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](https://www.amazon.com/StarTech-com-SATA-Drive-Adapter-Cable/dp/B00HJZJI84/ref=sr_1_3?keywords=startech+usb+to+sata&qid=1548102862&sr=8-3&th=1).
* 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](https://downloads.raspberrypi.org/raspios_full_armhf/images/raspios_full_armhf-2022-04-07/2022-04-04-raspios-bullseye-armhf-full.img.xz)).
* [Etcher](https://www.balena.io/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](https://downloads.raspberrypi.org/raspios_oldstable_armhf/images/raspios_oldstable_armhf-2022-09-26/2022-09-22-raspios-buster-armhf.img.xz) 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.

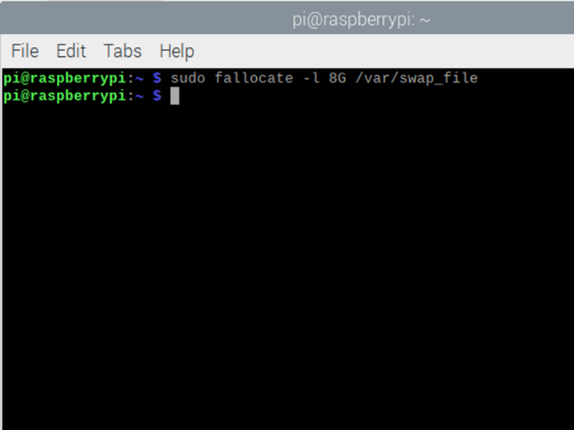
**OS Setup**

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.

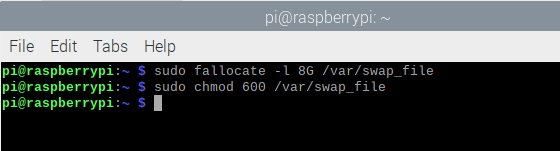
**Creation and Mounting of Swap Archive**

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**

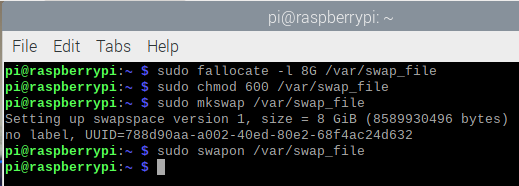
The fallocate command is used to preallocate space in a file; we specify the space with the -l option that indicates the length (it would be 8 GB). The last argument is the name of the file that we are going to host.

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**

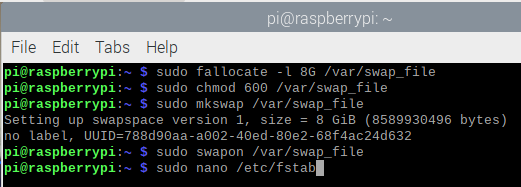
Now comes the final step and that is to convert our file into a swap file, using **mkswap.**

**sudo mkswap /var/swap\_file**

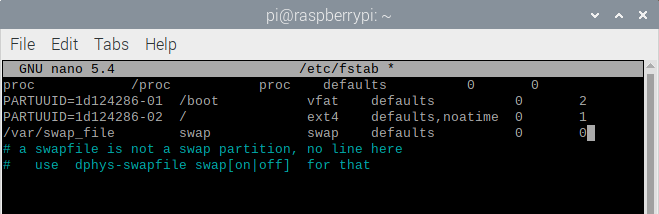
With that we have created the swap, but we still need to turn it on or activate it. If we want to turn it on, we execute:

**sudo swapon /var/swap\_file**

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**

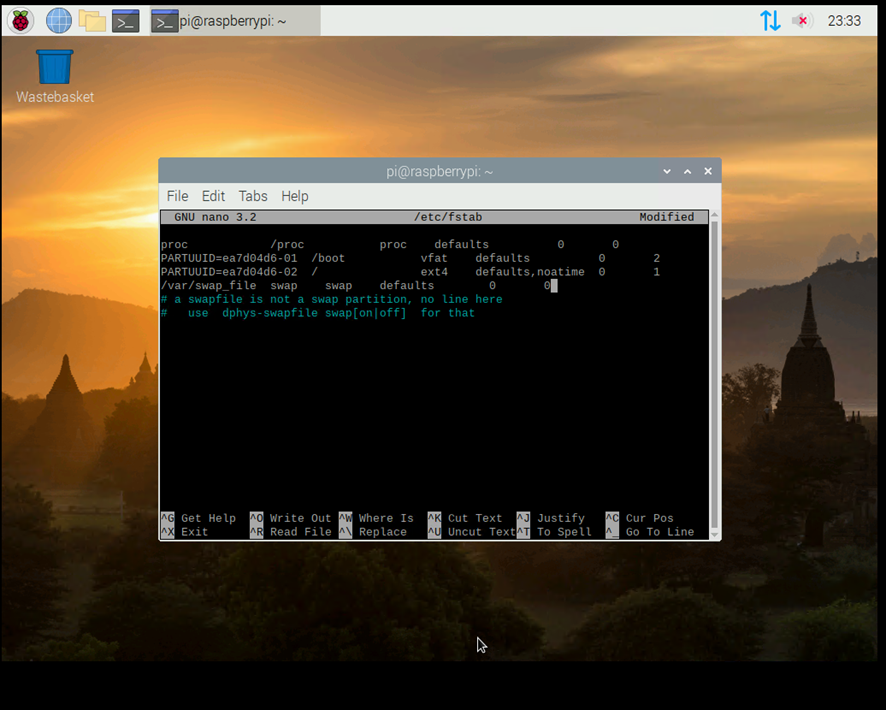
We add the following line:

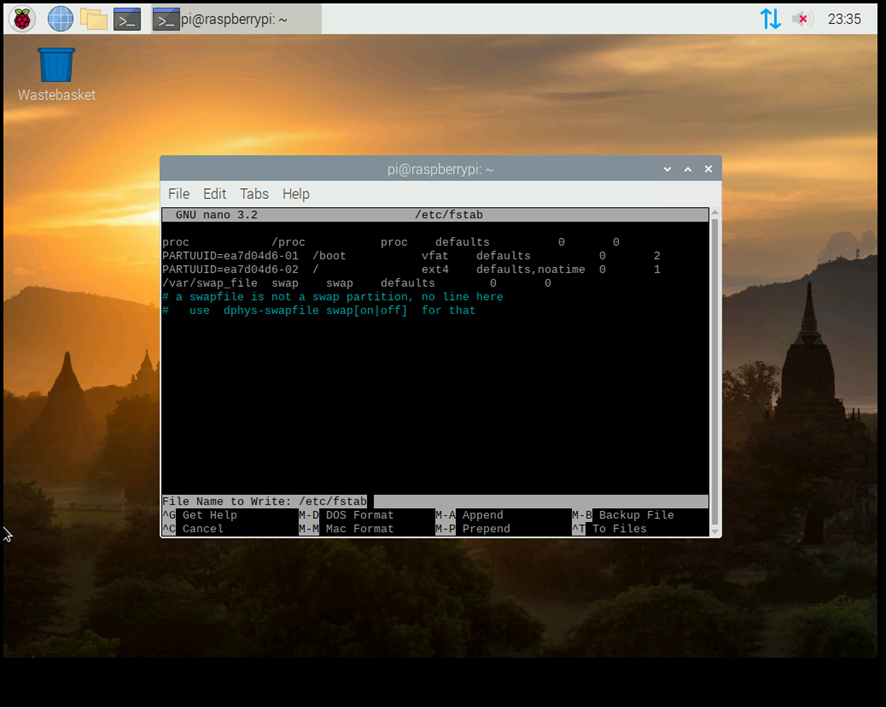
**/var/swap\_file swap swap defaults 0 0**

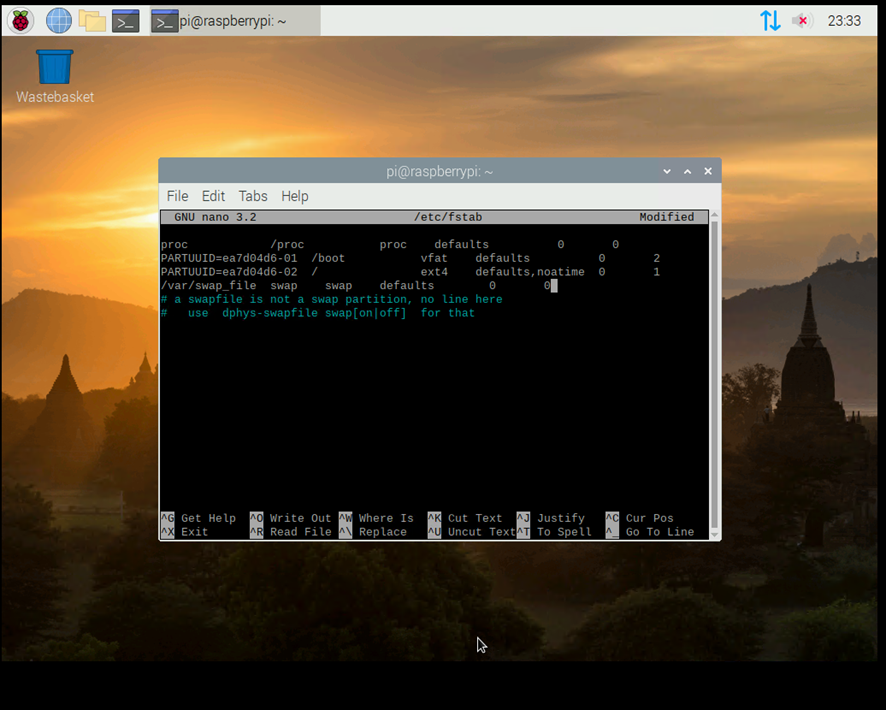
The first field indicates the location of the link (it would be the file we created), the second indicates where in the system it will be mounted (in **swap**). The third indicates the type of file system (that is why it is of type **swap**).

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.

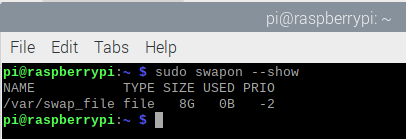
After editing we save **changes; if it's** *nano* then first press **Ctrl + O**, press **Enter** and finally **Ctrl + X**.



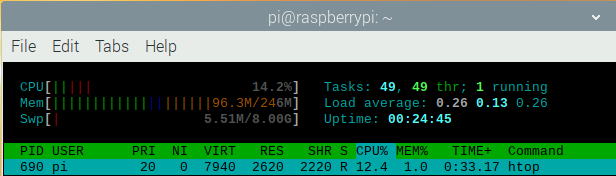




Check if we have created the swap file correctly.



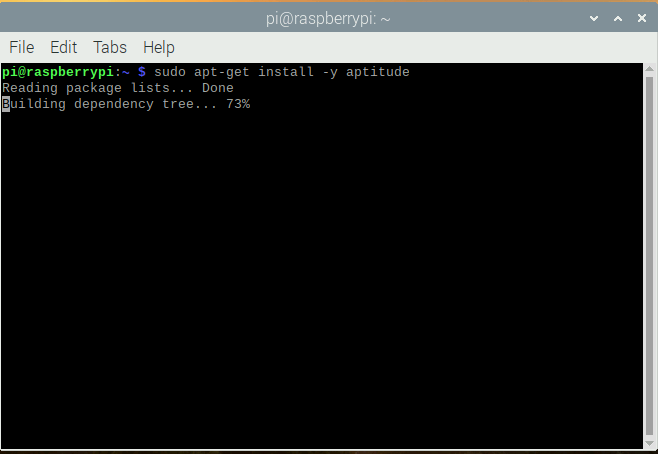
**sudo swapon --show**

or

**htop**

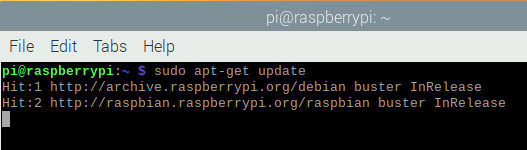
**Download Source Code and Compile Binaries of Zent Cash**

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**

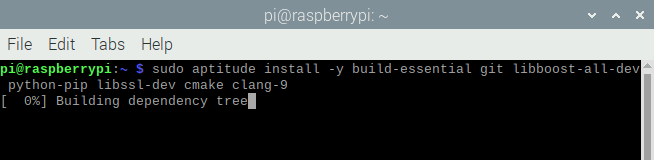
We enter the following command to download the package lists from the repositories and get information about the latest versions of the packages.

****

**sudo apt-get update**

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**

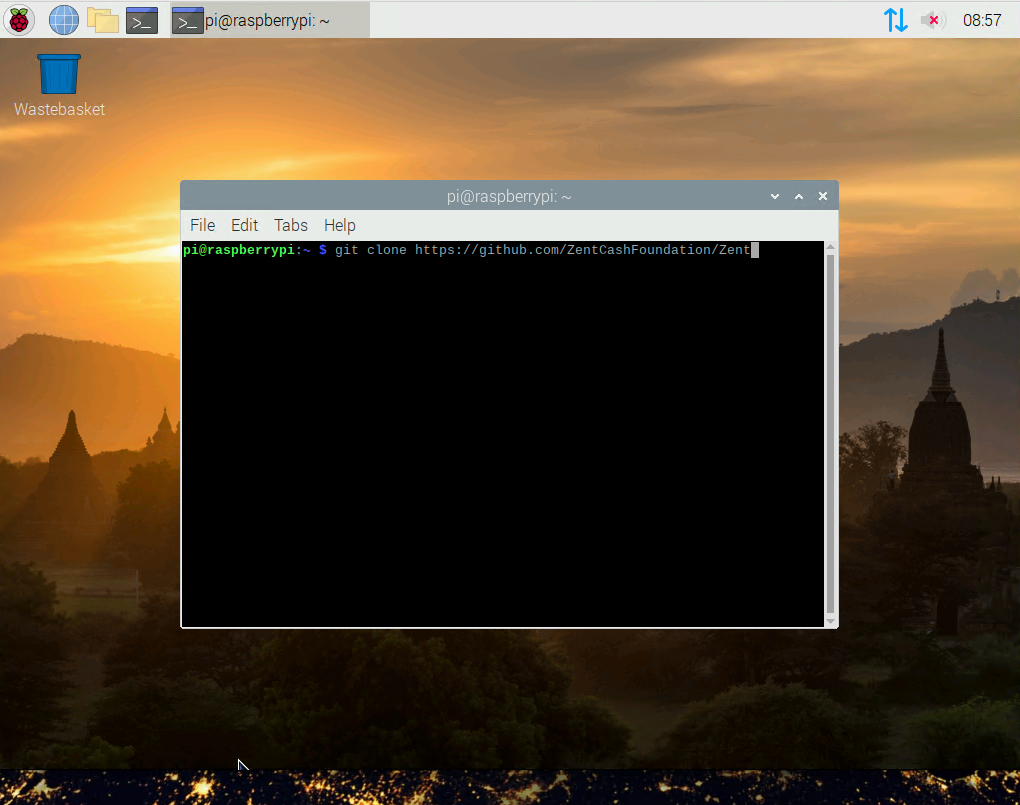
****

**DEPENDENCIES**

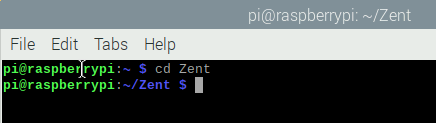
build-essential git libboost-all-dev python-pip libssl-dev cmake clang-9

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

**git clone** [**https://github.com/ZentCashFoundation/Zent**](https://github.com/ZentCashFoundation/Zent)

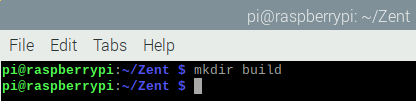


We access the folder that was generated with the previous command.

****

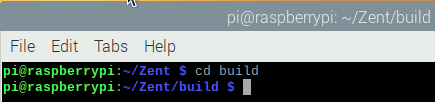
**cd Zent**

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

****

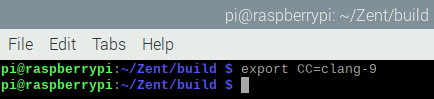
**mkdir build**

We access the directory build.

****

**cd build**

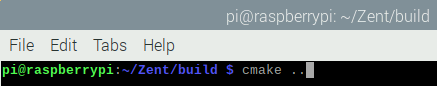
Before starting to compile the code we must assign the environment variables CC (C) and CXX (C++).

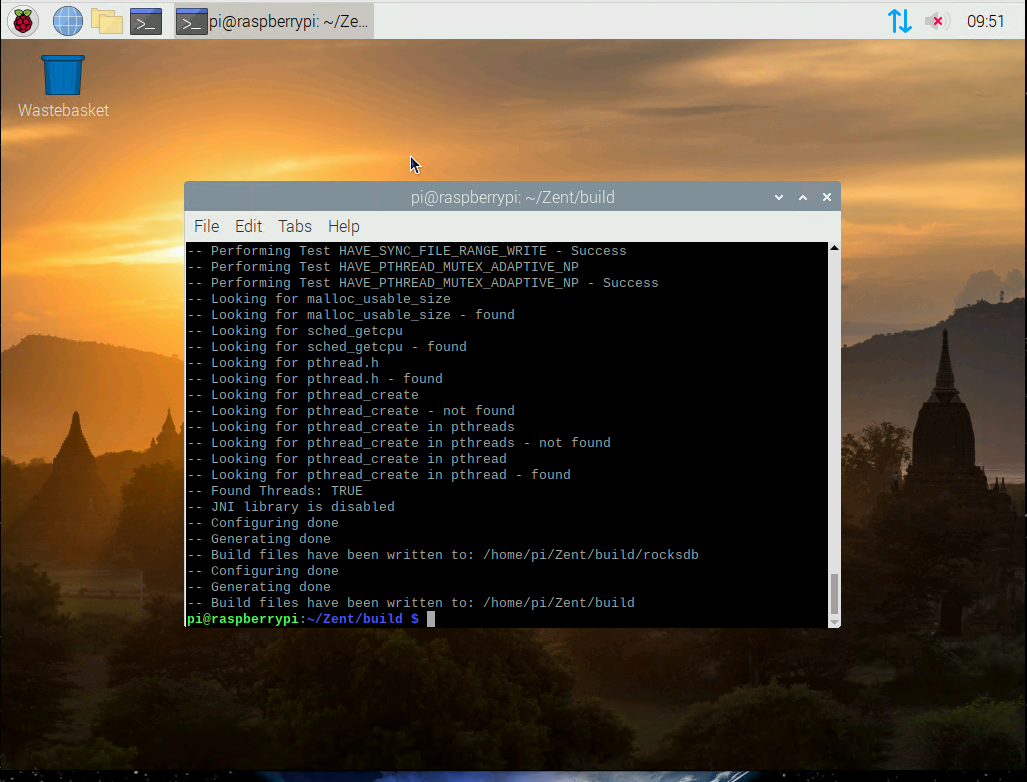
****We assign CC:

**export CC=clang-9**

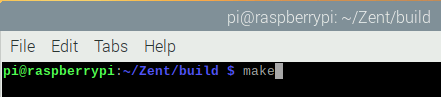
****We assign CXX:

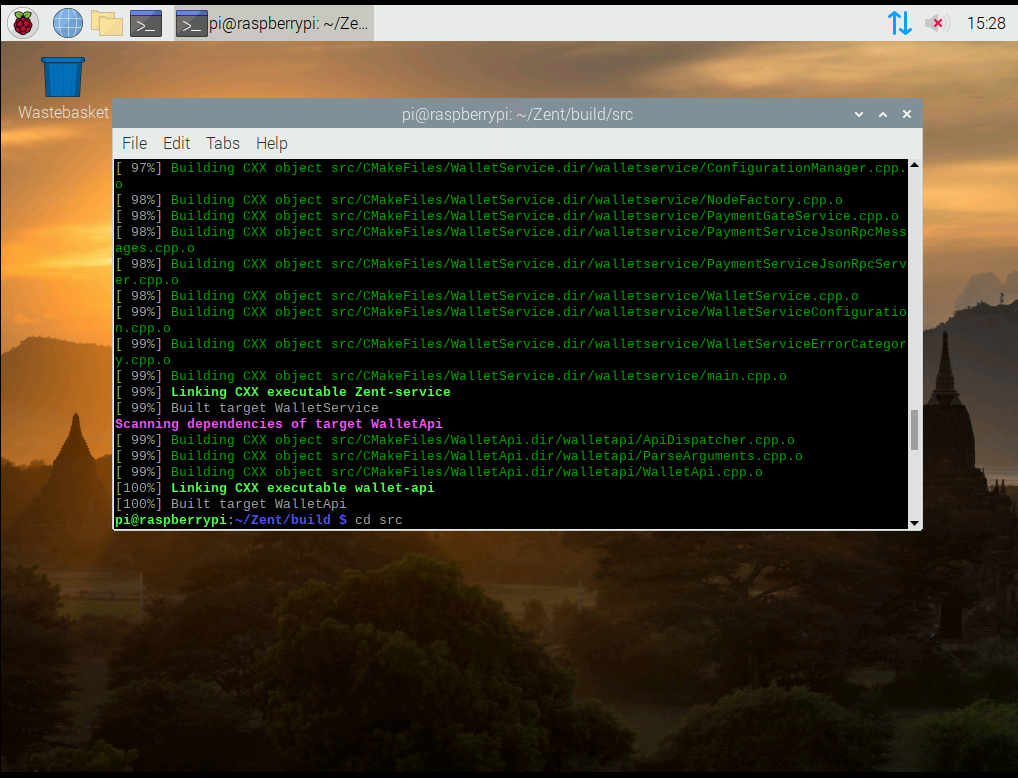
**export CXX=clang++-9**

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

**cmake ..**

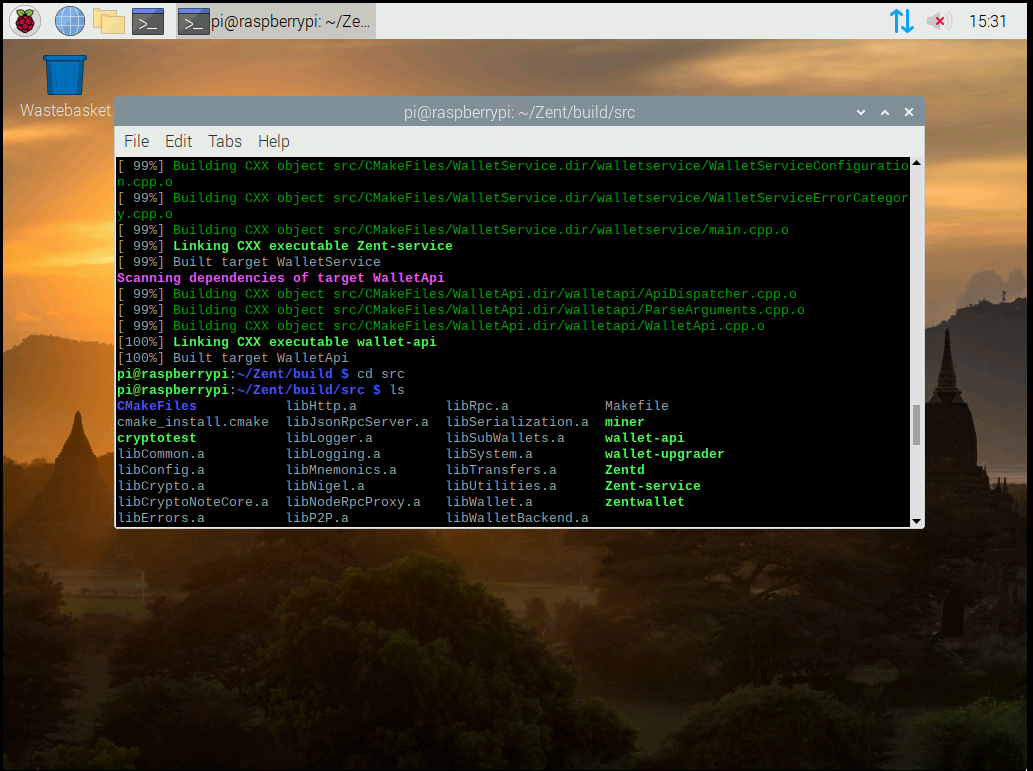
We type the following command to start the source code compilation process.

****



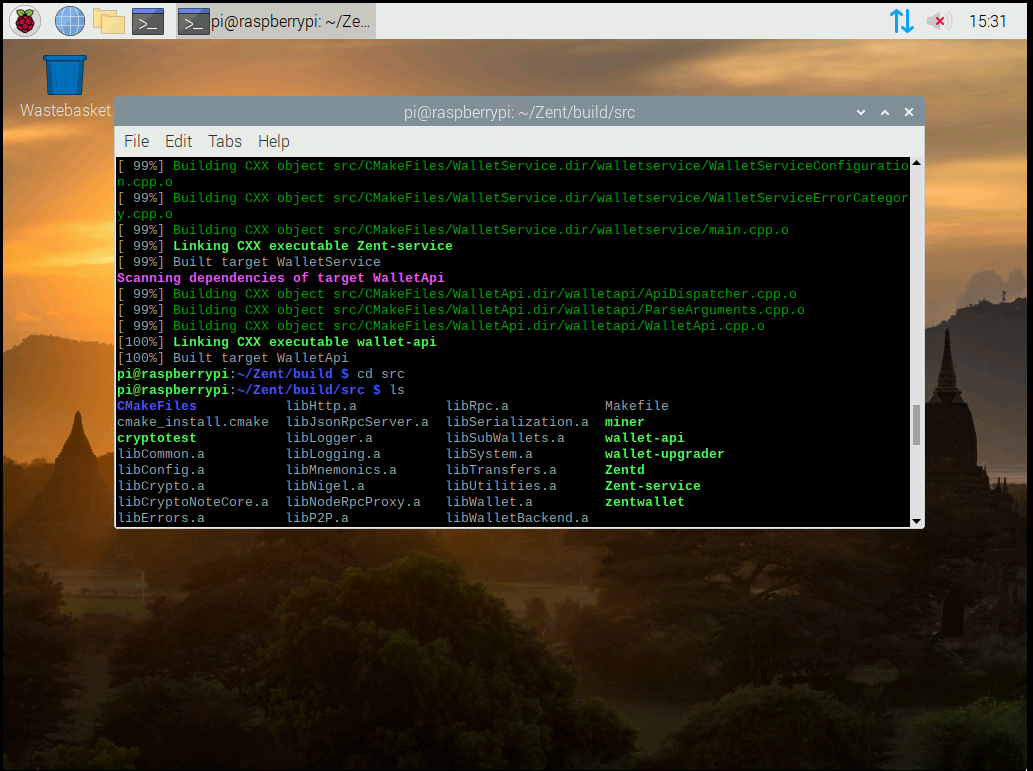
**make**

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



**cd src**

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



**ls**

**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.

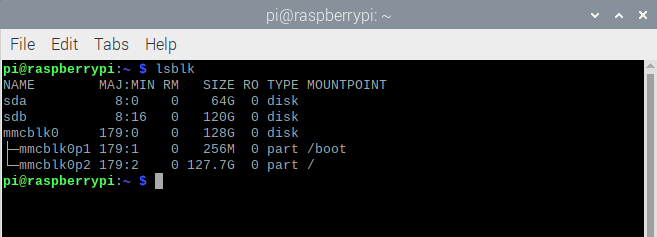
**miner =** It is used to mine the currency with your CPU.

**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.

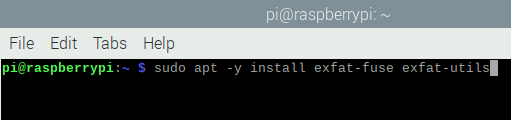
First, run this command to view your available disks.



**lsblk**

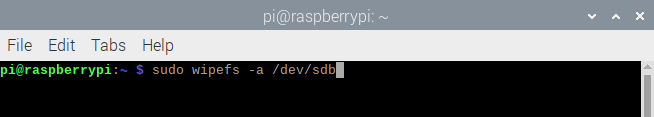
Look at the options and take note of which one is your SSD. It should say something about "Disk" and have the correct size as your drive. For me, it was located at /dev/sdb**.**

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.

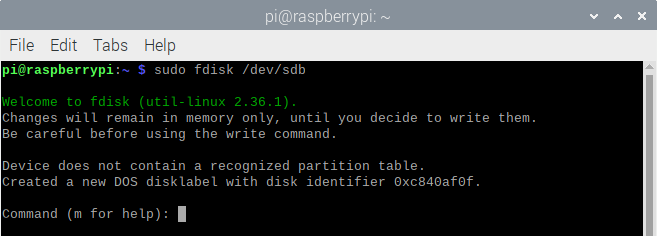


**sudo apt install exfat-fuse exfat-utils**

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.

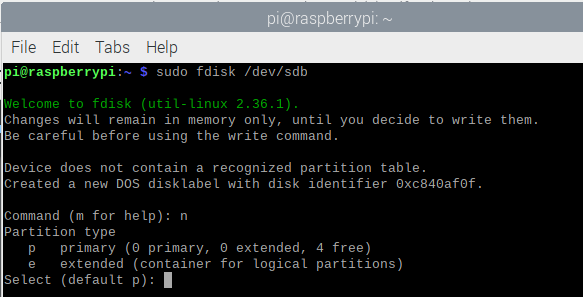


**sudo wipefs -a /dev/sdb**

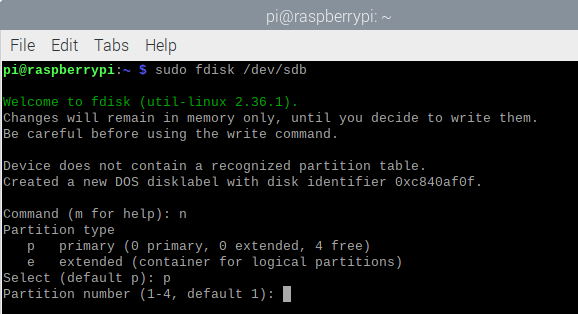
Next, you’ll want to type **sudo fdisk /dev/sdb** and push enter so you can make a new partition table. Please make sure you’re working with the right device file to avoid mucking anything up.

**sudo fdisk /dev/sdb**

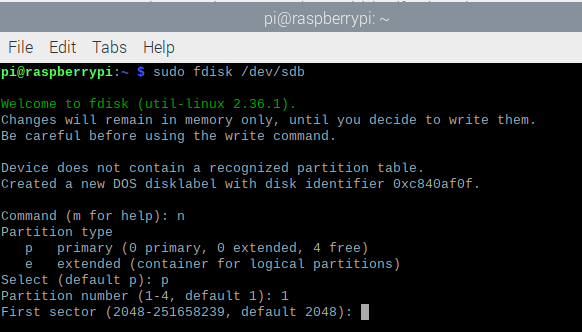
Now type the letter **n** and push **enter**.

****

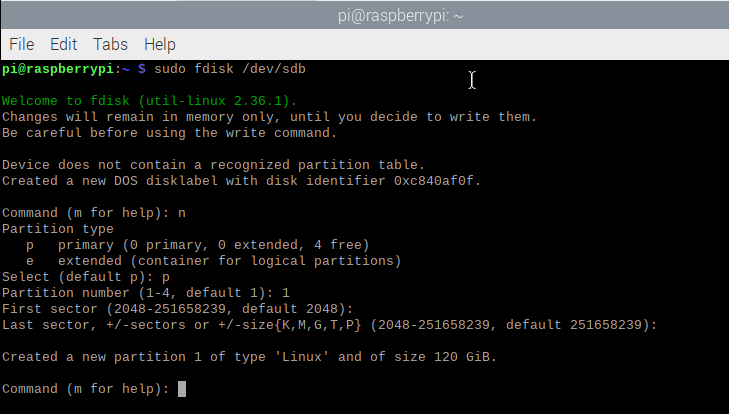
Now type the letter **p** to create a primary partition and press **enter**.

****

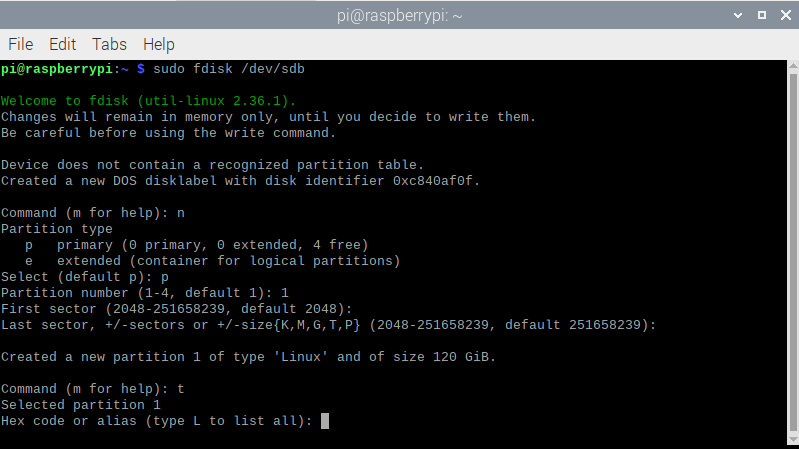
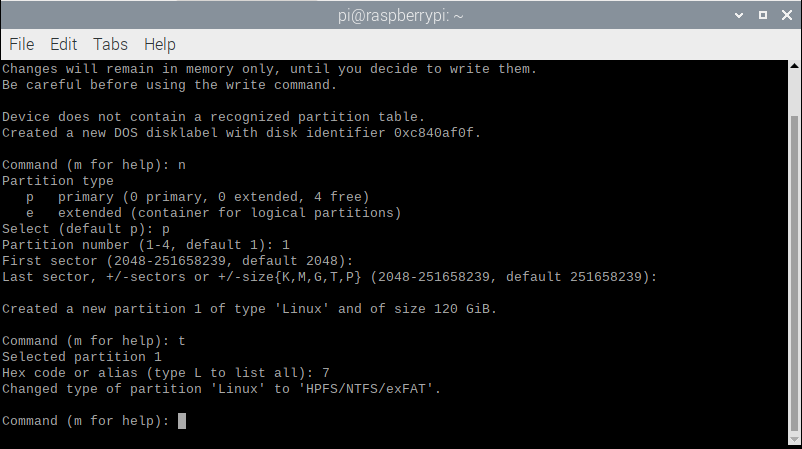
Now type the number **1** and push **enter**.

****

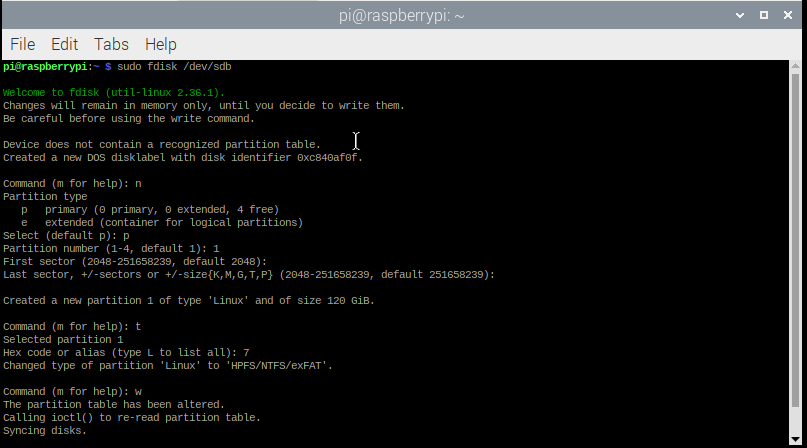
Now push enter and enter**.**



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.

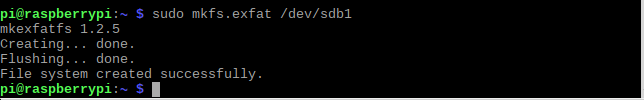


Finally, you simply have to push the **w** key.



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**.

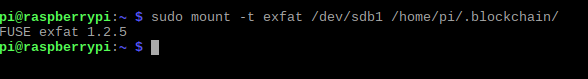
****

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

****

**cd ~**

**mkdir .blockchain**

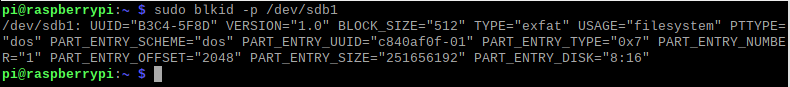
**sudo mount -t exfat /dev/sdb1 /home/pi/.blockchain/**

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**

****

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:

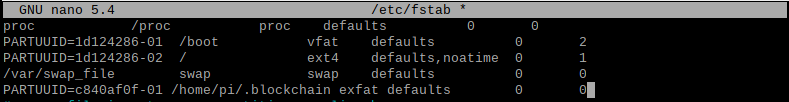
****

**sudo nano /etc/fstab**

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

****

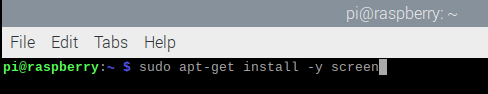
After editing the file we save the changes by pressing **Ctrl + O**, then press **Enter** and finally **Ctrl + X** to close the file.

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



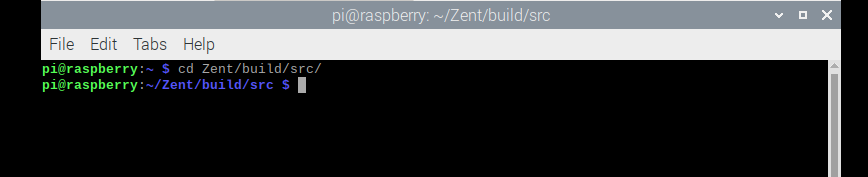
**Run Zentd and Keep it Running**

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**

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

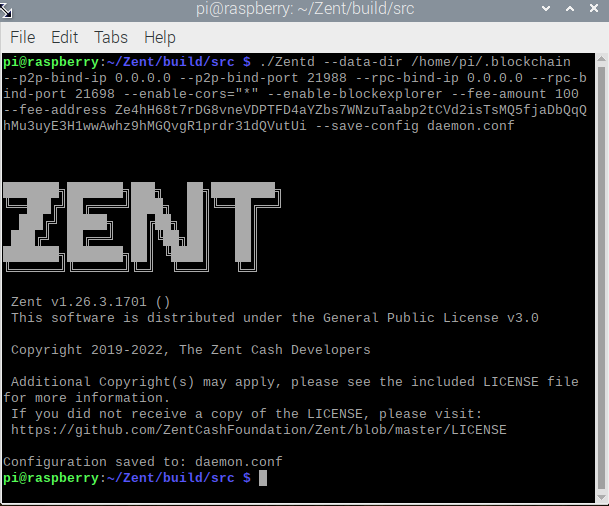


**cd Zent/build/src/**

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.

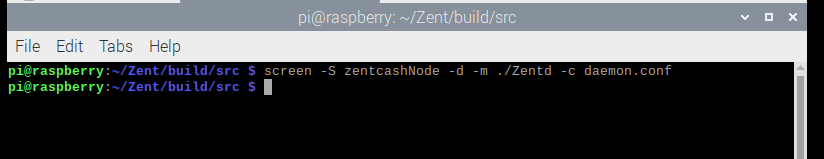
**./Zentd --data-dir /home/pi/.blockchain --p2p-bind-ip 0.0.0.0 --p2p-bind-port 21988 --rpc-bind-ip 0.0.0.0 --rpc-bind-port 21698 --enable-cors=”\*” --enable-blockexplorer --fee-amount 100 --fee-address Ze4hH68t7rDG8vneVDPTFD4aYZbs7WNzuTaabp2tCVd2isTsMQ5fjaDbQqQhMu3uyE3H1wwAwhz9hMGQvgR1prdr31dQVutUi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

****

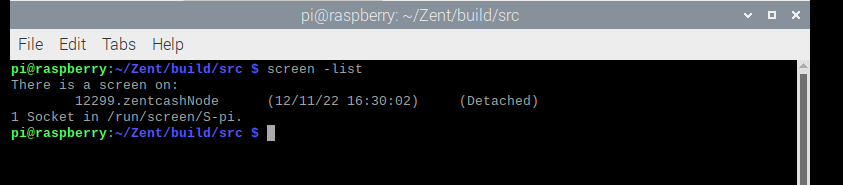
Now we only have to run the daemon with screen to be able to keep it started without having to keep our command console open.

To do this, we write the following command:

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

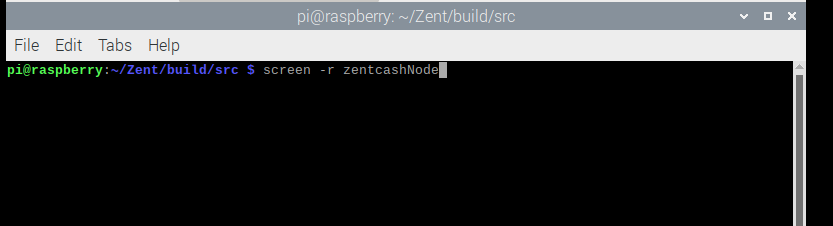
****

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

****

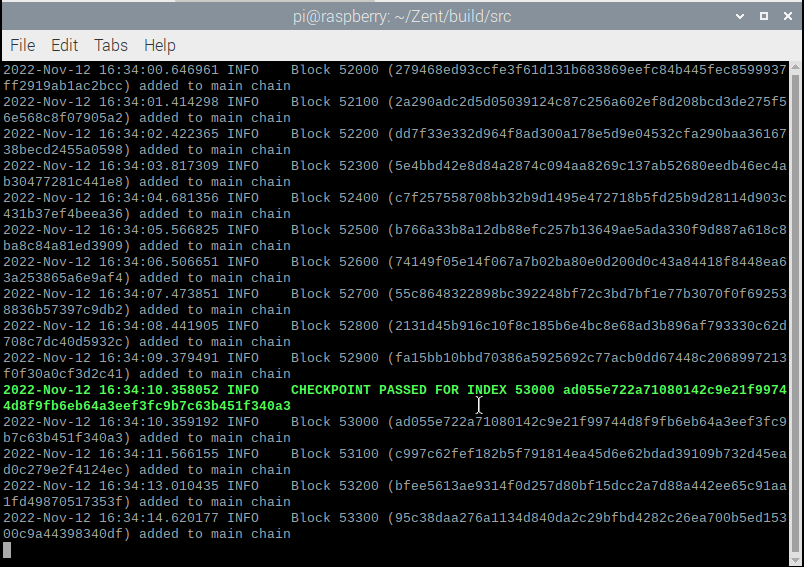
**screen -list**

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

****

**screen -r zentcashNode**

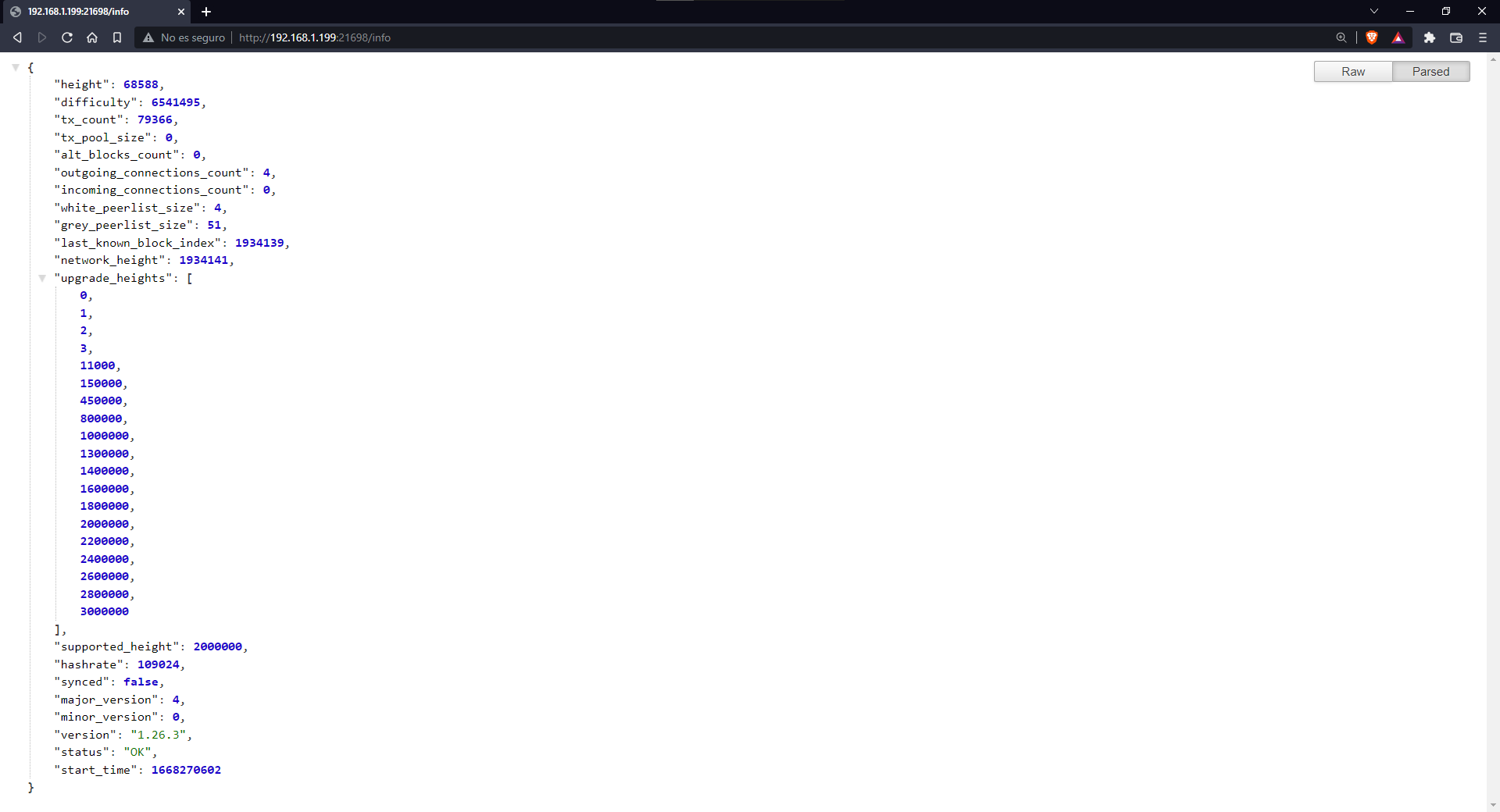
As we can see the node is already synchronizing with the network.

****

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**

****

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.

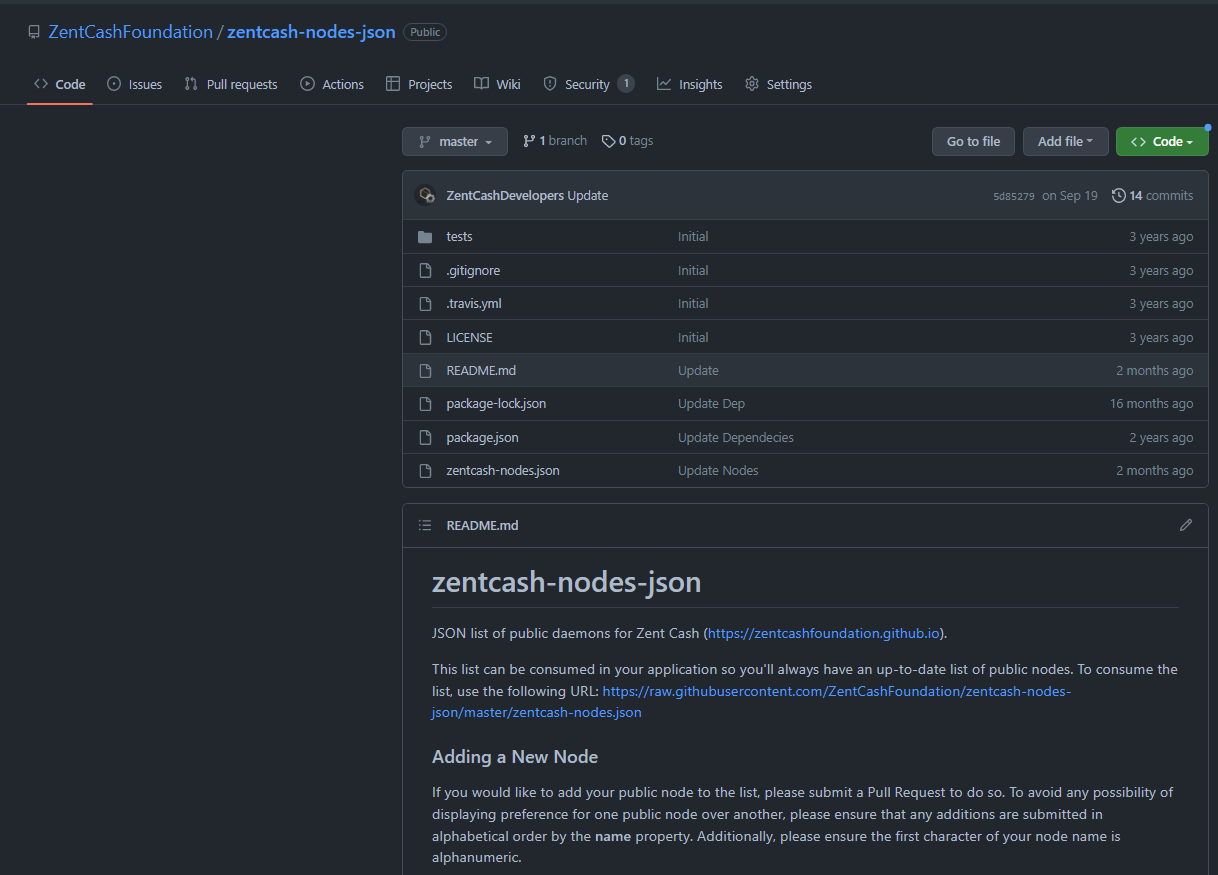


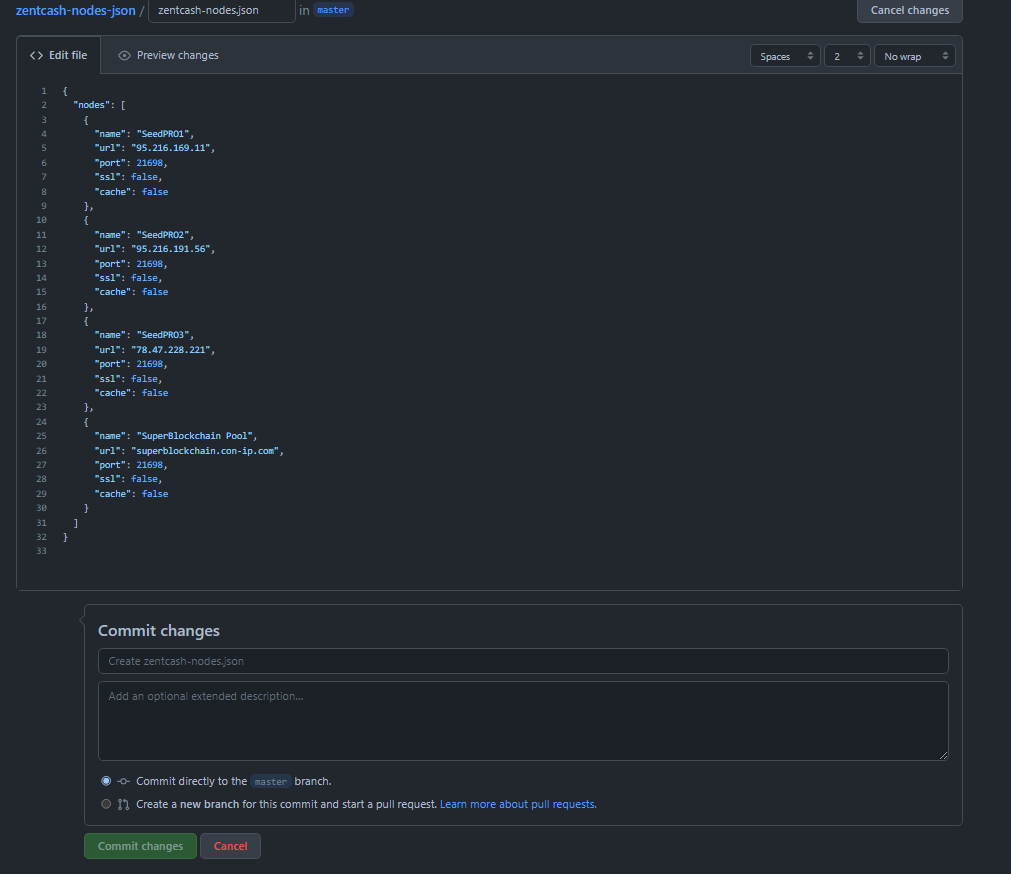
Finally, if you want to have your node available so that anyone can use it, you must allow TCP traffic through ports 21688 and 21698 in your firewall and router.

**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**](https://github.com/ZentCashFoundation/zentcash-nodes-json)

****

****

You can also promote your node through our social networks.

[**Discord**](https://discord.gg/tfaUE2G)[**Reddit**](https://www.reddit.com/r/ZentCashFoundation) [**Telegram**](https://t.me/zentcash_eng)[**Facebook**](https://www.facebook.com/Zent-Cash-Foundation-108069958362688)[**Twitter**](https://twitter.com/ZentCash)[**Youtube**](https://www.youtube.com/channel/UCRF0KXM-0UbovyGLpusYjVA?sub_confirmation=1)[**Odysee**](https://odysee.com/@ZentCash:7)