RUNNING A NODE ON A RASPBERRY PI

This guide will walk you through the process of installing and configuring a 32-bit operating system, compiling the Zent Cash ARM binaries, and synchronizing your node on a Raspberry Pi. ARM builds were recently patched, so this is great news for anyone who has ever wanted to run a node on one.

Requirements

You'll need:

- A Raspberry PI with a 32 bit processor (Raspberry Pi 1 “ARMv6”, Raspberry Pi Zero “ARMv6”).
- A microSD card for the OS (keep in mind faster is better here).
- A SATA to USB adapter such at this one.
- A 2.5" SATA SSD of at least 120GB (I recommend 240gb to future proof it).
- A 32 bit OS image file (I'm using Raspberry Pi OS edition for this tutorial, which you can find here).
- Etcher installed in order to flash the micro SD card easily.

Setting Up the OS

Make sure you've downloaded and installed Etcher and have the 2022-09-22-raspios-buster-armhf.img file saved somewhere on your hard drive. Insert your microSD card into your computer, select the image file in Etcher, click flash and wait for completion. When it's finished flashing and validating, remove the microSD from your computer and insert it into the pi.

OS Setup

Connect your ethernet cable and plug in the pi. Either install a keyboard, mouse and monitor in order to do preliminary setup or SSH into the pi from another computer (open-ssh server is installed and enabled by default on Raspberry Pi OS.) The default user is **pi** and its password **raspberry**, it has password less root privileges escalation through **sudo**.

Once logged in, you might want to run **sudo raspi-config** in order to get assisted with your setup! Otherwise, set up the linux environment to your liking.

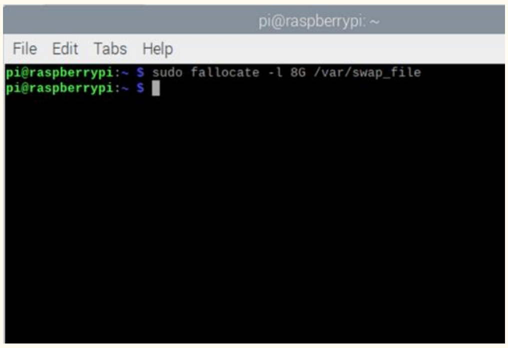
Creation and Mounting of Swap Archive

Since our Raspberry PI is low on RAM, we will need to create a swap archive in order to successfully compile our Zent Cash source code.

To create our swap file we will run the following command in Terminal.

sudo fallocate -l 8G /var/swap_file

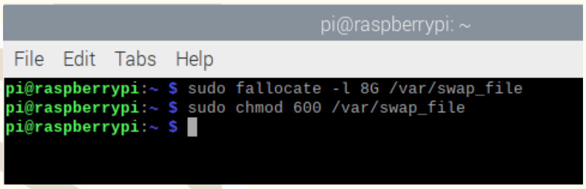
The fallocate command is used to preallocate space in a file; we specify the space with the -l option that indicates the length (it would be 8 GB). The last argument is the name of the file that we are going to host.



```
pi@raspberrypi:~ $ sudo fallocate -l 8G /var/swap_file
pi@raspberrypi:~ $ 
```

When the file has been created, it's time to protect it so that only root can modify it. In this case we use **chmod**.

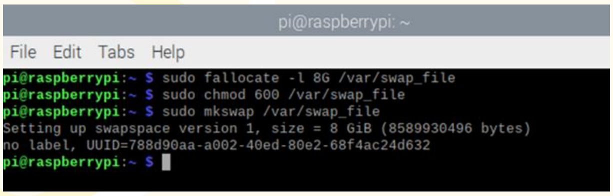
sudo chmod 600 /var/swap_file



```
pi@raspberrypi:~ $ sudo fallocate -l 8G /var/swap_file
pi@raspberrypi:~ $ sudo chmod 600 /var/swap_file
pi@raspberrypi:~ $ 
```

Now comes the final step and that is to convert our file into a swap file, using **mkswap.**

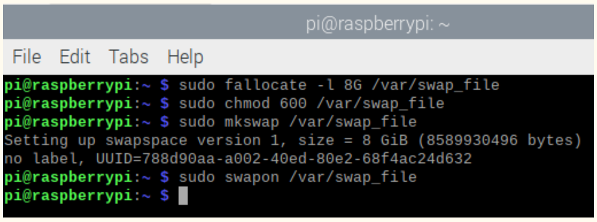
sudo mkswap /var/swap_file



```
pi@raspberrypi:~ $ sudo fallocate -l 8G /var/swap_file
pi@raspberrypi:~ $ sudo chmod 600 /var/swap_file
pi@raspberrypi:~ $ sudo mkswap /var/swap_file
Setting up swapspace version 1, size = 8 GiB (8589930496 bytes)
no label, UUID=788d90aa-a002-40ed-80e2-68f4ac24d632
pi@raspberrypi:~ $ 
```

With that we have created the swap, but we still need to turn it on or activate it. If we want to turn it on, we execute:

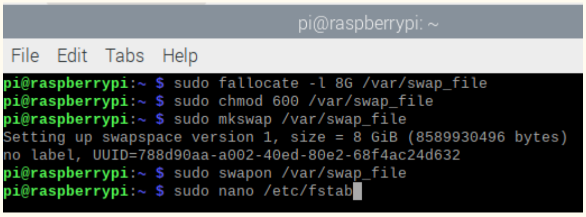
sudo swapon /var/swap_file



```
pi@raspberrypi:~ $ sudo fallocate -l 8G /var/swap_file
pi@raspberrypi:~ $ sudo chmod 600 /var/swap_file
pi@raspberrypi:~ $ sudo mkswap /var/swap_file
Setting up swapspace version 1, size = 8 GiB (8589930496 bytes)
no label, UUID=788d90aa-a002-40ed-80e2-68f4ac24d632
pi@raspberrypi:~ $ sudo swapon /var/swap_file
pi@raspberrypi:~ $ 
```

Finally we are going to make the change permanent even after a reboot. This is achieved by editing the **/etc/fstab** file that mounts the file systems. We edit it with **nano** or a text editor. If it is with nano, we execute**:**

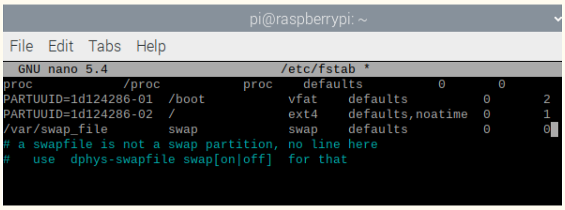
sudo nano /etc/fstab



```
pi@raspberrypi:~ $ sudo fallocate -l 8G /var/swap_file
pi@raspberrypi:~ $ sudo chmod 600 /var/swap_file
pi@raspberrypi:~ $ sudo mkswap /var/swap_file
Setting up swapspace version 1, size = 8 GiB (8589930496 bytes)
no label, UUID=788d90aa-a002-40ed-80e2-68f4ac24d632
pi@raspberrypi:~ $ sudo swapon /var/swap_file
pi@raspberrypi:~ $ sudo nano /etc/fstab
```

We add the following line:

/var/swap_file swap swap defaults 0 0

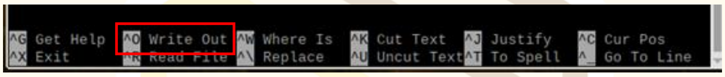


```
  GNU nano 5.4                           /etc/fstab *
proc            /proc           proc    defaults          0       0
PARTUUID=1d124286-01  /boot           vfat    defaults          0       2
PARTUUID=1d124286-02  /               ext4    defaults,noatime  0       1
/var/swap_file        swap            swap    defaults          0       0
# a swapfile is not a swap partition, no line here
#   use  dphys-swapfile swap[on|off]  for that
```

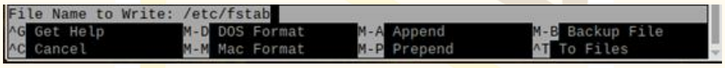
The first field indicates the location of the link (it would be the file we created), the second indicates where in the system it will be mounted (in **swap**). The third indicates the type of file system (that is why it is of type **swap**).

The fourth field indicates the options when mounting; We leave them in **defaults**. The fifth and sixth fields specify a behavior for dump and fsck, which we leave at 0.

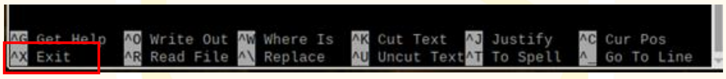
After editing we save **changes; if it's** *nano* then first press **Ctrl + O**, press **Enter** and finally **Ctrl + X**.



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



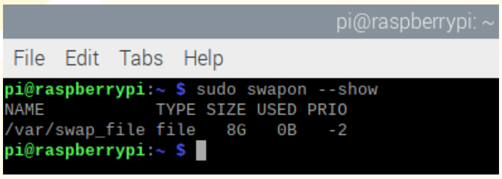
```
File Name to Write: /etc/fstab
^G Get Help            M-D DOS Format         M-A Append             M-B Backup File
^C Cancel              M-M Mac Format         M-P Prepend            ^T To Files
```



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

Check if we have created the swap file correctly.

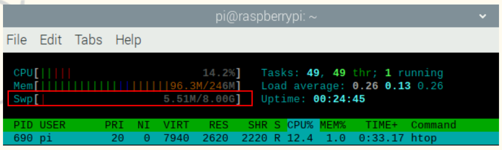
sudo swapon --show



```
pi@raspberrypi:~ $ sudo swapon --show
NAME           TYPE SIZE USED PRIO
/var/swap_file file   8G   0B   -2
pi@raspberrypi:~ $
```

or

htop



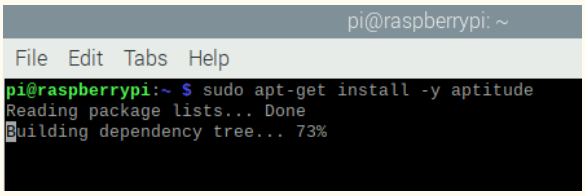
```
  CPU[|||||                     14.2%]   Tasks: 49, 49 thr; 1 running
  Mem[||||||||||||||||||||96.3M/246M]   Load average: 0.26 0.13 0.26
  Swp[|                    5.51M/8.00G]   Uptime: 00:24:45

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  690 pi         20   0  7940  2620  2220 R 12.4  1.0  0:33.17 htop
```

Download Source Code and Compile Binaries of Zent Cash

Before starting with the compilation of the source code you must download and install the necessary dependencies for it.

Next we will download the aptitude package manager to download and install the necessary dependencies.

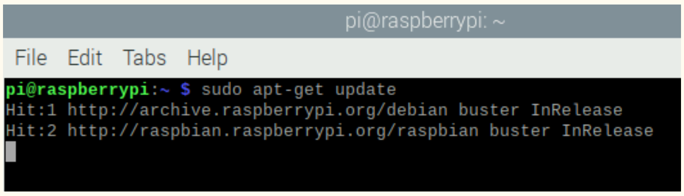
sudo apt-get install -y aptitude



```
pi@raspberrypi:~ $ sudo apt-get install -y aptitude
Reading package lists... Done
Building dependency tree... 73%
```

We enter the following command to download the package lists from the repositories and get information about the latest versions of the packages.

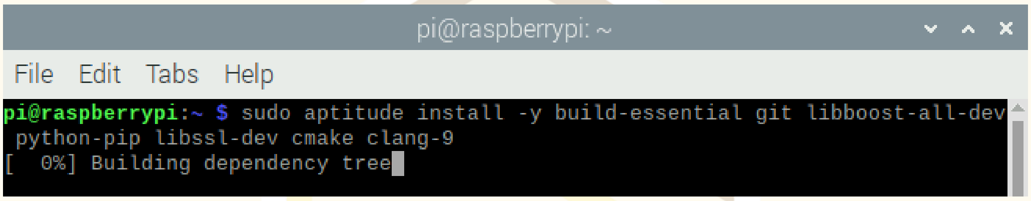
sudo apt-get update



```
pi@raspberrypi:~ $ sudo apt-get update
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian buster InRelease
```

We will install all the dependencies for the source code compilation with the following command:

sudo aptitude install -y build-essential git libboost-all-dev python-pip libssl-dev cmake clang-9



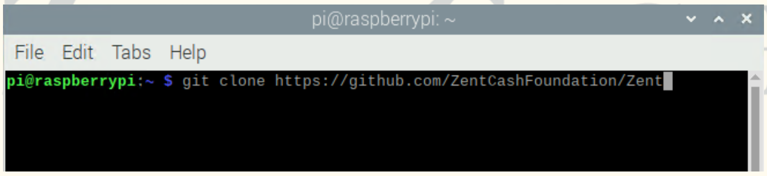
```
pi@raspberrypi:~ $ sudo aptitude install -y build-essential git libboost-all-dev
 python-pip libssl-dev cmake clang-9
[  0%] Building dependency tree
```

DEPENDENCIES

build-essential git libboost-all-dev python-pip libssl-dev cmake clang-9

We will download the source code of Zent Cash from the Github repository.

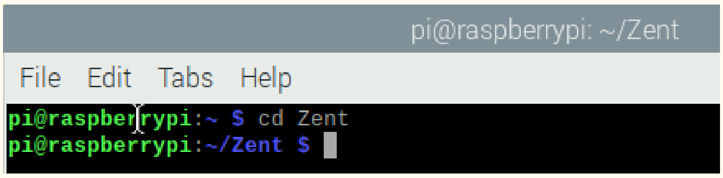
git clone https://github.com/ZentCashFoundation/Zent



```
pi@raspberrypi:~ $ git clone https://github.com/ZentCashFoundation/Zent
```

We access the folder that was generated with the previous command.

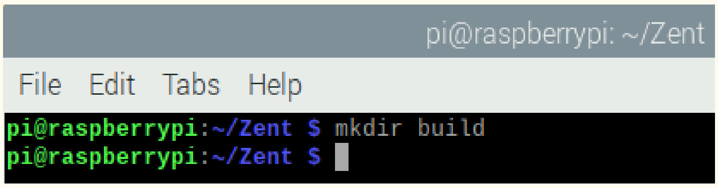
cd Zent



```
pi@raspberrypi:~ $ cd Zent
pi@raspberrypi:~/Zent $
```

We create the folder build. In this folder we will compile the source code.

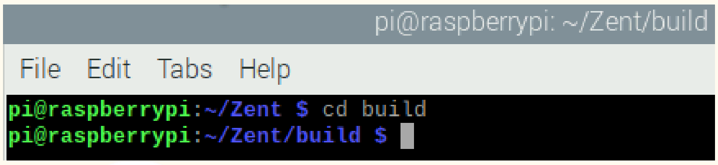
mkdir build



```
pi@raspberrypi:~/Zent $ mkdir build
pi@raspberrypi:~/Zent $ 
```

We access the directory build.

cd build

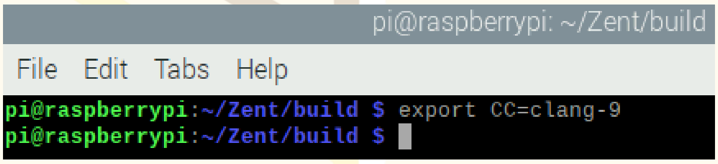


```
pi@raspberrypi:~/Zent $ cd build
pi@raspberrypi:~/Zent/build $ 
```

Before starting to compile the code we must assign the environment variables CC (C) and CXX (C++).

We assign CC:

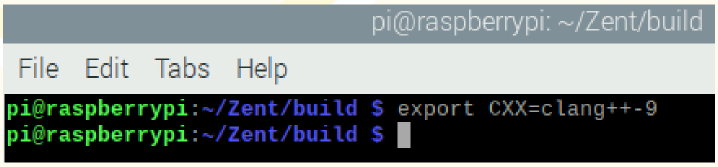
export CC=clang-9



```
pi@raspberrypi:~/Zent/build $ export CC=clang-9
pi@raspberrypi:~/Zent/build $ 
```

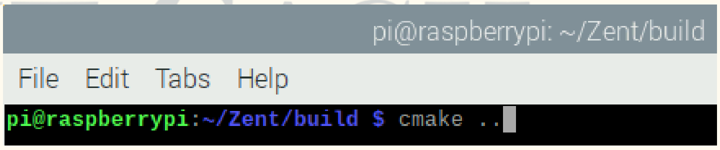
We assign CXX:

export CXX=clang++-9



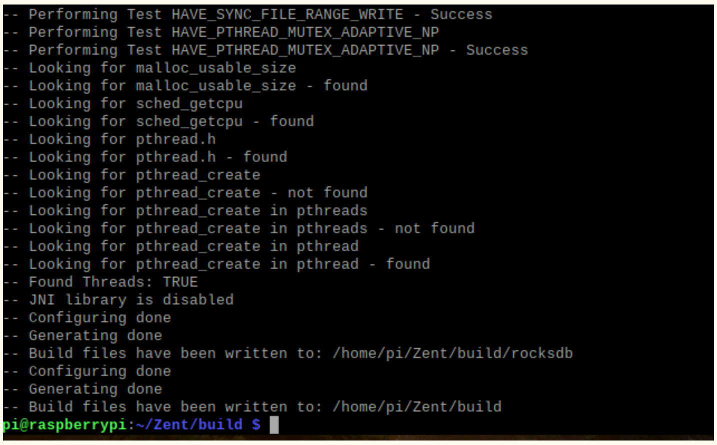
```
pi@raspberrypi:~/Zent/build $ export CXX=clang++-9
pi@raspberrypi:~/Zent/build $ 
```

We type the following command to generate the Makefile needed to start the build.



```
pi@raspberrypi:~/Zent/build $ cmake ..
```

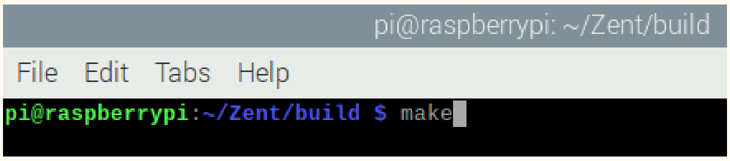
cmake ..



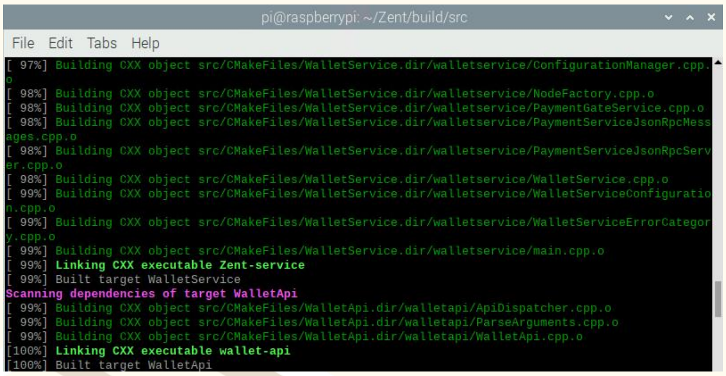
```
-- Performing Test HAVE_SYNC_FILE_RANGE_WRITE - Success
-- Performing Test HAVE_PTHREAD_MUTEX_ADAPTIVE_NP
-- Performing Test HAVE_PTHREAD_MUTEX_ADAPTIVE_NP - Success
-- Looking for malloc_usable_size
-- Looking for malloc_usable_size - found
-- Looking for sched_getcpu
-- Looking for sched_getcpu - found
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- JNI library is disabled
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pi/Zent/build/rocksdb
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pi/Zent/build
pi@raspberrypi:~/Zent/build $ 
```

We type the following command to start the source code compilation process.

make



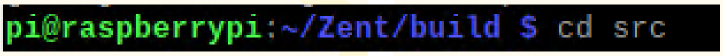
```
pi@raspberrypi:~/Zent/build $ make
```



```
[ 97%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/ConfigurationManager.cpp.
o
[ 98%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/NodeFactory.cpp.o
[ 98%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/PaymentGateService.cpp.o
[ 98%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/PaymentServiceJsonRpcMess
ages.cpp.o
[ 98%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/PaymentServiceJsonRpcServ
er.cpp.o
[ 98%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/WalletService.cpp.o
[ 99%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/WalletServiceConfiguratio
n.cpp.o
[ 99%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/WalletServiceErrorCategor
y.cpp.o
[ 99%] Building CXX object src/CMakeFiles/WalletService.dir/walletservice/main.cpp.o
[ 99%] Linking CXX executable Zent-service
[ 99%] Built target WalletService
Scanning dependencies of target WalletApi
[ 99%] Building CXX object src/CMakeFiles/WalletApi.dir/walletapi/ApiDispatcher.cpp.o
[ 99%] Building CXX object src/CMakeFiles/WalletApi.dir/walletapi/ParseArguments.cpp.o
[ 99%] Building CXX object src/CMakeFiles/WalletApi.dir/walletapi/WalletApi.cpp.o
[100%] Linking CXX executable wallet-api
[100%] Built target WalletApi
```

Once the compilation process is finished, we access the src directory to access the files generated in the compilation process.

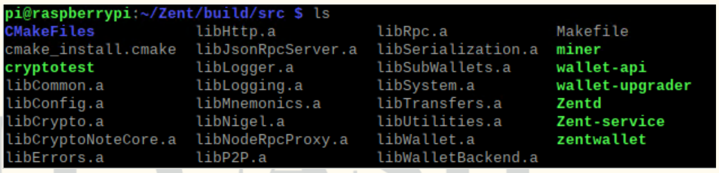
cd src



```
pi@raspberrypi:~/Zent/build $ cd src
```

To view the files resulting from the compilation we write the following command.

ls



```
pi@raspberrypi:~/Zent/build/src $ ls
CMakeFiles             libHttp.a            libRpc.a              Makefile
cmake_install.cmake    libJsonRpcServer.a   libSerialization.a    miner
cryptotest             libLogger.a          libSubWallets.a       wallet-api
libCommon.a            libLogging.a         libSystem.a           wallet-upgrader
libConfig.a            libMnemonics.a       libTransfers.a        Zentd
libCrypto.a            libNigel.a           libUtilities.a        Zent-service
libCryptoNoteCore.a    libNodeRpcProxy.a    libWallet.a           zentwallet
libErrors.a            libP2P.a             libWalletBackend.a
```

Zentd (Daemon) = It allows us to download the blockchain and keep it in sync.

zentwallet (Wallet CLI) = It is the main wallet via console.

Zent-service = Zent Cash RPC Wallet is an HTTP server that provides a JSON 2.0 RPC interface for Zent Cash payment operations and address management.

wallet-api = Zent Cash RPC Wallet is an HTTP server that provides a Swagger-based REST interface for Zent Cash payment operations and address management.

wallet-upgrader = It is used to convert Zent-service wallets to wallets compatible with wallet-api and zentwallet.

miner = It is used to mine the currency with your CPU.

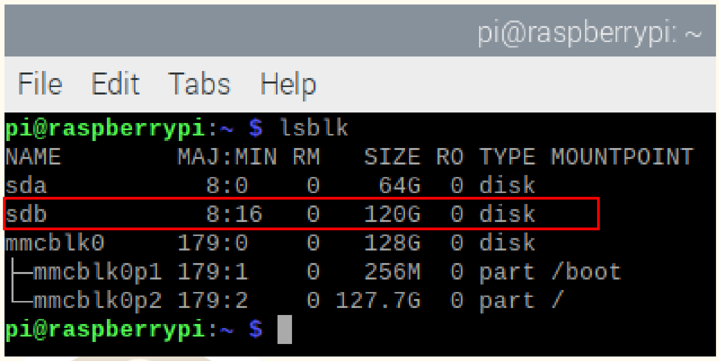
cryptotest = It is used to benchmark the different algorithms.

Mount your SSD

Zentd needs alot of space on a fast drive for the database. So plug in your SSD to one of the USB ports on the pi and we'll create a mount point for it.

First, run this command to view your available disks.

lsblk

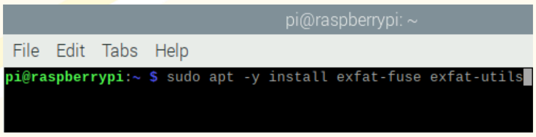


```
pi@raspberrypi:~ $ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda           8:0    0    64G  0 disk
sdb           8:16   0   120G  0 disk
mmcblk0     179:0    0   128G  0 disk
├─mmcblk0p1 179:1    0   256M  0 part /boot
└─mmcblk0p2 179:2    0 127.7G  0 part /
pi@raspberrypi:~ $
```

Look at the options and take note of which one is your SSD. It should say something about "Disk" and have the correct size as your drive. For me, it was located at /dev/sdb**.**

I recommend formatting the device to exFAT so that I can access it easily on both windows and linux systems. In order to mount this type of format, we'll need some additional packages.

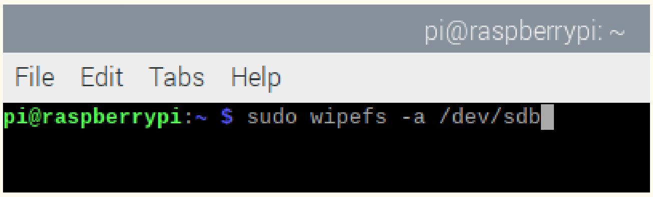
sudo apt install exfat-fuse exfat-utils



```
pi@raspberrypi:~ $ sudo apt -y install exfat-fuse exfat-utils
```

Assuming you wanted to destroy everything on it and start fresh with a new exFAT file system, type **sudo wipefs -a /dev/sdb** so you can start fresh.

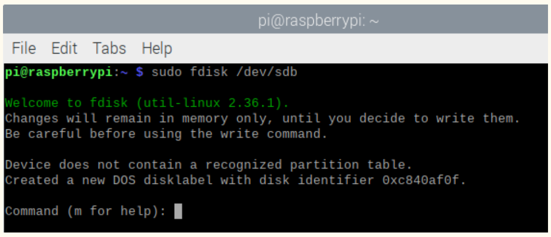
sudo wipefs -a /dev/sdb



```
pi@raspberrypi:~ $ sudo wipefs -a /dev/sdb
```

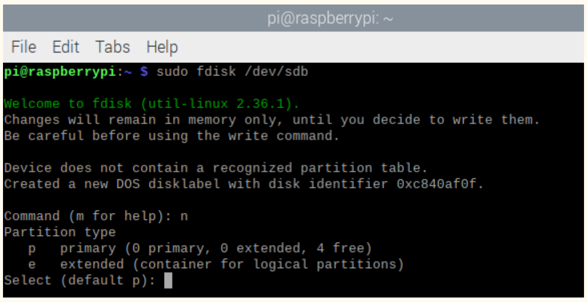
Next, you'll want to type **sudo fdisk /dev/sdb** and push enter so you can make a new partition table. Please make sure you're working with the right device file to avoid mucking anything up.

sudo fdisk /dev/sdb



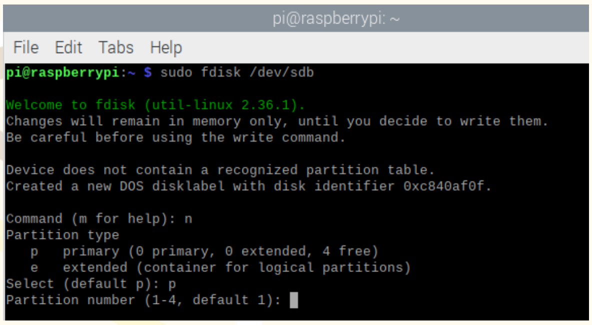
```
pi@raspberrypi:~ $ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc840af0f.

Command (m for help):
```

Now type the letter **n** and push **enter**.



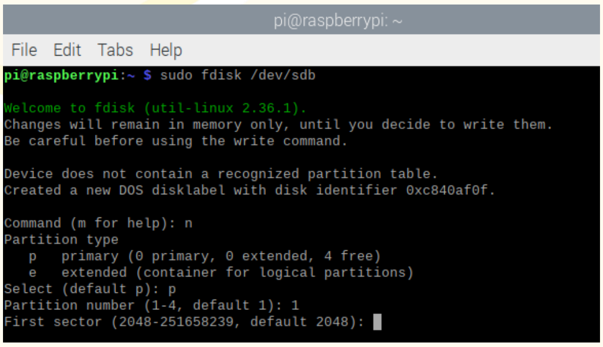
```
pi@raspberrypi:~ $ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc840af0f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): 
```

Now type the letter **p** to create a primary partition and press **enter**.



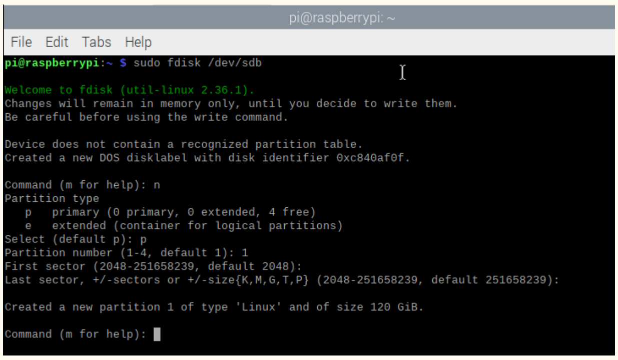
```
pi@raspberrypi:~ $ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc840af0f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 
```

Now type the number **1** and push **enter**.



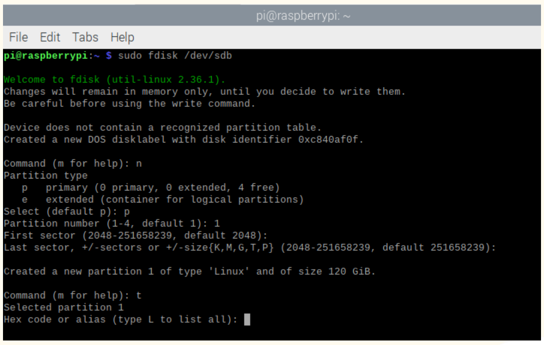
```
pi@raspberrypi:~ $ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc840af0f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-251658239, default 2048): 
```

Now push enter and enter**.**

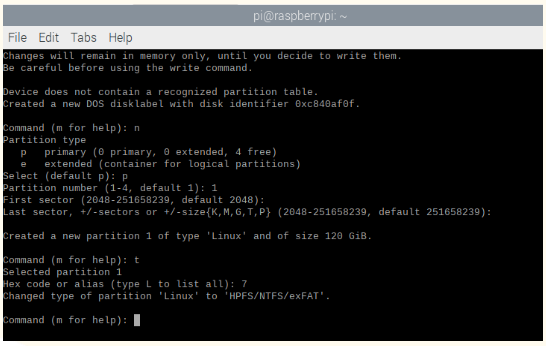


```
pi@raspberrypi:~ $ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc840af0f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-251658239, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-251658239, default 251658239):

Created a new partition 1 of type 'Linux' and of size 120 GiB.

Command (m for help): 
```

Push the letter **t** and then push the **enter** key so you can change the type. You can then push **7** to the expected type. It should give you a message about HPFS/NTFS/exFAT, which is exactly what you want if you're looking to format your external disk with exFAT on Linux.

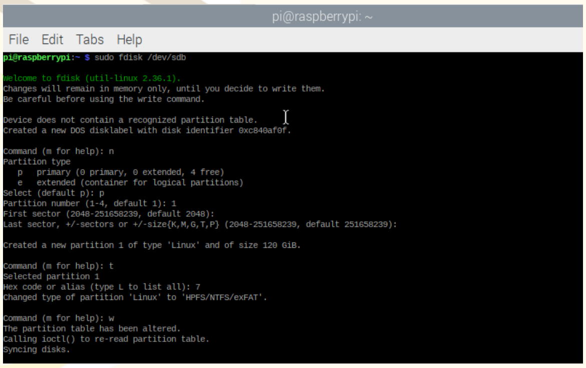


```
pi@raspberrypi:~ $ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc840af0f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-251658239, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-251658239, default 251658239):

Created a new partition 1 of type 'Linux' and of size 120 GiB.

Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 
```



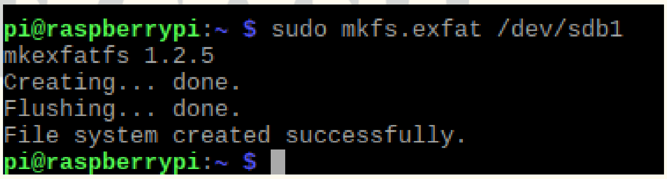
```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc840af0f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-251658239, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-251658239, default 251658239):

Created a new partition 1 of type 'Linux' and of size 120 GiB.

Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 7
Changed type of partition 'Linux' to 'HPFS/NTFS/exFAT'.

Command (m for help): 
```

Finally, you simply have to push the **w** key.



```
pi@raspberrypi:~ $ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc840af0f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-251658239, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-251658239, default 251658239):

Created a new partition 1 of type 'Linux' and of size 120 GiB.

Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 7
Changed type of partition 'Linux' to 'HPFS/NTFS/exFAT'.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

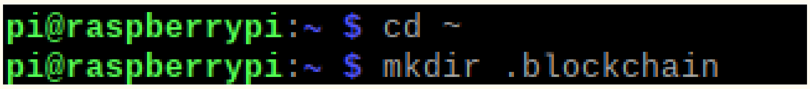
You're now ready for the format.
Assuming that the device file from before was /dev/sdb, you can now format your drive with **sudo mkfs.exfat /dev/sdb1**.



```
pi@raspberrypi:~ $ sudo mkfs.exfat /dev/sdb1
mkexfatfs 1.2.5
Creating... done.
Flushing... done.
File system created successfully.
pi@raspberrypi:~ $ 
```

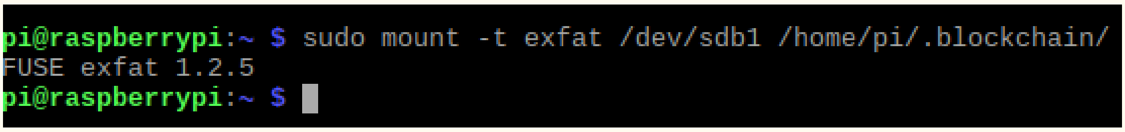
Now, we can create a folder called .blockchain in the home directory and mount the SSD there.

cd ~
mkdir .blockchain



```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ mkdir .blockchain
```

sudo mount -t exfat /dev/sdb1 /home/pi/.blockchain/



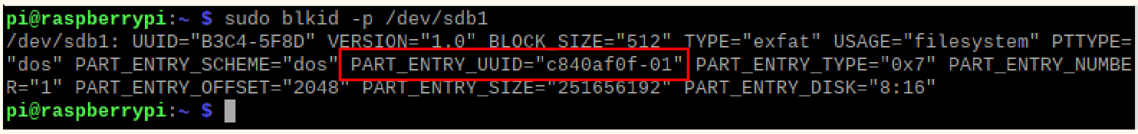
```
pi@raspberrypi:~ $ sudo mount -t exfat /dev/sdb1 /home/pi/.blockchain/
FUSE exfat 1.2.5
pi@raspberrypi:~ $ 
```

That will temporarily mount the drive to ~/.blockchain, but if we want it to be permanent. We will have to edit the fstab file.

Before we edit the fstab file, we first need to get the PARTUUID of the exfat partition we created earlier.

To do this we write the following command:

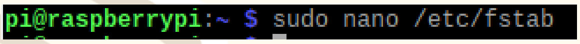
sudo blkid -p /dev/sdb1



```
pi@raspberrypi:~ $ sudo blkid -p /dev/sdb1
/dev/sdb1: UUID="B3C4-5F8D" VERSION="1.0" BLOCK_SIZE="512" TYPE="exfat" USAGE="filesystem" PTTYPE=
"dos" PART_ENTRY_SCHEME="dos" PART_ENTRY_UUID="c840af0f-01" PART_ENTRY_TYPE="0x7" PART_ENTRY_NUMBE
R="1" PART_ENTRY_OFFSET="2048" PART_ENTRY_SIZE="251656192" PART_ENTRY_DISK="8:16"
pi@raspberrypi:~ $
```

Now that we have the PARTUUID of the exfat partition we can edit the fstab file located in the /etc/fstab directory.
We write the following command for this:

sudo nano /etc/fstab

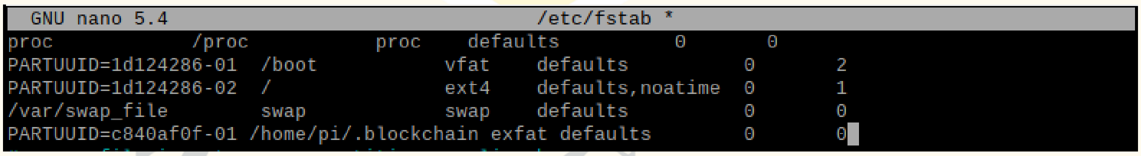


```
pi@raspberrypi:~ $ sudo nano /etc/fstab
```

We add the following line at the end of the fstab file.

PARTUUID=c840af0f-01   /home/pi/.blockchain   exfat   defaults   0   0

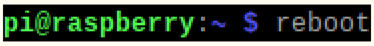
Partition   Mount   File System



```
  GNU nano 5.4                               /etc/fstab *
proc            /proc           proc    defaults          0       0
PARTUUID=1d124286-01  /boot           vfat    defaults          0       2
PARTUUID=1d124286-02  /               ext4    defaults,noatime  0       1
/var/swap_file        swap            swap    defaults          0       0
PARTUUID=c840af0f-01 /home/pi/.blockchain exfat defaults        0       0
```

After editing the file we save the changes by pressing **Ctrl + O**, then press **Enter** and finally **Ctrl + X** to close the file.

We restart the machine with the **reboot** command.

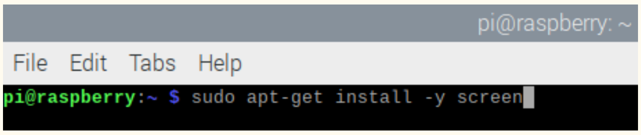


```
pi@raspberry:~ $ reboot
```

Run Zentd and Keep it Running

The first thing we are going to do is install screen to be able to run the daemon in background.

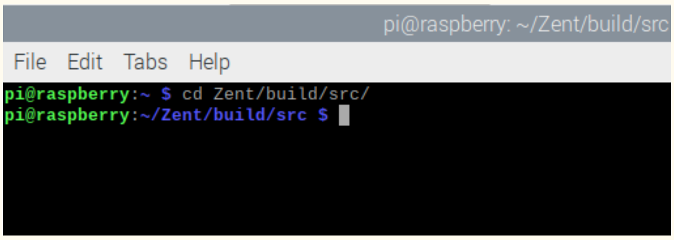
sudo apt-get install -y screen



```
pi@raspberry:~ $ sudo apt-get install -y screen
```

We access the directory where the Zent Cash binaries are located.

cd Zent/build/src/

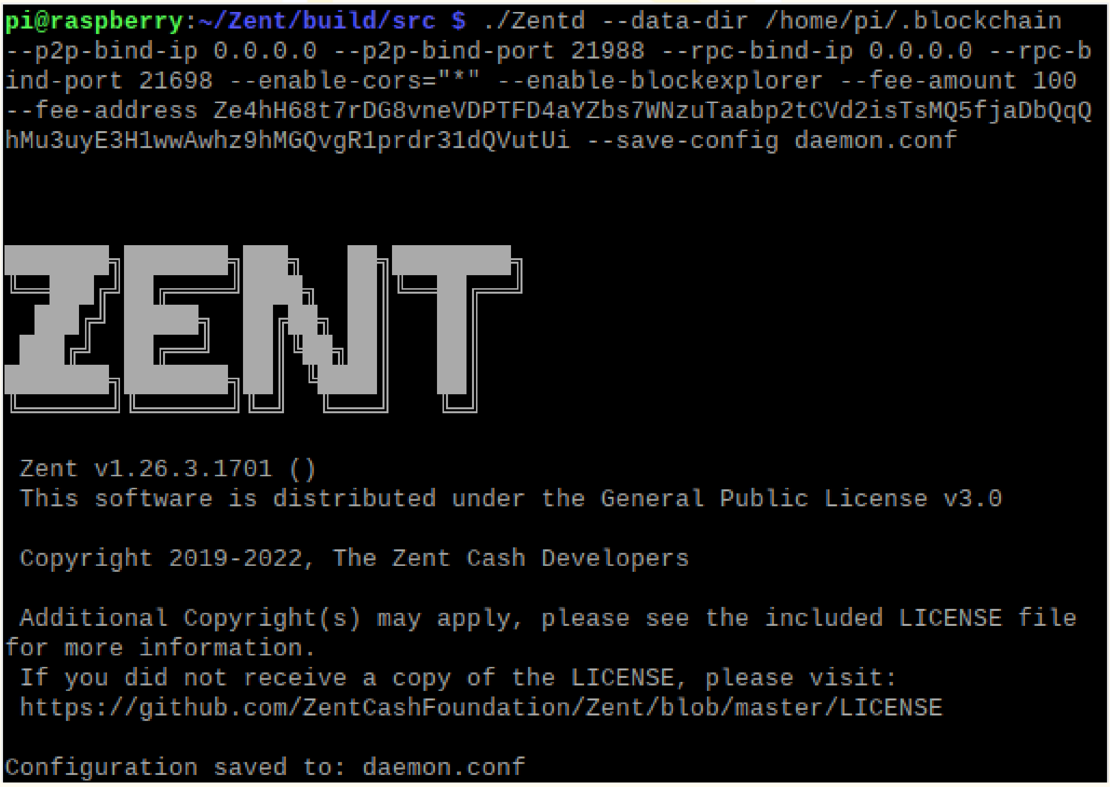


```
pi@raspberry:~ $ cd Zent/build/src/
pi@raspberry:~/Zent/build/src $ 
```

We run Zentd with the following parameters to generate the daemon.conf file.

> ⚠ Remember to change the address in the **--fee-address** parameter with your wallet address. You can also modify the **--fee-amount** parameter with the amount you want to charge for each transaction your node makes.

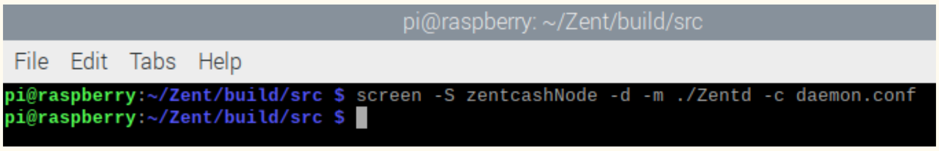
./Zentd --data-dir /home/pi/.blockchain --p2p-bind-ip 0.0.0.0 --p2p-bind-port 21988 --rpc-bind-ip 0.0.0.0 --rpc-bind-port 21698 --enable-cors=”*” --enable-blockexplorer --fee-amount 100 --fee-address Ze4hH68t7rDG8vneVDPTFD4aYZbs7WNzuTaabp2tCVd2isTsMQ5fjaDbQqQhMu3uyE3H1wwAwhz9hMGQvgR1prdr31dQVutUi --save-config daemon.conf



```
pi@raspberry:~/Zent/build/src $ ./Zentd --data-dir /home/pi/.blockchain
--p2p-bind-ip 0.0.0.0 --p2p-bind-port 21988 --rpc-bind-ip 0.0.0.0 --rpc-b
ind-port 21698 --enable-cors="*" --enable-blockexplorer --fee-amount 100
--fee-address Ze4hH68t7rDG8vneVDPTFD4aYZbs7WNzuTaabp2tCVd2isTsMQ5fjaDbQqQ
hMu3uyE3H1wwAwhz9hMGQvgR1prdr31dQVutUi --save-config daemon.conf


 Zent v1.26.3.1701 ()
 This software is distributed under the General Public License v3.0

 Copyright 2019-2022, The Zent Cash Developers

 Additional Copyright(s) may apply, please see the included LICENSE file
for more information.
 If you did not receive a copy of the LICENSE, please visit:
 https://github.com/ZentCashFoundation/Zent/blob/master/LICENSE

Configuration saved to: daemon.conf
```

Now we only have to run the daemon with screen to be able to keep it started without having to keep our command console open.
To do this, we write the following command:

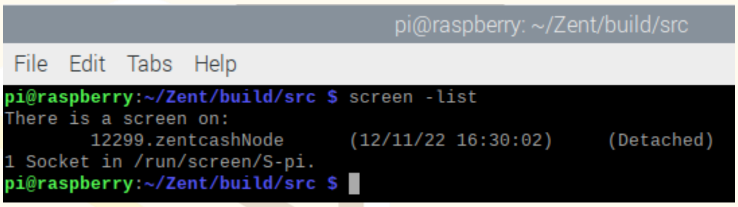
screen -S zentcashNode -d -m ./Zentd -c daemon.conf



```
pi@raspberry:~/Zent/build/src $ screen -S zentcashNode -d -m ./Zentd -c daemon.conf
pi@raspberry:~/Zent/build/src $ 
```

We can check that it is running in the background on screen by typing the following command:

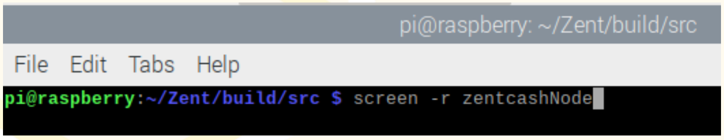
screen -list



```
pi@raspberry:~/Zent/build/src $ screen -list
There is a screen on:
        12299.zentcashNode      (12/11/22 16:30:02)     (Detached)
1 Socket in /run/screen/S-pi.
pi@raspberry:~/Zent/build/src $ 
```

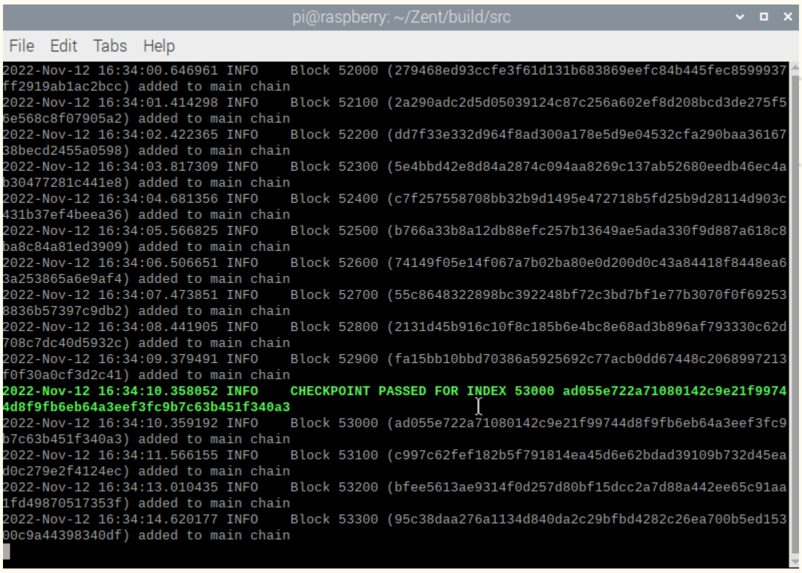
We can access the terminal in the background with the following command:

screen -r zentcashNode



```
pi@raspberry:~/Zent/build/src $ screen -r zentcashNode
```

As we can see the node is already synchronizing with the network.

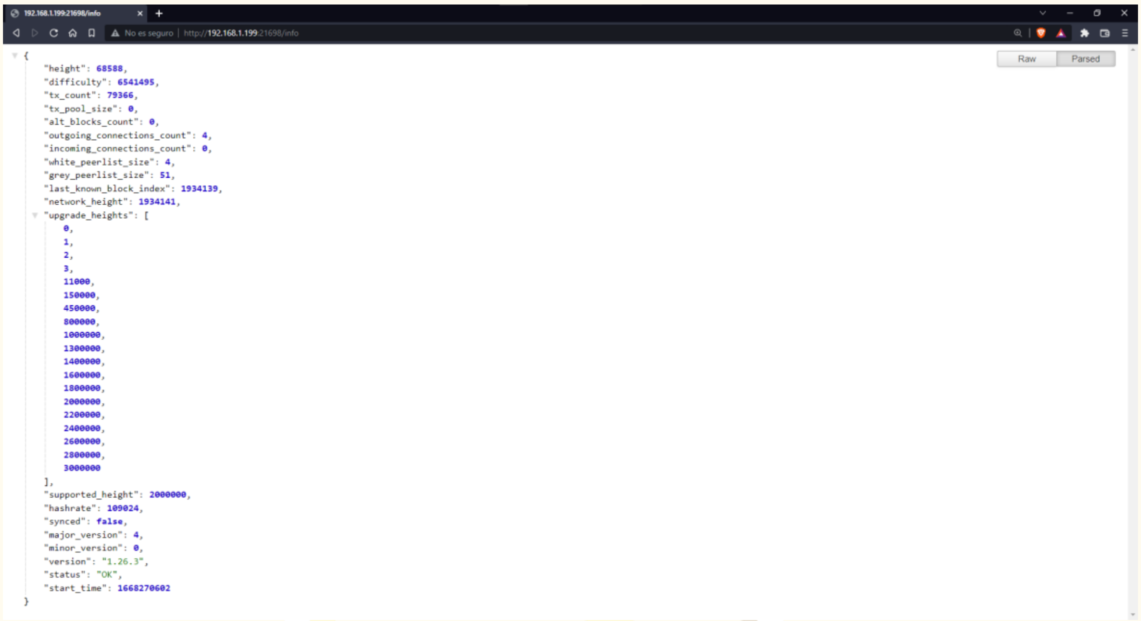


```
2022-Nov-12 16:34:00.646961 INFO    Block 52000 (279468ed93ccfe3f61d131b683869eefc84b445fec8599937
ff2919ab1ac2bcc) added to main chain
2022-Nov-12 16:34:01.414298 INFO    Block 52100 (2a290adc2d5d05039124c87c256a602ef8d208bcd3de275f5
6e568c8f07905a2) added to main chain
2022-Nov-12 16:34:02.422365 INFO    Block 52200 (dd7f33e332d964f8ad300a178e5d9e04532cfa290baa36167
38becd2455a0598) added to main chain
2022-Nov-12 16:34:03.817309 INFO    Block 52300 (5e4bbd42e8d84a2874c094aa8269c137ab52680eedb46ec4a
b30477281c441e8) added to main chain
2022-Nov-12 16:34:04.681356 INFO    Block 52400 (c7f257558708bb32b9d1495e472718b5fd25b9d28114d903c
431b37ef4beea36) added to main chain
2022-Nov-12 16:34:05.566825 INFO    Block 52500 (b766a33b8a12db88efc257b13649ae5ada330f9d887a618c8
ba8c84a81ed3909) added to main chain
2022-Nov-12 16:34:06.506651 INFO    Block 52600 (74149f05e14f067a7b02ba80e0d200d0c43a84418f8448ea6
3a253865a6e9af4) added to main chain
2022-Nov-12 16:34:07.473851 INFO    Block 52700 (55c8648322898bc392248bf72c3bd7bf1e77b3070f0f69253
8836b57397c9db2) added to main chain
2022-Nov-12 16:34:08.441905 INFO    Block 52800 (2131d45b916c10f8c185b6e4bc8e68ad3b896af793330c62d
708c7dc40d5932c) added to main chain
2022-Nov-12 16:34:09.379491 INFO    Block 52900 (fa15bb10bbd70386a5925692c77acb0dd67448c2068997213
f0f30a0cf3d2c41) added to main chain
2022-Nov-12 16:34:10.358052 INFO    CHECKPOINT PASSED FOR INDEX 53000 ad055e722a71080142c9e21f9974
4d8f9fb6eb64a3eef3fc9b7c63b451f340a3
2022-Nov-12 16:34:10.359192 INFO    Block 53000 (ad055e722a71080142c9e21f99744d8f9fb6eb64a3eef3fc9
b7c63b451f340a3) added to main chain
2022-Nov-12 16:34:11.566155 INFO    Block 53100 (c997c62fef182b5f791814ea45d6e62bdad39109b732d45ea
d0c279e2f4124ec) added to main chain
2022-Nov-12 16:34:13.010435 INFO    Block 53200 (bfee5613ae9314f0d257d80bf15dcc2a7d88a442ee65c91aa
1fd49870517353f) added to main chain
2022-Nov-12 16:34:14.620177 INFO    Block 53300 (95c38daa276a1134d840da2c29bfbd4282c26ea700b5ed153
00c9a44398340df) added to main chain
```

To return to background mode press **Ctrl + A + D**.

You can monitor how your node sync is going from your browser by typing the following.

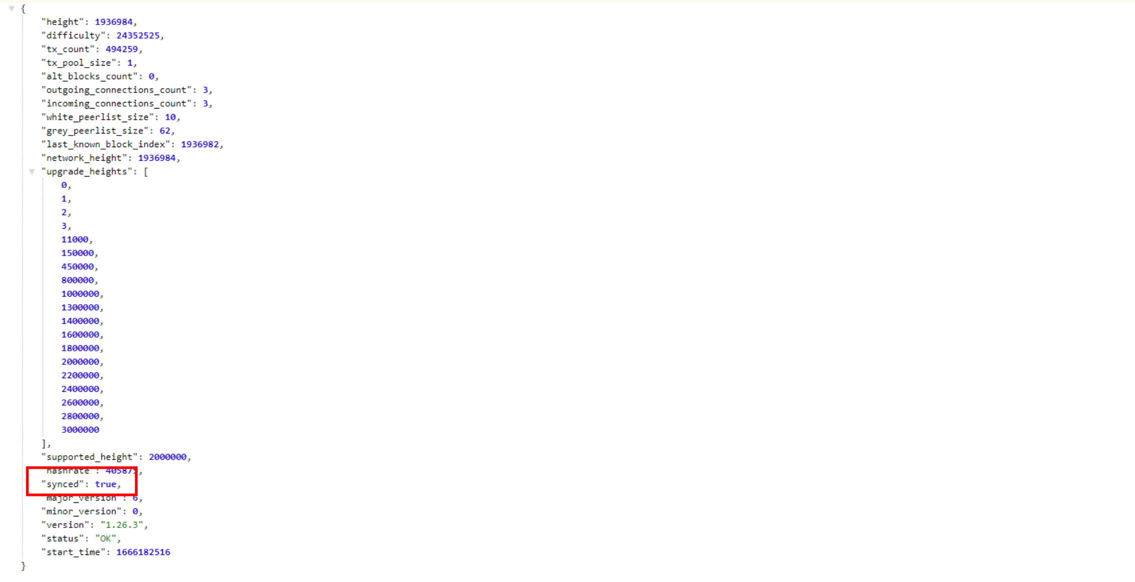
http://192.168.1.199:21698/info



```
{
    "height": 68588,
    "difficulty": 6541495,
    "tx_count": 79366,
    "tx_pool_size": 0,
    "alt_blocks_count": 0,
    "outgoing_connections_count": 4,
    "incoming_connections_count": 0,
    "white_peerlist_size": 4,
    "grey_peerlist_size": 51,
    "last_known_block_index": 1934139,
    "network_height": 1934141,
    "upgrade_heights": [
        0,
        1,
        2,
        3,
        11000,
        150000,
        450000,
        800000,
        1000000,
        1300000,
        1400000,
        1600000,
        1800000,
        2000000,
        2200000,
        2400000,
        2600000,
        2800000,
        3000000
    ],
    "supported_height": 2000000,
    "hashrate": 109024,
    "synced": false,
    "major_version": 4,
    "minor_version": 0,
    "version": "1.26.3",
    "status": "OK",
    "start_time": 1668270602
}
```

Note that you need to change the IP address to whatever address your machine has.

When your node is fully synced with the network the synced parameter will be true.



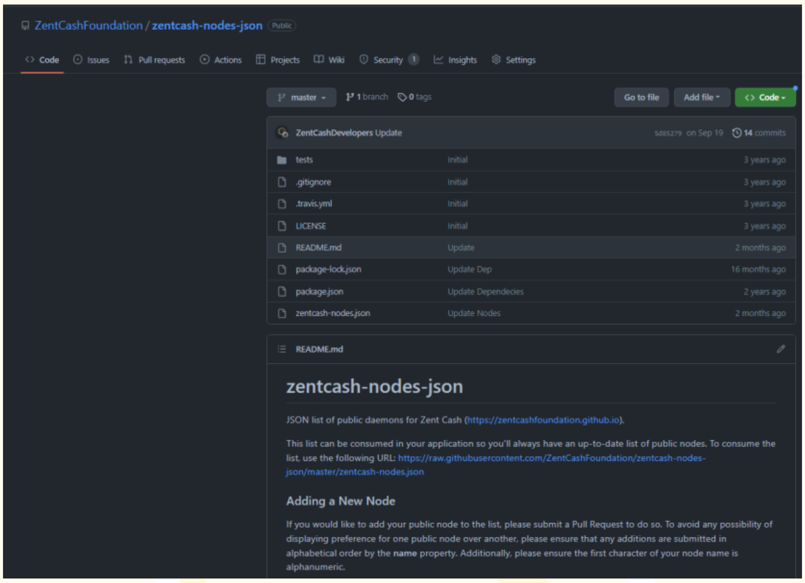
```
{
    "height": 1936984,
    "difficulty": 24352525,
    "tx_count": 494259,
    "tx_pool_size": 1,
    "alt_blocks_count": 0,
    "outgoing_connections_count": 3,
    "incoming_connections_count": 3,
    "white_peerlist_size": 10,
    "grey_peerlist_size": 62,
    "last_known_block_index": 1936982,
    "network_height": 1936984,
    "upgrade_heights": [
        0,
        1,
        2,
        3,
        11000,
        150000,
        450000,
        800000,
        1000000,
        1300000,
        1400000,
        1600000,
        1800000,
        2000000,
        2200000,
        2400000,
        2600000,
        2800000,
        3000000
    ],
    "supported_height": 2000000,
    "hashrate": 40587[illegible],
    "synced": true,
    "major_version": 6,
    "minor_version": 0,
    "version": "1.26.3",
    "status": "OK",
    "start_time": 1666182516
}
```

Finally, if you want to have your node available so that anyone can use it, you must allow TCP traffic through ports 21688 and 21698 in your firewall and router.

Promote your Node

If you want to promote your node you can commit your node data to the following github repository.

https://github.com/ZentCashFoundation/zentcash-nodes-json



ZentCashDevelopers Update — 5d85279 on Sep 19 — 14 commits

tests	Initial	3 years ago
.gitignore	Initial	3 years ago
.travis.yml	Initial	3 years ago
LICENSE	Initial	3 years ago
README.md	Update	2 months ago
package-lock.json	Update Dep	16 months ago
package.json	Update Dependecies	2 years ago
zentcash-nodes.json	Update Nodes	2 months ago

README.md

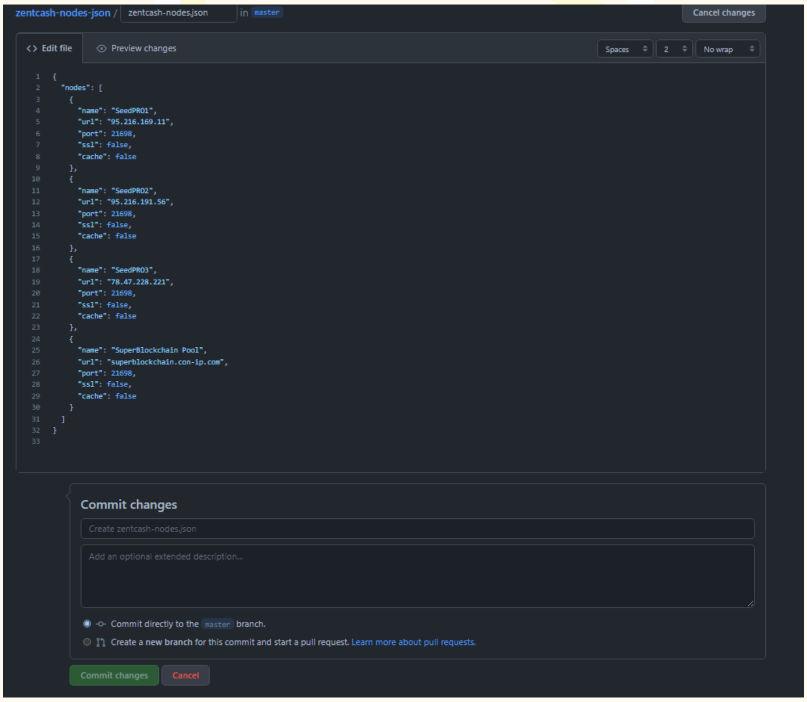
zentcash-nodes-json

JSON list of public daemons for Zent Cash (https://zentcashfoundation.github.io).

This list can be consumed in your application so you'll always have an up-to-date list of public nodes. To consume the list, use the following URL: https://raw.githubusercontent.com/ZentCashFoundation/zentcash-nodes-json/master/zentcash-nodes.json

Adding a New Node

If you would like to add your public node to the list, please submit a Pull Request to do so. To avoid any possibility of displaying preference for one public node over another, please ensure that any additions are submitted in alphabetical order by the **name** property. Additionally, please ensure the first character of your node name is alphanumeric.



```
{
  "nodes": [
    {
      "name": "SeedPR01",
      "url": "95.216.169.11",
      "port": 21698,
      "ssl": false,
      "cache": false
    },
    {
      "name": "SeedPR02",
      "url": "95.216.191.56",
      "port": 21698,
      "ssl": false,
      "cache": false
    },
    {
      "name": "SeedPR03",
      "url": "78.47.228.221",
      "port": 21698,
      "ssl": false,
      "cache": false
    },
    {
      "name": "SuperBlockchain Pool",
      "url": "superblockchain.con-ip.com",
      "port": 21698,
      "ssl": false,
      "cache": false
    }
  ]
}
```

Commit changes

Create zentcash-nodes.json

Add an optional extended description...

● Commit directly to the master branch.

○ Create a new branch for this commit and start a pull request. Learn more about pull requests.

You can also promote your node through our social networks.

Discord **Reddit** **Telegram** **Facebook** **Twitter** **Youtube** **Odysee**