**OS Setup for Raspberry Pi**

1. **Node Setup on Raspberry Pi:**

Prebuilt OS for Raspberry Pi3 is available @ <https://gitlabee.dt.renault.com/swlabs/blockchain/poo-benchmarking>

If we need to build a custom OS with specific focus on Ethereum / Multichain Node we can follow the instruction below.

1. **Ethereum Node:**
2. **Geth Node:**
3. Get the required arm 64 bit OS from : <https://ubuntu-mate.org/raspberry-pi/ubuntu-mate-18.04.2-beta1-desktop-armhf+raspi-ext4.img.xz>
4. For Ethereum Node, only the geth binary is required to be present for it to function as a Node which can be obtained from: (No version constraint is present, only Hardware constraint of Arm64 version is present)

<https://geth.ethereum.org/downloads/>

1. Binary can be un tared and can then be utilized.

**ii) Parity Node:**

1. Although the Parity Node is not available out of the box we have managed to cross-compile using Docker containers for Arm64.
2. Binaries as well as the instructions are available at**:**

<https://gitlabee.dt.renault.com/swlabs/blockchain/poo-benchmarking/tree/master/Raspberry_build_OS/Parity>

1. **Multichain Node:**
2. Multichain like Parity the binary is not available for Arm64 and we have managed to build.
3. Multichain binary 2.0.1 version is already compiled according to the instructions given in <https://gitlabee.dt.renault.com/swlabs/blockchain/multichain-rpi/tree/master/multichain_raspberrypi3>
4. This Binary along with Raspberry Pre-Built OS is also Present in <https://gitlabee.dt.renault.com/swlabs/blockchain/poo-benchmarking>
5. **SSH Configuration:**

sudo systemctl enable ssh

sudo sytemctl start ssh

1. **To transfer the public key:**

ssh-copy-id username@IP

1. **Swap Space Configuration:**

It is better to allocate swap space of 2GB since we have a limited RAM of 1GB on Raspberry Pi Model 3B +

Create the Disk Space:

sudo dd if=/dev/zero of=/swap\_2 bs=1024 count=2000000

sudo chmod 600 /swap\_2

sudo mkswap /swap\_2

sudo swapon /swap\_2

Add the swap permanently into fstab

**echo '/swap\_2 none swap sw 0 0' | sudo tee -a /etc/fstab**

1. **Proxy Configuration:**

**Proxy can be used with Squid or the with cosmos**

**I prefer to use cosmos since the time servers don’t sync well.**

*proxy=http://pxxxxx: password @cosmos-vip.intra.renault.fr:3128/*

*https-proxy=http:// pxxxxx:password@cosmos-vip.intra.renault.fr:3128/*

*http-proxy=http:// pxxxxx: password @cosmos-vip.intra.renault.fr:3128/*

*http\_proxy=http:// pxxxxx: password @cosmos-vip.intra.renault.fr:3128/*

*https\_proxy=http:// pxxxxx: password @cosmos-vip.intra.renault.fr:3128/*

*HTTP\_PROXY=http:// pxxxxx: password @cosmos-vip.intra.renault.fr:3128/*

*HTTPS\_PROXY=http:// pxxxxx: password @cosmos-vip.intra.renault.fr:3128*

Also, instead of cosmos you can use, [http://squid.sop.renault.fr:911](http://squid.sop.renault.fr:911/)

Also corresponding changes can be made in **apt/apt.conf if incase there are problem with package updation**

1. **Library Configuration for Multichain 2.0.1:**

**Following operations need to be performed:**

sudo apt-get update

sudo apt-get install -y build-essential

sudo apt-get install -y software-properties-common

sudo apt-get install libdb++-dev

sudo apt-get install libboost-all-dev

sudo apt-get install libssl-dev

sudo apt-get install libc++-dev

***In case of any problem also install:***

sudo apt-get install gcc-arm\*

1. ***Then the image can be saved to another disk and replicated across machines.***
2. ***To Run the middleware:***

**Install NodeJs and Angular:**

1. ***sudo apt-get install nodejs***

***sudo apt-get install npm***

Node Is installed

***Update the Node version:***

1. ***Sudo npm install -g n***
2. ***sudo n stable***

**Node is updated**

1. ***npm install -g @angular/cli***

***Also we need to add proxy configuration in ~/.npmrc for nodejs app to communicate with any external API:***

Location of npmrc file can be found by **npm config list**

proxy=http://pxxxx: password @cosmos-vip.intra.renault.fr:3128/

https-proxy=http:// pxxxx:password@cosmos-vip.intra.renault.fr:3128/

http-proxy=http:// pxxxx: password @cosmos-vip.intra.renault.fr:3128/

http\_proxy=http:// pxxxx: password @cosmos-vip.intra.renault.fr:3128/

https\_proxy=http:// pxxxx: password @cosmos-vip.intra.renault.fr:3128/

HTTP\_PROXY=http:// pxxxx: password @cosmos-vip.intra.renault.fr:3128/

HTTPS\_PROXY=http:// pxxxx: password @cosmos-vip.intra.renault.fr:3128/

***We are done with all the configurations and we are ready to run the node on Raspberry Pi.***