Setting up a web server load balancer on a computer cluster

A draft tutorial by Konstantinos Delistavrou

April 2018

We need at least 3 nodes to have a cluster.

We will use 2 "real" Raspberry Pis and a "virtual" Raspberry Pi as a virtual machine on the laptop.

Part 1: Setup virtual machine

Now, let's setup the virtual machine.

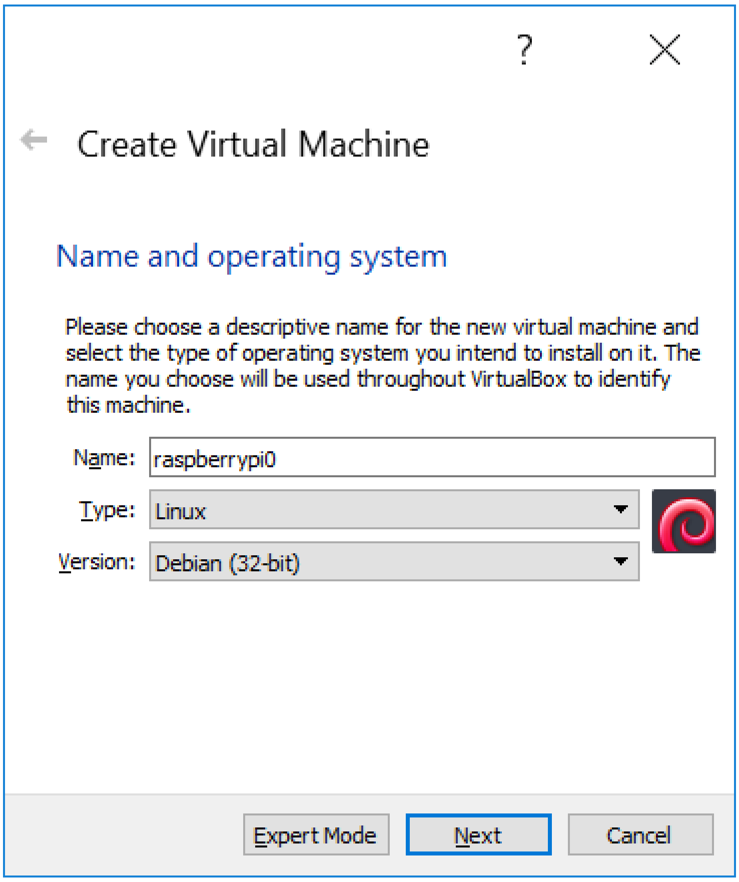
Go to https://www.virtualbox.org/wiki/Downloads and download VirtualBox for Windows hosts.

Install it.

Go to https://www.raspberrypi.org/downloads/raspberry-pi-desktop/ and download "The Raspberry Pi Desktop OS for PC and Mac - based on Debian Stretch".

On VirtualBox create a virtual machine (button New).

Then create a virtual machine like in the pictures.



Create Virtual Machine

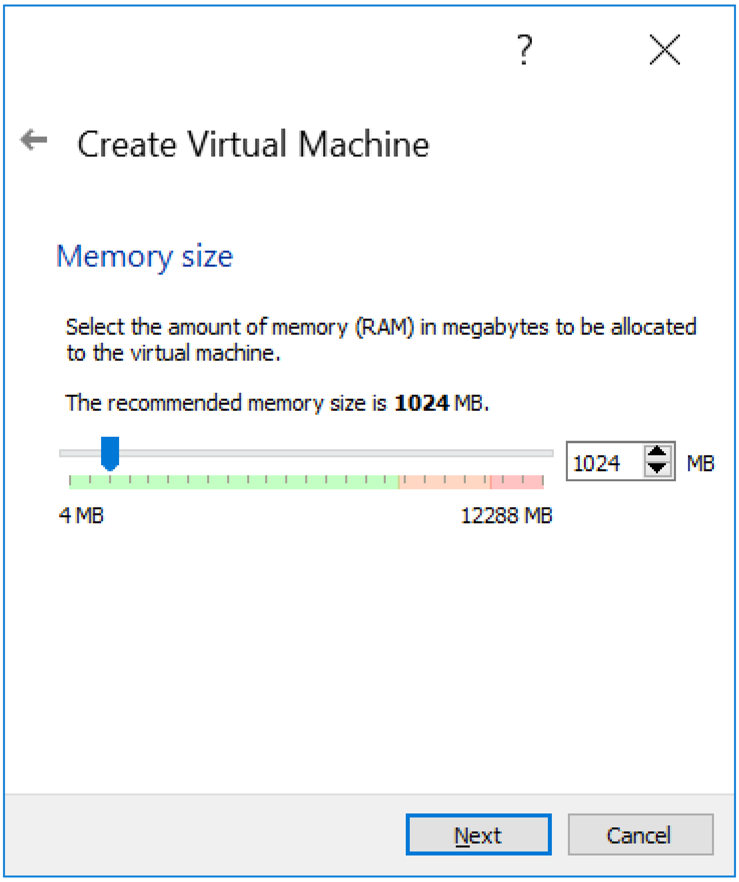
Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: raspberrypi0

Type: Linux

Version: Debian (32-bit)



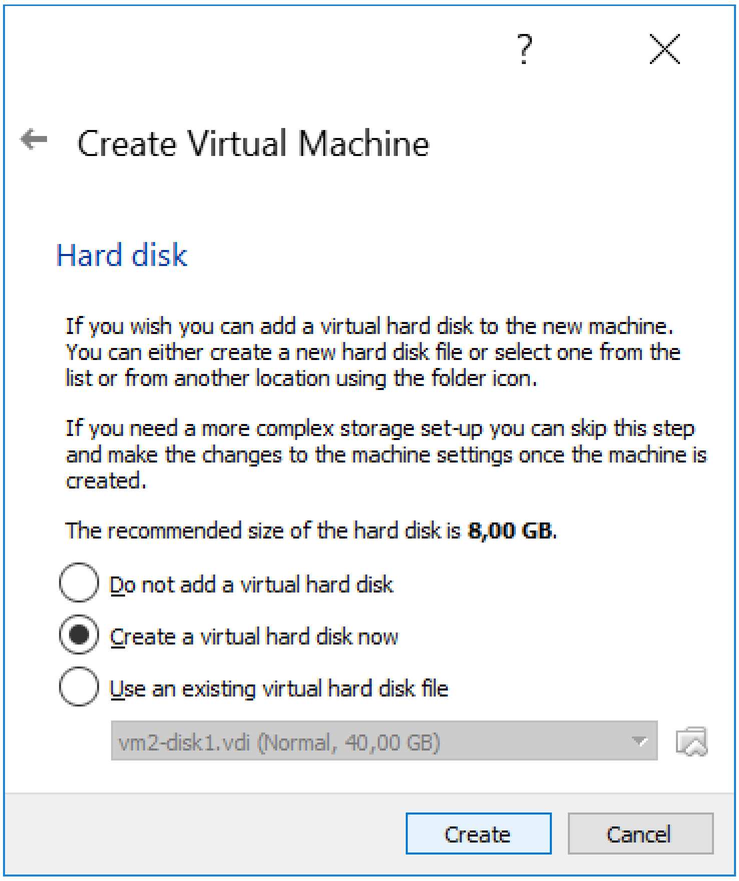
Create Virtual Machine

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

1024 MB



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

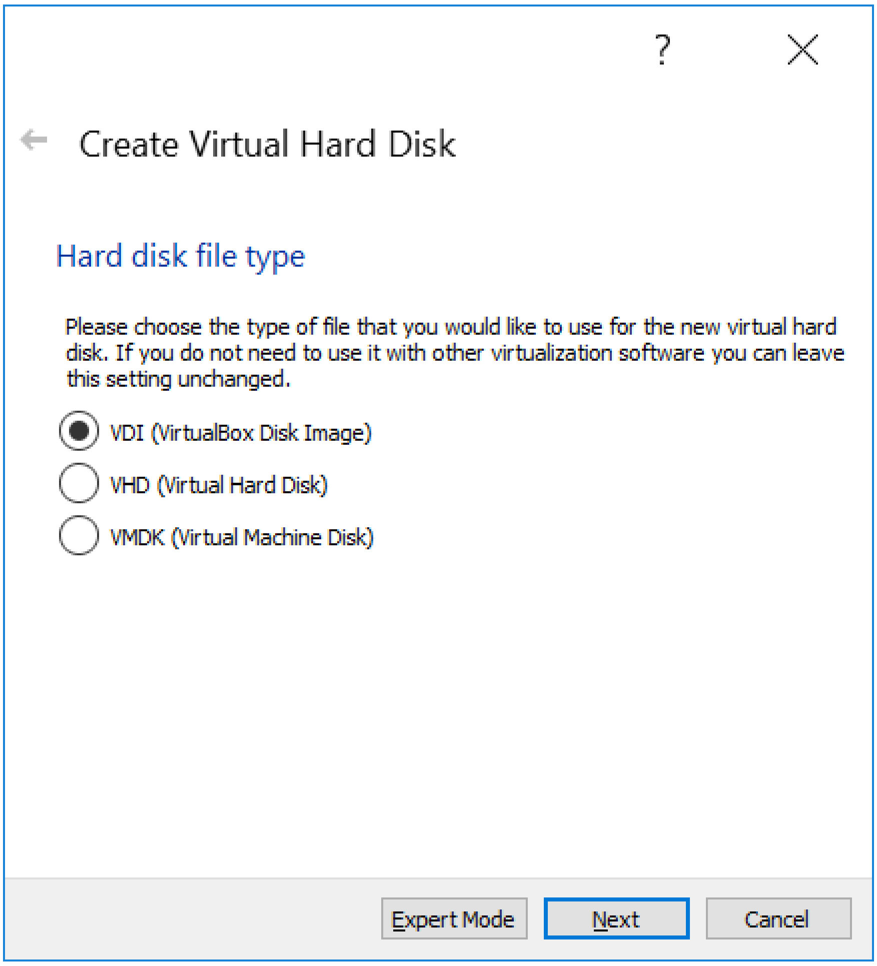
The recommended size of the hard disk is **8,00 GB**.

○ Do not add a virtual hard disk

● Create a virtual hard disk now

○ Use an existing virtual hard disk file

vm2-disk1.vdi (Normal, 40,00 GB)



Create Virtual Hard Disk

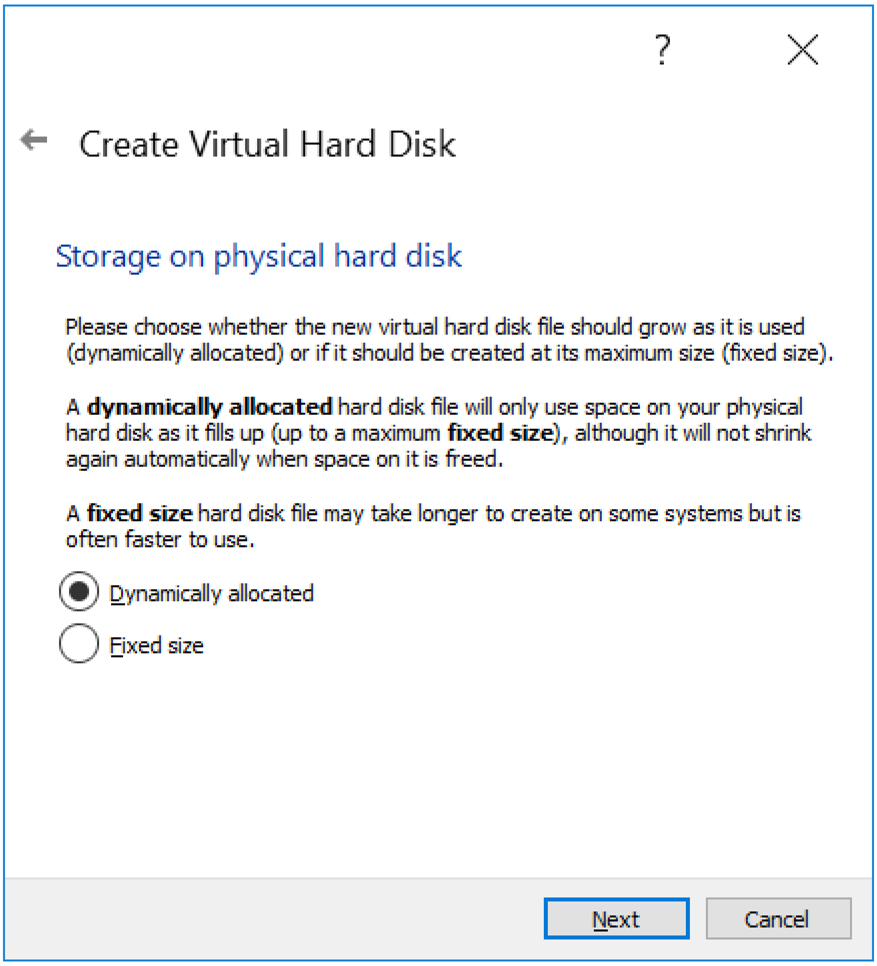
Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

● VDI (VirtualBox Disk Image)

○ VHD (Virtual Hard Disk)

○ VMDK (Virtual Machine Disk)



Create Virtual Hard Disk

Storage on physical hard disk

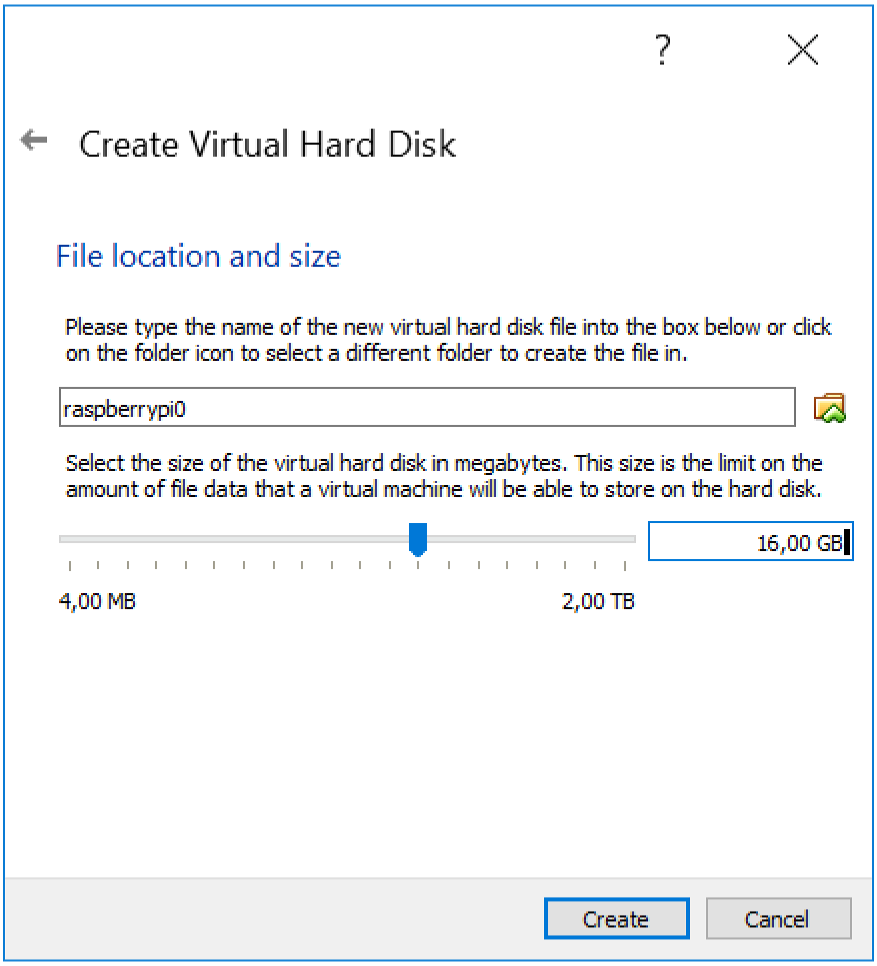
Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

● Dynamically allocated

○ Fixed size



Create Virtual Hard Disk

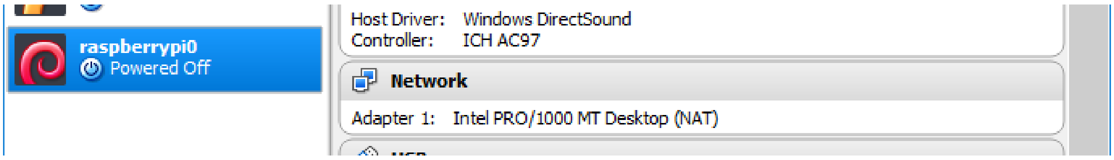
File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

raspberrypi0

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

16,00 GB



raspberrypi0
Powered Off

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

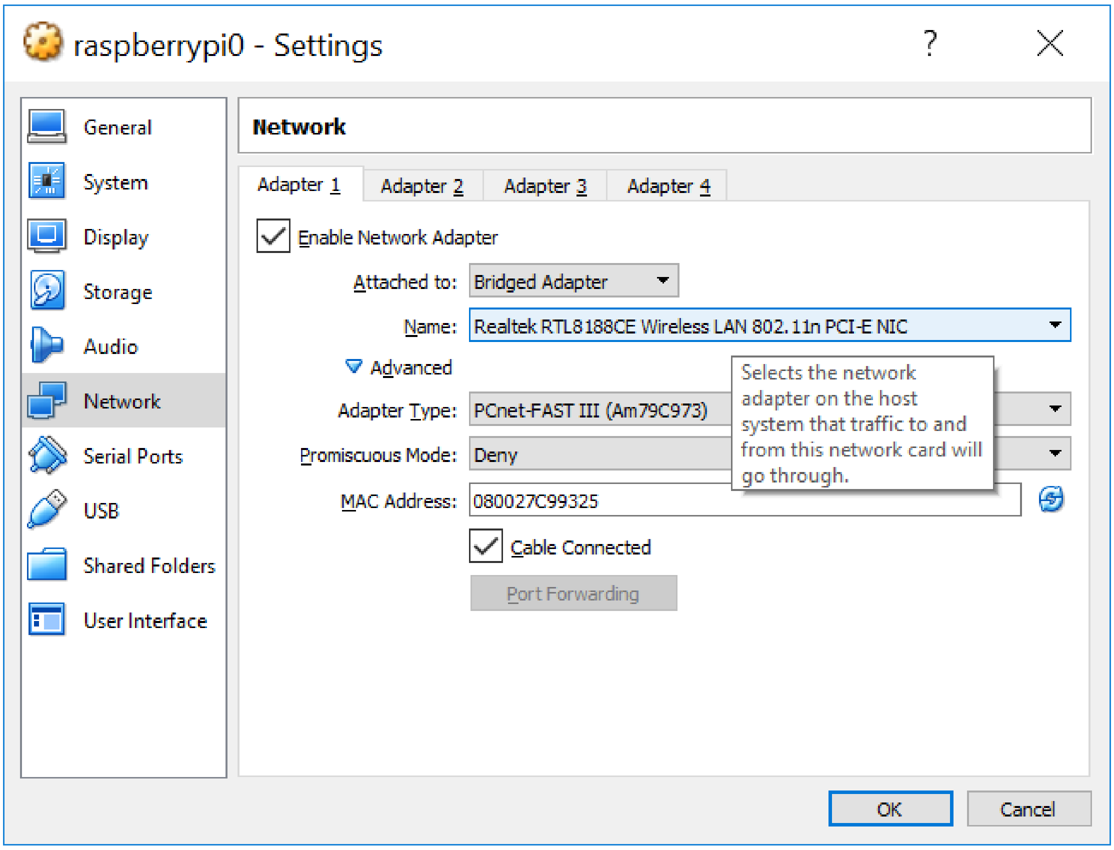
Right click the raspberrypi0 virtual machine and select Settings.

Go to Network tab.

Select Attached to Bridged Adapter.

On Name select the laptop’s Wireless network interface (This is important!).

On Adapter choose PCnet-Fast III.



raspberrypi0 - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Realtek RTL8188CE Wireless LAN 802.11n PCI-E NIC

Advanced

Adapter Type: PCnet-FAST III (Am79C973)

Promiscuous Mode: Deny

MAC Address: 080027C99325

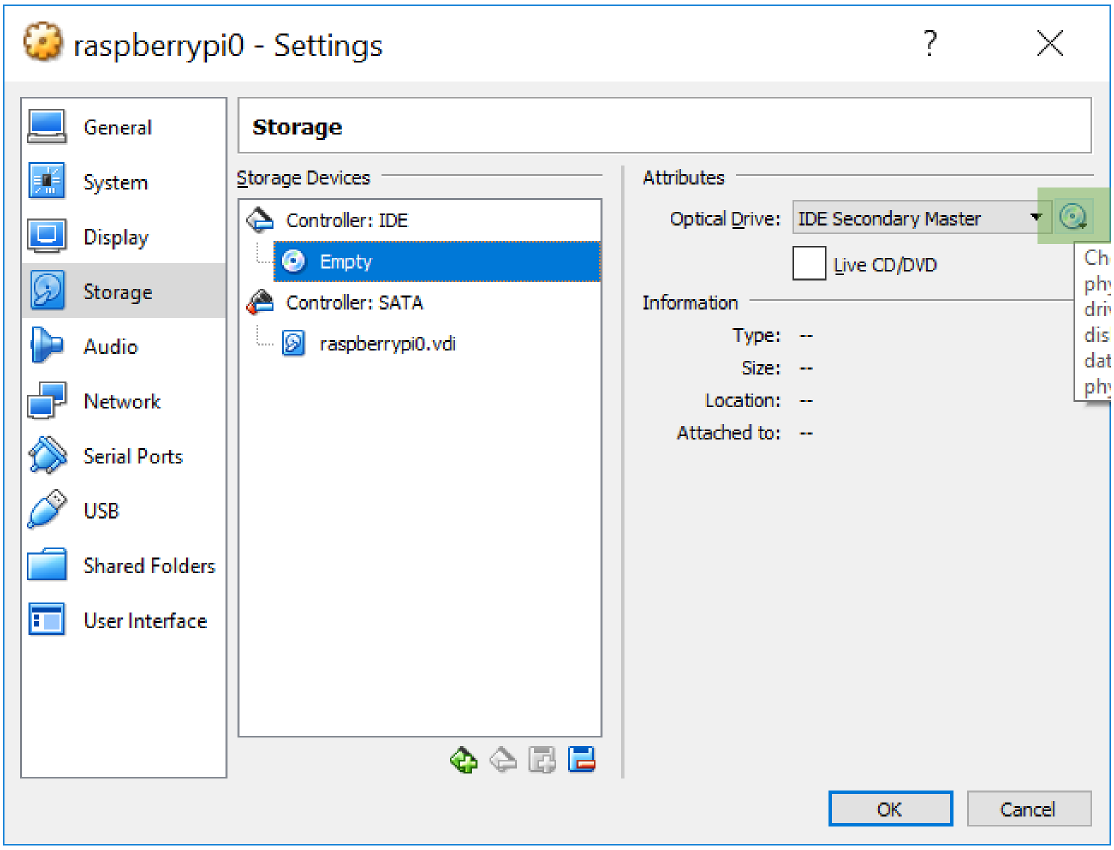
☑ Cable Connected

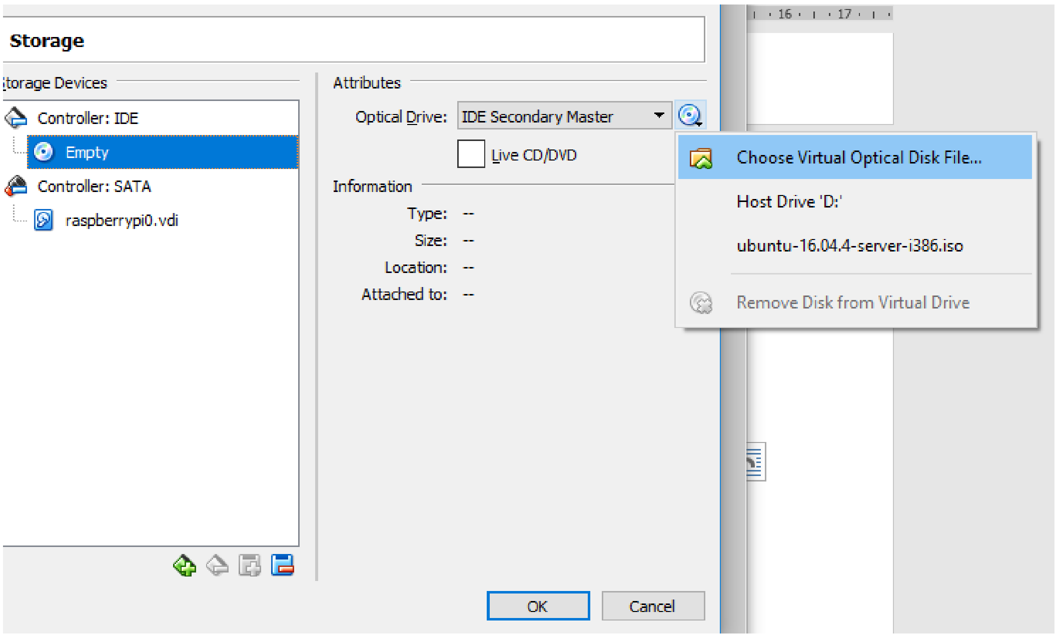
Selects the network adapter on the host system that traffic to and from this network card will go through.

Go to Storage tab.

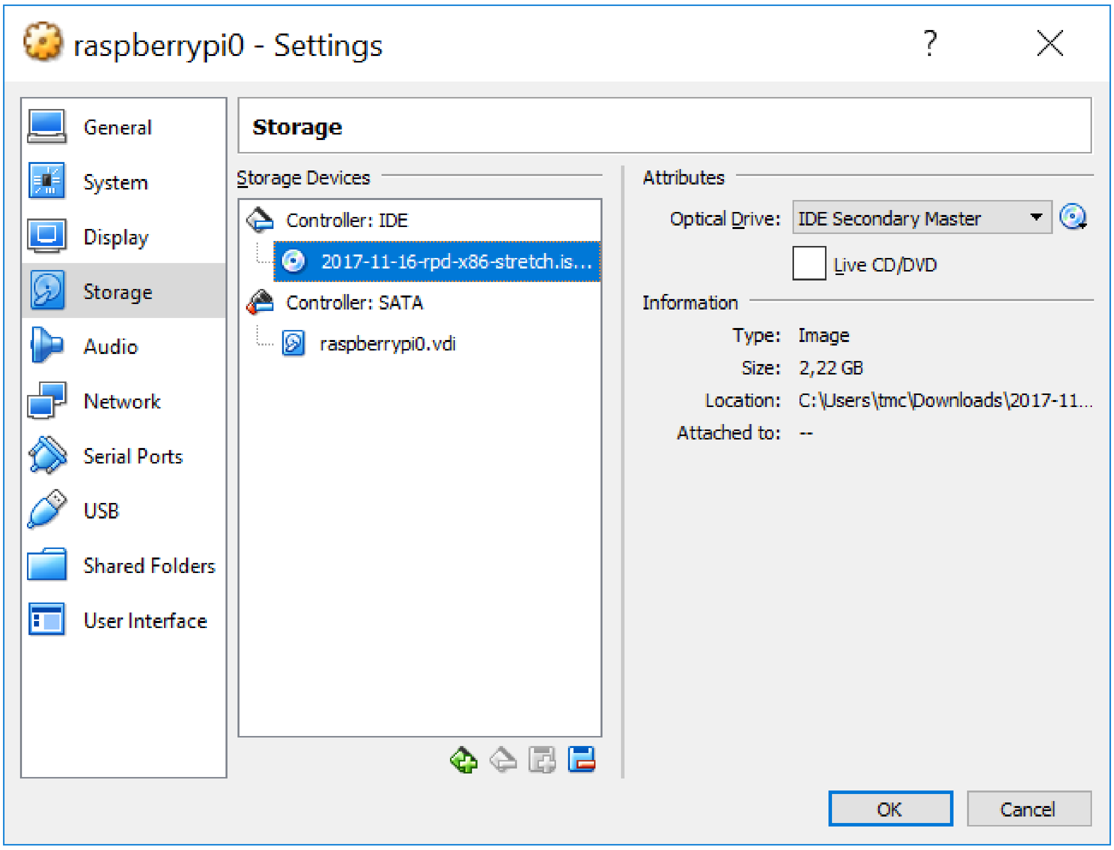
Select Empty disk and on the right panel click the disk icon.

Select Choose Virtual Optical Disk File.





Select the image “2017-11-16-rpd-x86-stretch.iso” downloaded earlier.



Storage

Storage Devices

Controller: IDE
2017-11-16-rpd-x86-stretch.is...
Controller: SATA
raspberrypi0.vdi

Attributes

Optical Drive: IDE Secondary Master

□ Live CD/DVD

Information

Type: Image
Size: 2,22 GB
Location: C:\Users\tmc\Downloads\2017-11...
Attached to: --

Press OK.

Start the virtual machine.

Choose Graphical Install.

Complete the installation using the default options (When asked to write changes to disk, and to write GRUB, answer Yes!).

See an example in this tutorial https://www.techrepublic.com/article/how-to-run-raspberry-pis-raspbian-os-on-a-pc/

At the end you will have a Raspberry Pi as a virtual machine.

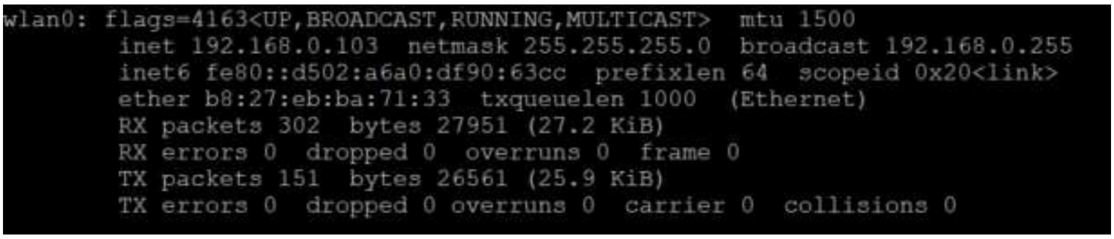
Part 2: Networking

Connect 1st Raspberry Pi on the HDMI monitor, keyboard, mouse.

Power it up.

Open a command terminal.

Give **ifconfig** and write to a paper its IP address (mine is 192.168.0.103).



```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.103  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::d502:a6a0:df90:63cc  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:ba:71:33  txqueuelen 1000  (Ethernet)
        RX packets 302  bytes 27951 (27.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 151  bytes 26561 (25.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The default username is pi.

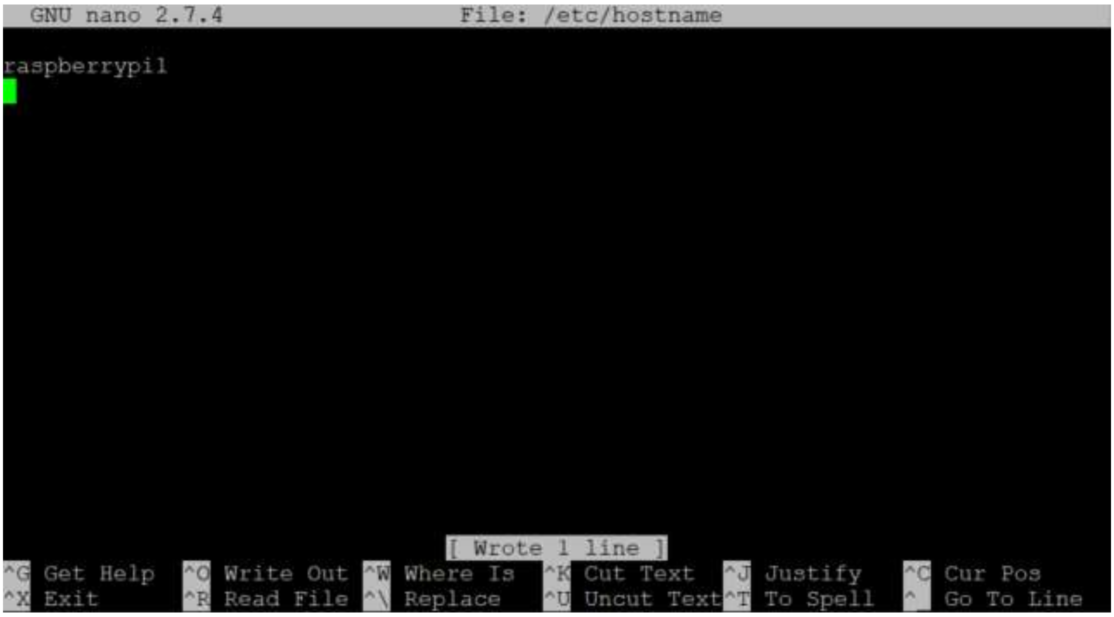
The default password is raspberry.

On the terminal type **sudo nano /etc/hostname**.

Change the name to **raspberrypi1**.

Press Ctrl + O to save.

Press Ctrl + X to exit nano.



```
  GNU nano 2.7.4                File: /etc/hostname

raspberrypi1

                              [ Wrote 1 line ]
^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
```

Type **sudo reboot** to reboot.

When it boots open a terminal again and type **ifconfig** to get its IP. Usually is the same.

Connect 2nd Raspberry Pi on the HDMI monitor, keyboard, mouse.

Do the same steps but name it raspberrypi2.

Start the raspberrypi0 virtual machine.

Do the same steps but name it raspberrypi0.

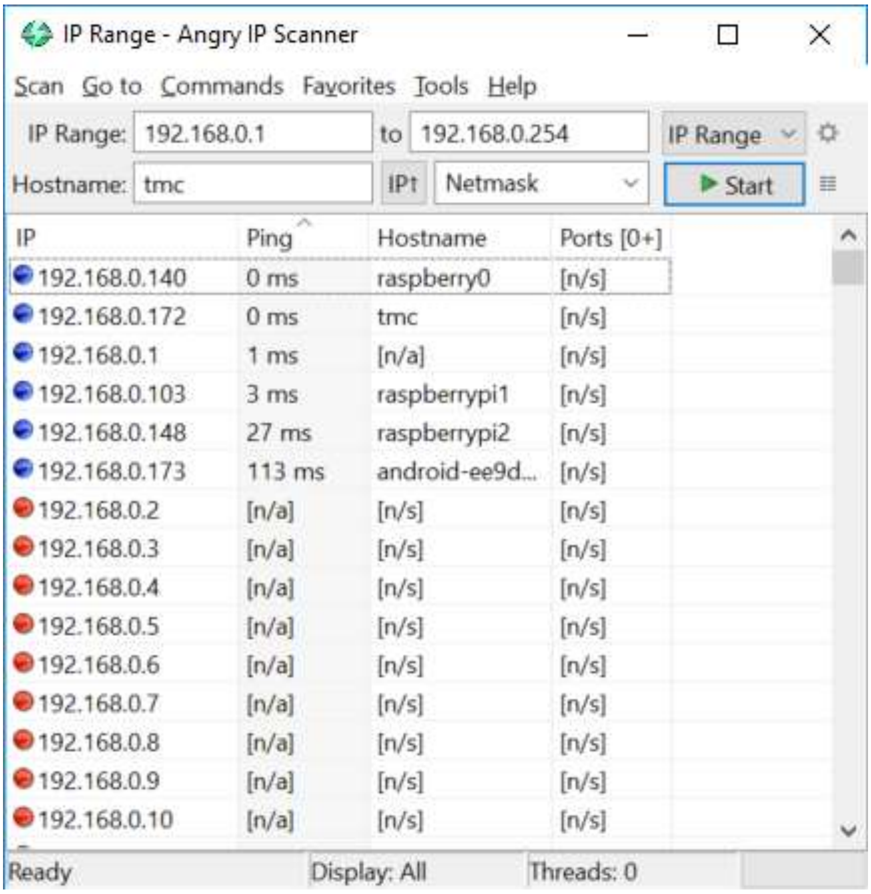
After reboot usually the Raspberry Pis don’t change IP address. If you are not lucky and they change, use Angry IP scanner to find their addresses.

Go to http://angryip.org/download/#windows, download and install it.

Scan the Wi-Fi LAN range (e.g. from 192.168.0.1 to 192.168.0.254) for all IP addresses (.1 or .254 is usually the router). Click on the Ping column and Sort by Ping.

You should see a list like the following. Mark Raspberry Pis addresses.

Do this every time you lose the IPs of the Raspberry Pis.


IP Range - Angry IP Scanner
Scan Go to Commands Favorites Tools Help
IP Range
IP↑
Start

IP Range: 192.168.0.1 to 192.168.0.254

Hostname: tmc

Netmask

IP	Ping	Hostname	Ports [0+]
192.168.0.140	0 ms	raspberry0	[n/s]
192.168.0.172	0 ms	tmc	[n/s]
192.168.0.1	1 ms	[n/a]	[n/s]
192.168.0.103	3 ms	raspberrypi1	[n/s]
192.168.0.148	27 ms	raspberrypi2	[n/s]
192.168.0.173	113 ms	android-ee9d...	[n/s]
192.168.0.2	[n/a]	[n/s]	[n/s]
192.168.0.3	[n/a]	[n/s]	[n/s]
192.168.0.4	[n/a]	[n/s]	[n/s]
192.168.0.5	[n/a]	[n/s]	[n/s]
192.168.0.6	[n/a]	[n/s]	[n/s]
192.168.0.7	[n/a]	[n/s]	[n/s]
192.168.0.8	[n/a]	[n/s]	[n/s]
192.168.0.9	[n/a]	[n/s]	[n/s]
192.168.0.10	[n/a]	[n/s]	[n/s]

Ready Display: All Threads: 0

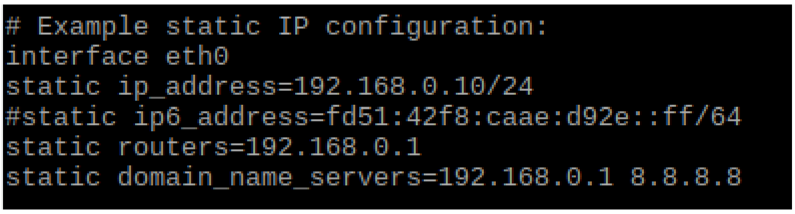
After these first steps you will have 2 Raspberry Pis connected to the Wi-Fi router, each one with a different IP.

Sometimes, after restart, the network interface doesn’t work.

In this case it is better to set a static IP address to the Raspberry Pi.

Edit the appropriate file by typing **`sudo nano /etc/dhcpcd.conf`**

Find the line “# Example static IP configuration:” and change the following lines as in the example of the next picture. Also remove the # character, so they are not a comment.



```
# Example static IP configuration:
interface eth0
static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8
```

Where …

- “static ip_address=” type a unique IP address for the Wi-Fi LAN subnet,
  prefer numbers after .200, like 192.168.0.201, etc.
- “static routers” is the address of the router.
- “static domain_name_servers” is the address of the router
  and 8.8.8.8 (Google’s public DNS server)

Leave the line #static ip6_address commented (don’t delete # in front of it).

Reboot the Raspberry Pi to apply changes by typing **`sudo reboot`**

Do this for each Raspberry Pi having problem, but each time with a different IP address (the router and domain name servers must be the same).

Part 3: Remote terminal

It is more convenient to work with a remote terminal, except if you have two HDMI monitors and mouse keyboard sets.

Otherwise you may enable SSH and connect to Raspberry Pis from the laptop.

In this case open a terminal on each Raspberry Pi you want to access remotely.

Type `sudo raspi-config`

Use the keyboard's arrow keys, TAB and Enter to browse the menu.

Select 5 Interfacing Options.

Select P2 SSH.

Select <Yes> to the question about enabling the SSH server.

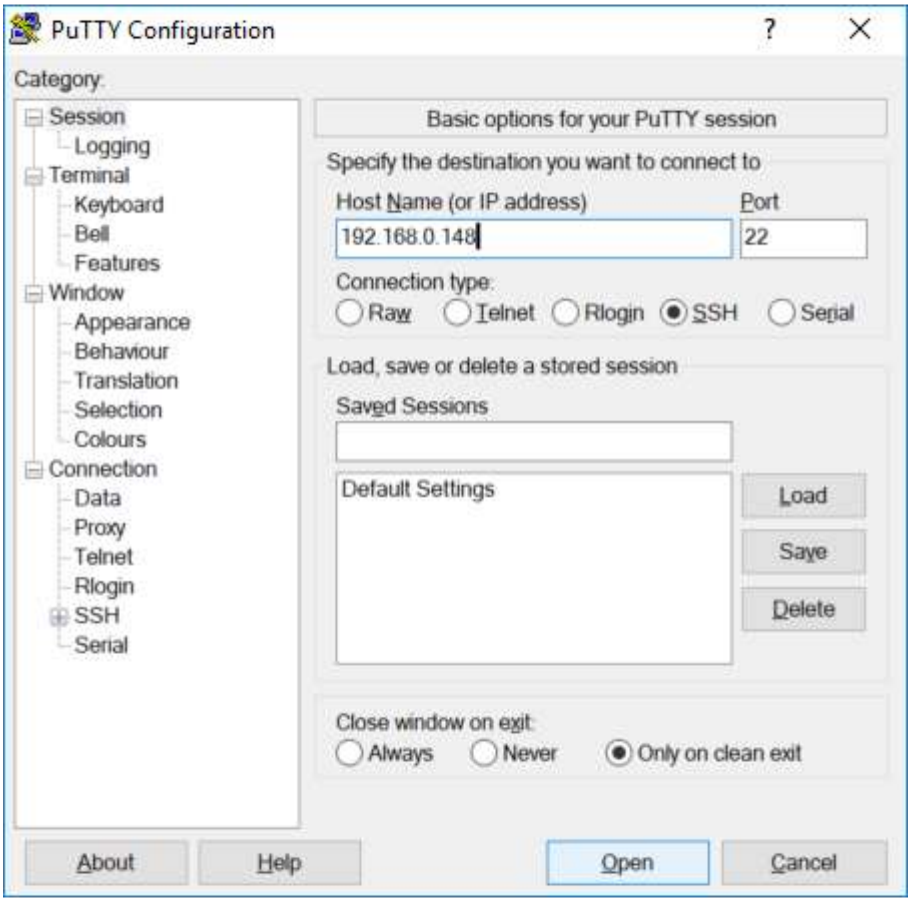
Select Finish.

Reboot Raspberry Pi

On the laptop, download and install PuTTY from this URL https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Install it.

Run PuTTY and just type the Raspberry Pi's IP address. Press Open.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.148

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

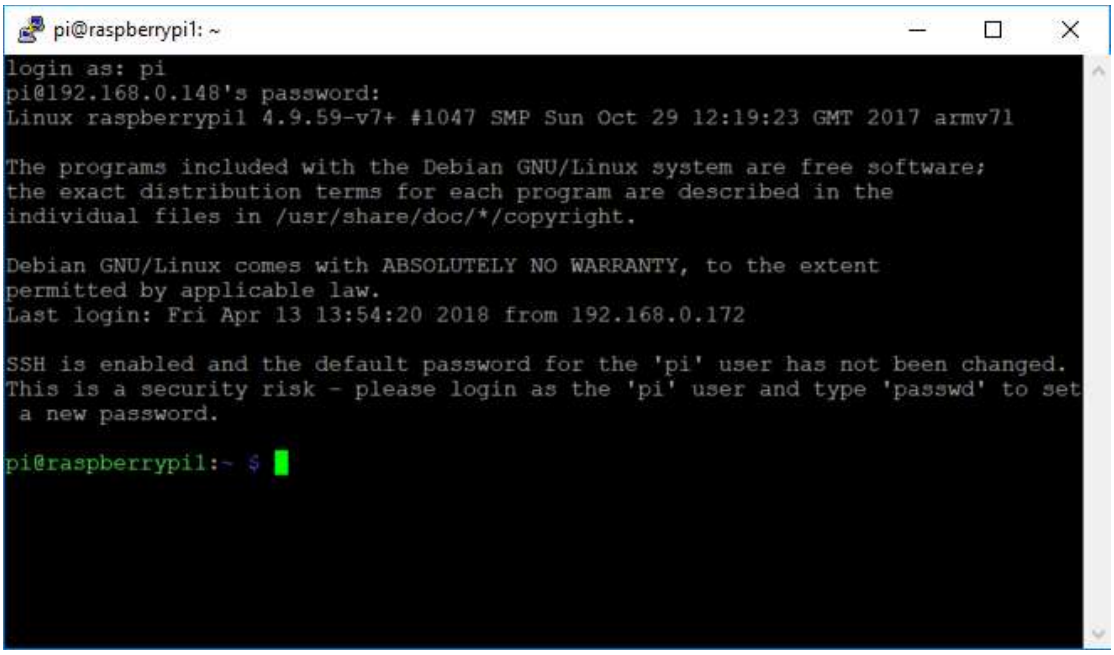
Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Reply Yes if you are asked about a certificate.

Then login with the username and password (usually pi, raspberry).



```
login as: pi
pi@192.168.0.148's password:
Linux raspberrypi1 4.9.59-v7+ #1047 SMP Sun Oct 29 12:19:23 GMT 2017 armv7l

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 13 13:54:20 2018 from 192.168.0.172

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi1:~ $
```

While using PuTTY you can Copy text by selecting it and left clicking on it. We can also Paste text by right clicking on the PuTTY window.

Part 4: Web Server

First update all the Raspberry Pis (0 virtual machine, 1 real, 2 real)

Type (on each raspberry)

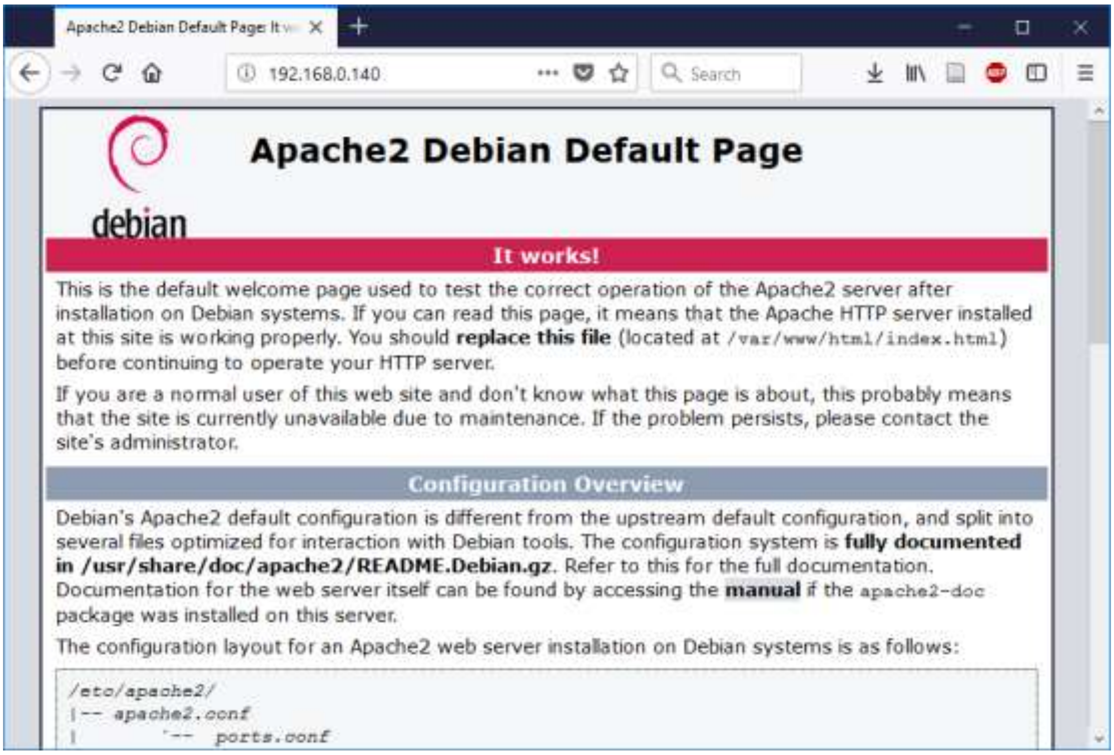
```
sudo apt-get update
```

Then install (on each raspberry) the Apache web server

```
sudo apt-get install apache2
```

Open a web browser and type the IP address of each Raspberry Pi.

We should see that apache is running, like in the following pictures.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

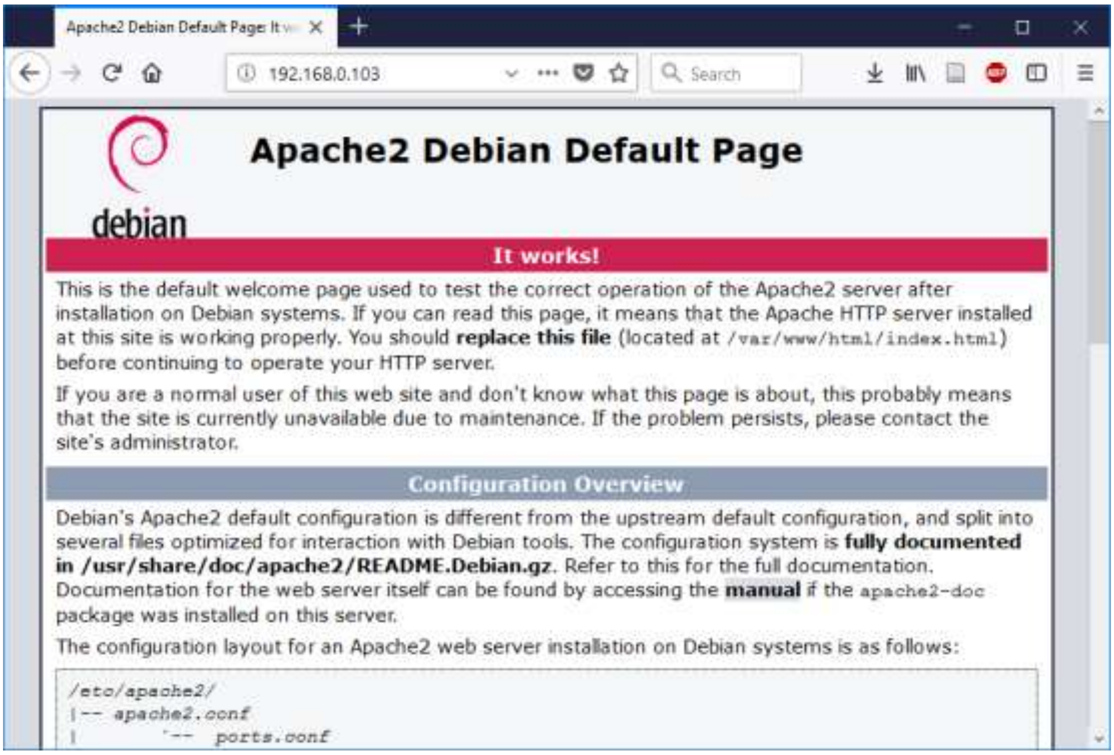
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
```



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

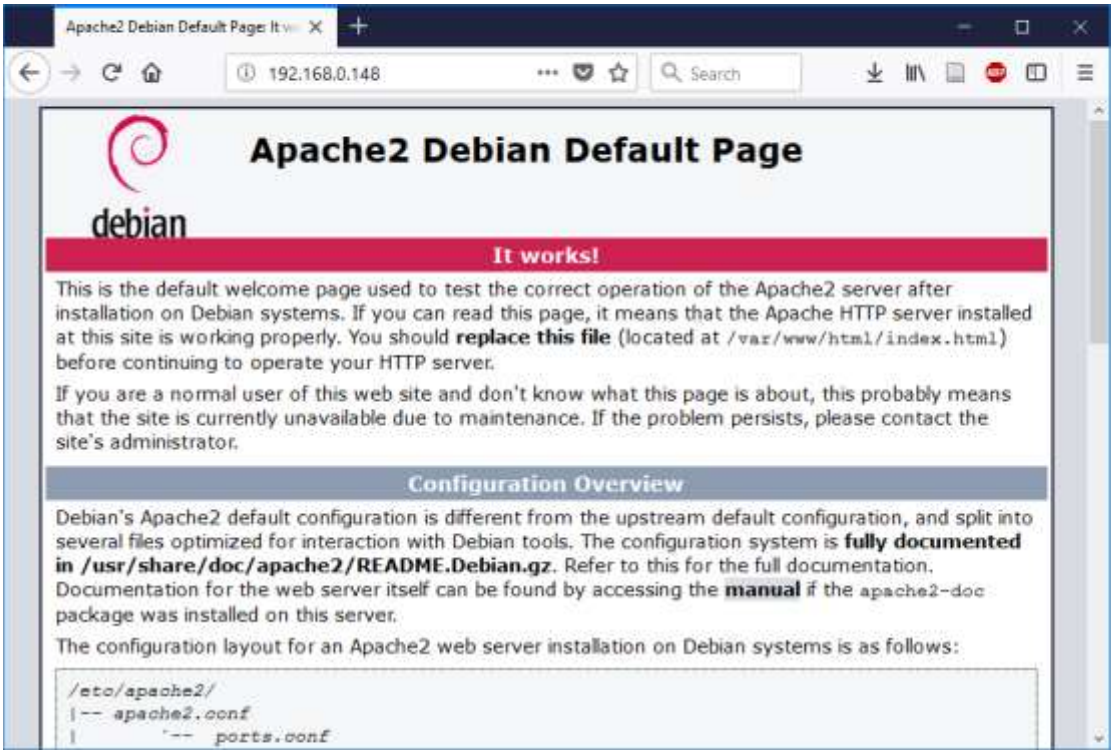
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
```



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
```

To know which Raspberry Pi web server is responding, you should add one message to each default web page.

Go to each Raspberry Pi and do the following…

Open a terminal and type

```
cd /var/www/html
```

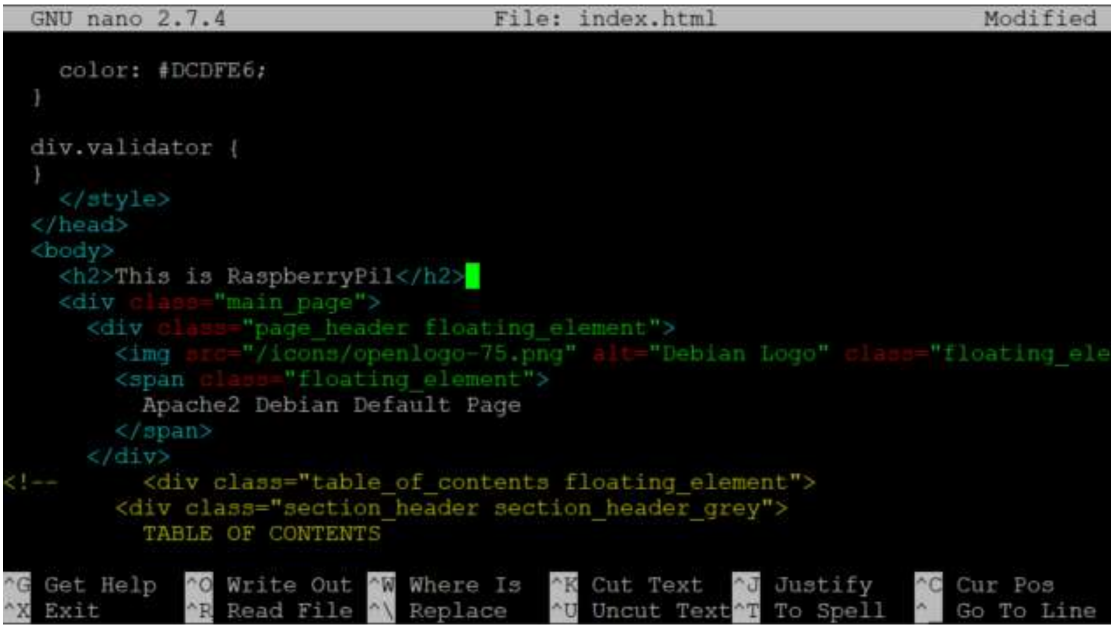
`sudo nano index.html`

Find the line <body> …

and bellow it type a new line…

`<h2>This is RaspberryPi1</h2>`

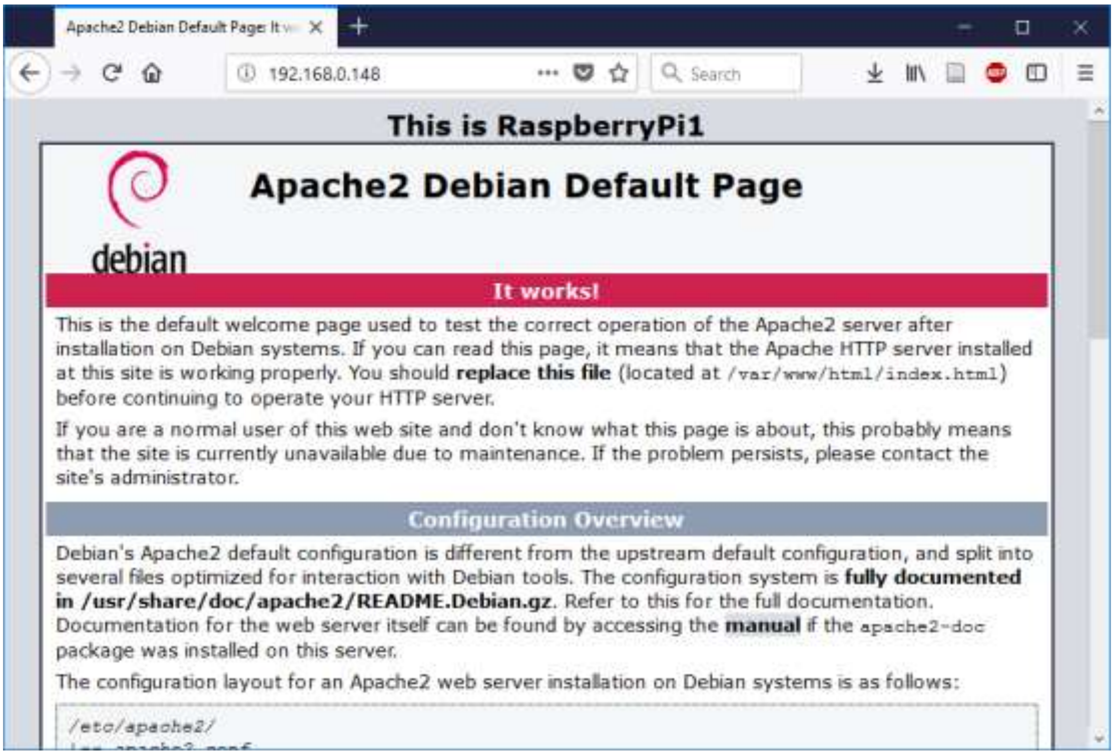
like in the picture.



```
  GNU nano 2.7.4                    File: index.html                      Modified

    color: #DCDFE6;
  }

  div.validator {
  }
    </style>
  </head>
  <body>
    <h2>This is RaspberryPi1</h2>
    <div class="main_page">
      <div class="page_header floating_element">
        <img src="/icons/openlogo-75.png" alt="Debian Logo" class="floating_ele
        <span class="floating_element">
          Apache2 Debian Default Page
        </span>
      </div>
<!--        <div class="table_of_contents floating_element">
        <div class="section_header section_header_grey">
          TABLE OF CONTENTS

^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
```

Press Ctrl + O to save and Ctrl + X to quit nano.

Go to the browser and type the Raspberry Pi's IP address. We should something like the following.



This is RaspberryPi1

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
```

Thus, you can recognize which Raspberry Pi is serving the web page.

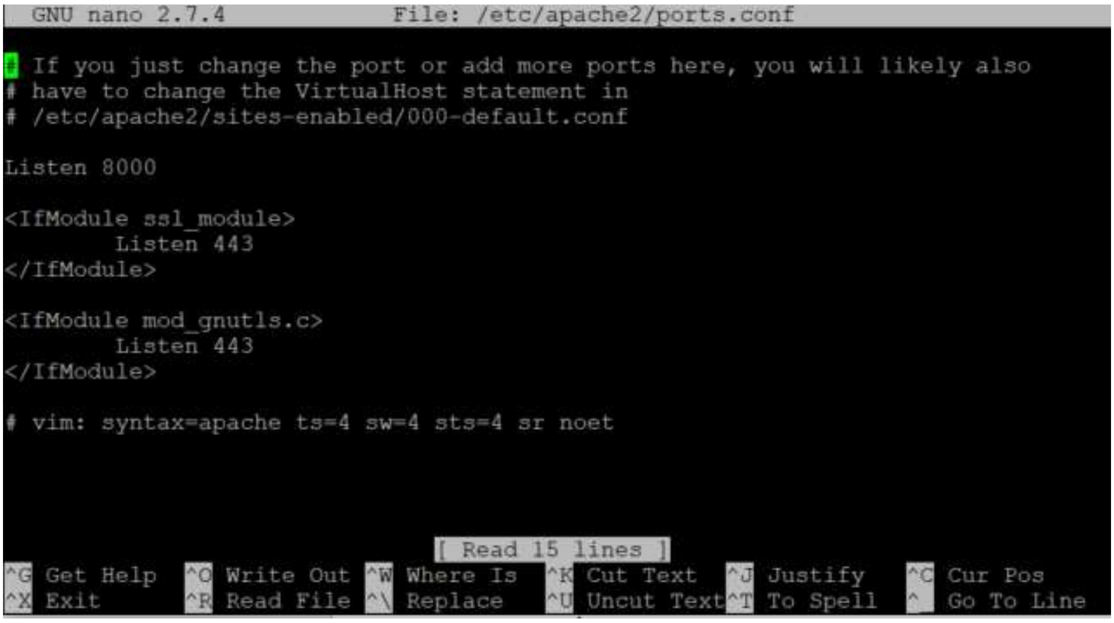
Do this for all the Raspberry Pis of the cluster, changing the number according to the number of each Raspberry Pi.

Change the port of each web server.

On each Raspberry Pi edit the ports configuration file and change the port from 80 to 8000.

Type **`sudo nano /etc/apache2/ports.conf`**

Change line Listen from 80 to 8000, like in the picture.



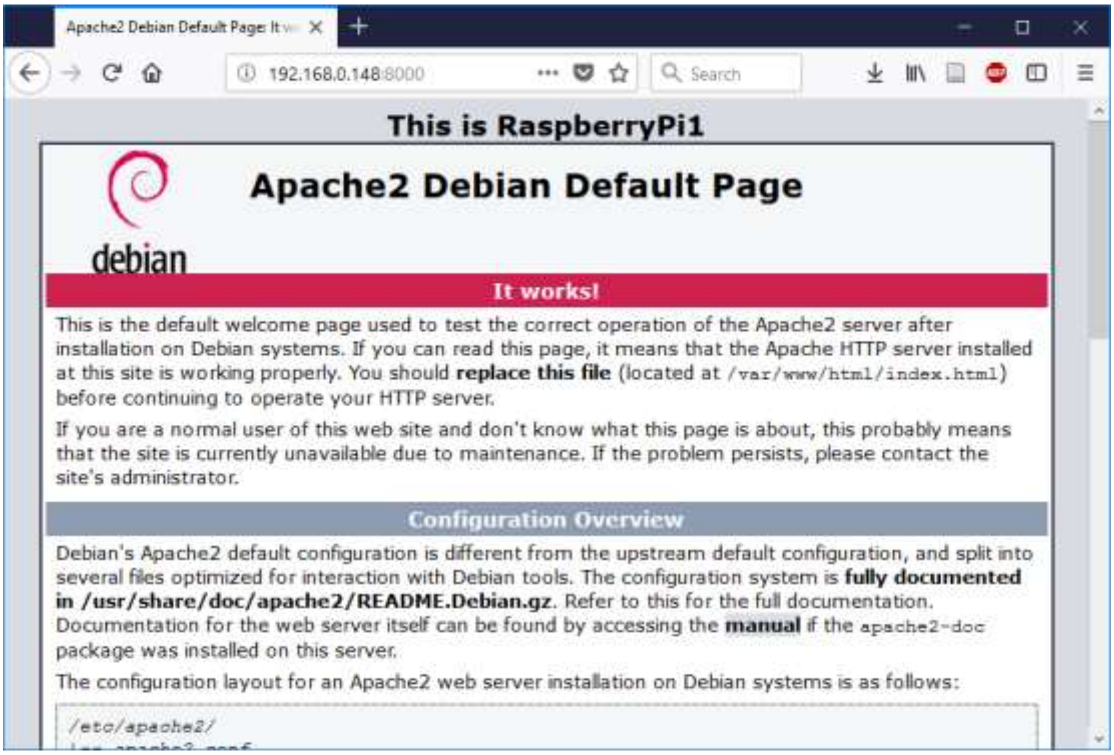
```
  GNU nano 2.7.4             File: /etc/apache2/ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8000

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet



                              [ Read 15 lines ]
^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
```

Press Ctrl + O to save, Ctrl + X to exit nano.

Reboot Raspberry Pi.

Do this for all Raspberry Pis.

After this, you need to supply the port to access the web server. Like in the picture.



This is RaspberryPi1

Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
```

Part 5: Load balancing

Open a terminal on the first Raspberry Pi.

Update system and install HAPROXY load balancer.

```
sudo apt-get update
```

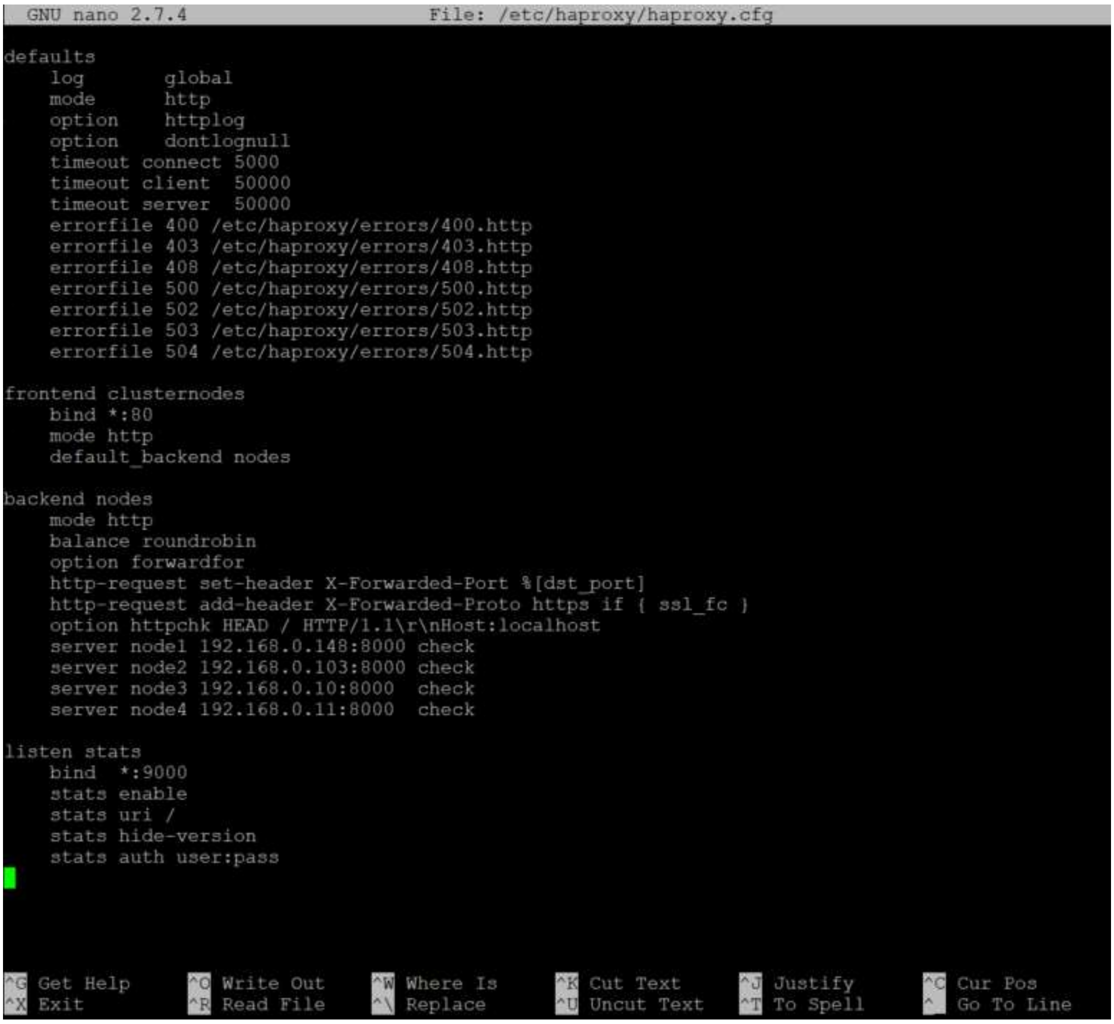
```
sudo apt-get install haproxy
```

Edit HAPROXY configuration

```
sudo nano /etc/haproxy/haproxy.cfg
```

At the end of the file add the following lines, like in the picture.

```
frontend clusternodes
    bind *:80
    mode http
    default_backend nodes

backend nodes
    mode http
    balance roundrobin
    option forwardfor
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request add-header X-Forwarded-Proto https if { ssl_fc }
    option httpchk HEAD / HTTP/1.1\r\nHost:localhost
    server node1 192.168.0.148:8000 check
    server node2 192.168.0.103:8000 check
    server node3 192.168.0.140:8000  check

listen stats
    bind  *:9000
    stats enable
    stats uri /
    stats hide-version
    stats auth user:pass
```



```
defaults
    log       global
    mode      http
    option    httplog
    option    dontlognull
    timeout connect 5000
    timeout client  50000
    timeout server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend clusternodes
    bind *:80
    mode http
    default_backend nodes

backend nodes
    mode http
    balance roundrobin
    option forwardfor
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request add-header X-Forwarded-Proto https if { ssl_fc }
    option httpchk HEAD / HTTP/1.1\r\nHost:localhost
    server node1 192.168.0.148:8000 check
    server node2 192.168.0.103:8000 check
    server node3 192.168.0.10:8000  check
    server node4 192.168.0.11:8000  check

listen stats
    bind  *:9000
    stats enable
    stats uri /
    stats hide-version
    stats auth user:pass
```

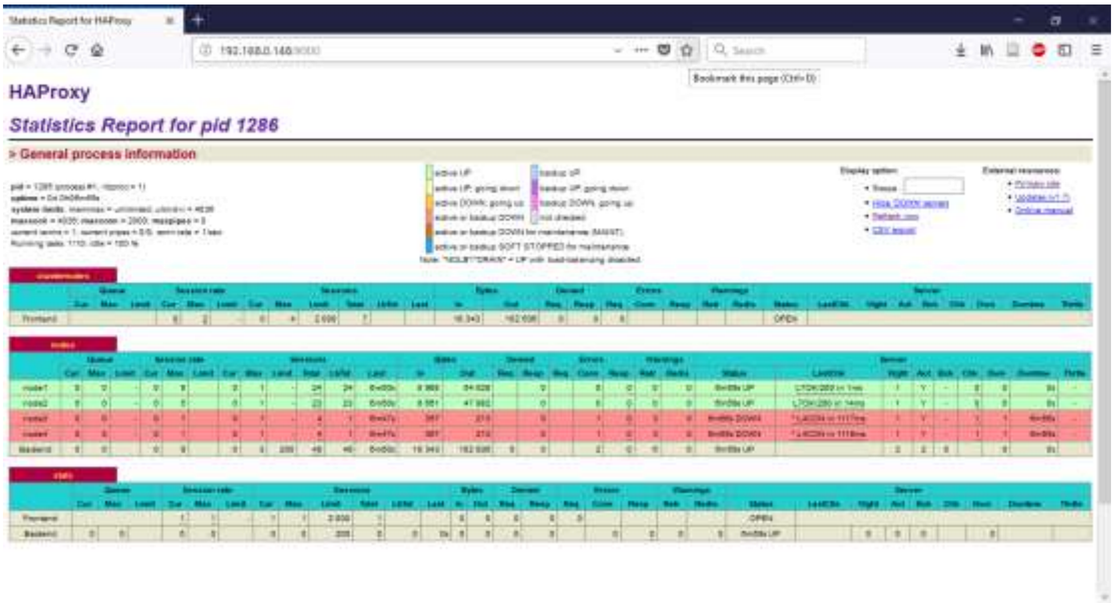
Restart HAPROXY service to apply changes.

```
sudo service haproxy restart
```

Afterwards open a browser and give the IP of node1 (raspberry1). We should see that if you refresh the page continuously, then sometimes it serves the page from Raspberry Pi 1, then from Raspberry Pi 2, and so on. This is load balancing. The workload is distributed among the nodes of the cluster.

If you give the IP of the first Raspberry Pi with the port 9000, then you have access to the statistics of the load balancer. The username is user and the password pass.

These are very useful measurements which must be utilized in the text.



HAProxy

Statistics Report for pid 1286

> General process information

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				6	2	-	0	4	2 000	7			16 343	162 606	0	0	0					OPEN								

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
node1	0	0	-	0	9		0	1	-	24	24	[illegible]	8 [illegible]	54 [illegible]		0		0	0	0	0	[illegible] UP	L7OK/200 in 1ms	1	Y	-	0	0	0s	-
node2	0	0	-	0	5		0	1	-	23	23	[illegible]	8 [illegible]	47 [illegible]		0		0	0	0	0	[illegible] UP	L7OK/200 in 14ms	1	Y	-	0	0	0s	-
node3	0	0	-	0	1		0	1	-	4	1	[illegible]	387	215		0		1	0	0	0	[illegible] DOWN	* L4CON in [illegible]	1	Y	-	1	1	[illegible]	-
node4	0	0	-	0	1		0	1	-	4	1	[illegible]	387	215		0		1	0	0	0	[illegible] DOWN	* L4CON in [illegible]	1	Y	-	1	1	[illegible]	-
Backend	0	0		0	9		0	3	200	48	48	[illegible]	16 343	162 606	0	0		2	0	0	0	[illegible] UP		2	2	0		0	0s	

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				1	1	-	1	1	2 000	1			0	0	0	0	0					OPEN								
Backend	0	0		0	0		0	0	200	0	0	0s	0	0	0	0		0	0	0	0	[illegible] UP		0	0	0		0		

References:

http://www.raspberrywebserver.com/raspberrypicluster/raspberry-pi-cluster.html

http://www.gregtrowbridge.com/setting-up-a-multiple-raspberry-pi-web-server-part-1/

http://www.gregtrowbridge.com/setting-up-a-multiple-raspberry-pi-web-server-part-2/

http://www.gregtrowbridge.com/setting-up-a-multiple-raspberry-pi-web-server-part-3/

http://gregtrowbridge.com/setting-up-a-multiple-raspberry-pi-web-server-part-4/

http://gregtrowbridge.com/setting-up-a-multiple-raspberry-pi-web-server-part-5/

https://serversforhackers.com/c/load-balancing-with-haproxy

https://stackoverflow.com/questions/42348849/configuration-for-haproxy-in-ubuntu