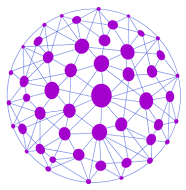
RAPTURE

prosper. foster. elevate.



Masternode Setup Guide

(HOT+COLD Wallet setup with Vultr.com Linux VPS)

What is a masternode?

Masternodes are computers on the Rapture network that provide network services and facilitate PrivateSend and InstantSend functionality. They also partake in the governance functions of the network and are allowed to submit and vote on proposals for the network. A masternode should have a fixed ip address and a stable uptime.

Since masternodes are allowed to govern and also facilitate transactions, there is a requirement for 1,000 Rapture to be held as collateral in order to operate a masternode. This makes it inherently difficult and expensive for an individual to setup and operate a majority of the masternodes on the network.

Rapture masternodes need to run with what's known as a hot/cold wallet setup. The basic premise is that the "cold" wallet is your local computer and the "hot" wallet is a remote VPS (virtual private server). The cold wallet effectively holds your 1,000 Rapture collateral and the rewards that are generated are sent to that wallet. This wallet is (or should be) password protected and can be closed once the masternode is started. The "hot" wallet is an empty wallet on the VPS. You don't send funds to it and don't even need to know its wallet address. It remains unlocked, but also empty….so if your VPS is ever compromised, there's no risk of losing funds.

With that said, let's get onto the process of setting up a Rapture masternode…

NOTE - If you see any error messages while working through this guide or certain commands aren't working, please retrace your steps or stop in at our Discord channel for support.

Also, there are other setup guides out there and they take different approaches and place files in different directories....jumping between guides is not recommended unless you're an experienced Linux user and can easily and account for the differences in paths and commands.

SCREENSHOTS SHOW v1.0.0 but the commands have been updated to v1.1.0.
BE SURE TO UPDATE YOUR LOCAL WALLET TO v1.1.0 PRIOR TO UPDATING THE MASTERNODE

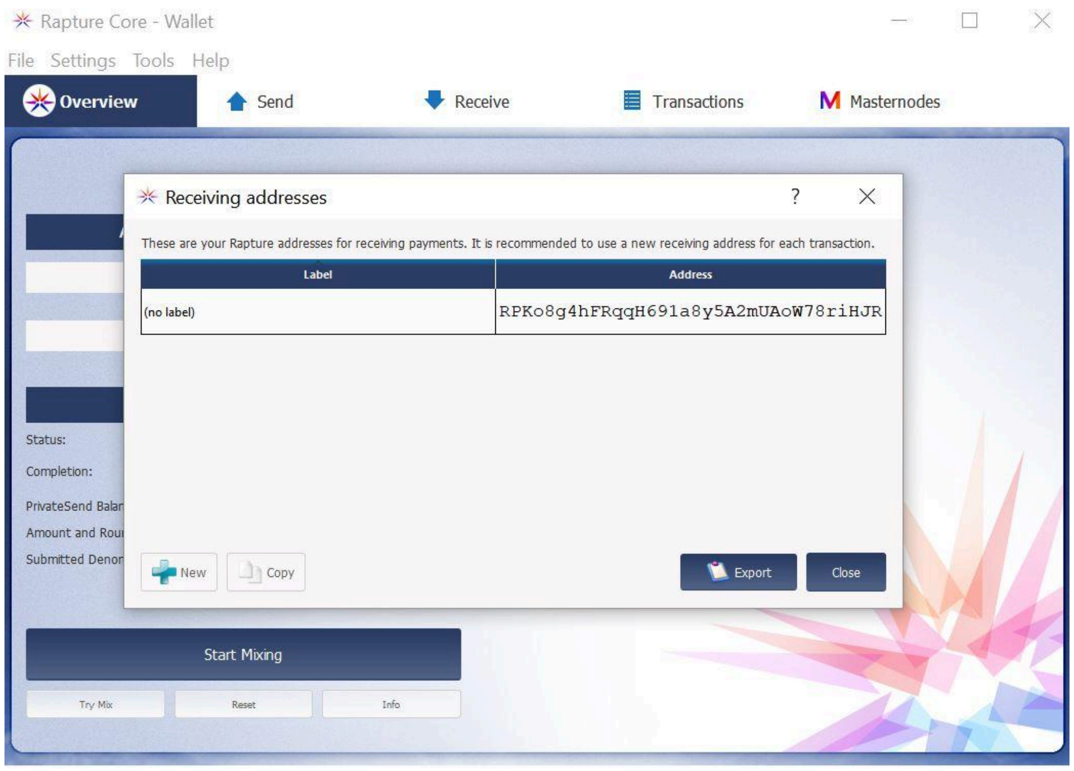
Setup Guide

Step 1 - Setting up the collateral transaction

The first step involves sending exactly 1,000 Rapture to a new wallet address. You'll want to have a small amount above 1,000 Rapture to cover the transaction fee, so you'll need to have a starting balance of at least, say 1,001.00 Rapture. First, we'll create a new wallet address to hold the 1,000 collateral. This will also be the address that the masternode rewards are sent to.

In the QT wallet, choose:

File->Receiving addresses…

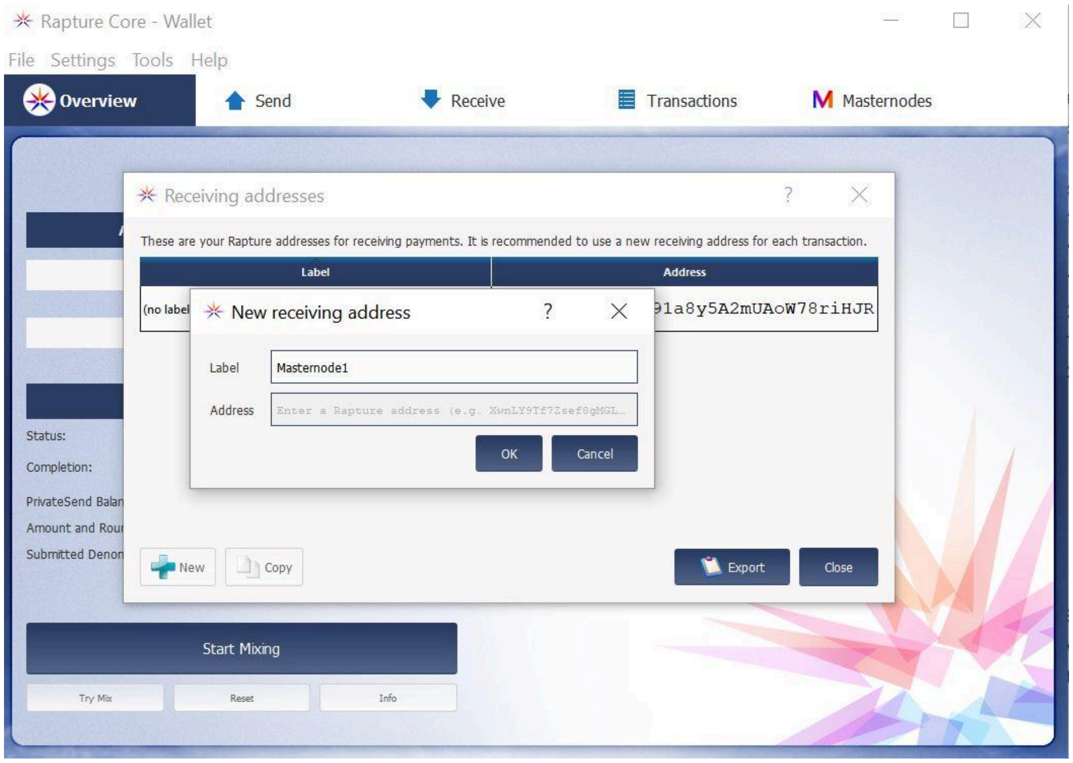


These are your Rapture addresses for receiving payments. It is recommended to use a new receiving address for each transaction.

Label	Address
(no label)	RPKo8g4hFRqqH691a8y5A2mUAoW78riHJR

This brings up a list of the receiving addresses managed by your wallet.

Click **+New**



These are your Rapture addresses for receiving payments. It is recommended to use a new receiving address for each transaction.

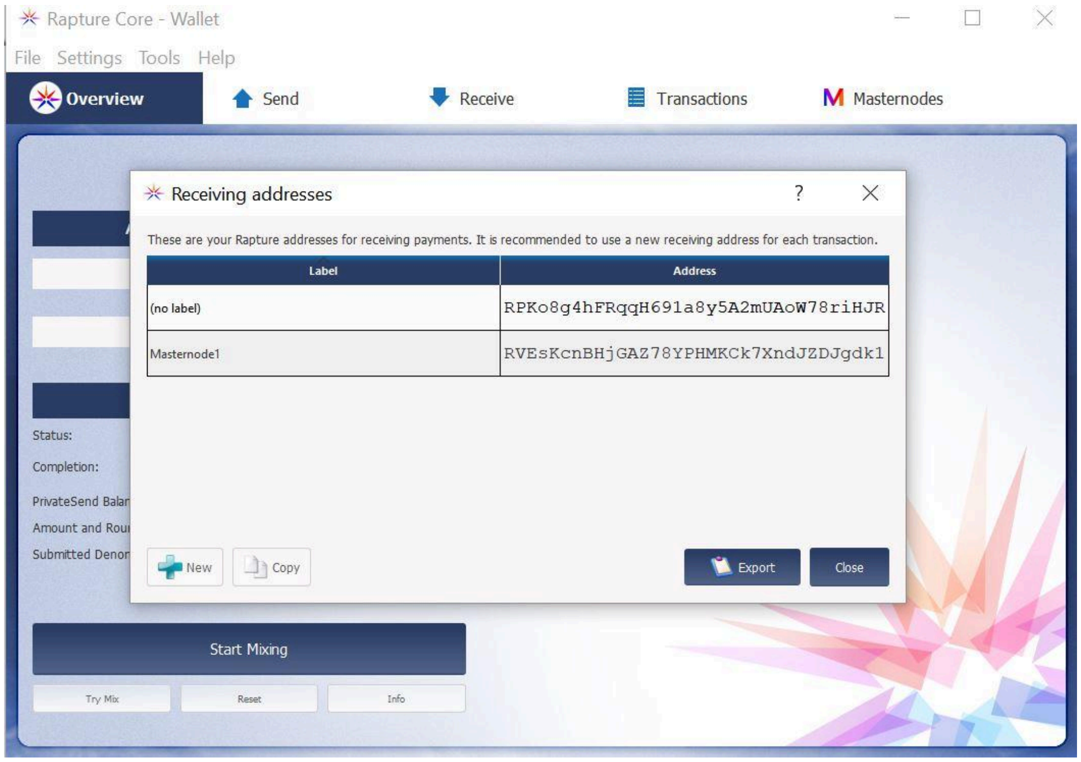
Label	Address
(no label	91a8y5A2mUAoW78riHJR

Label: Masternode1

Address: Enter a Rapture address (e.g. XwnLY9Tf7Zsef8gMGL...

...and give it a label, such as **Masternode1**.

Click **Ok** to create the address. You'll see the new address in the list, select it and choose **Copy** to store that address in the clipboard.

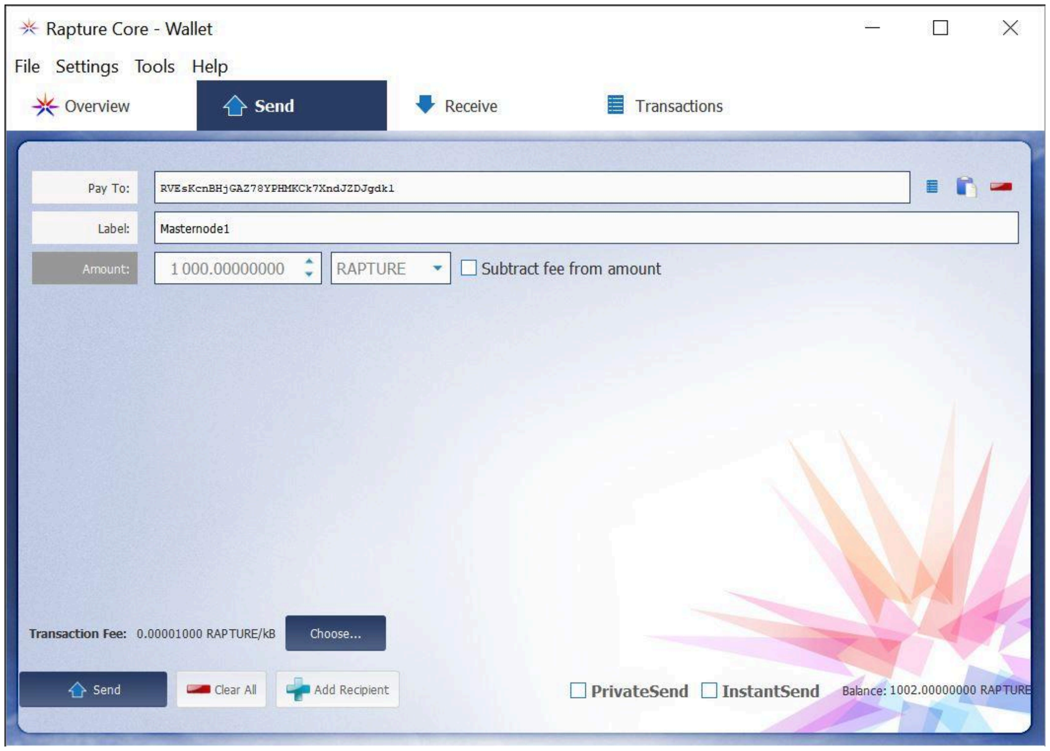


These are your Rapture addresses for receiving payments. It is recommended to use a new receiving address for each transaction.

Label	Address
(no label)	RPKo8g4hFRqqH691a8y5A2mUAoW78riHJR
Masternode1	RVEsKcnBHjGAZ78YPHMKCk7XndJZDJgdk1

Click **Close** to exit the receiving address dialog. Now we'll send the 1,000 collateral amount to the address you just created.

Under the **Send** tab, paste the address that you copied into the **Pay To:** field. The Label field should pre-populate with the label you gave it earlier, in this case, Masternode1.



Pay To: RVEsKcnBHjGAZ78YPHMKCk7XndJZDJgdk1

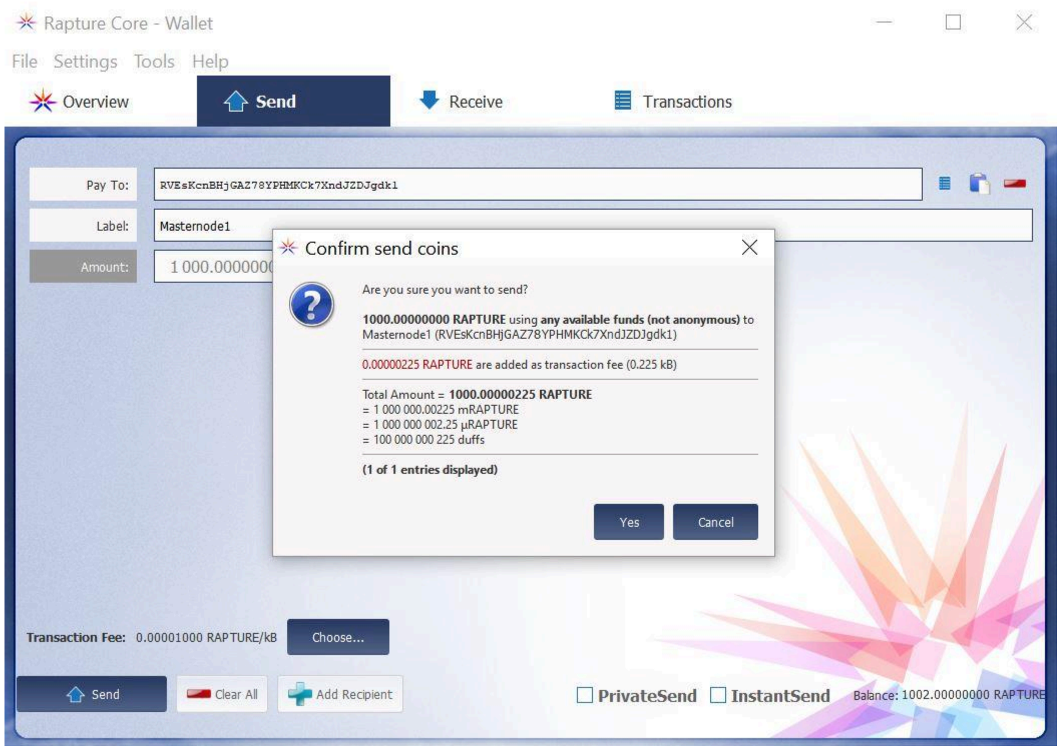
Label: Masternode1

Amount: 1 000.00000000 RAPTURE ☐ Subtract fee from amount

Transaction Fee: 0.00001000 RAPTURE/kB

☐ PrivateSend ☐ InstantSend Balance: 1002.00000000 RAPTURE

In the Amount: field, enter **1,000.** Do **NOT** check "Subtract fee from amount" as this will subtract the transaction fee from the Amount and your transaction will be below the required 1,000 Rapture.



Pay To: RVEsKcnBHjGAZ78YPHMKCk7XndJZDJgdk1

Label: Masternode1

Amount: 1 000.0000000

Are you sure you want to send?

1000.00000000 RAPTURE using **any available funds (not anonymous)** to Masternode1 (RVEsKcnBHjGAZ78YPHMKCk7XndJZDJgdk1)

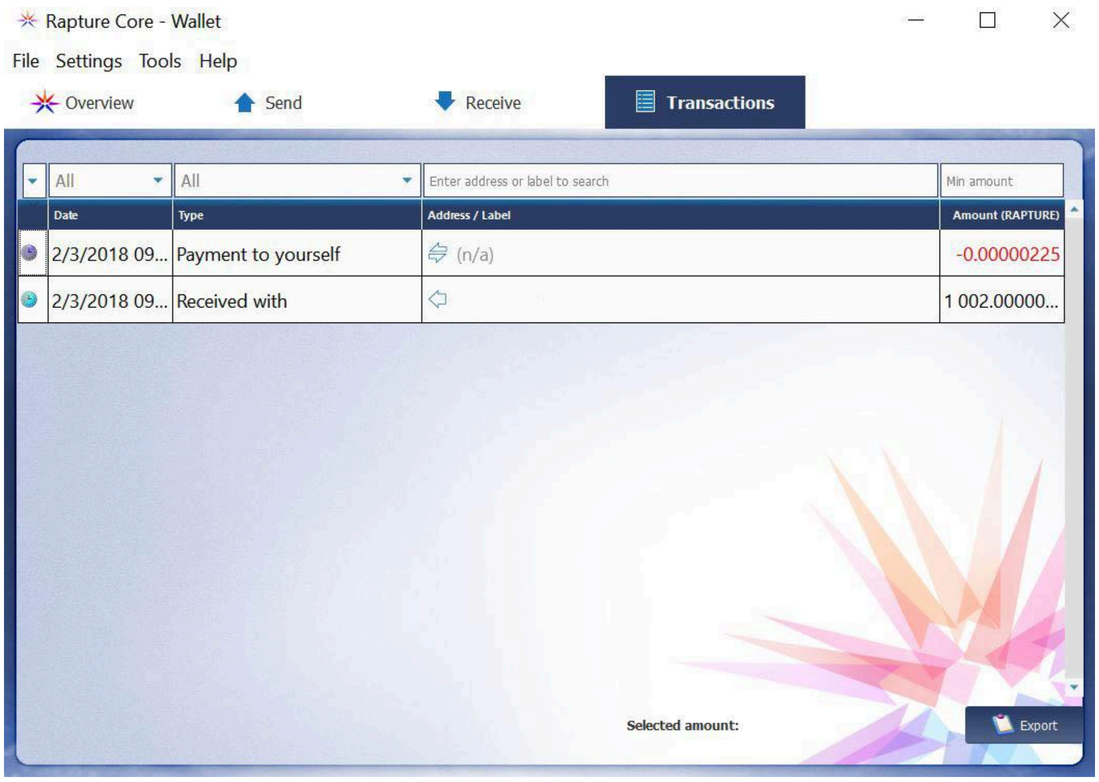
0.00000225 RAPTURE are added as transaction fee (0.225 kB)

Total Amount = **1000.00000225 RAPTURE**
= 1 000 000.00225 mRAPTURE
= 1 000 000 002.25 µRAPTURE
= 100 000 000 225 duffs

(1 of 1 entries displayed)

Transaction Fee: 0.00001000 RAPTURE/kB

☐ PrivateSend ☐ InstantSend Balance: 1002.00000000 RAPTURE



Date	Type	Address / Label	Amount (RAPTURE)
2/3/2018 09...	Payment to yourself	(n/a)	-0.00000225
2/3/2018 09...	Received with		1 002.00000...

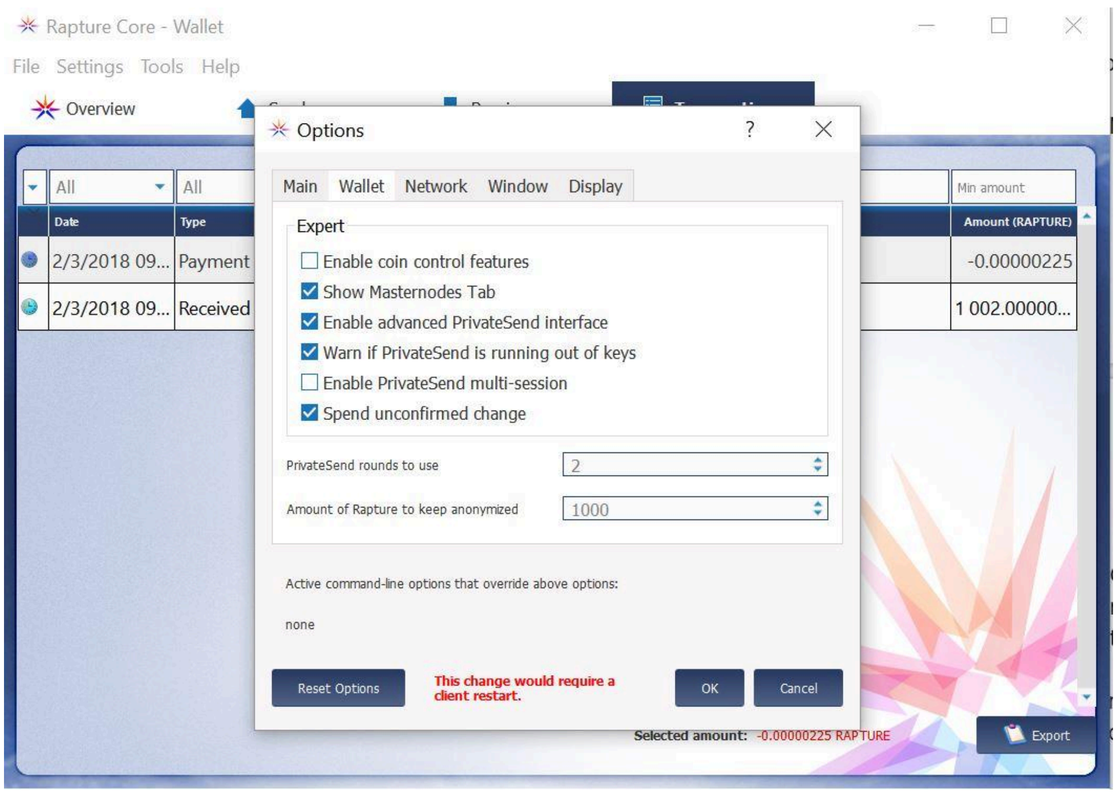
Click **Send** and your transaction will be broadcast to the network. You'll need to wait for 15 confirmations (the current number of confirmations is viewable in the Transactions tab) before the masternode will fully activate, but we can start working through the rest of the setup in the meantime.

Step 2 - Generate the masternode private key and txid

The next step is to enable the Masternodes tab in your local wallet and use the debug console to output a few important details that we'll need.

To enable the Masternodes tab in your local wallet, go to:

Settings->Options->Wallet->Show Masternodes Tab



Expert

- [ ] Enable coin control features
- [✓] Show Masternodes Tab
- [✓] Enable advanced PrivateSend interface
- [✓] Warn if PrivateSend is running out of keys
- [ ] Enable PrivateSend multi-session
- [✓] Spend unconfirmed change

PrivateSend rounds to use	2
Amount of Rapture to keep anonymized	1000

Active command-line options that override above options:

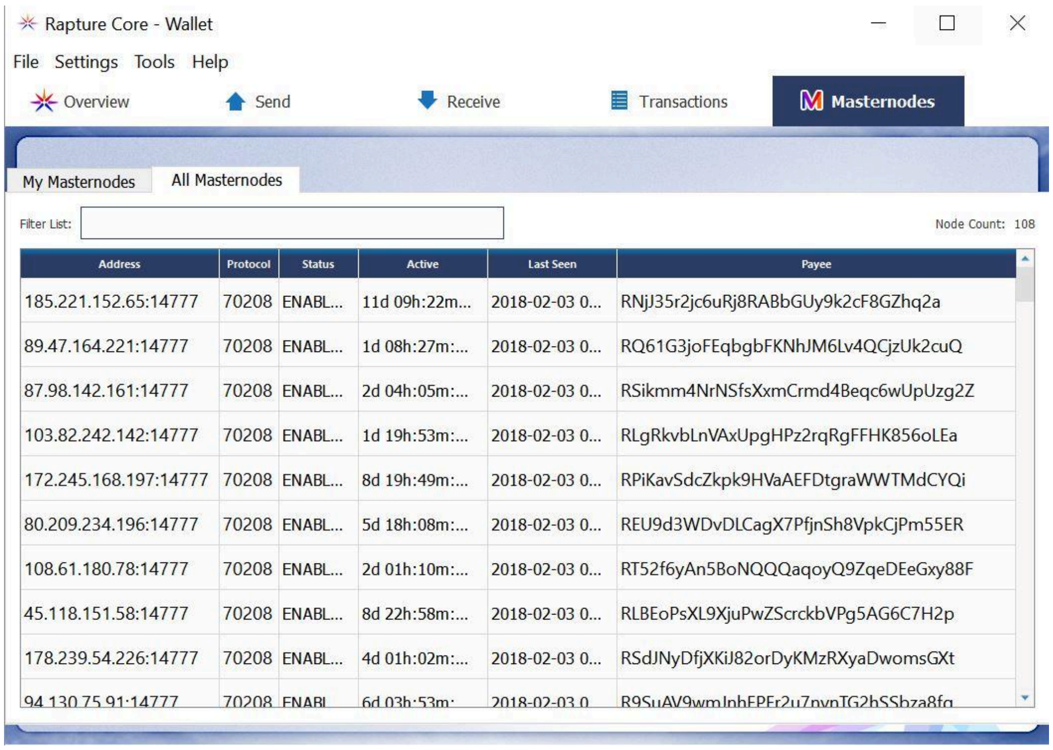
none

This change would require a client restart.

	Date	Type	Amount (RAPTURE)
	2/3/2018 09...	Payment	-0.00000225
	2/3/2018 09...	Received	1 002.00000...

Selected amount: -0.00000225 RAPTURE

Click **Ok** to apply the setting and then close and re-open your local wallet. After re-opening, you should see the Masternodes tab which will display the full list of masternodes on the network, as well as masternodes associated with your local wallet (My Masternodes is probably empty because we haven't added one yet).



Node Count: 108

Address	Protocol	Status	Active	Last Seen	Payee
185.221.152.65:14777	70208	ENABL...	11d 09h:22m...	2018-02-03 0...	RNjJ35r2jc6uRj8RABbGUy9k2cF8GZhq2a
89.47.164.221:14777	70208	ENABL...	1d 08h:27m:...	2018-02-03 0...	RQ61G3joFEqbgbFKNhJM6Lv4QCjzUk2cuQ
87.98.142.161:14777	70208	ENABL...	2d 04h:05m:...	2018-02-03 0...	RSikmm4NrNSfsXxmCrmd4Beqc6wUpUzg2Z
103.82.242.142:14777	70208	ENABL...	1d 19h:53m:...	2018-02-03 0...	RLgRkvbLnVAxUpgHPz2rqRgFFHK856oLEa
172.245.168.197:14777	70208	ENABL...	8d 19h:49m:...	2018-02-03 0...	RPiKavSdcZkpk9HVaAEFDtgraWWTMdCYQi
80.209.234.196:14777	70208	ENABL...	5d 18h:08m:...	2018-02-03 0...	REU9d3WDvDLCagX7PfjnSh8VpkCjPm55ER
108.61.180.78:14777	70208	ENABL...	2d 01h:10m:...	2018-02-03 0...	RT52f6yAn5BoNQQQaqoyQ9ZqeDEeGxy88F
45.118.151.58:14777	70208	ENABL...	8d 22h:58m:...	2018-02-03 0...	RLBEoPsXL9XjuPwZScrckbVPg5AG6C7H2p
178.239.54.226:14777	70208	ENABL...	4d 01h:02m:...	2018-02-03 0...	RSdJNyDfjXKiJ82orDyKMzRXyaDwomsGXt
94.130.75.91:14777	70208	ENABL	6d 03h:53m:	2018-02-03 0	R9SuAV9wmJnbFPEr2u7nvnTG2hSSbza8fq

Next, we need to access the **debug console** to output what's known as the **collateral txid** and the **masternode private key**. The collateral txid is an identifier for the transaction of 1,000 Rapture that you made previously. The masternode private key is a key that is specific to your wallet and is used to validate your masternode on the network. Like any private key, you want to keep this secret...**there's no reason to share this number with anyone or post it publically.**

To access the debug console, go to:

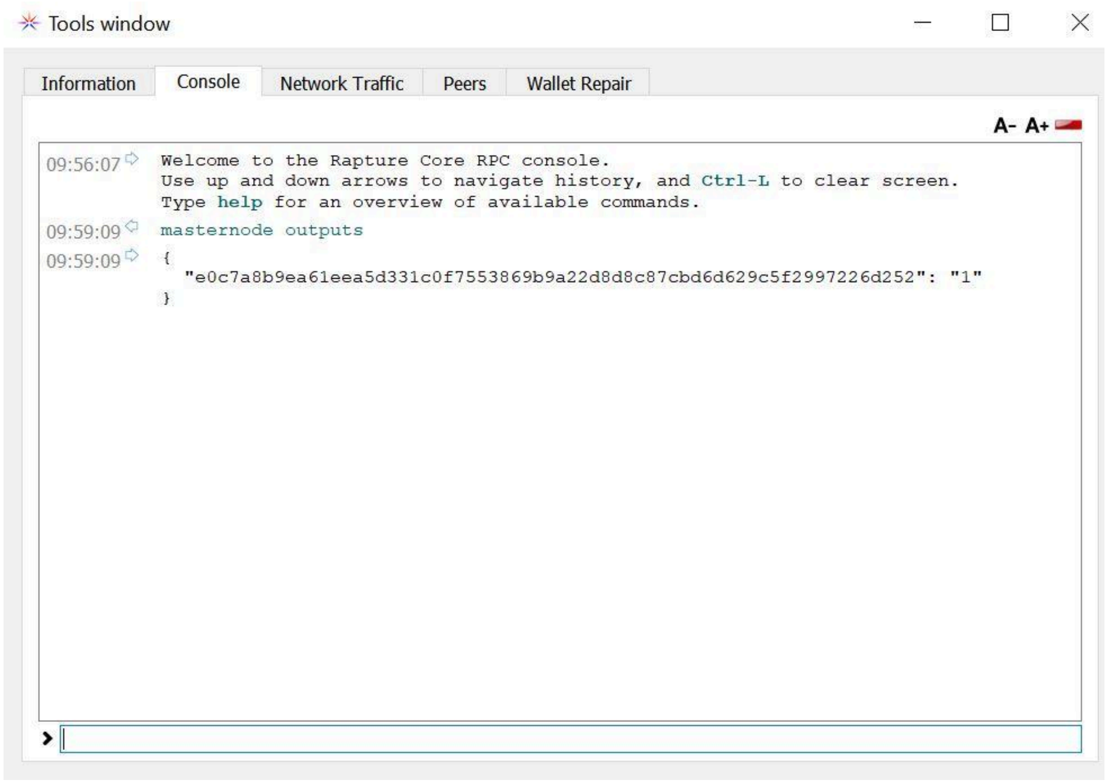
Tools->Debug console

The debug console has a wide range of commands that provide all sorts of helpful information and provide convenient functions. In this case, we'll use it to obtain your masternode txid and private key.

In the debug console, type in:

masternode outputs

This will list all transactions that meet the parameters for starting a masternode. Since we sent a transaction to a new local wallet address for 1,000 Rapture, we should see the transaction id listed here (if you're not seeing any output, verify that you have in fact sent a transaction to a local wallet address for exactly 1,000 Rapture):



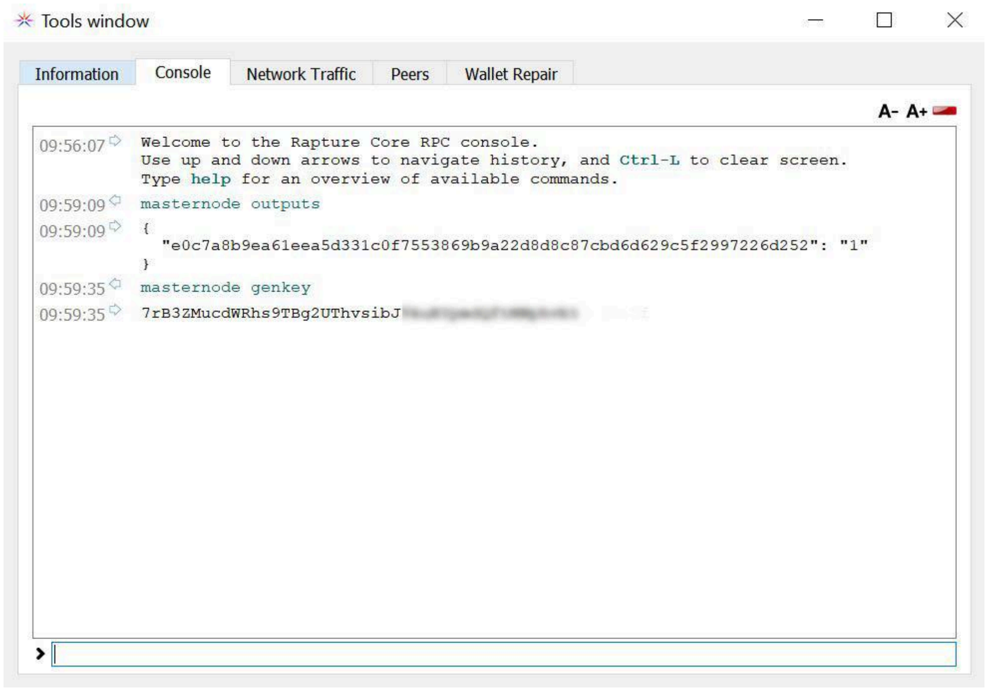
```
09:56:07  Welcome to the Rapture Core RPC console.
          Use up and down arrows to navigate history, and Ctrl-L to clear screen.
          Type help for an overview of available commands.
09:59:09  masternode outputs
09:59:09  {
            "e0c7a8b9ea61eea5d331c0f7553869b9a22d8d8c87cbd6d629c5f2997226d252": "1"
          }
```

Copy that output and paste it somewhere convenient (in an empty Notepad session is fine).

Next, we'll generate a private key for you masternode. Type in:

masternode genkey

Copy and paste this number somewhere convenient as well, we'll use it shortly. With these two numbers output and pasted in a text document, we can proceed with the VPS setup.



```
09:56:07  Welcome to the Rapture Core RPC console.
          Use up and down arrows to navigate history, and Ctrl-L to clear screen.
          Type help for an overview of available commands.
09:59:09  masternode outputs
09:59:09  {
            "e0c7a8b9ea61eea5d331c0f7553869b9a22d8d8c87cbd6d629c5f2997226d252": "1"
          }
09:59:35  masternode genkey
09:59:35  7rB3ZMucdWRhs9TBg2UThvsibJ[redacted]
```

Step 3 - Setting up a Linux VPS with Vultr.com

This part of the guide will take you through setting up a VPS with vultr.com. There are a few other good VPS services out there, but Vultr has basic servers that can be run for $2.50/month and provide the resources necessary to run a masternode.

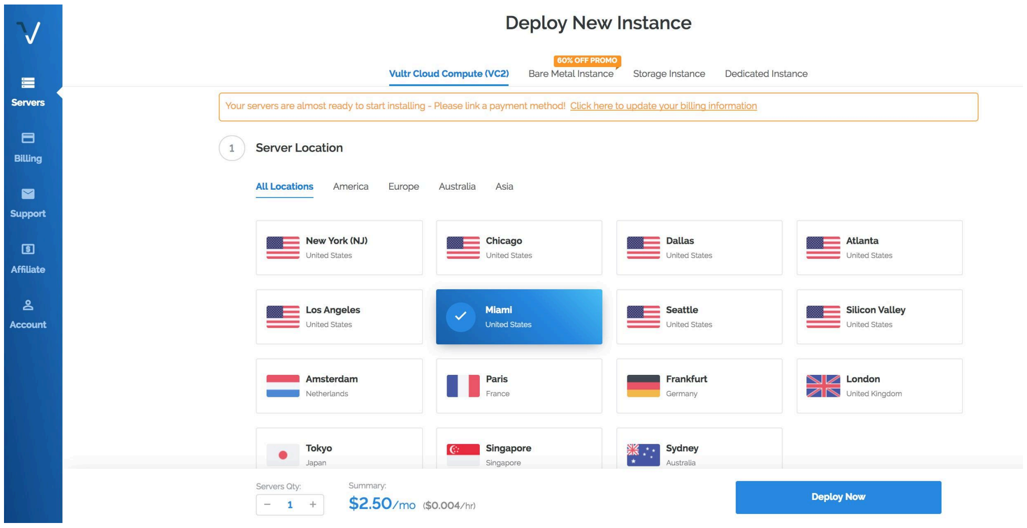
https://vultr.com

Create an account with Vultr.com and setup a payment method. Once you've created an account with Vultr and linked a credit card or payment method, navigate to:

https://my.vultr.com/

This is your main console for Vultr and where you can deploy and monitor servers. Click on the **Servers** tab on the left. If you have servers running already, you'll see them listed here and can click **+** to launch a new server. Otherwise, you should already be seeing the server deployment page.

Choose a location:



Deploy New Instance

Vultr Cloud Compute (VC2) | Bare Metal Instance (60% OFF PROMO) | Storage Instance | Dedicated Instance

Your servers are almost ready to start installing - Please link a payment method! Click here to update your billing information

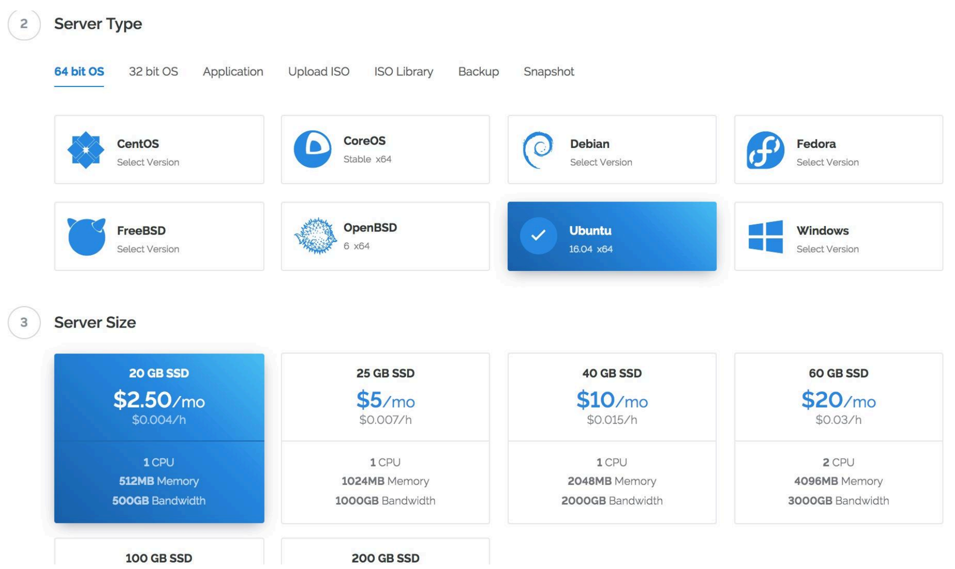
1 Server Location

All Locations | America | Europe | Australia | Asia

New York (NJ) United States	Chicago United States	Dallas United States	Atlanta United States
Los Angeles United States	✓ Miami United States	Seattle United States	Silicon Valley United States
Amsterdam Netherlands	Paris France	Frankfurt Germany	London United Kingdom
Tokyo Japan	Singapore Singapore	Sydney Australia	

Servers Qty: 1 Summary: $2.50/mo ($0.004/hr) Deploy Now

...and then choose a type. For Rapture, we want a **64-bit Ubuntu 16.04 x64** installation. Next, choose a Server Size. The standard $2.50/month server should be adequate.



2 Server Type

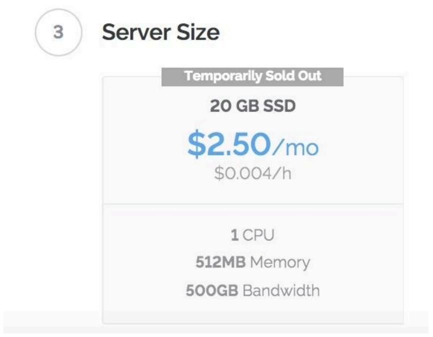
64 bit OS | 32 bit OS | Application | Upload ISO | ISO Library | Backup | Snapshot

CentOS Select Version	CoreOS Stable x64	Debian Select Version	Fedora Select Version
FreeBSD Select Version	OpenBSD 6 x64	✓ Ubuntu 16.04 x64	Windows Select Version

3 Server Size

✓ 20 GB SSD $2.50/mo $0.004/h 1 CPU 512MB Memory 500GB Bandwidth	25 GB SSD $5/mo $0.007/h 1 CPU 1024MB Memory 1000GB Bandwidth	40 GB SSD $10/mo $0.015/h 1 CPU 2048MB Memory 2000GB Bandwidth	60 GB SSD $20/mo $0.03/h 2 CPU 4096MB Memory 3000GB Bandwidth
100 GB SSD	200 GB SSD		

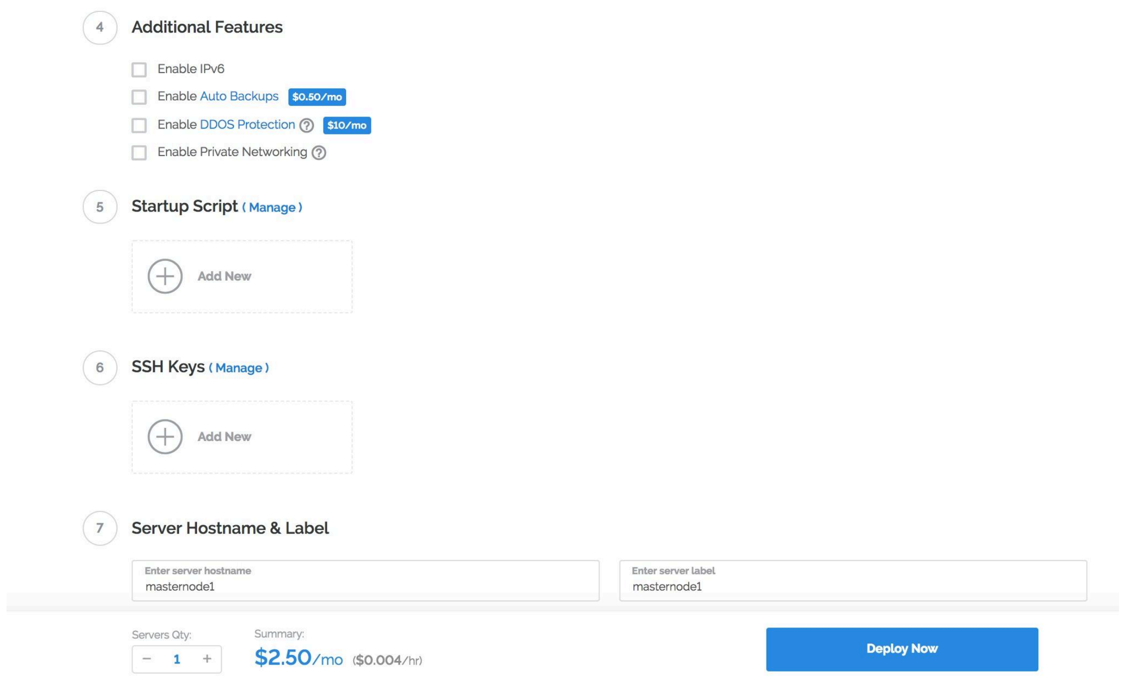
If it’s listed as Temporarily Sold Out:



3 Server Size

Temporarily Sold Out
20 GB SSD
$2.50/mo
$0.004/h
1 CPU
512MB Memory
500GB Bandwidth

...then scroll back up and choose a different location. It seems that the United States based locations have the best availability in the $2.50/month server size.


4 Additional Features

- Enable IPv6
- Enable Auto Backups $0.50/mo
- Enable DDOS Protection $10/mo
- Enable Private Networking

5 Startup Script (Manage)

Add New

6 SSH Keys (Manage)

Add New

7 Server Hostname & Label

Enter server hostname: masternode1

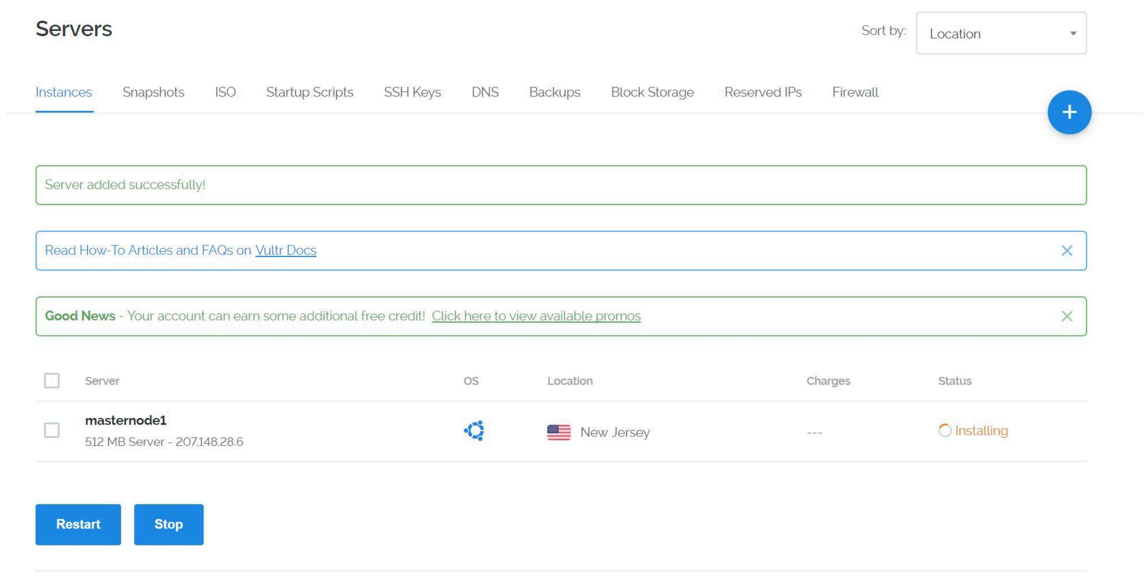
Enter server label: masternode1

Servers Qty: 1

Summary: $2.50/mo ($0.004/hr)

Deploy Now

All other options can remain at their defaults, but assign the server a hostname and label, in this case 'masternode1'. Then click Deploy Now to launch your server.


Servers

Sort by: Location

Instances | Snapshots | ISO | Startup Scripts | SSH Keys | DNS | Backups | Block Storage | Reserved IPs | Firewall

Server added successfully!

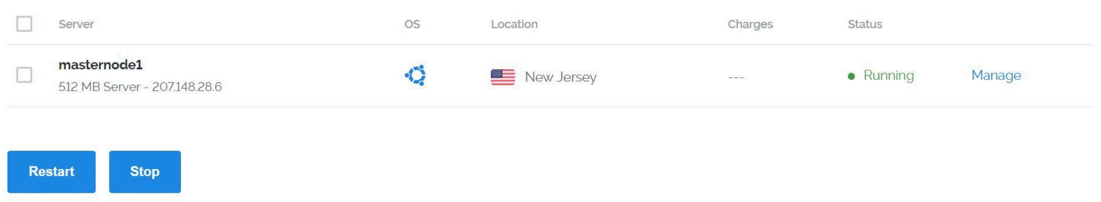
Read How-To Articles and FAQs on Vultr Docs

Good News - Your account can earn some additional free credit! Click here to view available promos

Server	OS	Location	Charges	Status
masternode1 512 MB Server - 207.148.28.6		New Jersey	---	Installing

Restart | Stop

You'll be taken back to the list of your servers and you'll see that the server you just deployed is installing. Once it's listed as Running, we can login for the first time.

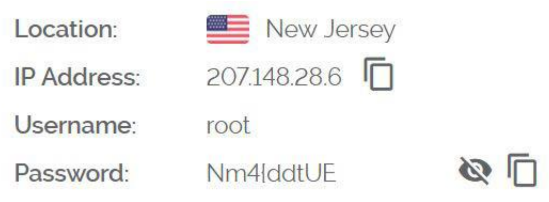


Server	OS	Location	Charges	Status	
masternode1 512 MB Server - 207.148.28.6		New Jersey	---	● Running	Manage

Restart | Stop

We need to retrieve the login credentials for the initial login. Click on the name of the server, in this case 'masternode1'.

This will bring you to the Server Information page for your VPS:



Location:	New Jersey
IP Address:	207.148.28.6
Username:	root
Password:	Nm4{ddtUE

Down in the bottom left you'll see the Username: root and Password. Click on the icon of the eye to reveal the password for the root account. This is the set of credentials that we'll use to login.

Step 4 - Logging into your VPS with PuTTY

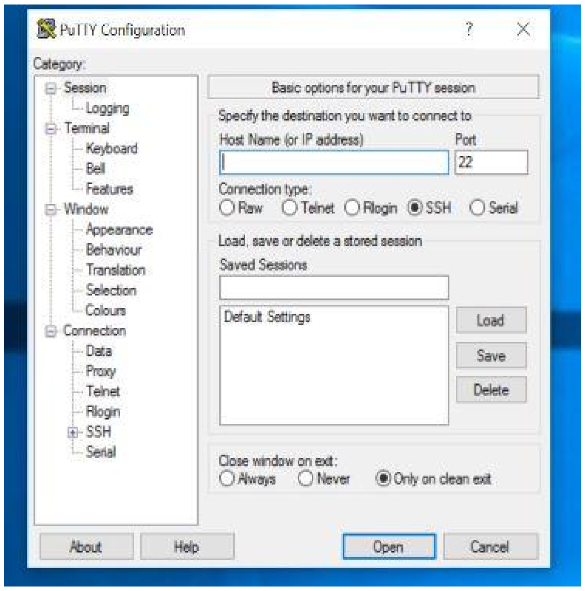
To access the VPS, we need to use a protocol called SSH. An SSH session will give you command-line access to your VPS and will be the mechanism we use for running commands on the Linux machine. We'll use a program called PuTTY. This is a popular and widely used ssh client in Windows. (For OSX users, you can use your Terminal and just run the command ssh root@<YOUR VPS IP ADDRESS>)

The latest version of PuTTY can be downloaded from here:
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

...or the most recent version at the time of this writing, 0.70, can be downloaded directly from here:

https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.70-installer.msi

Install PuTTY and run it from the Start Menu. Once open, you'll see the standard PuTTy interface:



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

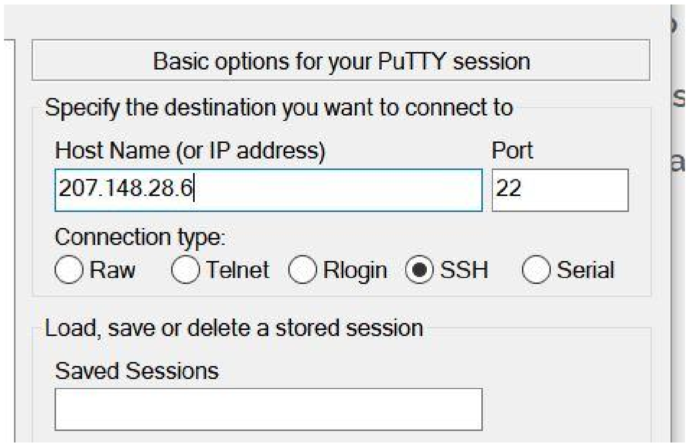
Connection type: ( ) Raw ( ) Telnet ( ) Rlogin [✓ SSH] ( ) Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ( ) Always ( ) Never [✓ Only on clean exit]

Under Host Name, enter the IP address for your VPS. In this case we use 207.148.28.6:



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 207.148.28.6

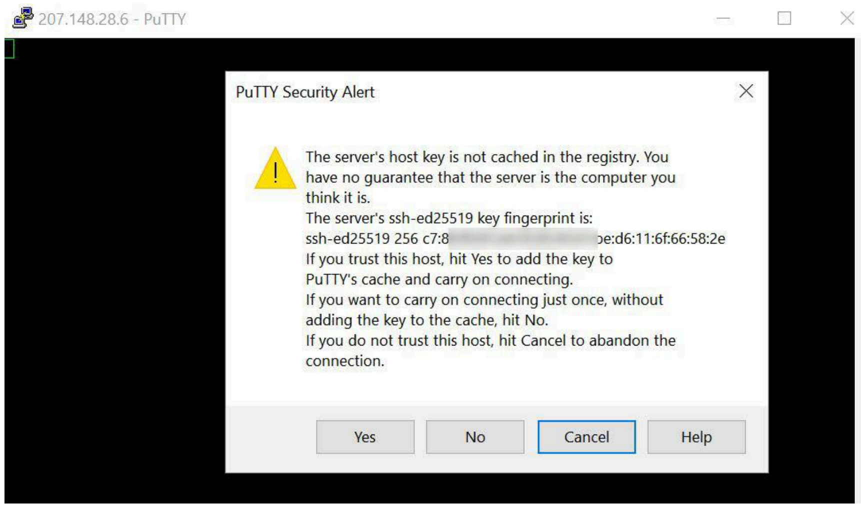
Port: 22

Connection type: ( ) Raw ( ) Telnet ( ) Rlogin [✓ SSH] ( ) Serial

Load, save or delete a stored session

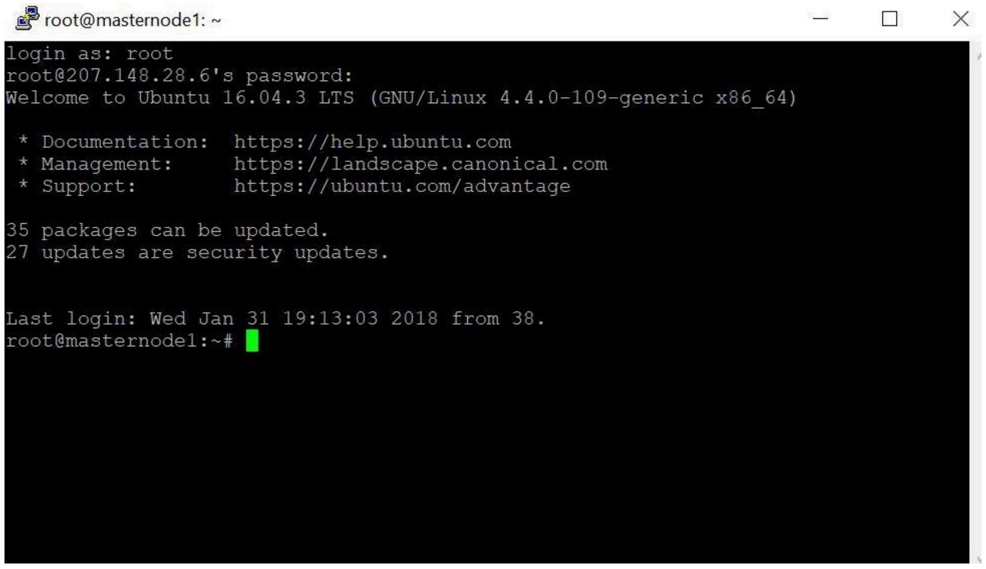
Saved Sessions

For convenience, you can also type a name (like “vultr masternode1”) under Saved Sessions and click save to store your VPN details. Click Open and PuTTY will initiate the first connection to your VPS. You’ll likely see a security alert listing the ssh key fingerprint, choose Yes.



```
The server's host key is not cached in the registry. You
have no guarantee that the server is the computer you
think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 256 c7:8[redacted]e:d6:11:6f:66:58:2e
If you trust this host, hit Yes to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the
connection.
```

Next up, you'll see the login prompt for your VPS. The username is **root** and the password is the password that was listed on your Server Information page in Vultr:



```
login as: root
root@207.148.28.6's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-109-generic x86_64)

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

35 packages can be updated.
27 updates are security updates.


Last login: Wed Jan 31 19:13:03 2018 from 38.
root@masternode1:~#
```

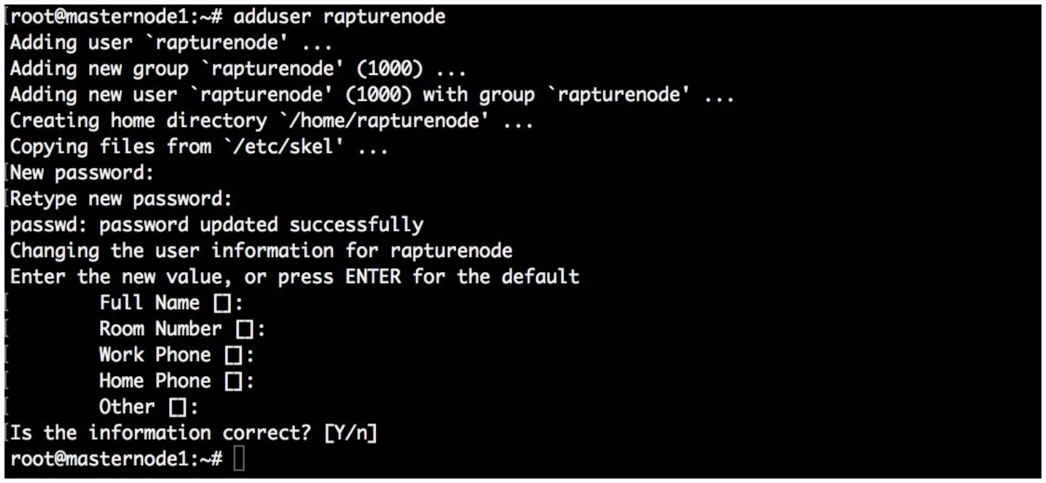
Note that in PuTTY, a right-click will paste the contents of your clipboard....so if you're copying and pasting the root password, a right-click will paste. You won't see any *'s or feedback for the password when you paste it, so just hit Enter after pasting. If you get an Access Denied message, double-check your root password and try typing it in manually rather than pasting.

Step 5 - Setting up a new user account

Now that we're logged in as root, the first thing we want to do is create a separate user to run the masternode under. The root account has the highest level of system access and it's generally not good to run as root if you can avoid it, so let's make a user named '**rapturenode**'.

adduser rapturenode

Assign a password that you'll remember, and then accept the remainder of the defaults:

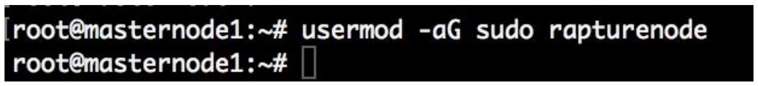


```
root@masternode1:~# adduser rapturenode
Adding user `rapturenode' ...
Adding new group `rapturenode' (1000) ...
Adding new user `rapturenode' (1000) with group `rapturenode' ...
Creating home directory `/home/rapturenode' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for rapturenode
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
root@masternode1:~#
```

Next, we'll give this user what's known as sudo access. This allows that user to run commands with elevated privileges and to do things like install software and system updates.

Type the following:

usermod -aG sudo rapturenode



```
root@masternode1:~# usermod -aG sudo rapturenode
root@masternode1:~#
```

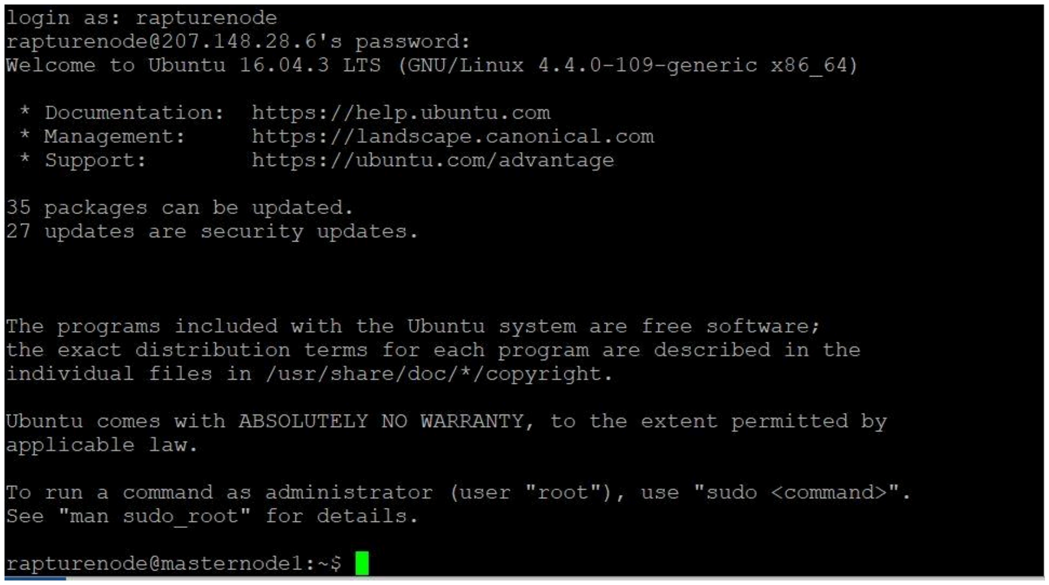
You won't see any output, but you've just given the user 'rapturenode' sudo access. Now we can logoff our root account from the VPS and re-connect as our new 'rapturenode' user. Type:

exit

...to close the current ssh connection. This will drop you back to the main PuTTY interface. Re-connect to the VPS the same way we did before, by typing the IP address into the Host Name field and click Open.

You'll be presented with the login prompt again, this time we'll log in as our 'rapturenode' user.

login as: **rapturenode**
password: **<the password that you set earlier>**

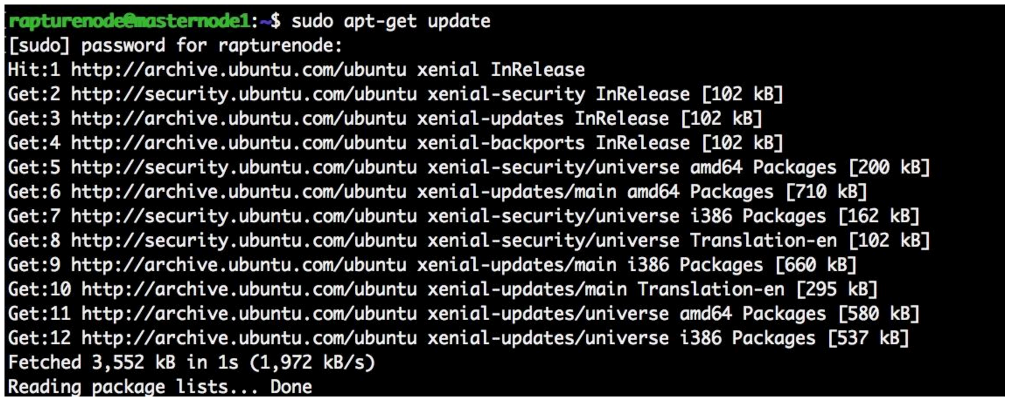


```
login as: rapturenode
rapturenode@207.148.28.6's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-109-generic x86_64)

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

35 packages can be updated.
27 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

rapturenode@masternode1:~$
```

Step 6 - Installing Rapture and dependencies

At this point, you should be logged into the vps and at a command prompt. The first thing we'll do is install updates to Ubuntu. The installation and management of software in Ubuntu is done using the 'apt-get' command. First run:

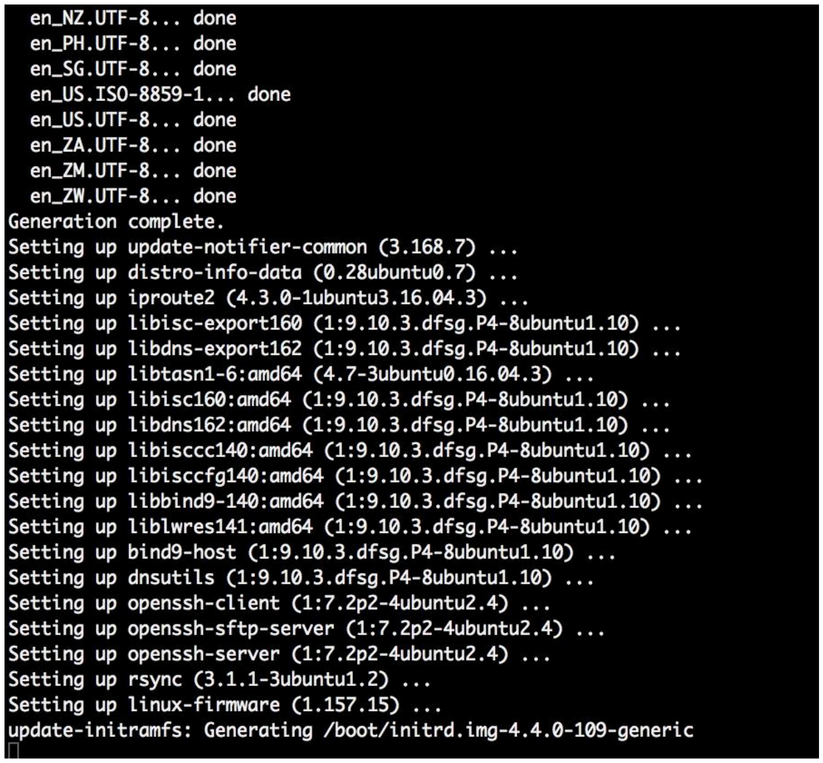
sudo apt-get update



```
rapturenode@masternode1:~$ sudo apt-get update
[sudo] password for rapturenode:
Hit:1 http://archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:4 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:5 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [200 kB]
Get:6 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [710 kB]
Get:7 http://security.ubuntu.com/ubuntu xenial-security/universe i386 Packages [162 kB]
Get:8 http://security.ubuntu.com/ubuntu xenial-security/universe Translation-en [102 kB]
Get:9 http://archive.ubuntu.com/ubuntu xenial-updates/main i386 Packages [660 kB]
Get:10 http://archive.ubuntu.com/ubuntu xenial-updates/main Translation-en [295 kB]
Get:11 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [580 kB]
Get:12 http://archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packages [537 kB]
Fetched 3,552 kB in 1s (1,972 kB/s)
Reading package lists... Done
```

Enter your password if prompted. Then run:

sudo apt-get upgrade

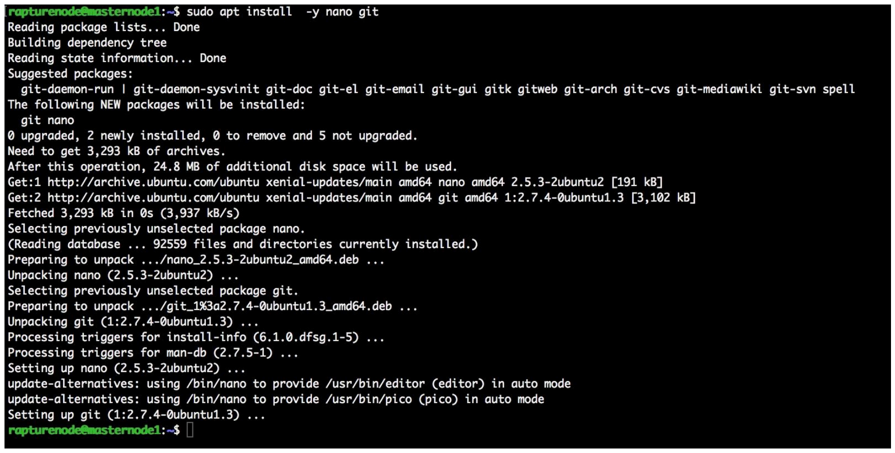


```
  en_NZ.UTF-8... done
  en_PH.UTF-8... done
  en_SG.UTF-8... done
  en_US.ISO-8859-1... done
  en_US.UTF-8... done
  en_ZA.UTF-8... done
  en_ZM.UTF-8... done
  en_ZW.UTF-8... done
Generation complete.
Setting up update-notifier-common (3.168.7) ...
Setting up distro-info-data (0.28ubuntu0.7) ...
Setting up iproute2 (4.3.0-1ubuntu3.16.04.3) ...
Setting up libisc-export160 (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up libdns-export162 (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up libtasn1-6:amd64 (4.7-3ubuntu0.16.04.3) ...
Setting up libisc160:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up libdns162:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up libisccc140:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up libisccfg140:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up libbind9-140:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up liblwres141:amd64 (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up bind9-host (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up dnsutils (1:9.10.3.dfsg.P4-8ubuntu1.10) ...
Setting up openssh-client (1:7.2p2-4ubuntu2.4) ...
Setting up openssh-sftp-server (1:7.2p2-4ubuntu2.4) ...
Setting up openssh-server (1:7.2p2-4ubuntu2.4) ...
Setting up rsync (3.1.1-3ubuntu1.2) ...
Setting up linux-firmware (1.157.15) ...
update-initramfs: Generating /boot/initrd.img-4.4.0-109-generic
```

This will list a group of packages in Ubuntu that can be upgraded. Type 'y' to start upgrading the packages. Sit back and watch as packages and applications are updated. It may not look exactly like this, but you'll see some similar messages streaming by. Once that's finished, you'll be back at the command prompt.

Now we'll install **nano** (a convenient text editor), **git** (a tool used for accessing github repositories) and **fail2ban** (a tool to help keep the vps secure from false login attempts). Type in:

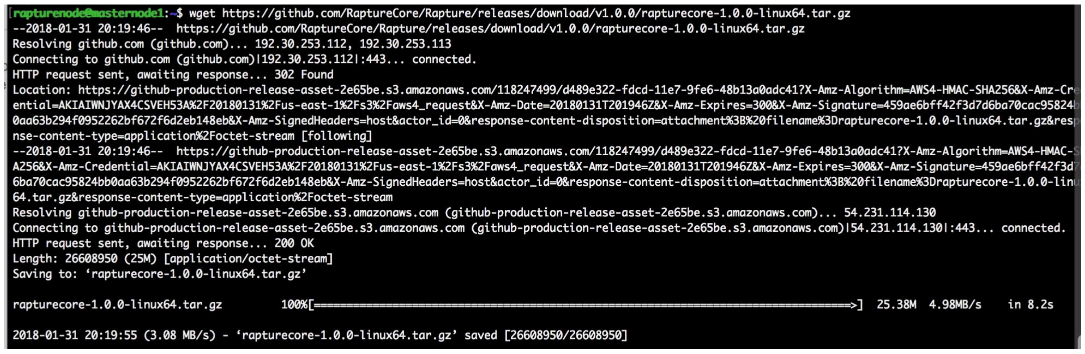
sudo apt install -y nano git fail2ban



```
rapturenode@masternode1:~$ sudo apt install  -y nano git
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch git-cvs git-mediawiki git-svn spell
The following NEW packages will be installed:
  git nano
0 upgraded, 2 newly installed, 0 to remove and 5 not upgraded.
Need to get 3,293 kB of archives.
After this operation, 24.8 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 nano amd64 2.5.3-2ubuntu2 [191 kB]
Get:2 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 git amd64 1:2.7.4-0ubuntu1.3 [3,102 kB]
Fetched 3,293 kB in 0s (3,937 kB/s)
Selecting previously unselected package nano.
(Reading database ... 92559 files and directories currently installed.)
Preparing to unpack .../nano_2.5.3-2ubuntu2_amd64.deb ...
Unpacking nano (2.5.3-2ubuntu2) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.7.4-0ubuntu1.3_amd64.deb ...
Unpacking git (1:2.7.4-0ubuntu1.3) ...
Processing triggers for install-info (6.1.0.dfsg.1-5) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up nano (2.5.3-2ubuntu2) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
Setting up git (1:2.7.4-0ubuntu1.3) ...
rapturenode@masternode1:~$
```

Now it's time to download the Rapture binaries. Type:

wget https://github.com/RaptureCore/Rapture/releases/download/v1.1.2.2/rapturecore-1.1.2-linux64.tar.gz



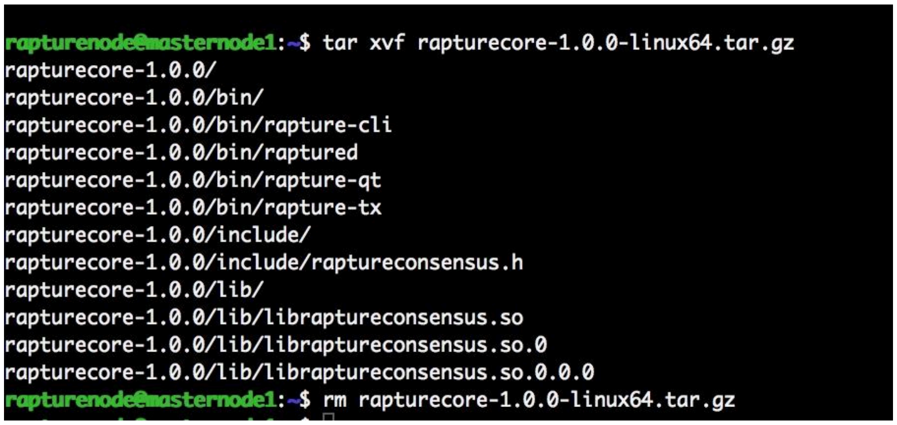
```
[rapturenode@masternode1:~$ wget https://github.com/RaptureCore/Rapture/releases/download/v1.0.0/rapturecore-1.0.0-linux64.tar.gz
--2018-01-31 20:19:46--  https://github.com/RaptureCore/Rapture/releases/download/v1.0.0/rapturecore-1.0.0-linux64.tar.gz
Resolving github.com (github.com)... 192.30.253.112, 192.30.253.113
Connecting to github.com (github.com)|192.30.253.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/118247499/d489e322-fdcd-11e7-9fe6-48b13a0adc41?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Cr
ential=AKIAIWNJYAX4CSVEH53A%2F20180131%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20180131T201946Z&X-Amz-Expires=300&X-Amz-Signature=459ae6bff42f3d7d6ba70cac95824
0aa63b294f0952262bf672f6d2eb148eb&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Drapturecore-1.0.0-linux64.tar.gz&res
nse-content-type=application%2Foctet-stream [following]
--2018-01-31 20:19:46--  https://github-production-release-asset-2e65be.s3.amazonaws.com/118247499/d489e322-fdcd-11e7-9fe6-48b13a0adc41?X-Amz-Algorithm=AWS4-HMAC-
A256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20180131%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20180131T201946Z&X-Amz-Expires=300&X-Amz-Signature=459ae6bff42f3d
6ba70cac95824bb0aa63b294f0952262bf672f6d2eb148eb&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Drapturecore-1.0.0-lin
64.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 54.231.114.130
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|54.231.114.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26608950 (25M) [application/octet-stream]
Saving to: ‘rapturecore-1.0.0-linux64.tar.gz’

rapturecore-1.0.0-linux64.tar.gz        100%[=========================================================================>]  25.38M  4.98MB/s    in 8.2s

2018-01-31 20:19:55 (3.08 MB/s) - ‘rapturecore-1.0.0-linux64.tar.gz’ saved [26608950/26608950]
```

The wget command will download the binaries straight from github. The binaries are in an archive, so we need to extract them using the command:

tar -xvf rapturecore-1.1.2-linux64.tar.gz

They'll extract quickly, and we can delete the archive by running:

rm rapturecore-1.1.2-linux64.tar.gz



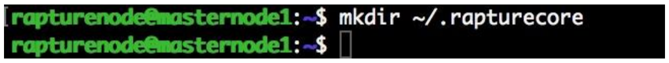
```
rapturenode@masternode1:~$ tar xvf rapturecore-1.0.0-linux64.tar.gz
rapturecore-1.0.0/
rapturecore-1.0.0/bin/
rapturecore-1.0.0/bin/rapture-cli
rapturecore-1.0.0/bin/raptured
rapturecore-1.0.0/bin/rapture-qt
rapturecore-1.0.0/bin/rapture-tx
rapturecore-1.0.0/include/
rapturecore-1.0.0/include/raptureconsensus.h
rapturecore-1.0.0/lib/
rapturecore-1.0.0/lib/libraptureconsensus.so
rapturecore-1.0.0/lib/libraptureconsensus.so.0
rapturecore-1.0.0/lib/libraptureconsensus.so.0.0.0
rapturenode@masternode1:~$ rm rapturecore-1.0.0-linux64.tar.gz
```

Step 7 - Editing rapture.conf and running Rapture

Now we'll create a configuration file for Rapture that has a few important details. We'll use a tool in Linux called 'nano' to edit the file.

First, we'll create the data directory for Rapture. Type in:

mkdir ~/.rapturecore



```
[rapturenode@masternode1:~$ mkdir ~/.rapturecore
rapturenode@masternode1:~$ 
```

This creates a directory for Rapture to store its configuration files, logs, and blockchain data. Next, we'll create the rapture.conf file that Rapture pulls its settings from. To do this, type:

nano ~/.rapturecore/rapture.conf

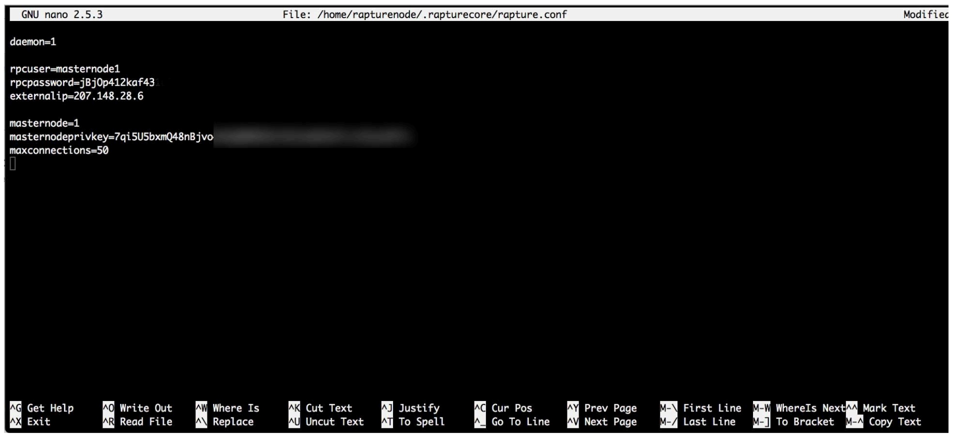
This will open an empty file in the text editor nano. Enter in the following lines (**NOTE:** Do not leave any trailing spaces at the end of this file, this has caused issues for some users):

```
daemon=1

rpcuser=masternode1
rpcpassword=jBjOp412kaf431pS
externalip=207.148.28.6

masternode=1
masternodeprivkey=7qi5U5bxmQ48nBjvo4mPgBMNX9kUrKBJewb6hAFCvc53qjsMfYe
maxconnections=50
```

rpcpassword can be any combination of letters and numbers. You don't need to remember this password and won't need to enter it anywhere. **Just be sure to NOT use any special characters or symbols in here**…..**letters and numbers only!**

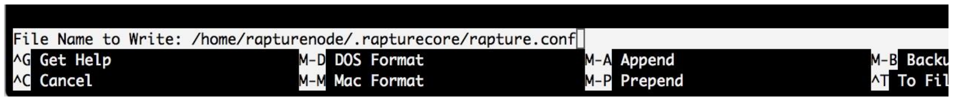
externalip should match the IP address of your vps

masternodeprivkey should match the private key that we generated earlier in the debug console.



```
  GNU nano 2.5.3                    File: /home/rapturenode/.rapturecore/rapture.conf                    Modified

daemon=1

rpcuser=masternode1
rpcpassword=jBjOp412kaf43
externalip=207.148.28.6

masternode=1
masternodeprivkey=7qi5U5bxmQ48nBjvo[redacted]
maxconnections=50
```

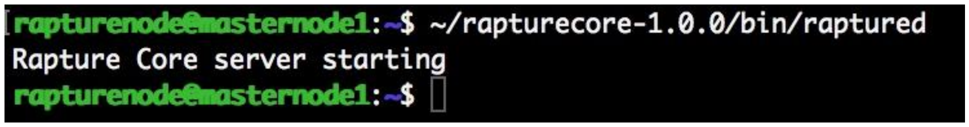
Press **Ctrl-O** to "Write out" the file in nano and press Enter to accept the default path:



```
File Name to Write: /home/rapturenode/.rapturecore/rapture.conf
```

Then press **Ctrl-X** to exit nano. You should now be back at a command line and ready to start Rapture on your vps. Type in:

~/rapturecore-1.1.2/bin/raptured

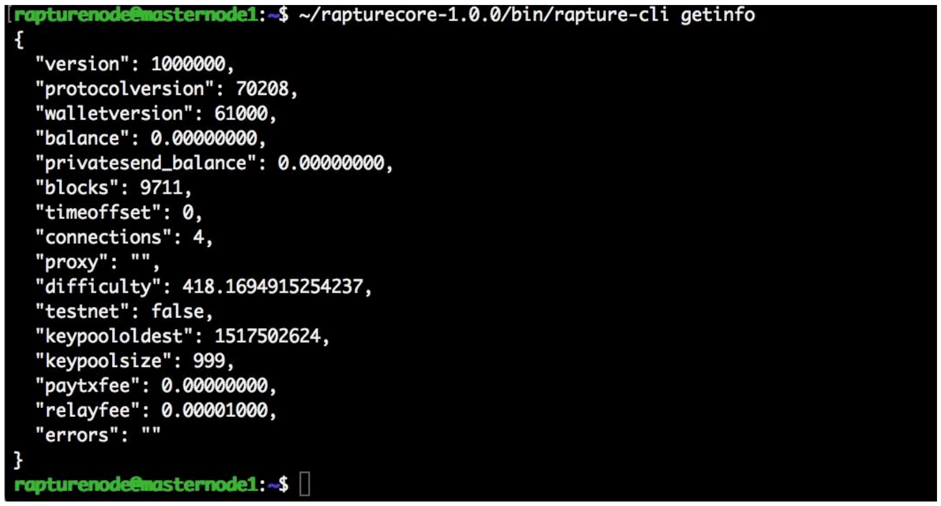


```
rapturenode@masternode1:~$ ~/rapturecore-1.0.0/bin/raptured
Rapture Core server starting
rapturenode@masternode1:~$
```

This starts Rapture and it should now be running in the background. If you ever need to restart your VPS or restart Rapture, this same command can be used to start the Rapture daemon.

To send commands to Rapture, we use the binary **rapture-cli**. The same commands that are available in the debug console of your local wallet are also available with rapture-cli. To check that Rapture is properly connected to the network, type in:

~/rapturecore-1.1.2/bin/rapture-cli getinfo

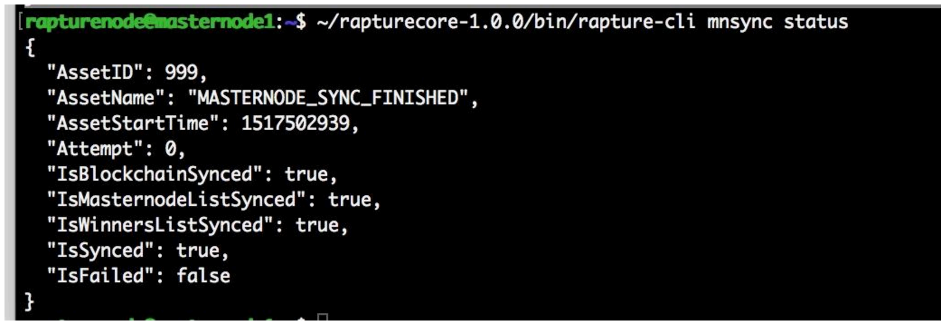


```
rapturenode@masternode1:~$ ~/rapturecore-1.0.0/bin/rapture-cli getinfo
{
  "version": 1000000,
  "protocolversion": 70208,
  "walletversion": 61000,
  "balance": 0.00000000,
  "privatesend_balance": 0.00000000,
  "blocks": 9711,
  "timeoffset": 0,
  "connections": 4,
  "proxy": "",
  "difficulty": 418.1694915254237,
  "testnet": false,
  "keypoololdest": 1517502624,
  "keypoolsize": 999,
  "paytxfee": 0.00000000,
  "relayfee": 0.00001000,
  "errors": ""
}
rapturenode@masternode1:~$
```

This issues rapture-cli the ‘getinfo’ command, which lists some basic information. Mainly, we’re looking to see that “connections” has at least 1 connection (hopefully more).

Another helpful command is:

~/rapturecore-1.1.2/bin/rapture-cli mnsync status



```
rapturenode@masternode1:~$ ~/rapturecore-1.0.0/bin/rapture-cli mnsync status
{
  "AssetID": 999,
  "AssetName": "MASTERNODE_SYNC_FINISHED",
  "AssetStartTime": 1517502939,
  "Attempt": 0,
  "IsBlockchainSynced": true,
  "IsMasternodeListSynced": true,
  "IsWinnersListSynced": true,
  "IsSynced": true,
  "IsFailed": false
}
```

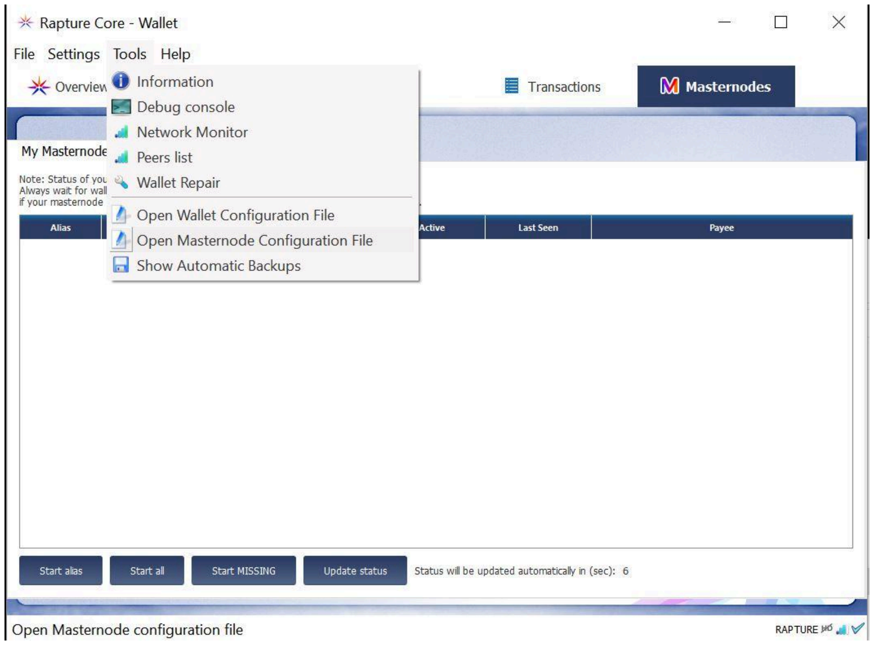
This checks the current status of the wallet sync. When you launch your local wallet, you’ll notice that it takes a minute or so to catch up to the latest block and also sync important information about the network, such as the current list of masternodes on the network. Your vps wallet does the exact same thing, but because it’s run through the command line, there’s no status meter to watch. This mnsync status command can be run multiple times to check how far along the sync is. You’re looking for the line “isSynced” to be true, which means that your wallet is fully synced with the blockchain and network.

Step 8 - Creating masternode entry in your local wallet

With Rapture up and running on your VPS, we'll jump back over to your local wallet to get it communicating with the VPS. Keep your VPS PuTTY session up and running, we'll be back to take care of one more important task there shortly.

On your local wallet, we need to give it the configuration information for your masternode. This is done by editing your masternode.conf file. This file is generally stored in %APPDATA%/RaptureCore/masternode.conf, but an easier way to edit it is to go to:

Tools->Open Masternode Configuration file



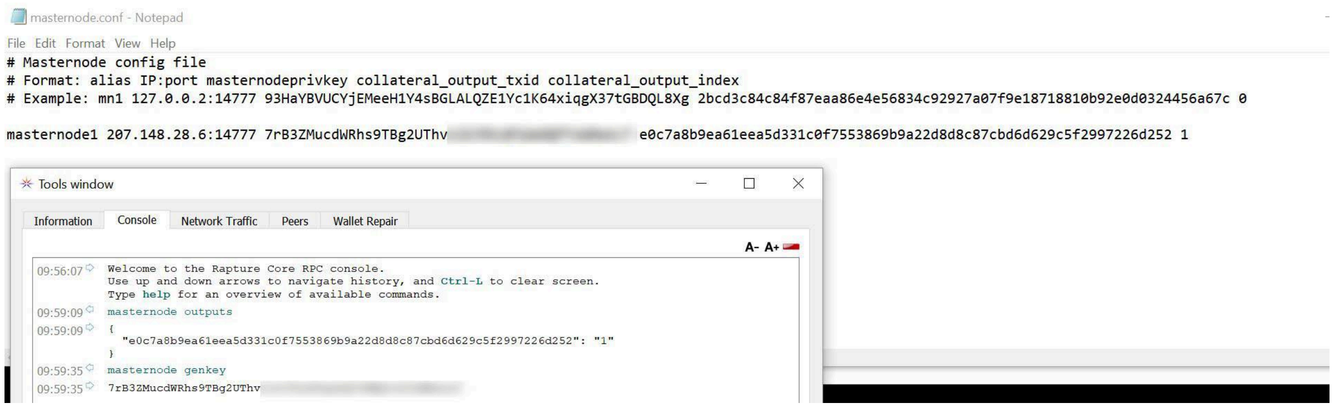
This should open up your masternode.conf file in Notepad.

This file has an example line in that can be used as a guide. The # in front of the line indicates that it's "commented out", which means it's effectively ignored. Following this example, we see that the structure is:

alias - a name for your masternode
ip address - the IP address for your VPS that is hosting the node
masternode private key - the key that we generated previously using 'masternode genkey'
txid - the transaction id for your 1,000 Rapture collateral transaction
index - the block position of the transaction…this is the number that appears at the end of of the txid listed by 'masternode outputs'

Following this structure, for our example we're going to add a line that looks like this:

masternode1 207.148.28.6:14777 7rDZwYbEfHoFpGHKPJ e0c7a8b9ea61eea5d331c0f7553869b9a22d8d8c87cbd6d629c5f2997226d252 1

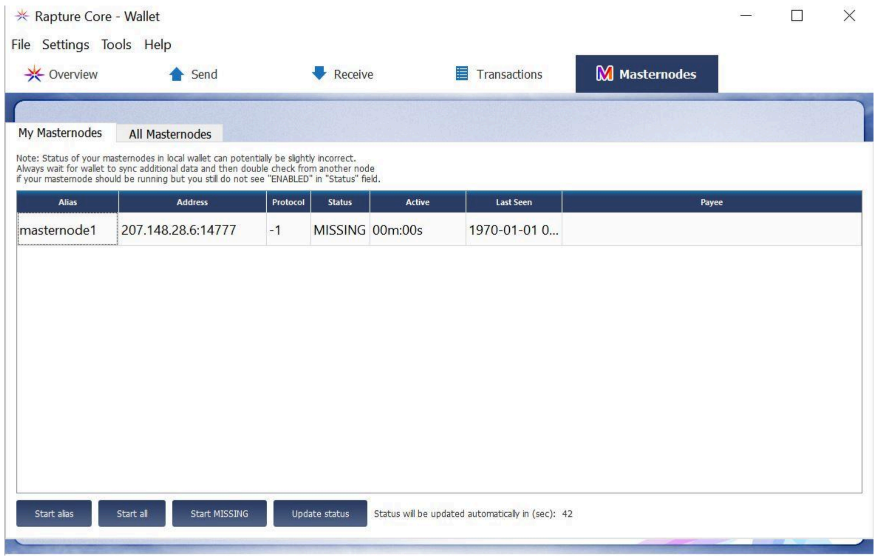


```
# Masternode config file
# Format: alias IP:port masternodeprivkey collateral_output_txid collateral_output_index
# Example: mn1 127.0.0.2:14777 93HaYBVUCYjEMeeH1Y4sBGLALQZE1Yc1K64xiqgX37tGBDQL8Xg 2bcd3c84c84f87eaa86e4e56834c92927a07f9e18718810b92e0d0324456a67c 0

masternode1 207.148.28.6:14777 7rB3ZMucdWRhs9TBg2UThv[redacted] e0c7a8b9ea61eea5d331c0f7553869b9a22d8d8c87cbd6d629c5f2997226d252 1
```

```
09:56:07  Welcome to the Rapture Core RPC console.
          Use up and down arrows to navigate history, and Ctrl-L to clear screen.
          Type help for an overview of available commands.
09:59:09  masternode outputs
09:59:09  {
            "e0c7a8b9ea61eea5d331c0f7553869b9a22d8d8c87cbd6d629c5f2997226d252": "1"
          }
09:59:35  masternode genkey
09:59:35  7rB3ZMucdWRhs9TBg2UThv[redacted]
```

Double check your entries to make sure they all match up, and then Save and close notepad. Every time you edit or make changes to the masternode.conf file (or any conf file), you need to close and re-open the wallet to initialize the changes. So close your local wallet, and then re-open.

Once you've re-opened, over on the Masternodes tab you should see your node listed.

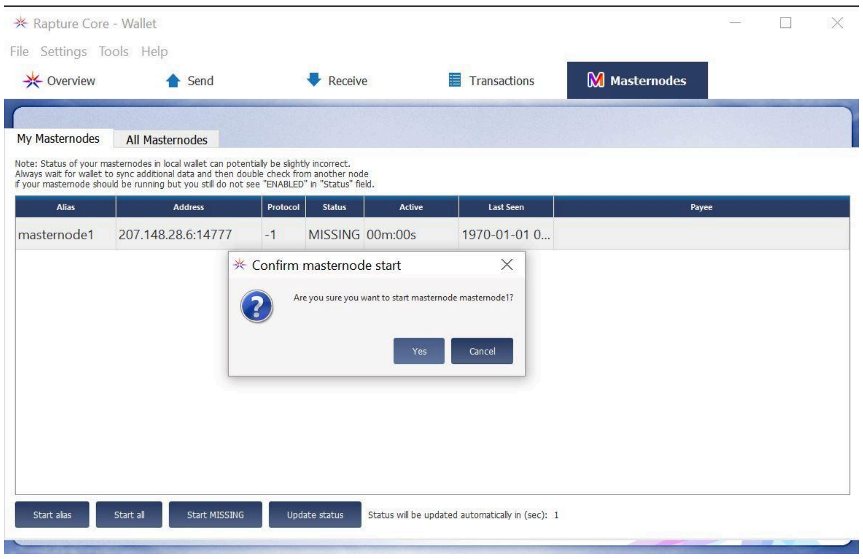


Note: Status of your masternodes in local wallet can potentially be slightly incorrect.
Always wait for wallet to sync additional data and then double check from another node
if your masternode should be running but you still do not see "ENABLED" in "Status" field.

Alias	Address	Protocol	Status	Active	Last Seen	Payee
masternode1	207.148.28.6:14777	-1	MISSING	00m:00s	1970-01-01 0...	

Status will be updated automatically in (sec): 42

Click the Start alias button and then **click Yes** to confirm. (**NOTE**, before clicking start, make sure your transaction of 1,000 Rapture has **AT LEAST 15 confirmations**, you can check the number of confirmations by hovering the mouse cursor over the transaction in the **Transactions** tab)


Rapture Core - Wallet
File Settings Tools Help
Overview Send Receive Transactions Masternodes
My Masternodes All Masternodes

Note: Status of your masternodes in local wallet can potentially be slightly incorrect.
Always wait for wallet to sync additional data and then double check from another node
if your masternode should be running but you still do not see "ENABLED" in "Status" field.

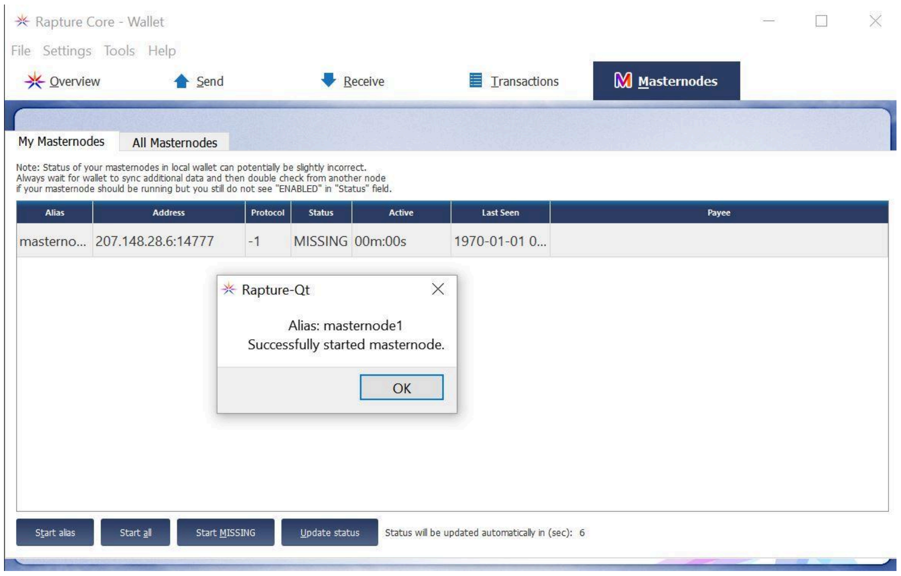
Alias	Address	Protocol	Status	Active	Last Seen	Payee
masternode1	207.148.28.6:14777	-1	MISSING	00m:00s	1970-01-01 0...	

Confirm masternode start

Are you sure you want to start masternode masternode1?

Status will be updated automatically in (sec): 1

This remotely connects to your VPS and uses your private key and txid to activate the masternode. If everything has been configured properly, you should see the following message:


Rapture Core - Wallet
File Settings Tools Help
Overview Send Receive Transactions Masternodes
My Masternodes All Masternodes

Note: Status of your masternodes in local wallet can potentially be slightly incorrect.
Always wait for wallet to sync additional data and then double check from another node
if your masternode should be running but you still do not see "ENABLED" in "Status" field.

Alias	Address	Protocol	Status	Active	Last Seen	Payee
masterno...	207.148.28.6:14777	-1	MISSING	00m:00s	1970-01-01 0...	

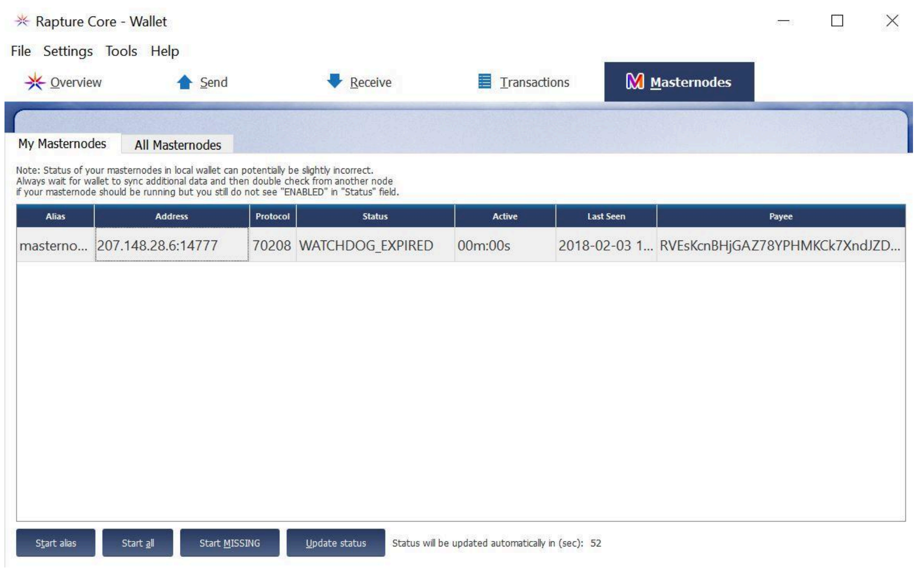
Rapture-Qt

Alias: masternode1
Successfully started masternode.

Status will be updated automatically in (sec): 6

Step 9 - Setting up sentinel

At this point, your masternode has been enabled and will start broadcasting itself to the network, but will likely be showing WATCHDOG_EXPIRED status.



Note: Status of your masternodes in local wallet can potentially be slightly incorrect.
Always wait for wallet to sync additional data and then double check from another node
if your masternode should be running but you still do not see "ENABLED" in "Status" field.

Alias	Address	Protocol	Status	Active	Last Seen	Payee
masterno...	207.148.28.6:14777	70208	WATCHDOG_EXPIRED	00m:00s	2018-02-03 1...	RVEsKcnBHjGAZ78YPHMKCk7XndJZD...

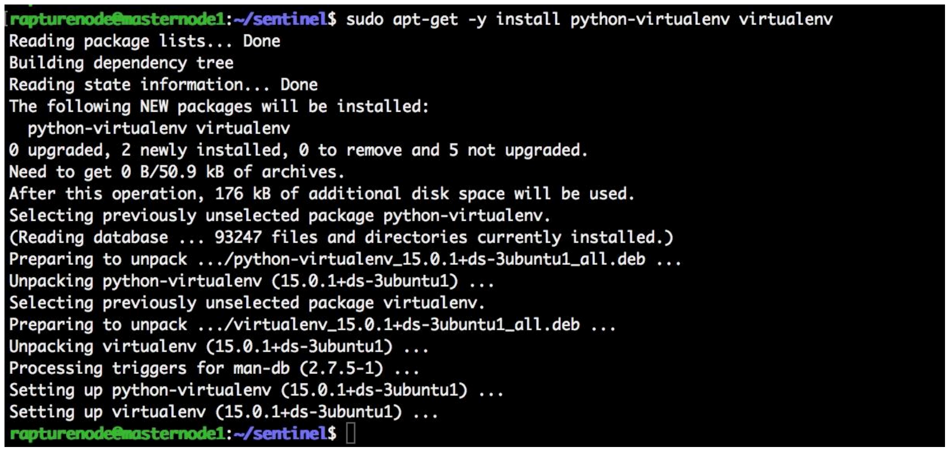
Status will be updated automatically in (sec): 52

There's one more important step before your node is fully enabled and eligible for masternode rewards and voting. We need to configure sentinel, which is a service that interfaces with the masternode. If this step is skipped, your masternode will continue to display WATCHDOG_EXPIRED status and will not receive rewards....so onwards with this important step!

Back over to your VPS now (if your PuTTY session disconnected, just reconnect and login with your 'rapturenode' user as before).

First, we want to install a package called virtualenv. From the command line, type:

sudo apt-get -y install python-virtualenv virtualenv

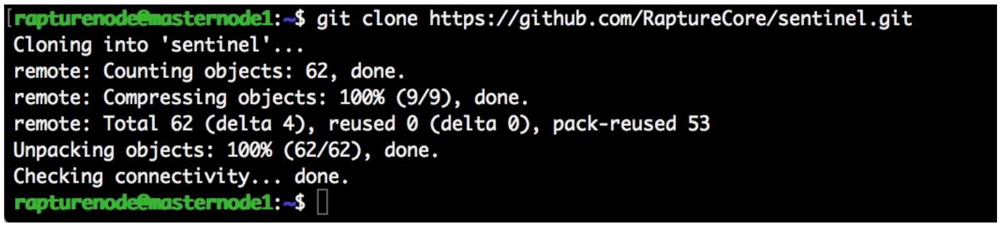


```
rapturenode@masternode1:~/sentinel$ sudo apt-get -y install python-virtualenv virtualenv
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  python-virtualenv virtualenv
0 upgraded, 2 newly installed, 0 to remove and 5 not upgraded.
Need to get 0 B/50.9 kB of archives.
After this operation, 176 kB of additional disk space will be used.
Selecting previously unselected package python-virtualenv.
(Reading database ... 93247 files and directories currently installed.)
Preparing to unpack .../python-virtualenv_15.0.1+ds-3ubuntu1_all.deb ...
Unpacking python-virtualenv (15.0.1+ds-3ubuntu1) ...
Selecting previously unselected package virtualenv.
Preparing to unpack .../virtualenv_15.0.1+ds-3ubuntu1_all.deb ...
Unpacking virtualenv (15.0.1+ds-3ubuntu1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up python-virtualenv (15.0.1+ds-3ubuntu1) ...
Setting up virtualenv (15.0.1+ds-3ubuntu1) ...
rapturenode@masternode1:~/sentinel$
```

This should output some information and ultimately finish back at the command line.

Next we'll clone sentinel from the RaptureCore github repository using the following command:

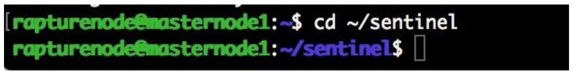
git clone https://github.com/RaptureCore/sentinel.git



```
rapturenode@masternode1:~$ git clone https://github.com/RaptureCore/sentinel.git
Cloning into 'sentinel'...
remote: Counting objects: 62, done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 62 (delta 4), reused 0 (delta 0), pack-reused 53
Unpacking objects: 100% (62/62), done.
Checking connectivity... done.
rapturenode@masternode1:~$ 
```

This command places sentinel under ~/sentinel, so we’ll change to that directory:

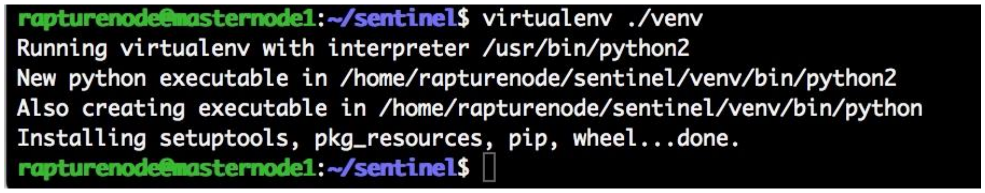
cd ~/sentinel



```
rapturenode@masternode1:~$ cd ~/sentinel
rapturenode@masternode1:~/sentinel$ 
```

Next, we’ll run a command to setup a python virtual environment in the sentinel directory. Type in:

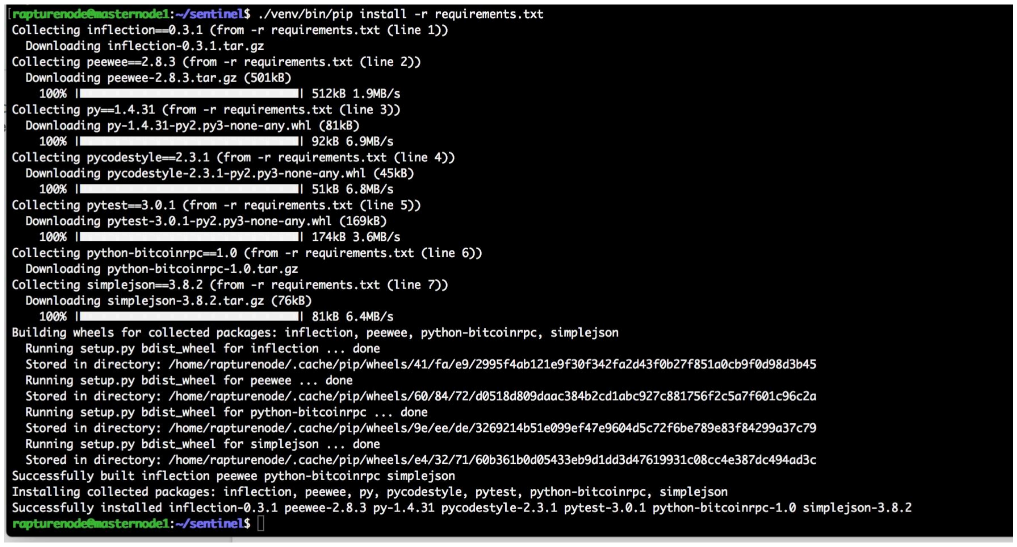
virtualenv ./venv



```
rapturenode@masternode1:~/sentinel$ virtualenv ./venv
Running virtualenv with interpreter /usr/bin/python2
New python executable in /home/rapturenode/sentinel/venv/bin/python2
Also creating executable in /home/rapturenode/sentinel/venv/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
rapturenode@masternode1:~/sentinel$ 
```

This command creates a directory in your current sentinel directory that contains the python executables and libraries. Next, we’ll install the requirements and dependencies for sentinel. Run:

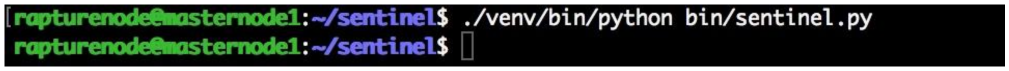
./venv/bin/pip install -r requirements.txt



```
rapturenode@masternode1:~/sentinel$ ./venv/bin/pip install -r requirements.txt
Collecting inflection==0.3.1 (from -r requirements.txt (line 1))
  Downloading inflection-0.3.1.tar.gz
Collecting peewee==2.8.3 (from -r requirements.txt (line 2))
  Downloading peewee-2.8.3.tar.gz (501kB)
    100% |████████████████████████████████| 512kB 1.9MB/s
Collecting py==1.4.31 (from -r requirements.txt (line 3))
  Downloading py-1.4.31-py2.py3-none-any.whl (81kB)
    100% |████████████████████████████████| 92kB 6.9MB/s
Collecting pycodestyle==2.3.1 (from -r requirements.txt (line 4))
  Downloading pycodestyle-2.3.1-py2.py3-none-any.whl (45kB)
    100% |████████████████████████████████| 51kB 6.8MB/s
Collecting pytest==3.0.1 (from -r requirements.txt (line 5))
  Downloading pytest-3.0.1-py2.py3-none-any.whl (169kB)
    100% |████████████████████████████████| 174kB 3.6MB/s
Collecting python-bitcoinrpc==1.0 (from -r requirements.txt (line 6))
  Downloading python-bitcoinrpc-1.0.tar.gz
Collecting simplejson==3.8.2 (from -r requirements.txt (line 7))
  Downloading simplejson-3.8.2.tar.gz (76kB)
    100% |████████████████████████████████| 81kB 6.4MB/s
Building wheels for collected packages: inflection, peewee, python-bitcoinrpc, simplejson
  Running setup.py bdist_wheel for inflection ... done
  Stored in directory: /home/rapturenode/.cache/pip/wheels/41/fa/e9/2995f4ab121e9f30f342fa2d43f0b27f851a0cb9f0d98d3b45
  Running setup.py bdist_wheel for peewee ... done
  Stored in directory: /home/rapturenode/.cache/pip/wheels/60/84/72/d0518d809daac384b2cd1abc927c881756f2c5a7f601c96c2a
  Running setup.py bdist_wheel for python-bitcoinrpc ... done
  Stored in directory: /home/rapturenode/.cache/pip/wheels/9e/ee/de/3269214b51e099ef47e9604d5c72f6be789e83f84299a37c79
  Running setup.py bdist_wheel for simplejson ... done
  Stored in directory: /home/rapturenode/.cache/pip/wheels/e4/32/71/60b361b0d05433eb9d1dd3d47619931c08cc4e387dc494ad3c
Successfully built inflection peewee python-bitcoinrpc simplejson
Installing collected packages: inflection, peewee, py, pycodestyle, pytest, python-bitcoinrpc, simplejson
Successfully installed inflection-0.3.1 peewee-2.8.3 py-1.4.31 pycodestyle-2.3.1 pytest-3.0.1 python-bitcoinrpc-1.0 simplejson-3.8.2
rapturenode@masternode1:~/sentinel$ 
```

That will display information on the packages being installed and then drop you back to the command line. At this point, we can run sentinel for the first time and make sure that it runs properly. Type:

./venv/bin/python bin/sentinel.py



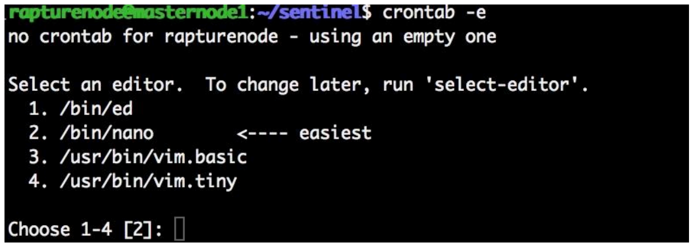
```
rapturenode@masternode1:~/sentinel$ ./venv/bin/python bin/sentinel.py
rapturenode@masternode1:~/sentinel$ 
```

When sentinel is running properly, it will process for a few seconds and then return to the command line without any output. If you receive any error messages, take a look at the Troubleshooting section of the guide.

If sentinel runs without any errors, then we can proceed with the final step, which is setting up what's known as a cron job that schedules the sentinel command to run every minute. To create the cron job, we'll run the following command:

crontab -e

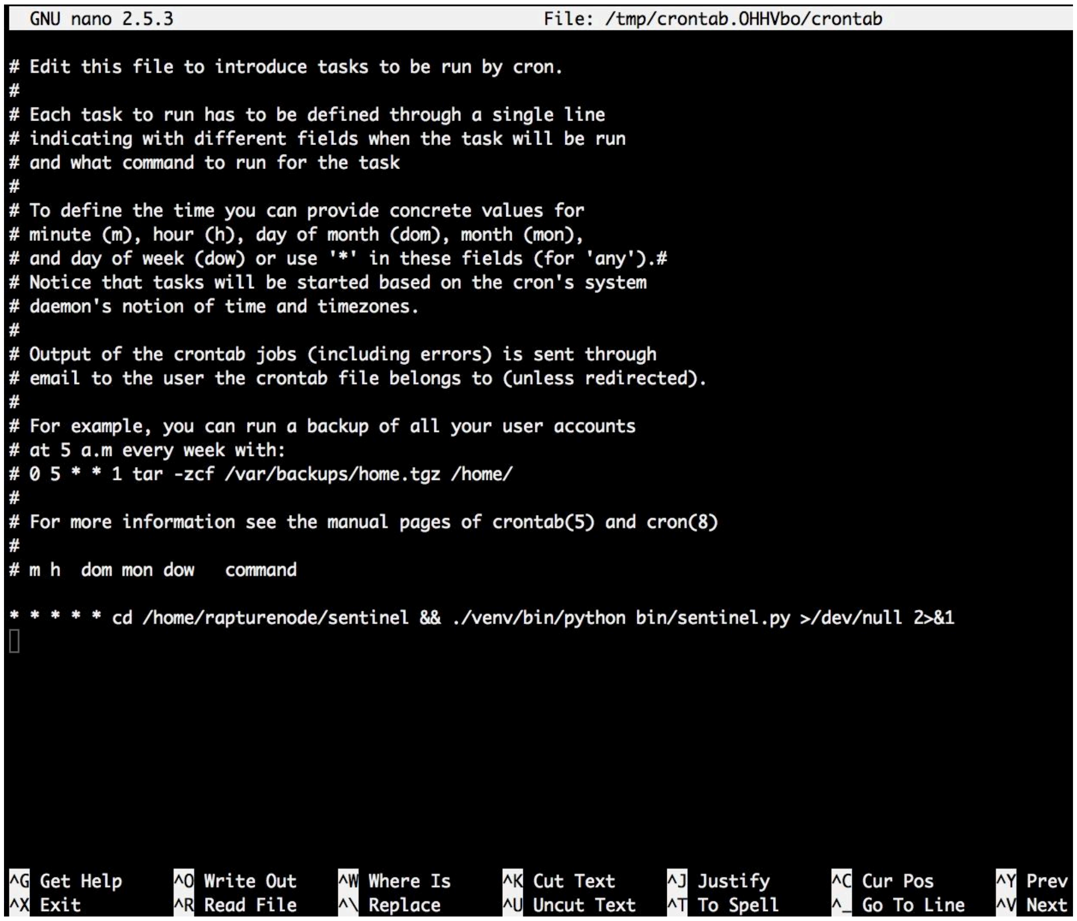
You'll be prompted to choose an editor, we'll use **nano**:



```
rapturenode@masternode1:~/sentinel$ crontab -e
no crontab for rapturenode - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/ed
  2. /bin/nano        <---- easiest
  3. /usr/bin/vim.basic
  4. /usr/bin/vim.tiny

Choose 1-4 [2]: 
```

Down bellow all of the # (commented) lines, we'll create a new line and insert (**NOTE:** If you selected a username OTHER than 'rapturenode', edit this path and replace 'rapturenode' with your username. If you've deviated from the guide and are logged in and installing as **root**, then the path should be /root/sentinel):

*** * * * * cd /home/rapturenode/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1**



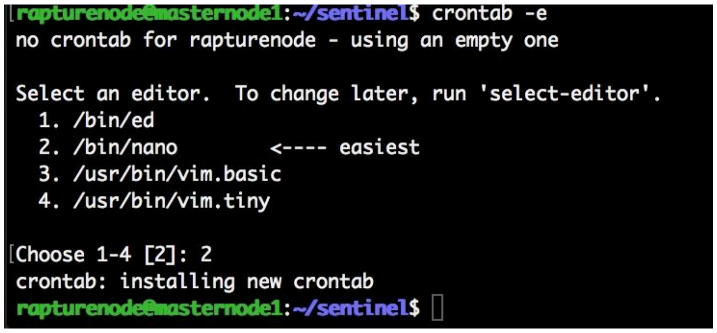
```
  GNU nano 2.5.3                                    File: /tmp/crontab.OHHVbo/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

* * * * * cd /home/rapturenode/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1







^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  ^V Next
```

This basically tells the sentinel command to run every minute. Type:

Ctrl-O and then **Enter**

to write out the file. Then:

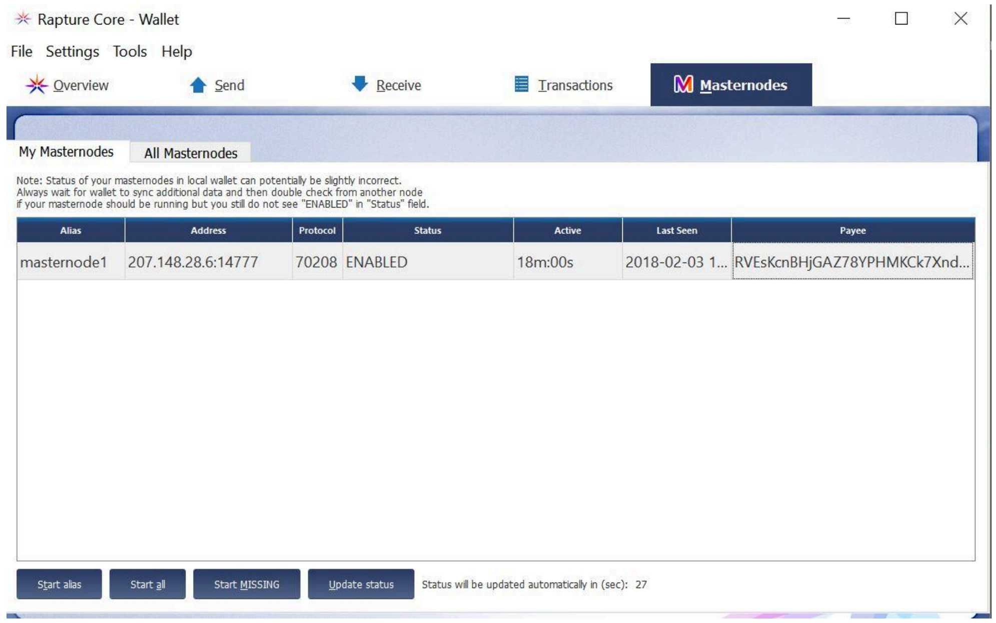
Ctrl-X to exit



```
rapturenode@masternode1:~/sentinel$ crontab -e
no crontab for rapturenode - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/ed
  2. /bin/nano        <---- easiest
  3. /usr/bin/vim.basic
  4. /usr/bin/vim.tiny

Choose 1-4 [2]: 2
crontab: installing new crontab
rapturenode@masternode1:~/sentinel$ 
```

Step 10 - Wait for ENABLED status

At this point we've performed all of the steps in setting up a properly functioning masternode. It will take about 15-20 minutes for the node to switch over to **ENABLED** status:



Note: Status of your masternodes in local wallet can potentially be slightly incorrect.
Always wait for wallet to sync additional data and then double check from another node
if your masternode should be running but you still do not see "ENABLED" in "Status" field.

Alias	Address	Protocol	Status	Active	Last Seen	Payee
masternode1	207.148.28.6:14777	70208	ENABLED	18m:00s	2018-02-03 1...	RVEsKcnBHjGAZ78YPHMKCk7Xnd...

Status will be updated automatically in (sec): 27

Once it has switched to ENABLED, you can close your local wallet if you prefer. It does not need to be open in order to receive rewards. If after about 20 minutes, you're seeing anything other than ENABLED, proceed to the Troubleshooting section below.

TROUBLESHOOTING

Sentinel debugging

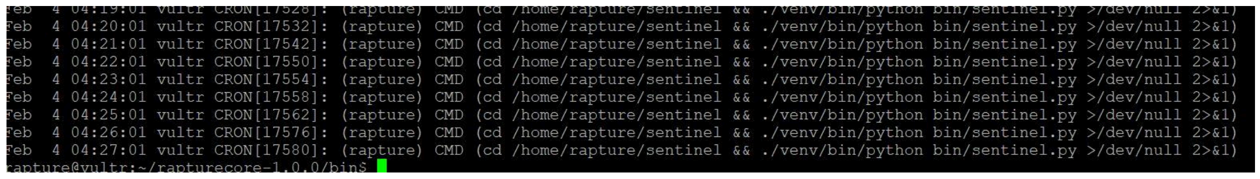
The number one issue that users run into tends to be a misconfiguration with sentinel. It could be that the cron task that keeps sentinel running in the background wasn't setup properly, or it could be that sentinel wasn't installed properly to being with.

Checking the cron log

Whenever a cron task runs, it writes to the system log. You can easily check this by typing:

sudo grep CRON /var/log/syslog

This command checks the system log for entries with CRON in the name. You should see an output like this:



```
Feb  4 04:19:01 vultr CRON[17528]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
Feb  4 04:20:01 vultr CRON[17532]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
Feb  4 04:21:01 vultr CRON[17542]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
Feb  4 04:22:01 vultr CRON[17550]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
Feb  4 04:23:01 vultr CRON[17554]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
Feb  4 04:24:01 vultr CRON[17558]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
Feb  4 04:25:01 vultr CRON[17562]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
Feb  4 04:26:01 vultr CRON[17576]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
Feb  4 04:27:01 vultr CRON[17580]: (rapture) CMD (cd /home/rapture/sentinel && ./venv/bin/python bin/sentinel.py >/dev/null 2>&1)
rapture@vultr:~/rapturecore-1.0.0/bin$
```

Here we can see that the cron task we setup in Step 9 is running every minute. If you're not seeing something similar, or you're seeing error messages, go back to here and repeat the crontab procedure.

Checking that sentinel runs properly

We can double-check that sentinel is running without issue by running the following commands:

cd ~/sentinel

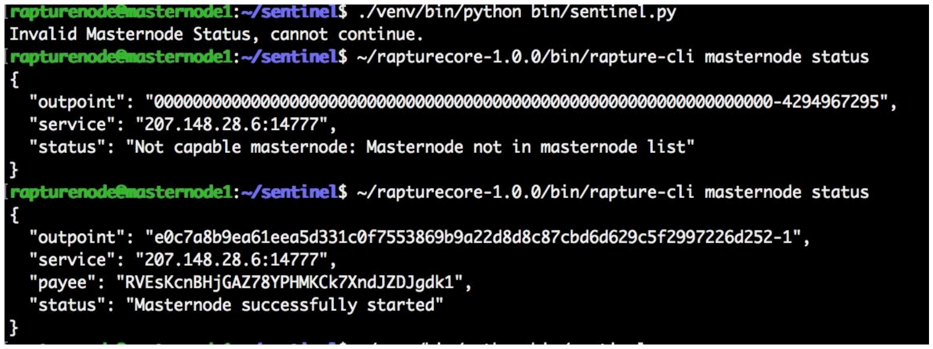
./venv/bin/python bin/sentinel.py

This should process for a second or two and then return to the command line without any messages or errors. If you receive the message: **Invalid Masternode Status, cannot continue**….then your masternode was not started successfully.

We can check the status of the masternode by running:

~/rapturecore-1.1.2/bin/rapture-cli masternode status

If the status says anything other than "**Masternode successfully started**", try restarting the masternode by clicking the **Start all** button on the Masternodes tab in your local wallet. Re-check the status by running the previous command:



```
rapturenode@masternode1:~/sentinel$ ./venv/bin/python bin/sentinel.py
Invalid Masternode Status, cannot continue.
rapturenode@masternode1:~/sentinel$ ~/rapturecore-1.0.0/bin/rapture-cli masternode status
{
  "outpoint": "0000000000000000000000000000000000000000000000000000000000000000-4294967295",
  "service": "207.148.28.6:14777",
  "status": "Not capable masternode: Masternode not in masternode list"
}
rapturenode@masternode1:~/sentinel$ ~/rapturecore-1.0.0/bin/rapture-cli masternode status
{
  "outpoint": "e0c7a8b9ea61eea5d331c0f7553869b9a22d8d8c87cbd6d629c5f2997226d252-1",
  "service": "207.148.28.6:14777",
  "payee": "RVEsKcnBHjGAZ78YPHMKCk7XndJZDJgdk1",
  "status": "Masternode successfully started"
}
```

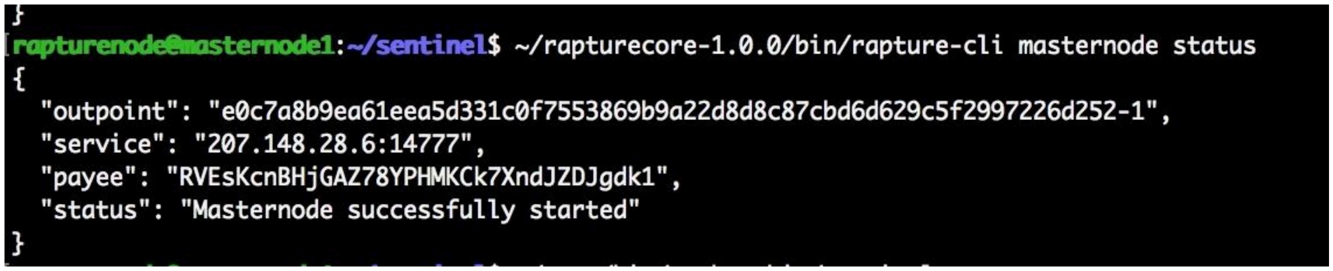
if all is configured correctly, then try running sentinel to verify that the problem is fixed:

./venv/bin/python bin/sentinel.py

Checking the masternode status

The status of the masternode can be checked on the VPS by running:

~/rapturecore-1.1.2/bin/rapture-cli masternode status



```
}
rapturenode@masternode1:~/sentinel$ ~/rapturecore-1.0.0/bin/rapture-cli masternode status
{
  "outpoint": "e0c7a8b9ea61eea5d331c0f7553869b9a22d8d8c87cbd6d629c5f2997226d252-1",
  "service": "207.148.28.6:14777",
  "payee": "RVEsKcnBHjGAZ78YPHMKCk7XndJZDJgdk1",
  "status": "Masternode successfully started"
}
```

If everything is configured correctly, this should return the status: **Masternode successfully started**

If you receive another status, such as "**Not capable masternode: Masternode not in masternode list**", check the following:

- Click **Start all** from the Masternodes tab in your local wallet
- Verify that your collateral transaction of 1,000 Rapture has at least 15 confirmations. The masternode will only start successfully once that transaction has 15 confirmations (you can check this by hovering the mouse cursor over the transaction in the Transactions tab of the local wallet)
- Check your conf files for any errors

Masternode failed to start - CONF file errors

If you're having issues starting the masternode via the Masternodes tab in your local wallet, best to check your local masternode.conf file and your VPS rapture.conf file to make sure that they match up.

You can open your local masternode.conf file by choosing:

Tools->Open Masternode Configuration File...

...in your local wallet. On your VPS, you can open the rapture.conf file by typing:

nano ~/.rapturecore/rapture.conf

Arrange both of these side-by-side and double check that the IP address and masternode private key match.

UPGRADING

Upgrading masternode from v1.0.0 to v1.1.0

On February 22, 2017, a mandatory update was committed to the github repository bringing Rapture up to v1.1.0. This update includes a re-balance between proof-of-work and masternode rewards after block height 29,000. As with any adjustment to block subsidies, the update is mandatory and users MUST upgrade both their LOCAL and MASTERNODE wallets. The local update is very straight forward, just download your preferred binary from:

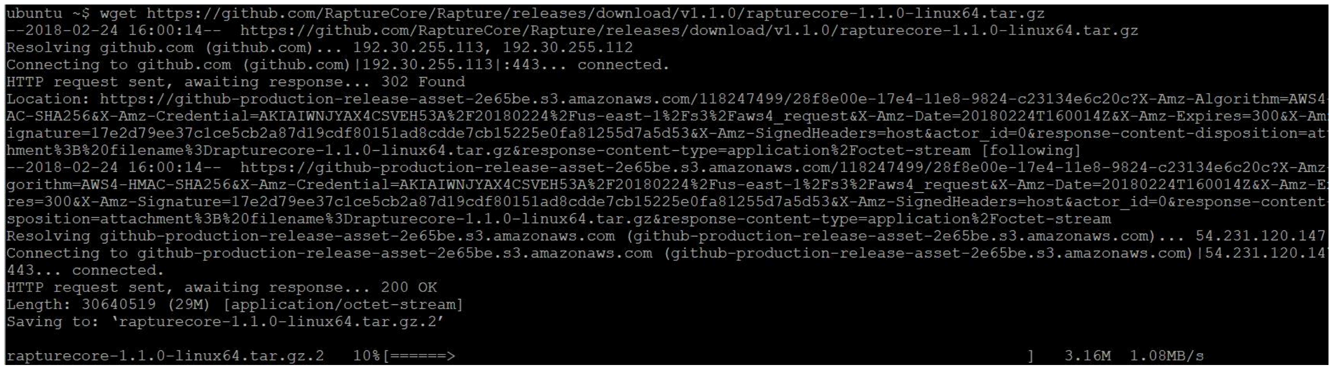
https://github.com/RaptureCore/Rapture/releases/tag/v1.1.0

If you're using the Windows installer, just install the new version and you should be all set. If you're using the zipped binaries, just delete your old rapturecore-1.0.0 directory and replace it with rapturecore-1.1.0.

To update your Linux VPS masternode, login to your VPS via putty (see the final commands in Step 5 if you forget how to do this) and run the following commands:

First, download the v1.1.0 binaries from github:

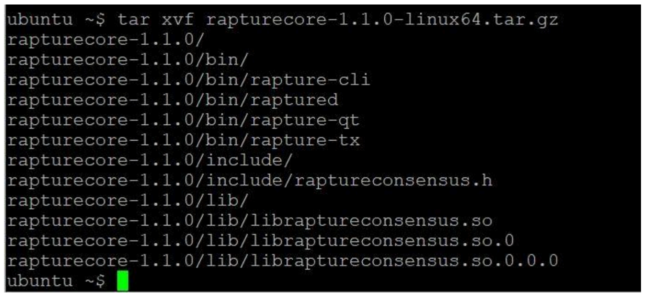
wget https://github.com/RaptureCore/Rapture/releases/download/v1.1.0/rapturecore-1.1.0-linux64.tar.gz



```
ubuntu ~$ wget https://github.com/RaptureCore/Rapture/releases/download/v1.1.0/rapturecore-1.1.0-linux64.tar.gz
--2018-02-24 16:00:14--  https://github.com/RaptureCore/Rapture/releases/download/v1.1.0/rapturecore-1.1.0-linux64.tar.gz
Resolving github.com (github.com)... 192.30.255.113, 192.30.255.112
Connecting to github.com (github.com)|192.30.255.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/118247499/28f8e00e-17e4-11e8-9824-c23134e6c20c?X-Amz-Algorithm=AWS4
AC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20180224%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20180224T160014Z&X-Amz-Expires=300&X-Am
ignature=17e2d79ee37c1ce5cb2a87d19cdf80151ad8cdde7cb15225e0fa81255d7a5d53&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=at
hment%3B%20filename%3Drapturecore-1.1.0-linux64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2018-02-24 16:00:14--  https://github-production-release-asset-2e65be.s3.amazonaws.com/118247499/28f8e00e-17e4-11e8-9824-c23134e6c20c?X-Amz
gorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20180224%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20180224T160014Z&X-Amz-E
res=300&X-Amz-Signature=17e2d79ee37c1ce5cb2a87d19cdf80151ad8cdde7cb15225e0fa81255d7a5d53&X-Amz-SignedHeaders=host&actor_id=0&response-content
sposition=attachment%3B%20filename%3Drapturecore-1.1.0-linux64.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 54.231.120.147
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|54.231.120.14
443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30640519 (29M) [application/octet-stream]
Saving to: 'rapturecore-1.1.0-linux64.tar.gz.2'

rapturecore-1.1.0-linux64.tar.gz.2   10%[======>                                                            ]   3.16M  1.08MB/s
```

Then extract those binaries:

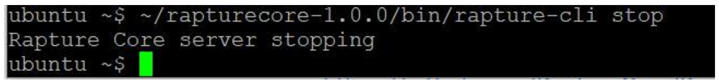
tar xvf rapturecore-1.1.0-linux64.tar.gz



```
ubuntu ~$ tar xvf rapturecore-1.1.0-linux64.tar.gz
rapturecore-1.1.0/
rapturecore-1.1.0/bin/
rapturecore-1.1.0/bin/rapture-cli
rapturecore-1.1.0/bin/raptured
rapturecore-1.1.0/bin/rapture-qt
rapturecore-1.1.0/bin/rapture-tx
rapturecore-1.1.0/include/
rapturecore-1.1.0/include/raptureconsensus.h
rapturecore-1.1.0/lib/
rapturecore-1.1.0/lib/libraptureconsensus.so
rapturecore-1.1.0/lib/libraptureconsensus.so.0
rapturecore-1.1.0/lib/libraptureconsensus.so.0.0.0
ubuntu ~$ 
```

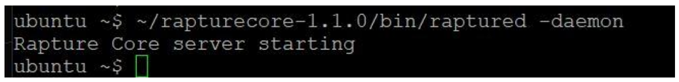
Next, we'll shutdown the previous version of the Rapture daemon:

~/rapturecore-1.0.0/bin/rapture-cli stop


```
ubuntu ~$ ~/rapturecore-1.0.0/bin/rapture-cli stop
Rapture Core server stopping
ubuntu ~$
```

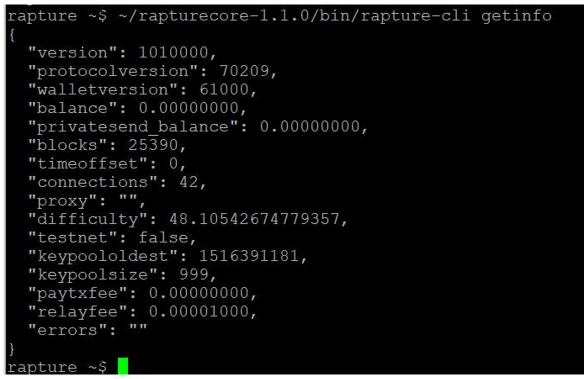
Wait a few seconds for the previous daemon to fully shutdown, and then start up v1.1.0:

~/rapturecore-1.1.0/bin/raptured -daemon


```
ubuntu ~$ ~/rapturecore-1.1.0/bin/raptured -daemon
Rapture Core server starting
ubuntu ~$
```

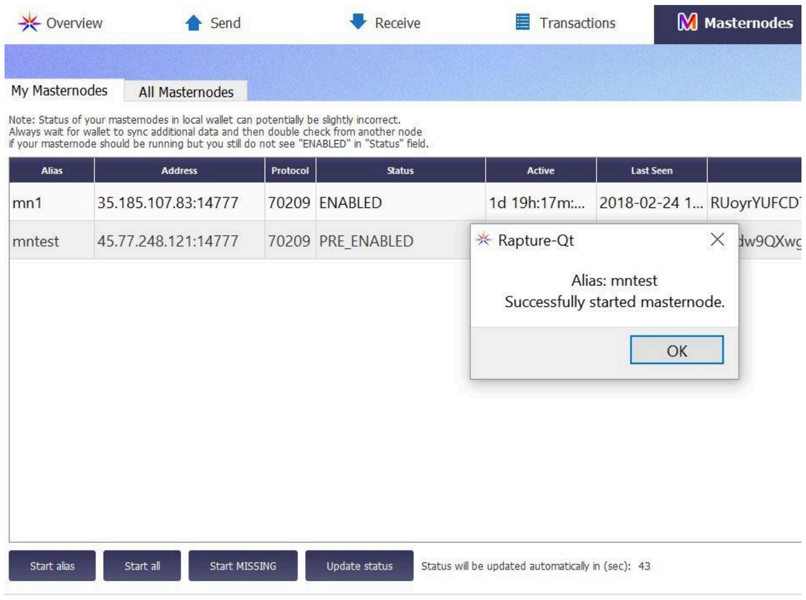
You should now be running v1.0.0, you can verify this by running:

~/rapturecore-1.1.0/bin/rapture-cli getinfo


```
rapture ~$ ~/rapturecore-1.1.0/bin/rapture-cli getinfo
{
  "version": 1010000,
  "protocolversion": 70209,
  "walletversion": 61000,
  "balance": 0.00000000,
  "privatesend_balance": 0.00000000,
  "blocks": 25390,
  "timeoffset": 0,
  "connections": 42,
  "proxy": "",
  "difficulty": 48.10542674779357,
  "testnet": false,
  "keypoololdest": 1516391181,
  "keypoolsize": 999,
  "paytxfee": 0.00000000,
  "relayfee": 0.00001000,
  "errors": ""
}
rapture ~$
```

You'll see the version listed as 1010000 and the protocol version as 70209.

At this point, head back over to your local wallet (which you should have already update to v1.1.0) and start your masternode using **Start alias**:



Note: Status of your masternodes in local wallet can potentially be slightly incorrect.
Always wait for wallet to sync additional data and then double check from another node
if your masternode should be running but you still do not see "ENABLED" in "Status" field.

Alias	Address	Protocol	Status	Active	Last Seen	
mn1	35.185.107.83:14777	70209	ENABLED	1d 19h:17m:...	2018-02-24 1...	RUoyrYUFCD
mntest	45.77.248.121:14777	70209	PRE_ENABLED			dw9QXwg

Rapture-Qt

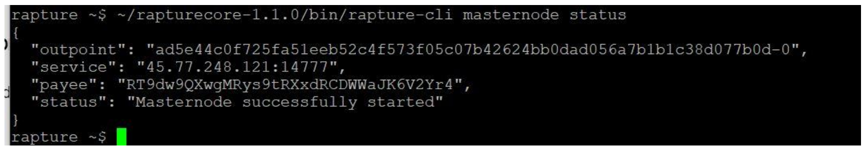
Alias: mntest
Successfully started masternode.

Status will be updated automatically in (sec): 43

Your node should show up as successfully started and will switch to **PRE_ENABLED**. PRE_ENABLED should last for ~10 minutes and then switch to **ENABLED**.

You can double-check the status of the masternode on the VPS using:

~/rapturecore-1.1.0/bin/rapture-cli masternode status



```
rapture ~$ ~/rapturecore-1.1.0/bin/rapture-cli masternode status
{
  "outpoint": "ad5e44c0f725fa51eeb52c4f573f05c07b42624bb0dad056a7b1b1c38d077b0d-0",
  "service": "45.77.248.121:14777",
  "payee": "RT9dw9QXwgMRys9tRXxdRCDWWaJK6V2Yr4",
  "status": "Masternode successfully started"
}
rapture ~$
```

The status should say “**Masternode successfully started**”. If the status reads as “**Not capable masternode: Invalid protocol version**", make sure that the local wallet that you’re activating from has been upgraded to v1.1.0. If so, then this message will clear momentarily and no additional actions are required.

When everything looks good, you can remove v1.0.0 binaries from your VPS by running (paste this exactly as-is):

rm ~/rapturecore-1.0.0/bin/*

(NOTE, this command previously removed the entire rapturecore-1.0.0 directory, but several people have their sentinel directory under there rather than in their home directory, so this version leaves any other data under rapturecore-1.0.0 in place).

At this point, you’re fully updated to v1.1.0 and ready for block 29,001!

APPENDIX

Understanding your Masternode

Information provided by Discord users @secsi#1493 and edited by @RaptureDev#7473

Masternode Status

ENABLED: Working properly.

WATCHDOG_EXPIRED: Sentinel is not reporting properly. Check crontab -e and ensure all install file paths are correct. Specifically the path to sentinel needs to match your sentinel location. Find the full path to sentinel by typing: **cd ~/sentinel;pwd**

WATCHDOG_EXPIRED Troubleshooting can be found here

NEW_START_REQUIRED: Several possible causes. Try restarting the alias and wait up to 30 minutes for ENABLED. If not begin troubleshooting.

PRE_ENABLED: New starts require time (approximately 10 minutes) to connect to peers and verify connection, be patient.

Real-time Masternode status can be checked online here:
http://explorer2.our-rapture.com/masternodes
https://masternodes.online/monitoring/

Local Wallet Terminology

Collateral Transaction: This is the amount of coins required to operate the Masternode. 1000 RAP are required for a Rapture Masternode.

[Sending your Collateral Transaction]: You must send exactly 1000 coins to generate a valid Transaction ID(txid) to be used for a Masternode. (When sending a transaction to yourself you will see it in your history "payment to self"as a small fee. 1000 coins will not show here.)

Transaction ID(txid): This value is generated when you successfully send exactly 1000 RAP to your new receiving address. This process ensures you only have 1000 RAP in the specified address and allows the wallet to lock those funds in place. These will now be used for your Masternode Collateral.
vout: This can be confusing to understand. This number is often 0, 1 or 2.
(The... blockchain uses a per-output transactional model, in which every transaction has a set of inputs and a set of outputs. Each input “spends” one output of a previous transaction, with the blockchain ensuring that this output cannot be spent elsewhere. The full history of transactions forms a multiway connected chain... All of the [assets] in a transaction’s inputs flow into that transaction, which are then distributed across its outputs in accordance with the quantities written within. As a result most regular payments require two outputs – one with

the intended amount for the recipient, and the other containing “change” which goes back to the sender for use in a subsequent transaction.)

Private Key: This is essentially a password that your VPS will use to access your local wallet and verify your Collateral Transaction. KEEP THIS SAFE. DO NOT SHARE IT. PEOPLE CAN ACCESS YOUR WALLET WITH THIS KEY.

VPS

A VPS is essentially a computer you rent. There are several providers to choose from. I personally use DigitalOcean but Vultr is widely used as well.
($10 free credit at Digitalocean with this referral link: https://m.do.co/c/22b7cd6629d3)

Renting a VPS is vital to running a Masternode to ensure security/stability/quality of life. When running a "hot/cold wallet" you are able to close your wallet and turn off your local machine while the Masternode stays active. Masternodes also require a static IP and broadcast their IP address to the network.For this reason it's best not host from your local machine.

Hot/Cold Wallet: This term refers to your VPS and Local Machine respectively. Your VPS(Hot) will run your Masternode and stay online while your Local wallet(Cold) is free to go offline.

Vultr
DigitalOcean
Time4VPS
Amazon EC2
Google Cloud Compute

Putty

Putty is a tool used to access your VPS via ssh. You can input your IP address and connect remotely through the Putty console. This program is relaying the information you provide to your VPS. It allows you to act as if you were sitting in front of a keyboard at your hosts site. The latest version of Putty can be downloaded here:
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Ubuntu Commands

adduser : creates a new user profile
usermod -aG sudo rapturenode : gives user profile (in this case, 'rapturenode') admin power
exit : closes current ssh session
sudo apt-get update : updates the list of packages and versions listed by the apt repositories
sudo apt-get upgrade : upgrades any packages that have a new version reported by apt-get update
sudo apt install -y nano git : installs nano, a text editor, and git, a tool for downloading github repositories
wget : downloads from the specified target
~/rapturecore-1.1.0/bin/rapture-cli mnsync status : Because the VPS wallet does not have a visible progress bar, we use this command to see the current state of synchronization.
crontab -e : edits the cronjob file. Cronjobs are automated request that you setup. In this instance we ask the cronjob to tell the sentinel command every 60 seconds.