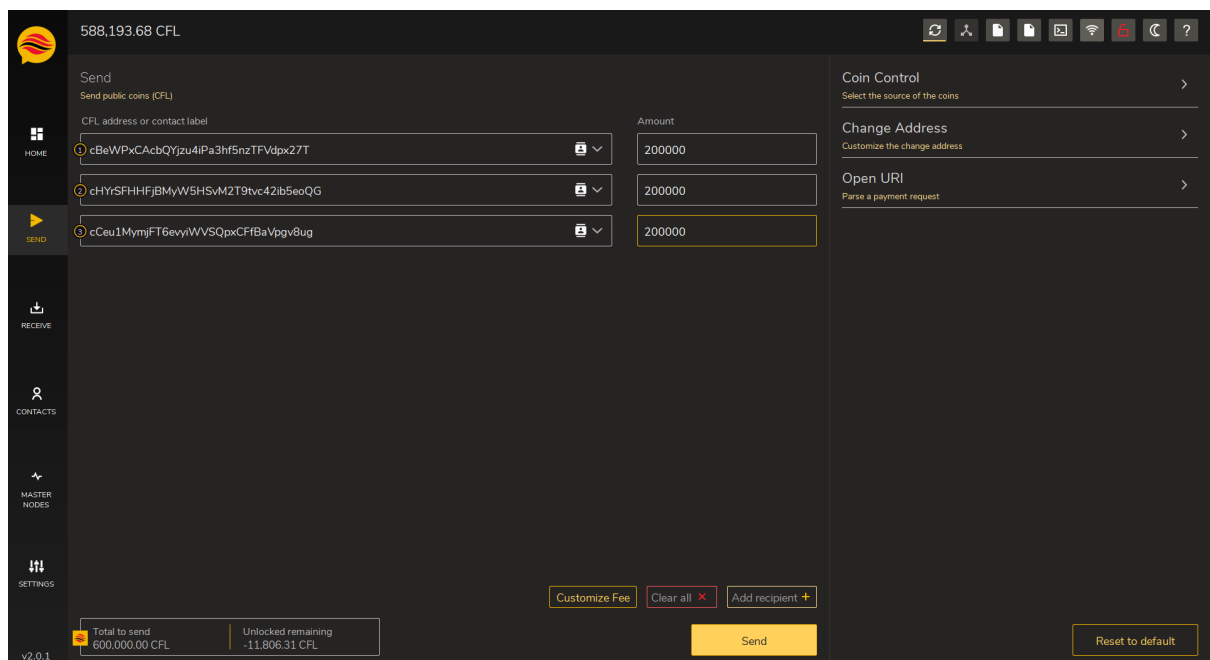
Then we type the **netplan apply** command to activate the new configuration.

Once this is done we can start sending the collateral transactions to configure the three new masternodes (in our case CFL).

In the wallet we generate three addresses and assign a label to them.

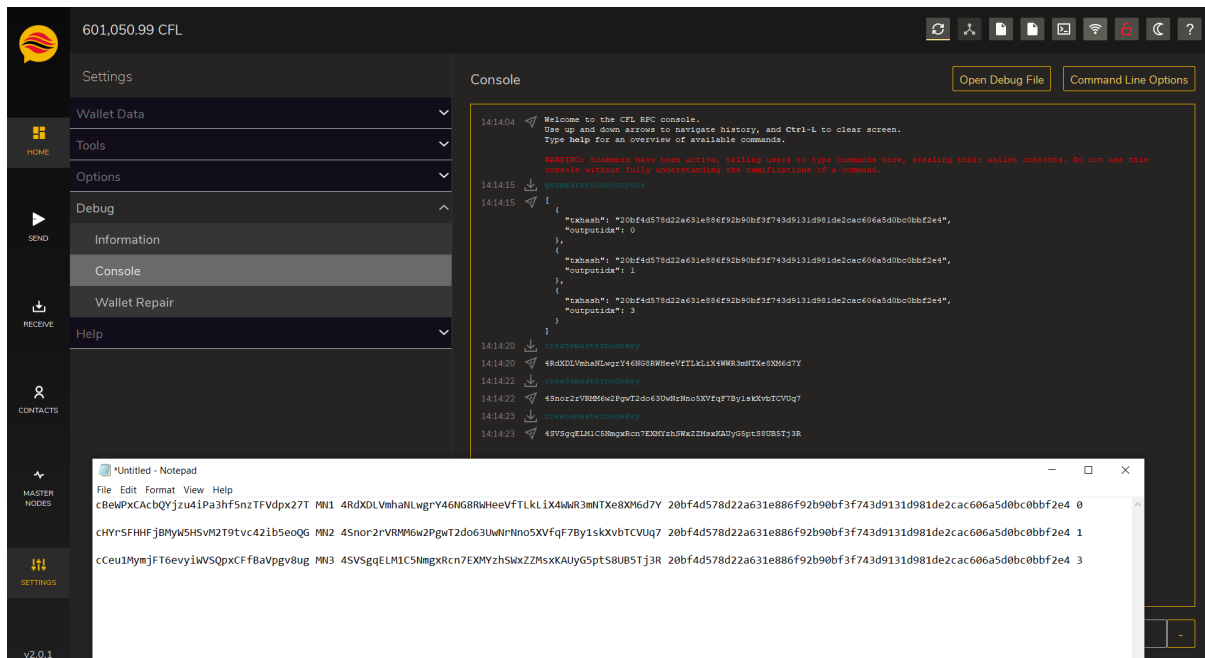


Now we can send the three collaterals, we can do that in a single transaction.



Now we can verify the transaction id and index, and generate three masternode keys. We use the following commands

- `getmasternodeoutputs`
- `createmasternodekey`



Now we can start configuring the masternode.conf file just adding our server IP addresses. Now download the linux wallet on our server and copy it to have three different instances. We use the following set of commands in the home directory, in our case /root

- `wget https://github.com/CryptoFlowCoin/CryptoFlowCoin/releases/download/v2.0.1.0/CryptoFlow-2.0.1.0-Linux.zip`
- `mkdir cfl1 cfl2 cfl3`
- `unzip CryptoFlow-2.0.1.0-Linux.zip`
- `cp cryptoflowd cryptoflow-cli cfl1/`
- `cp cryptoflowd cryptoflow-cli cfl2`
- `cp cryptoflowd cryptoflow-cli cfl3`
- `rm CryptoFlow-2.0.1.0-Linux.zip cryptoflow*`

Before starting the wallets, we manually create the data directories and related configuration files.

- `mkdir .cryptoflow .cryptoflow2 .cryptoflow3`

For each data directory we create a cryptoflow.conf configuration file

Example

- nano ~/.cryptoflow/cryptoflow.conf
- nano ~/.cryptoflow2/cryptoflow.conf
- nano ~/.cryptoflow3/cryptoflow.conf

Here we can enter the configuration parameters and save the file.

Below we show two examples

The first refers to the IPv4 configuration, the second refers to the IPv6 configuration.

```
rpcuser=cfl3
rpcpassword=cflmn3
rpcport=13336
rpcallowip=127.0.0.1
port=13333
listen=1
server=1
daemon=1
maxconnections=64
bind=70.34.203.123
masternode=1
masternodeaddr=70.34.203.123:13333
masternodeprivkey=4SVSgqELM1C5NmngxRcn7EXMYzhSWxZZMsxKAUyG5ptS8UB5Tj3R
```

```
rpcuser=cfl1
rpcpassword=cflmn1
rpcport=13334
rpcallowip>:::1
port=13333
listen=1
server=1
daemon=1
maxconnections=64
bind=[2a05:f480:2000:160c:0000:0000:0000:0000]
masternode=1
masternodeaddr=[2a05:f480:2000:160c:0000:0000:0000:0000]:13333
masternodeprivkey=4RdXDLVmhaNLwgrY46NG8RWHeeVfTLkLiX4WWR3mNTXe8XM6d7
Y
```

Now we can start our wallets taking care to specify the path of the data directory when it is different from the default one.

It is also suggested to start the IPv6 wallets first and then the IPv4 one.

- `~/cfl1/cryptoflowd`
- `~/cfl1/cryptoflowd -datadir=/root/.cryptoflow2 -conf=/root/.cryptoflow2/cryptoflow.conf`
- `~/cfl1/cryptoflowd -datadir=/root/.cryptoflow3 -conf=/root/.cryptoflow3/cryptoflow.conf`

We now wait for the wallets to be fully synchronized.

Once the synchronization is complete we can restart our local wallet to apply the changes made previously to the `masternode.conf` file and then start the masternodes from the debug console using **startmasternode** command, as in the picture.

The screenshot shows a terminal window titled "Console" with two buttons: "Open Debug File" and "Command Line Options". The terminal output is as follows:

```
14:55:04 Welcome to the CFL RPC console.
Use up and down arrows to navigate history, and Ctrl-L to clear screen.
Type help for an overview of available commands.

WARNING: Scammers have been active, telling users to type commands here, stealing their wallet contents. Do not use this
console without fully understanding the ramifications of a command.

15:24:16 startmasternode alias false MN1
15:24:16 {
  "alias": "MN1",
  "result": "success",
  "error": ""
}

15:24:25 startmasternode alias false MN2
15:24:25 {
  "alias": "MN2",
  "result": "success",
  "error": ""
}

15:24:27 startmasternode alias false MN3
15:24:27 {
  "alias": "MN3",
  "result": "success",
  "error": ""
}
```

At the bottom, there is a "Console input" field with a cursor and a "-" button.