

Raspberry Pi 3

Installing Debian Jessie set-up

First Win32DiskImager has to be installed. Prior to the installation insert a micro SD into an USB adaptor and plug it into the computer. Double click the win32diskimager-1.0.0-install.exe located in /Instalation Files and Required Software and a window will pop up as show on figure 1. Accept the license agreement and select "Next".

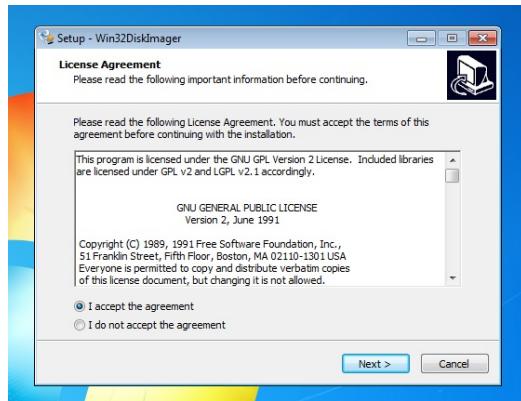


Figure 1: Win32DiskImager installation process part 1

Next on figure 2 choose the installation directory and select "Next".

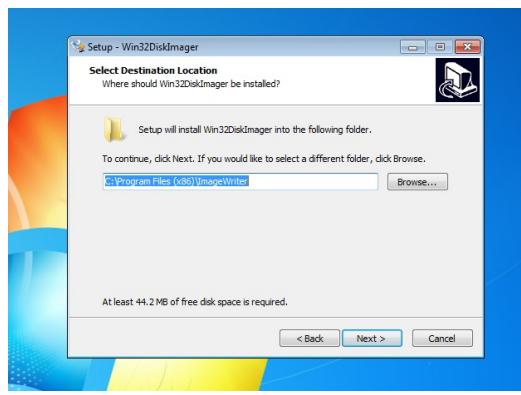


Figure 2: Win32DiskImager installation process part 2

Next on figure 3 choose the Start Menu directory and select "Next".

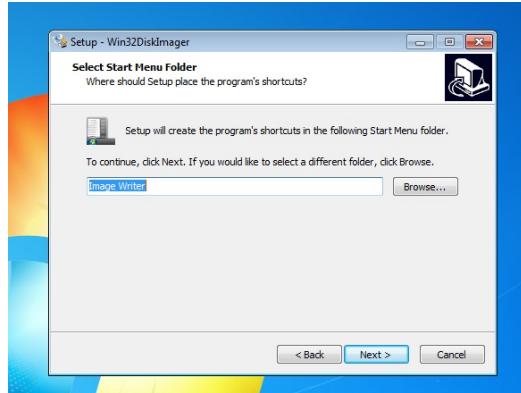


Figure 3: Win32DiskImager installation process part 3

Next on figure 4 this window gives the option to create a desktop shortcut by checking the check box. Select "Next".

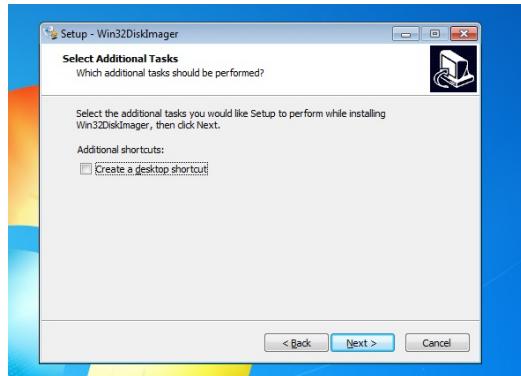


Figure 4: Win32DiskImager installation process part 4

Next on figure 5 verify all the details are correct and select "Next".

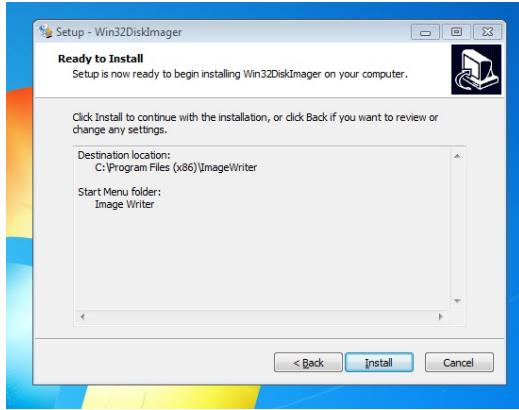


Figure 5: Win32DiskImager installation process part 5

Next on figure 6 choose choose to "Launch Win32DiskImager" and select "Finish".

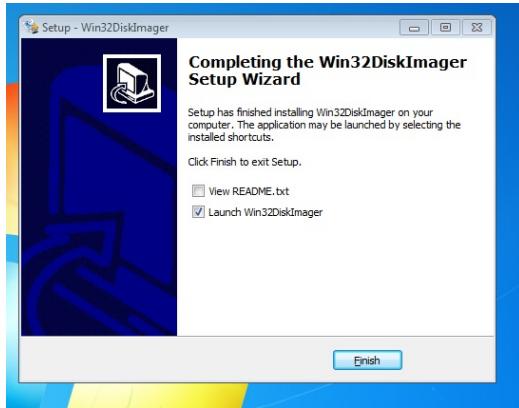


Figure 6: Win32DiskImager installation process part 6

Next on a window will open as shown on figure 7 Select the folder icon to browse for image files.

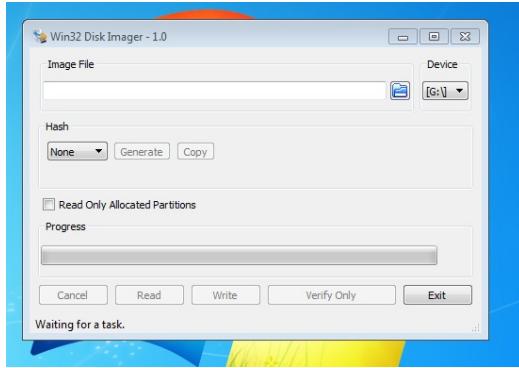


Figure 7: Debian Jessie installation process part 1

Navigate to the image file as shown on figure 8 In this case the image file is located in the /Instalation Files and Required Software directory. Select the image file and click "Open".

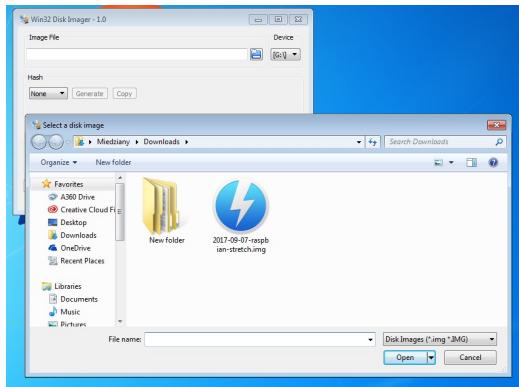


Figure 8: Debian Jessie installation process part 2

Select the micro SD card drive as shown on figure 9.

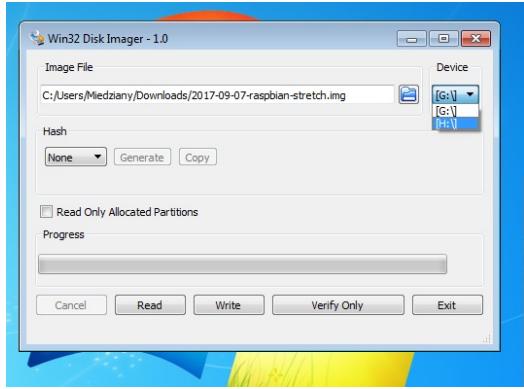


Figure 9: Debian Jessie installation process part 3

Select "Yes" as shown on figure 10.

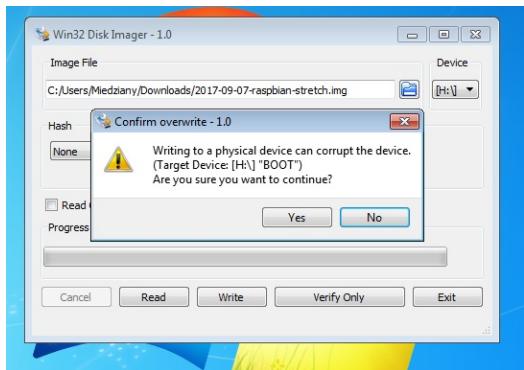


Figure 10: Debian Jessie installation process part 4

Figure 11 shows what the writing of the image file to the micro SD card looks like.

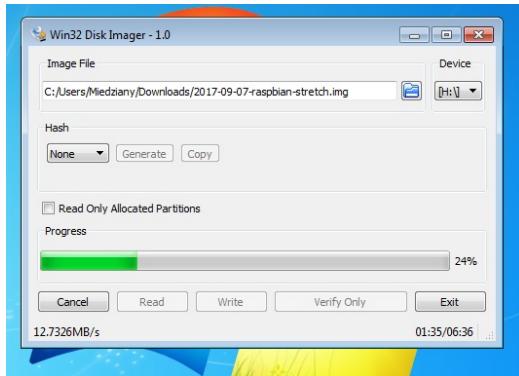


Figure 11: Debian Jessie installation process part 5

As shown on figure ?? this pop up message will display once the writing of the image file has been completed.

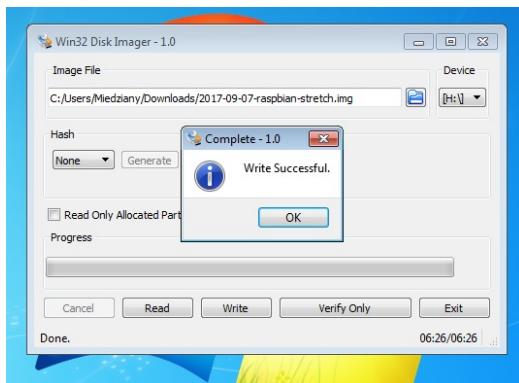


Figure 12: Debian Jessie installation process part 6

This concludes a successfully installation of Debian Jessie.

Debian Jessie configuration

In order to run the Raspberry Pi Jessie headless SSH has to be enabled. Create a new text file as shown by figure 13, and select "Save As..." as shown by figure 14.

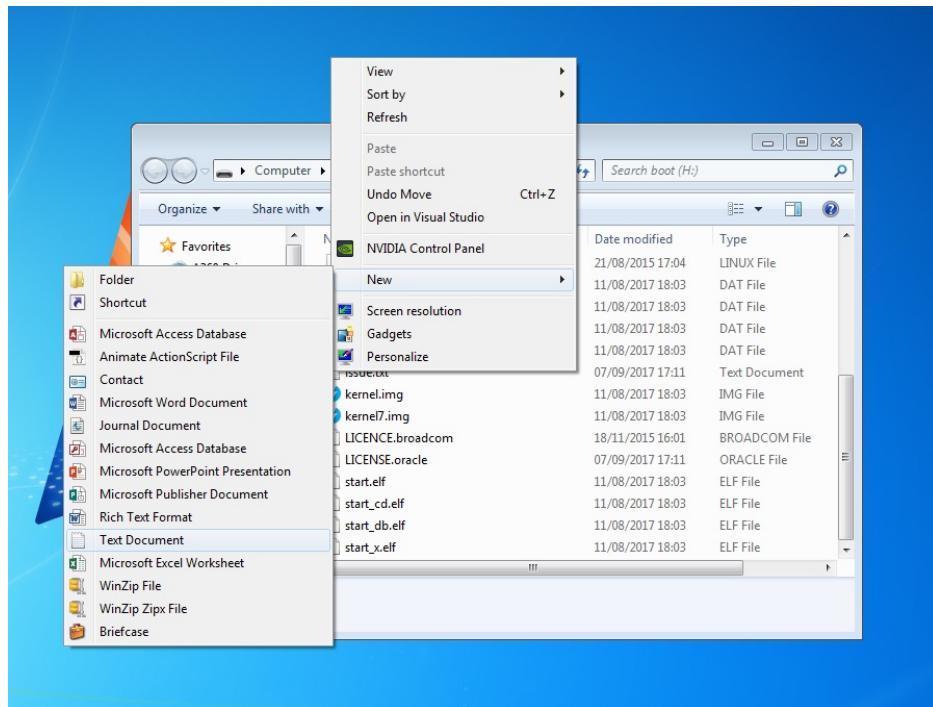


Figure 13: Debian Jessie configuration part 0

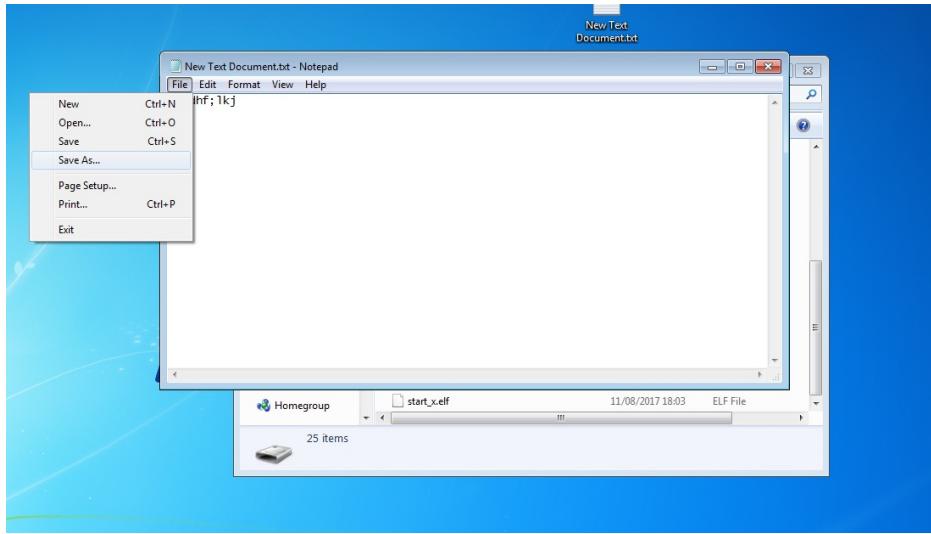


Figure 14: Debian Jessie configuration part 1

Select type as "All Files", and the file name "ssh" as shown by figure 15.

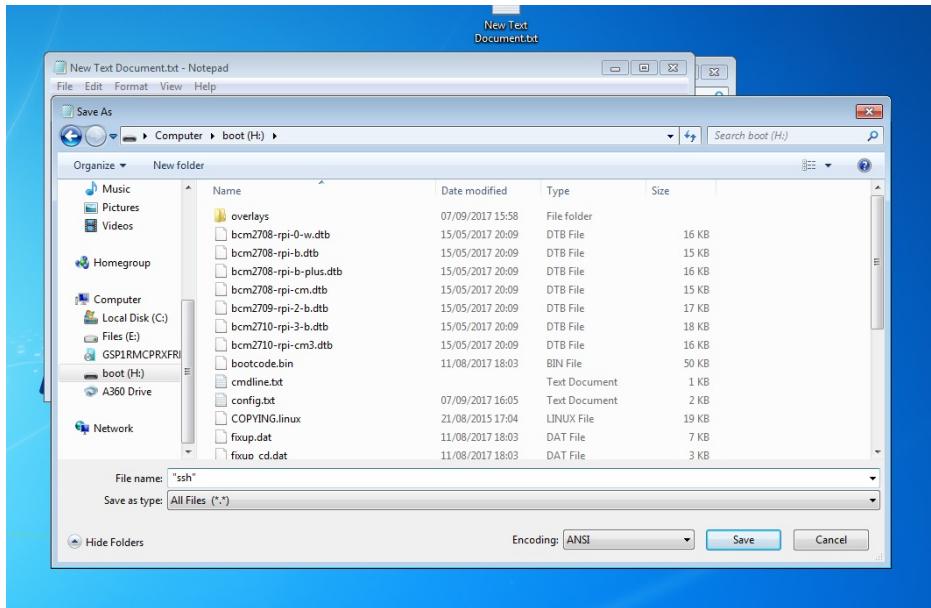


Figure 15: Debian Jessie configuration part 2

Figure 16 shows what the micro SD card directory should look like after the ssh file has been copied to the drive.

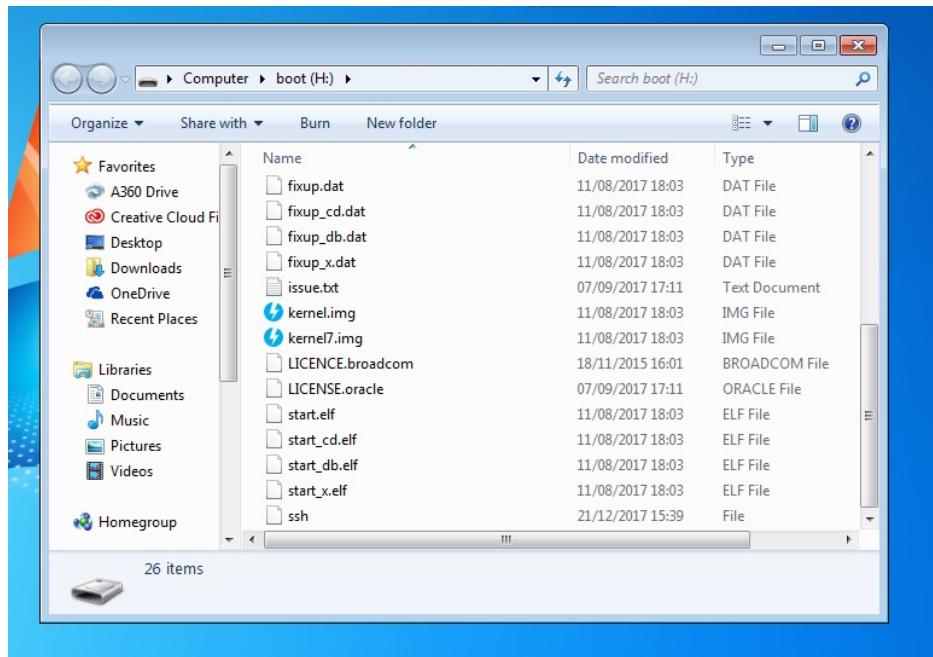


Figure 16: Debian Jessie configuration part 3

Connect Raspberry pi to the router and go to the router con-fig web page as shown by figure 17, and log into the router.

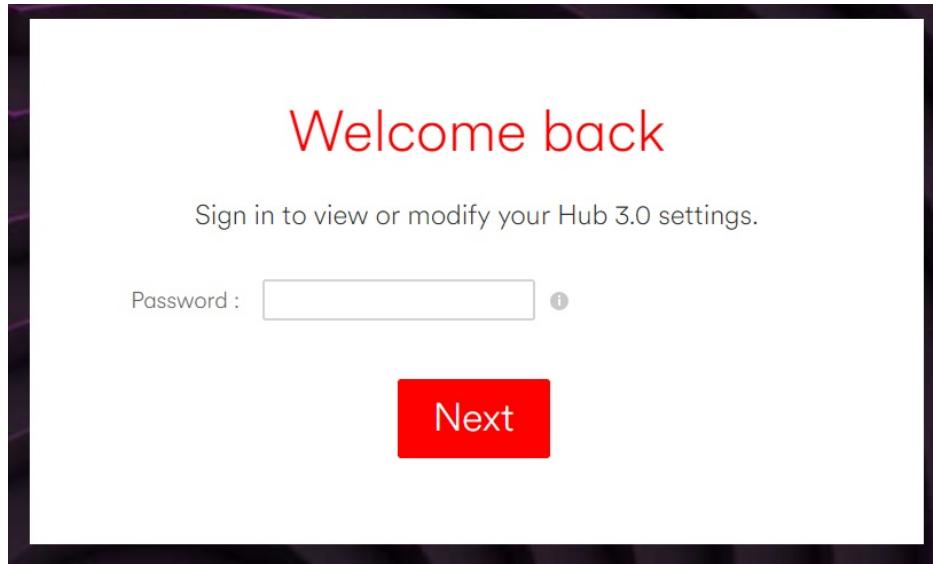


Figure 17: Debian Jessie configuration part 4

Select "Connected Devices" as shown by figure 18.



Home



Connected devices



Modem mode



Advanced settings



Admin

Figure 18: Debian Jessie configuration part 5

Raspberry Pi IP address can be seen as shown by figure 19.

raspberrypi	B8:27:EB:4B:7A:8D	192.168.0.41/ 24	1000	Ethernet
-------------	-------------------	---------------------	------	----------

Figure 19: Debian Jessie configuration part 6

Next open the Putty terminal located in the /Instalation Files and Required Software directory, and enter Raspberry PI IP address as shown by figure 20. And click "Open".

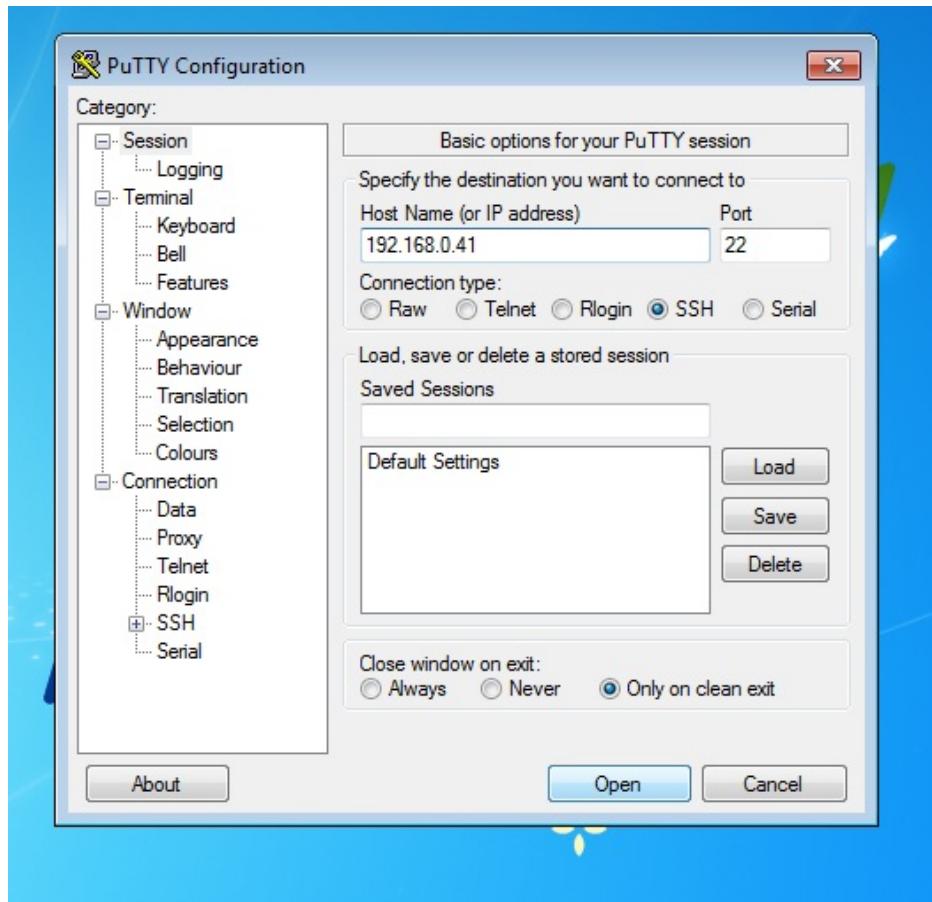


Figure 20: Debian Jessie configuration part 7

On the pop up window select "Yes" as shown by figure 21.

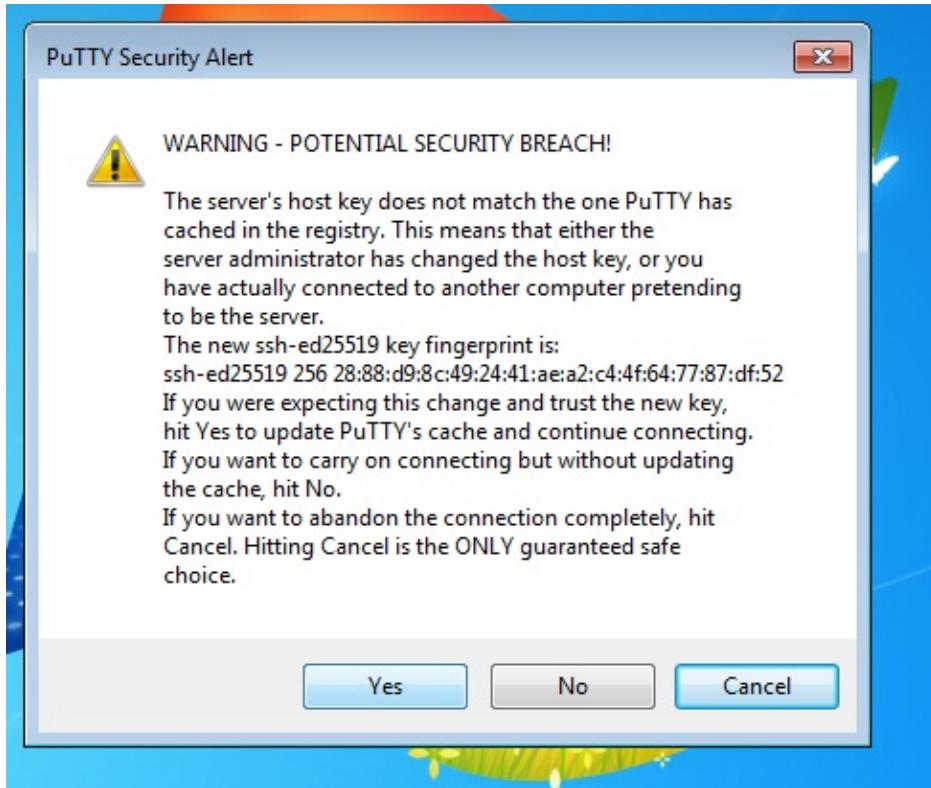
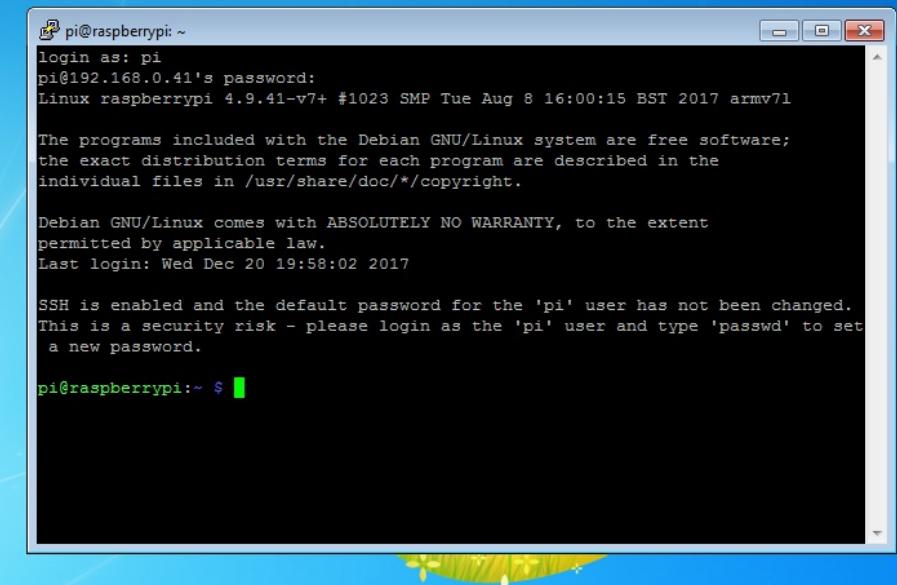


Figure 21: Debian Jessie configuration part 8

Next the terminal will ask for user-name, type "pi" and then the terminal will ask for the password, type "raspberry". This will result in a successful log in, as shown by figure 22.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window displays the following text:

```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.41's password:
Linux raspberrypi 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 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.

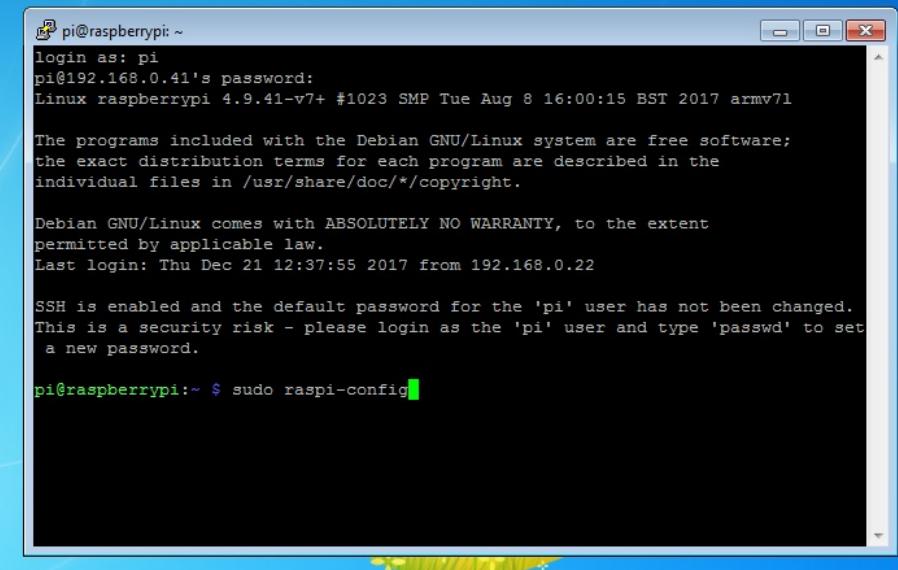
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec 20 19:58:02 2017

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@raspberrypi:~ $
```

Figure 22: Debian Jessie configuration part 9

Next enter "sudo raspi-config" as shown by figure 23.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window displays the following text:

```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.41's password:
Linux raspberrypi 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 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: Thu Dec 21 12:37:55 2017 from 192.168.0.22

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@raspberrypi: ~ $ sudo raspi-config
```

Figure 23: Debian Jessie configuration part 10

Navigate to "Advanced Options" as shown by figure 24.

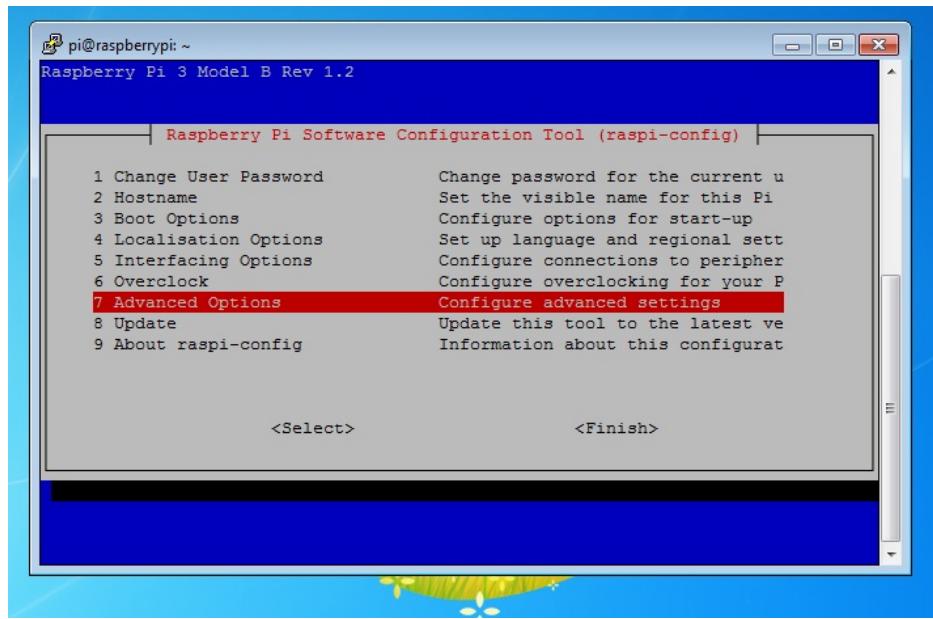


Figure 24: Debian Jessie configuration part 11

And select "Expand Filesystem" as shown by figure 25.

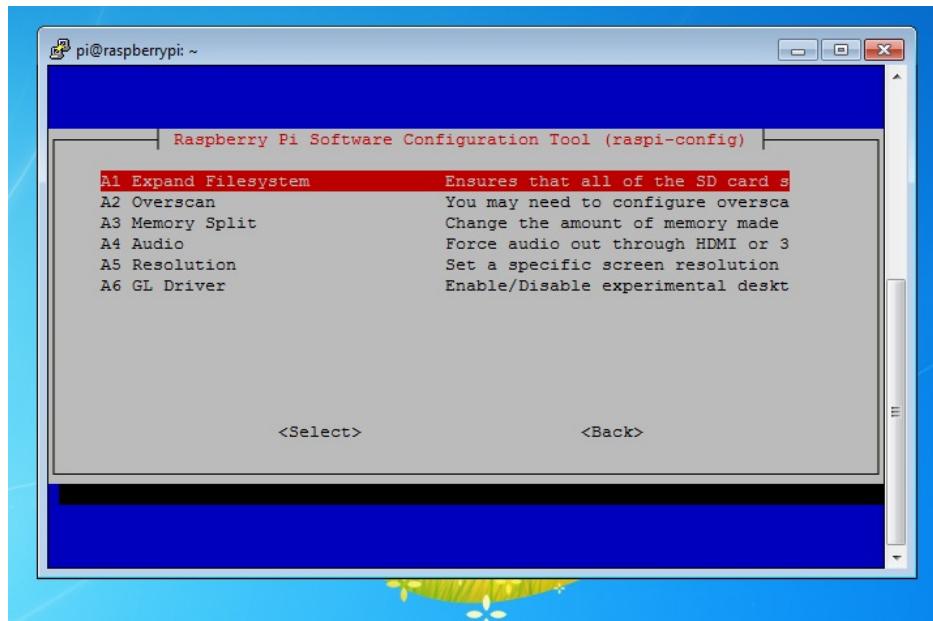


Figure 25: Debian Jessie configuration part 12

Select "OK" as shown by figure 26.

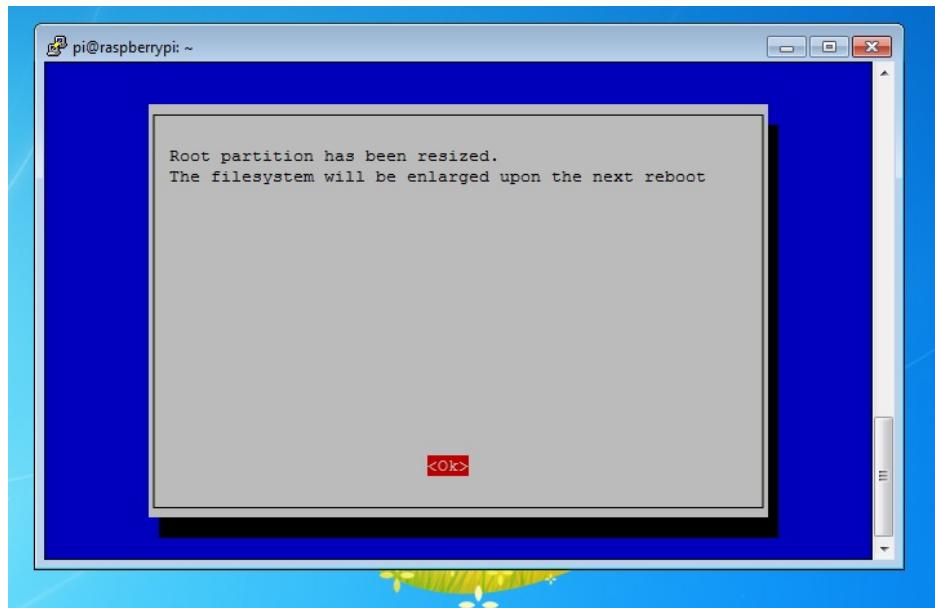


Figure 26: Debian Jessie configuration part 13

Select "Finish" as shown by figure 27.

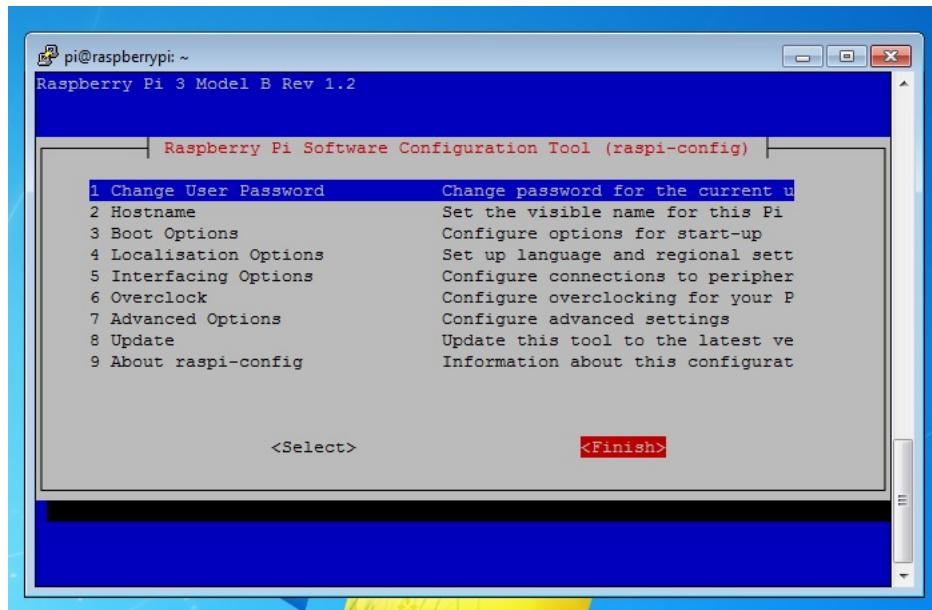


Figure 27: Debian Jessie configuration part 14

Select "Yes" as shown by figure 28.

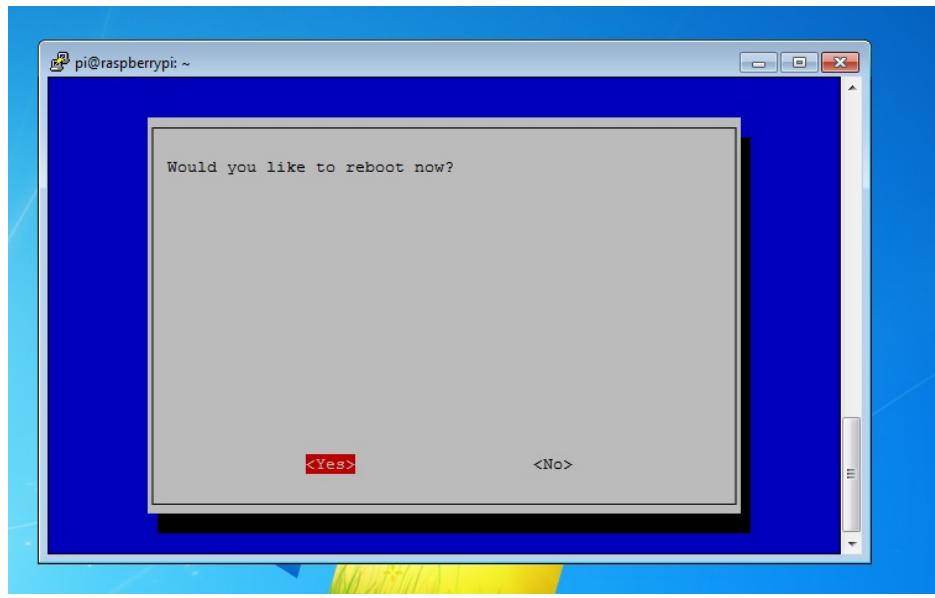


Figure 28: Debian Jessie configuration part 15

This will close the connection with the Server as shown by figure 29.

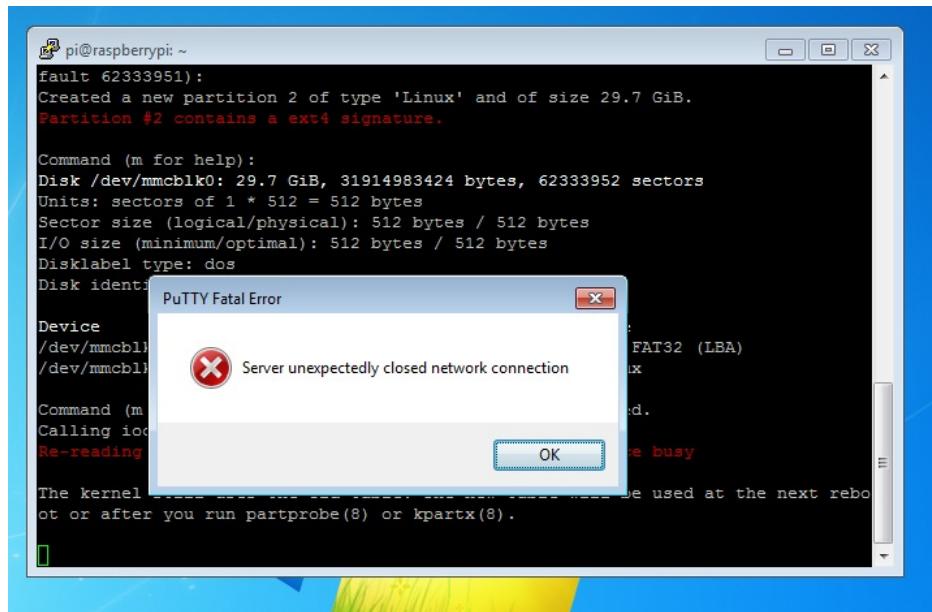
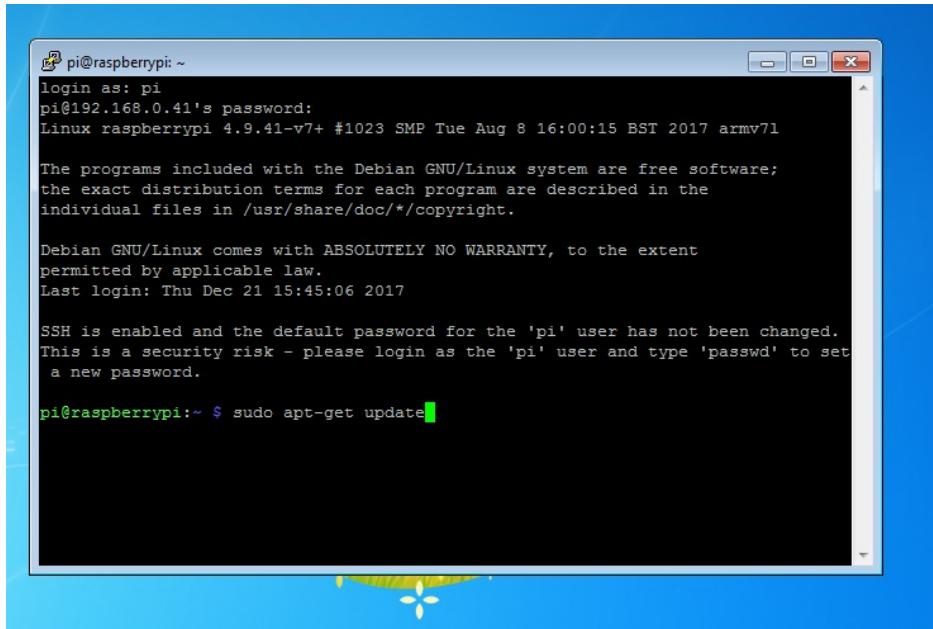


Figure 29: Debian Jessie configuration part 16

Restart the putty connection repeating the process from part 7. Next type "sudo apt-get update" as shown by figure 30. This will update the software and all the repositories.

A screenshot of a terminal window titled "pi@raspberrypi: ~". The window shows a standard Debian Jessie login screen with the following text:

```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.41's password:
Linux raspberrypi 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 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: Thu Dec 21 15:45:06 2017

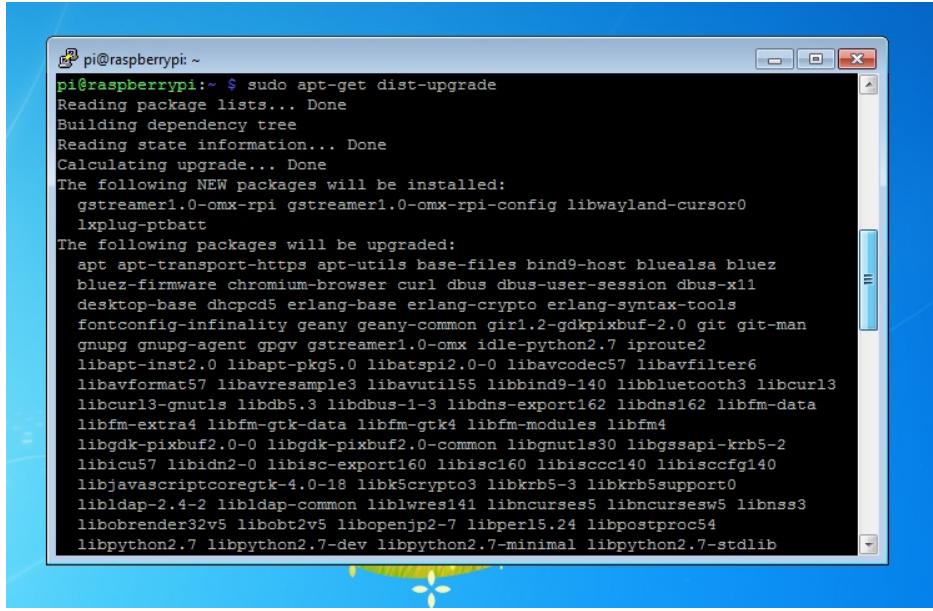
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ sudo apt-get update
```

The terminal window is set against a blue background with a small yellow flower icon at the bottom center.

Figure 30: Debian Jessie configuration part 17

This will take some time. Next type ”sudo apt-get dist-upgrade” the terminal will look like figure 31. This will update the software and all the repositories.



```
pi@raspberrypi: ~ $ sudo apt-get dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  gstreamer1.0-omx-rpi gstreamer1.0-omx-rpi-config libwayland-cursor0
  lxplug-ptbatt
The following packages will be upgraded:
  apt apt-transport-https apt-utils base-files bind9-host bluealsa bluez
  bluez-firmware chromium-browser curl dbus dbus-user-session dbus-x11
  desktop-base dhcpcd5 erlang-base erlang-crypto erlang-syntax-tools
  fontconfig-infinality geany geany-common girl1.2-gdkpixbuf-2.0 git git-man
  gnupg gnupg-agent gpgv gstreamer1.0-omx idle-python2.7 iproute2
  libapt-inst2.0 libapt-pkg5.0 libatspi2.0-0 libavcodec57 libavfilter6
  libavformat57 libavresample3 libavutil155 libbind9-140 libbluetooth3 libcurl3
  libcurl3-gnutls libdb5.3 libdbus-1-3 libdns-export162 libdns162 libfm-data
  libfm-extra4 libfm-gtk-data libfm-gtk4 libfm-modules libfm4
  libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common libgnutls30 libgssapi-krb5-2
  libicu57 libidn2-0 libisc-export160 libisc160 libisccc140 libisccfg140
  libjavascriptcoregtk-4.0-18 libk5crypto3 libkrb5-3 libkrb5support0
  libldap-2.4-2 libldap-common liblwres141 libncurses5 libncursesw5 libnss3
  libobrender32v5 libobt2v5 libopenjp2-7 libperl5.24 libpostproc54
  libpython2.7 libpython2.7-dev libpython2.7-minimal libpython2.7-stdlib
```

Figure 31: Debian Jessie configuration part 18

Next install Apache web server by typing the following command "sudo apt-get install apache2 -y" figure 32.

```

pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,654 kB/1,948 kB of archives.
After this operation, 6,263 kB of additional disk space will be used.
Get:1 http://mirror.mythic-beasts.com/raspbian/raspbian stretch/main armhf apache2-bin armhf 2.4.25-3+deb9u3 [1,038 kB]
Get:2 http://mirror.mythic-beasts.com/raspbian/raspbian stretch/main armhf apache2-utils armhf 2.4.25-3+deb9u3 [218 kB]
Get:3 http://mirror.mythic-beasts.com/raspbian/raspbian stretch/main armhf apache2-data all 2.4.25-3+deb9u3 [162 kB]
Get:4 http://mirror.mythic-beasts.com/raspbian/raspbian stretch/main armhf apache2 armhf 2.4.25-3+deb9u3 [235 kB]

```

Figure 32: MariaDB instalation part 1

After the sucessfull instalation of Apache web server, by entering Raspberry PI IP into the browser, the browser should look like figure 33.

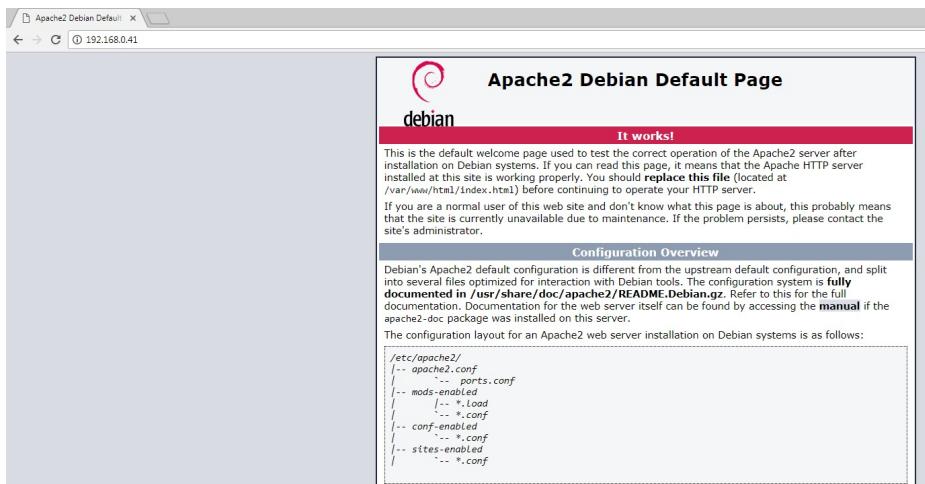
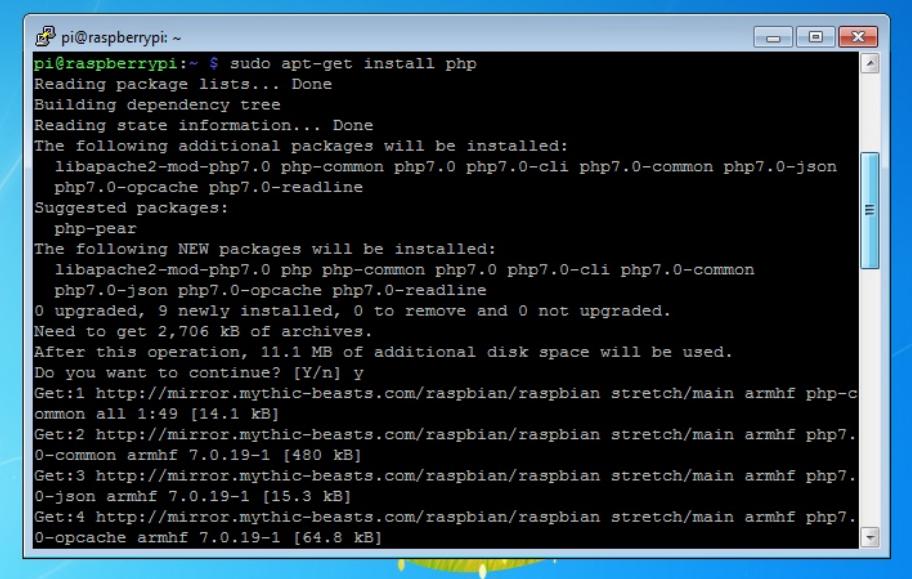


Figure 33: MariaDB instalation part 2

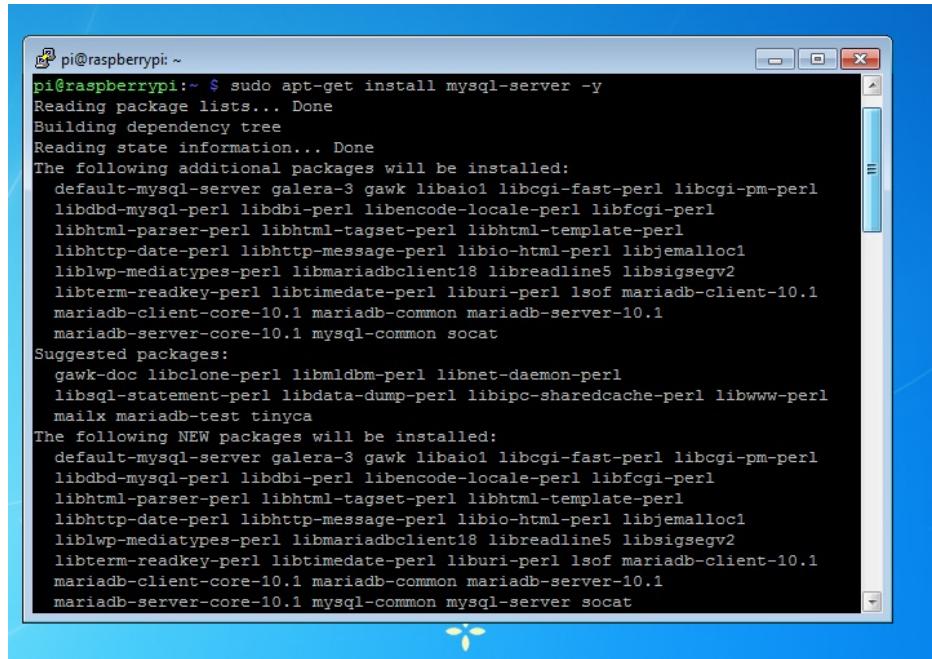
Next install PHP with the command "sudo apt-get install php" as show by figure 34.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window shows the output of the command "sudo apt-get install php". The terminal displays the package list, dependency tree, state information, additional packages to be installed (including libapache2-mod-php7.0, php-common, php7.0-cli, php7.0-common, php7.0-json, php7.0-opcache, and php7.0-readline), suggested packages (php-pear), and new packages to be installed (libapache2-mod-php7.0, php-common, php7.0-cli, php7.0-common, php7.0-json, php7.0-opcache, and php7.0-readline). It also shows that 0 packages were upgraded, 9 were newly installed, and 0 were removed. The total size of the archives to be downloaded is 2,706 kB. After this operation, 11.1 MB of additional disk space will be used. The user is prompted with "Do you want to continue? [Y/n] y". The terminal then shows the download process with four GET requests from "http://mirror.mythic-beasts.com/raspbian/raspbian stretch/main armhf".

Figure 34: MariaDB instalation part 3

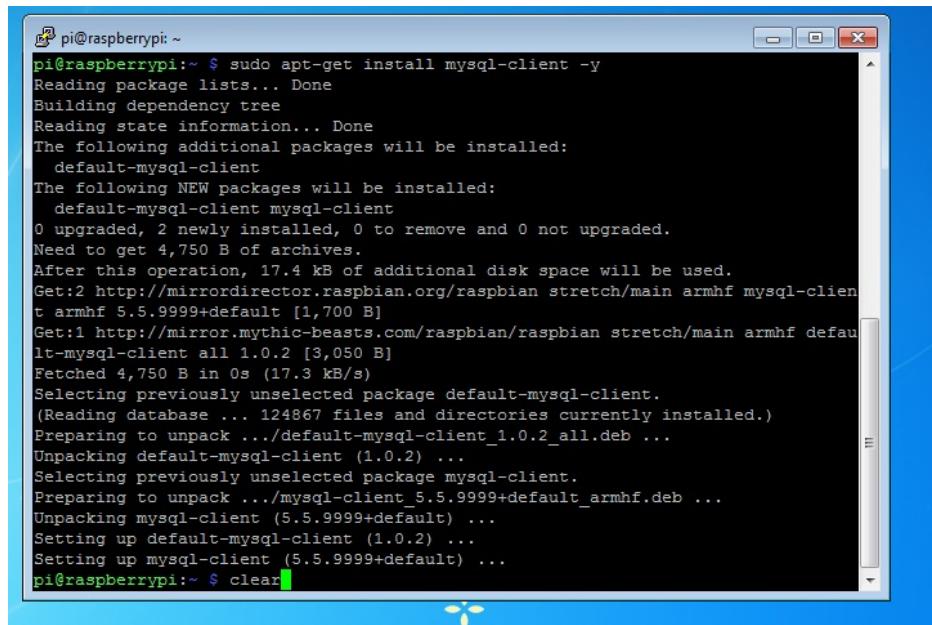
Next install MySQL Server with the command "sudo apt-get install mysql-server -y" as show by figure 35.



```
pi@raspberrypi: ~ $ sudo apt-get install mysql-server -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  default-mysql-server galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl
  libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl libjemalloc1
  liblwp-mediatypes-perl libmariadbclient18 libreadline5 libsigsegv2
  libterm-readkey-perl libtimedate-perl liburi-perl lsof mariadb-client-10.1
  mariadb-client-core-10.1 mariadb-common mariadb-server-10.1
  mariadb-server-core-10.1 mysql-common socat
Suggested packages:
  gawk-doc libclone-perl libmldb-perl libnet-daemon-perl
  libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl libwww-perl
  mailx mariadb-test tinyca
The following NEW packages will be installed:
  default-mysql-server galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl
  libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl libjemalloc1
  liblwp-mediatypes-perl libmariadbclient18 libreadline5 libsigsegv2
  libterm-readkey-perl libtimedate-perl liburi-perl lsof mariadb-client-10.1
  mariadb-client-core-10.1 mariadb-common mariadb-server-10.1
  mariadb-server-core-10.1 mysql-common mysql-server socat
```

Figure 35: MariaDB instalation part 4

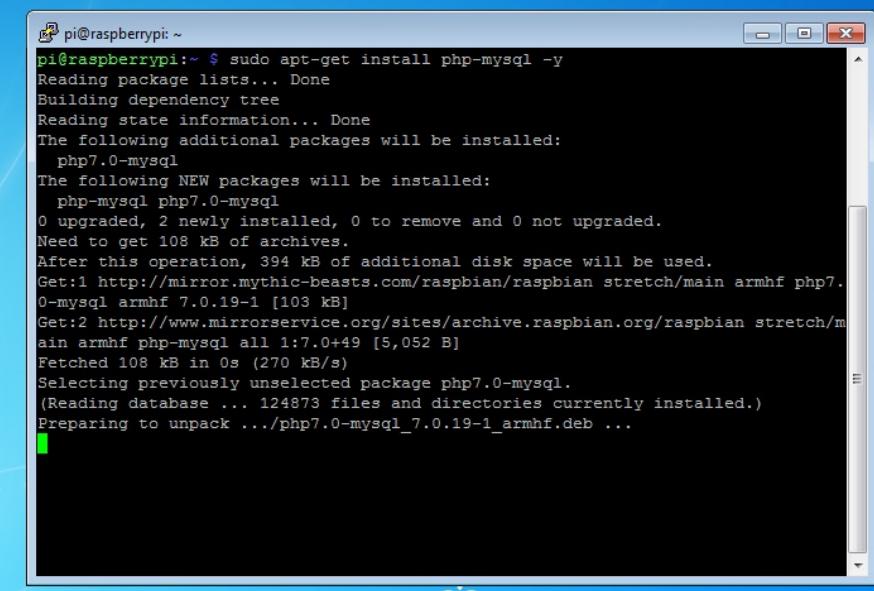
Next install MySQL Client with the command ”sudo apt-get install mysql-client -y” as show by figure 36.



```
pi@raspberrypi:~ $ sudo apt-get install mysql-client -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  default-mysql-client
The following NEW packages will be installed:
  default-mysql-client mysql-client
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,750 B of archives.
After this operation, 17.4 kB of additional disk space will be used.
Get:2 http://mirrordirector.raspbian.org/raspbian stretch/main armhf mysql-client armhf 5.5.9999+default [1,700 B]
Get:1 http://mirror.mythic-beasts.com/raspbian/raspbian stretch/main armhf default-mysql-client all 1.0.2 [3,050 B]
Fetched 4,750 B in 0s (17.3 kB/s)
Selecting previously unselected package default-mysql-client.
(Reading database ... 124867 files and directories currently installed.)
Preparing to unpack .../default-mysql-client_1.0.2_all.deb ...
Unpacking default-mysql-client (1.0.2) ...
Selecting previously unselected package mysql-client.
Preparing to unpack .../mysql-client_5.5.9999+default_armhf.deb ...
Unpacking mysql-client (5.5.9999+default) ...
Setting up default-mysql-client (1.0.2) ...
Setting up mysql-client (5.5.9999+default) ...
pi@raspberrypi:~ $ clear
```

Figure 36: MariaDB instalation part 5

Next install MySQL repository for PHP with the command ”sudo apt-get install php-mysql -y” as show by figure 37.

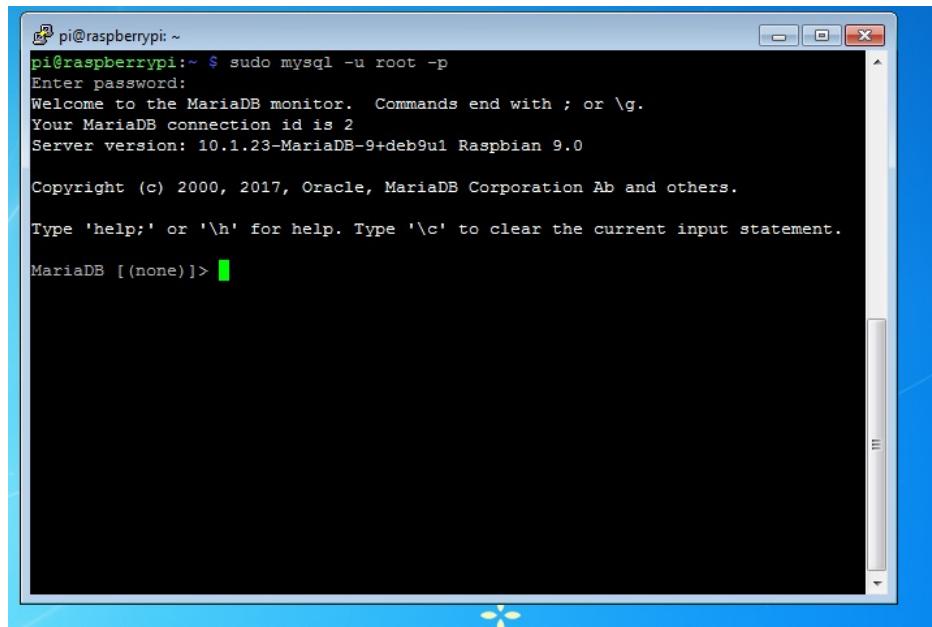


```
pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo apt-get install php-mysql -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  php7.0-mysql
The following NEW packages will be installed:
  php-mysql php7.0-mysql
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 108 kB of archives.
After this operation, 394 kB of additional disk space will be used.
Get:1 http://mirror.mythic-beasts.com/raspbian/raspbian stretch/main armhf php7.0-mysql armhf 7.0.19-1 [103 kB]
Get:2 http://www.mirrorservice.org/sites/archive.raspbian.org/raspbian stretch/main armhf php-mysql all 1:7.0+49 [5,052 B]
Fetched 108 kB in 0s (270 kB/s)
Selecting previously unselected package php7.0-mysql.
(Reading database ... 124873 files and directories currently installed.)
Preparing to unpack .../php7.0-mysql_7.0.19-1_armhf.deb ...

```

Figure 37: MariaDB instalation part 6

Log into MySQL using "root" as user as show by figure 38. When the terminal asks for password just click "Enter" as the root user after installation has no password.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window contains the following text:

```
pi@raspberrypi:~ $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

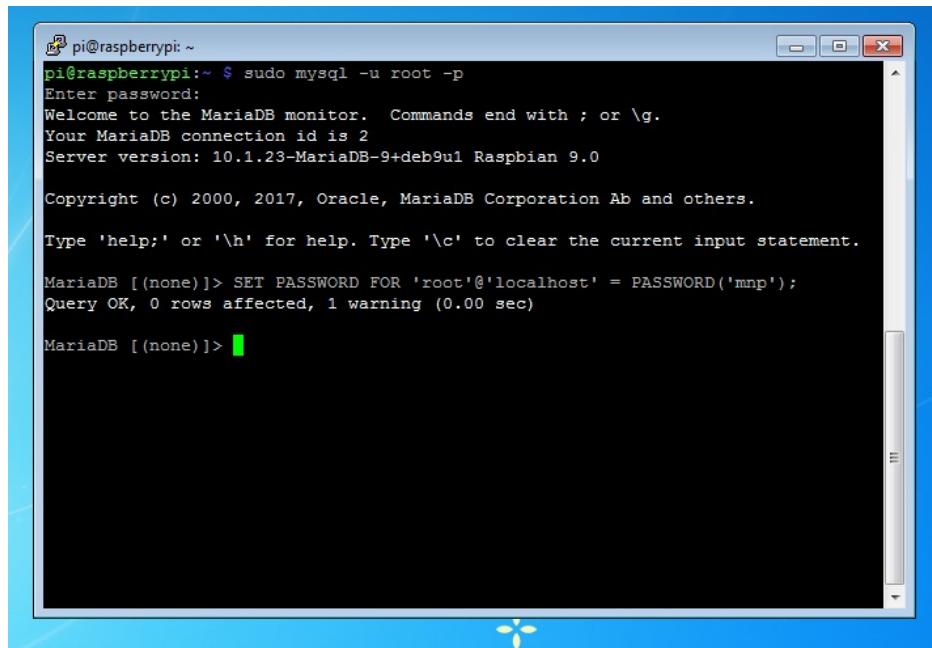
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Figure 38: Database configuration part 1

This is an optional step to create a password for the root user as show by figure 39. This will increase server security. Exit the server and restart it by typing "sudo service mysql restart" and then login to the server using the new password as show by figure 40.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window contains the following text:

```
pi@raspberrypi:~ $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

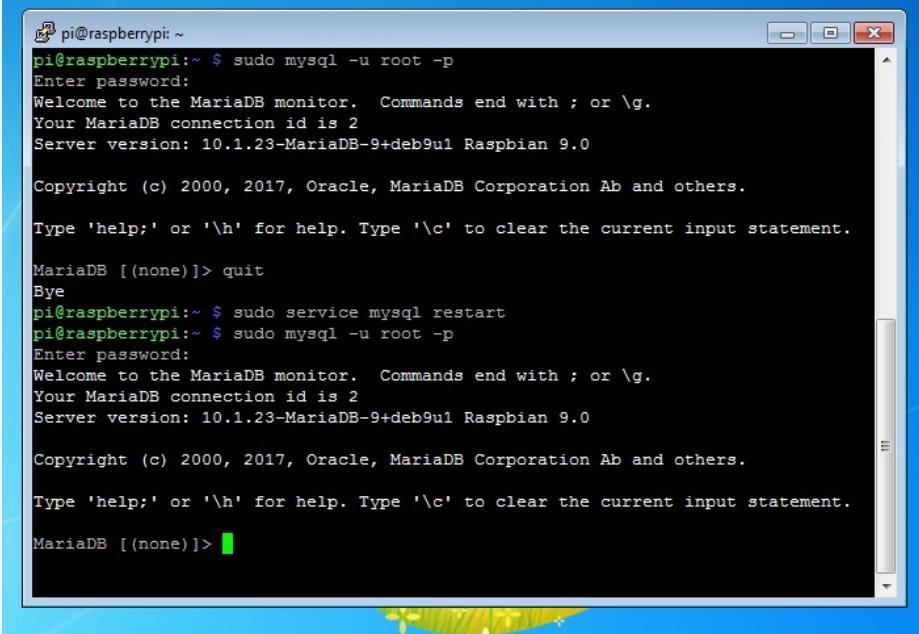
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('mnp');
Query OK, 0 rows affected, 1 warning (0.00 sec)

MariaDB [(none)]>
```

Figure 39: Database configuration part 2



```
pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> quit
Bye
pi@raspberrypi:~ $ sudo service mysql restart
pi@raspberrypi:~ $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Figure 40: Database configuration part 3

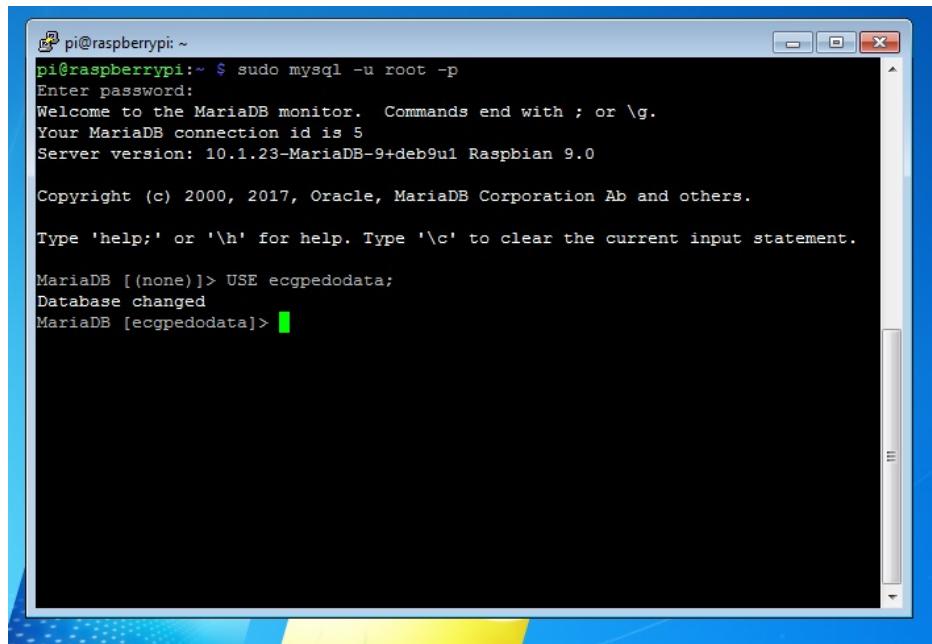
Create the "ecgpedodata" database as show by figure 41.

```
MariaDB [(none)]> create database ecgpedodata;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]>
```

Figure 41: Database configuration part 4

Change the current database by using the command "USE ecgpedodata" as show by figure 42.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window shows the MySQL monitor interface. The user has run the command `sudo mysql -u root -p` and entered their password. The MySQL prompt is at the bottom of the screen.

```
pi@raspberrypi:~ $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

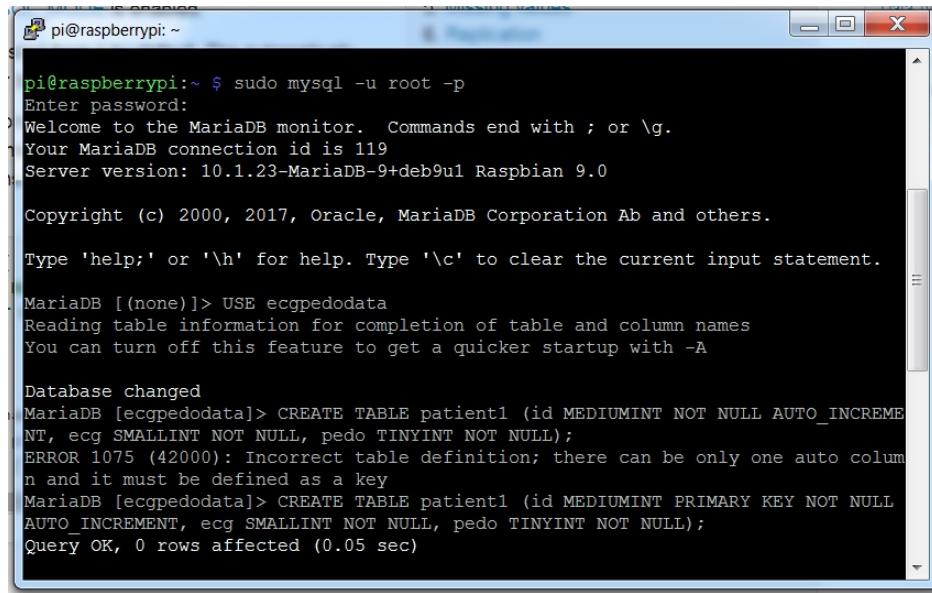
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE ecgpodata;
Database changed
MariaDB [ecgpodata]> █
```

Figure 42: Database configuration part 5

Create table "patient1" as shown by figure 43.



The screenshot shows a terminal window titled 'pi@raspberrypi: ~' running on a Raspberry Pi. The user has run the command `sudo mysql -u root -p` to connect to the MariaDB monitor. The password was entered. The prompt shows the connection details: MariaDB version 10.1.23, connection id 119, and Raspbian 9.0. The user then runs `USE ecgpedodata` to switch to the database. A warning message indicates that table completion is enabled, which can be disabled with the `-A` option. The user attempts to create a table named `patient1` with three columns: `id`, `ecg`, and `pedo`. The first attempt uses `MEDIUMINT` for all columns and `AUTO_INCREMENT` for `id`, but fails due to an error (ERROR 1075) stating that there can only be one auto column per table. The second attempt changes `id` to `PRIMARY KEY` and succeeds, returning `Query OK, 0 rows affected (0.05 sec)`.

```
pi@raspberrypi:~ $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 119
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

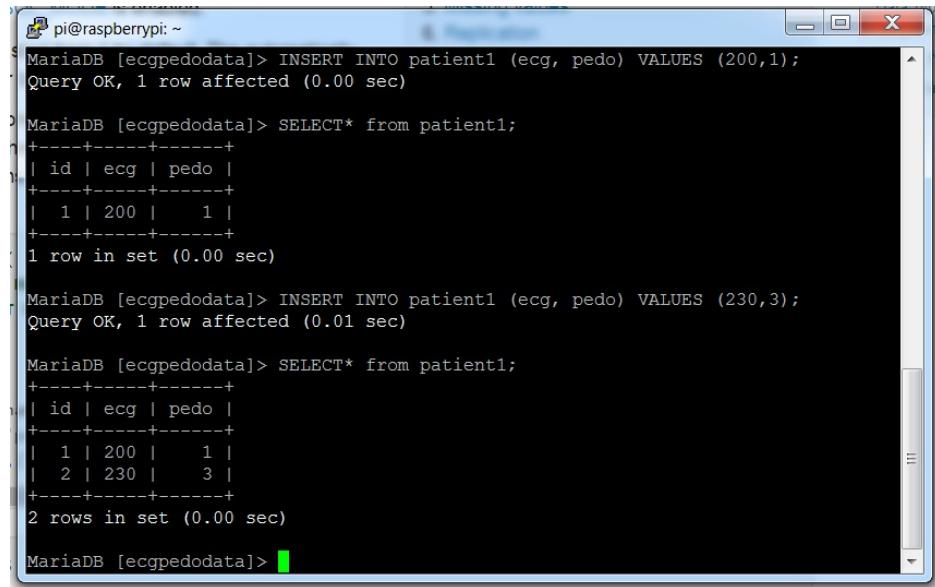
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE ecgpedodata
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [ecgpedodata]> CREATE TABLE patient1 (id MEDIUMINT NOT NULL AUTO_INCREMENT, ecg SMALLINT NOT NULL, pedo TINYINT NOT NULL);
ERROR 1075 (42000): Incorrect table definition; there can be only one auto column and it must be defined as a key
MariaDB [ecgpedodata]> CREATE TABLE patient1 (id MEDIUMINT PRIMARY KEY NOT NULL AUTO_INCREMENT, ecg SMALLINT NOT NULL, pedo TINYINT NOT NULL);
Query OK, 0 rows affected (0.05 sec)
```

Figure 43: Database configuration part 6

Use the INSERT command to enter data into the table as show by figure 44.



A terminal window titled "pi@raspberrypi: ~" showing MySQL command-line interface. The session starts with an INSERT INTO command followed by a SELECT* from patient1 query, then another INSERT INTO command, and finally another SELECT* from patient1 query. The data is displayed in a tabular format.

```
pi@raspberrypi: ~
MariaDB [ecgpedodata]> INSERT INTO patient1 (ecg, pedo) VALUES (200,1);
Query OK, 1 row affected (0.00 sec)

MariaDB [ecgpedodata]> SELECT* from patient1;
+---+---+---+
| id | ecg | pedo |
+---+---+---+
| 1 | 200 | 1 |
+---+---+---+
1 row in set (0.00 sec)

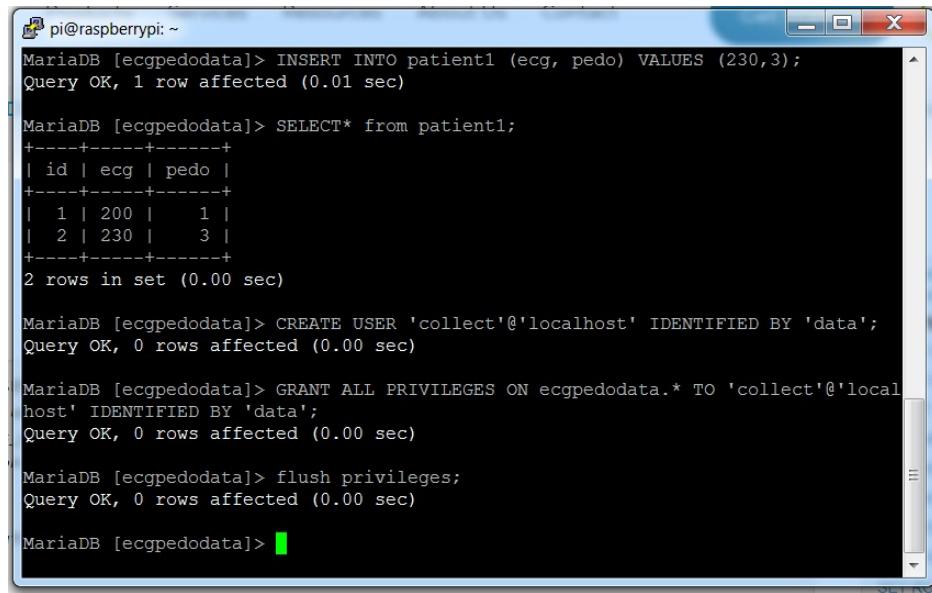
MariaDB [ecgpedodata]> INSERT INTO patient1 (ecg, pedo) VALUES (230,3);
Query OK, 1 row affected (0.01 sec)

MariaDB [ecgpedodata]> SELECT* from patient1;
+---+---+---+
| id | ecg | pedo |
+---+---+---+
| 1 | 200 | 1 |
| 2 | 230 | 3 |
+---+---+---+
2 rows in set (0.00 sec)

MariaDB [ecgpedodata]>
```

Figure 44: Database configuration part 7

Create a new user that will have access to ecgpedodata database from local-host as show by figure 45.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window contains the following MySQL session:

```
MariaDB [ecgpedodata]> INSERT INTO patient1 (ecg, pedo) VALUES (230,3);
Query OK, 1 row affected (0.01 sec)

MariaDB [ecgpedodata]> SELECT* from patient1;
+---+---+---+
| id | ecg | pedo |
+---+---+---+
| 1 | 200 | 1 |
| 2 | 230 | 3 |
+---+---+---+
2 rows in set (0.00 sec)

MariaDB [ecgpedodata]> CREATE USER 'collect'@'localhost' IDENTIFIED BY 'data';
Query OK, 0 rows affected (0.00 sec)

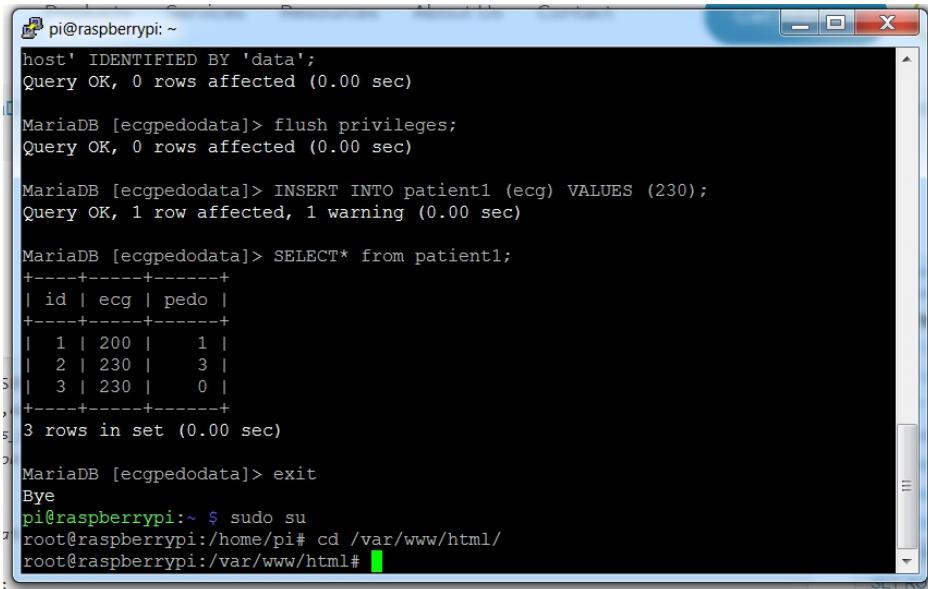
MariaDB [ecgpedodata]> GRANT ALL PRIVILEGES ON ecgpedodata.* TO 'collect'@'localhost' IDENTIFIED BY 'data';
Query OK, 0 rows affected (0.00 sec)

MariaDB [ecgpedodata]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [ecgpedodata]>
```

Figure 45: Database configuration part 8

Exit the database and navigate as superuser to /var/www/html/ directory using cd command as show by figure 46.



```
pi@raspberrypi: ~
host' IDENTIFIED BY 'data';
Query OK, 0 rows affected (0.00 sec)

MariaDB [ecgpedodata]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

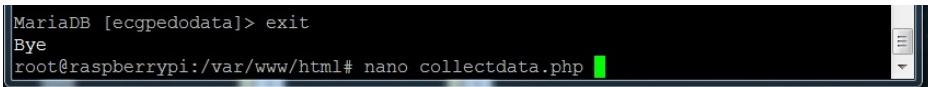
MariaDB [ecgpedodata]> INSERT INTO patient1 (ecg) VALUES (230);
Query OK, 1 row affected, 1 warning (0.00 sec)

MariaDB [ecgpedodata]> SELECT* from patient1;
+---+---+
| id | ecg | pedo |
+---+---+
| 1 | 200 | 1 |
| 2 | 230 | 3 |
| 3 | 230 | 0 |
+---+---+
3 rows in set (0.00 sec)

MariaDB [ecgpedodata]> exit
Bye
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# cd /var/www/html/
root@raspberrypi:/var/www/html#
```

Figure 46: PHP file configuration part 1

Create a 'collectdata.php' file using nano editor as show by figure 47.



```
MariaDB [ecgpedodata]> exit
Bye
root@raspberrypi:/var/www/html# nano collectdata.php
```

Figure 47: PHP file configuration part 2

Enter the PHP code as show by figure 48.

The screenshot shows a terminal window titled "pi@raspberrypi: ~" with the file "collectdata.php" open in the nano editor. The code in the file is as follows:

```
?php
$ecgString = $_POST['ecg'];
$pedo = $_POST['pedo'];

$ecgArray = explode(',',$ecgString);

$link = mysqli_connect('localhost', 'collect', 'data','ecgpedodata');
if(!$link)
{
    die('Could not connect: ' . mysql_error());
}

foreach($ecgArray as $ecg_Array)
{
    $sq = "INSERT INTO patient1 (ecg, pedo) VALUES ('$ecg_Array','$pedo');";
    $result = mysqli_query($link,$sq");
    if(!$result)
    {
        die('Invalid query: ' . mysql_error());
    }
}
mysqli_close($link);
?>
```

At the bottom of the terminal window, there is a menu bar with the following options: [Read 23 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, ^L Go To Line.

Figure 48: PHP file configuration part 3

Use an online POST generator to simulate POST request to collect-data.php as shown by figure 49. And check if the values have been entered into the table as shown by figure 50.

Figure 49: PHP file configuration part 4

```

pi@raspberrypi: ~
root@raspberrypi:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 127
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE ecgpedodata
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [ecgpedodata]> SELECT* from patient1;
+---+---+
| id | ecg | pedo |
+---+---+
| 1 | 200 | 1 |
| 2 | 230 | 3 |
| 3 | 230 | 0 |
| 4 | 250 | 2 |
| 5 | 0 | 2 |
| 6 | 36 | 2 |
| 7 | 250 | 2 |
| 8 | 36 | 2 |
+---+---+
8 rows in set (0.00 sec)

MariaDB [ecgpedodata]>

```

Figure 50: PHP file configuration part 5

Create a new user that will be able to log into the server remotely as shown by figure 51. And grant all privalages to the new user but at an IP adress not a localhost as shown by figure 52.

```

MariaDB [(none)]>
MariaDB [(none)]> CREATE USER 'proj'@'localhost' IDENTIFIED BY 'mnp';

```

Figure 51: Web application user creation part 1

```

MariaDB [(none)]> GRANT ALL PRIVILEGES ON ecgpedodata.* TO 'proj'@'192.168.0.%'
IDENTIFIED BY 'mnp';

```

Figure 52: Web application user creation part 2

This concludes the Raspberry Pi 3 set-up.