RaspberryPi - From Start to Advanced

The idea is to provide the full instructions for not only loading the Raspbian operating system on your RaspberryPi to configuring it and your PC to be able to configure it from your PC, then to adding a WebServer, then to adding a database (mySQL), and then using your web page to display data from the SQL Server. The last part includes adding PHP scripting and the use of Python programming.

The purpose is that you can take this a basic framework. Think about it, once you install and understand the basic framework, you could then add web pages, or more scripts, or use the database, etc. You could even take the next step where your PHP script fires an LED or switch on the GPIO!

Let’s get started with loading the operating system on your Pi.

## Loading the Raspbian Operating System on your RaspberryPi

### Prepare Your MicroSD card

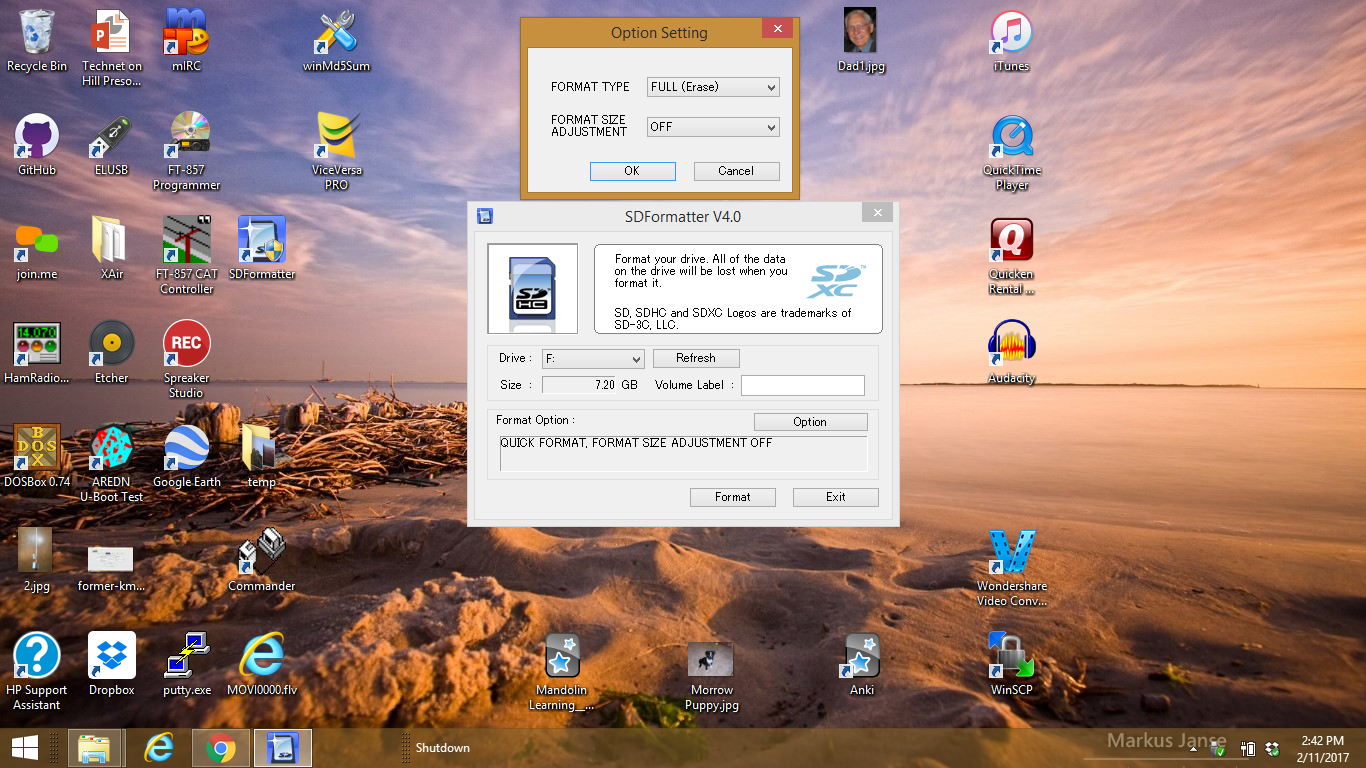
Be sure you have an 8GB card.

Download SD Formatter: <https://www.sdcard.org/downloads/formatter_4/>

Possibly install it on your desktop.

Place your SD card in the PC.

Launch SD Formatter, recommend to pick “FULL (Erase)”. Ensure you have the right drive selected! Click Format.



### Download the Operating System

Here is where to find the Pi Operating Systems: <https://www.raspberrypi.org/downloads/raspbian/>

Download the RASPBIAN JESSIE WITH PIXEL in the ZIP form, it is about 1.4 GB.

When the lower left tab indicates the file is complete, click the tab at the bottom of the browser and click “Show in folder”. Suggest that you also save the zip file on a Flash Drive for possible future use (so that you don’t have to download again). Also suggest you change the name of the zip file to include the Raspbian version number (like 4-4).

### Unzip the Operating System

Suggest that you unzip the file to your desktop

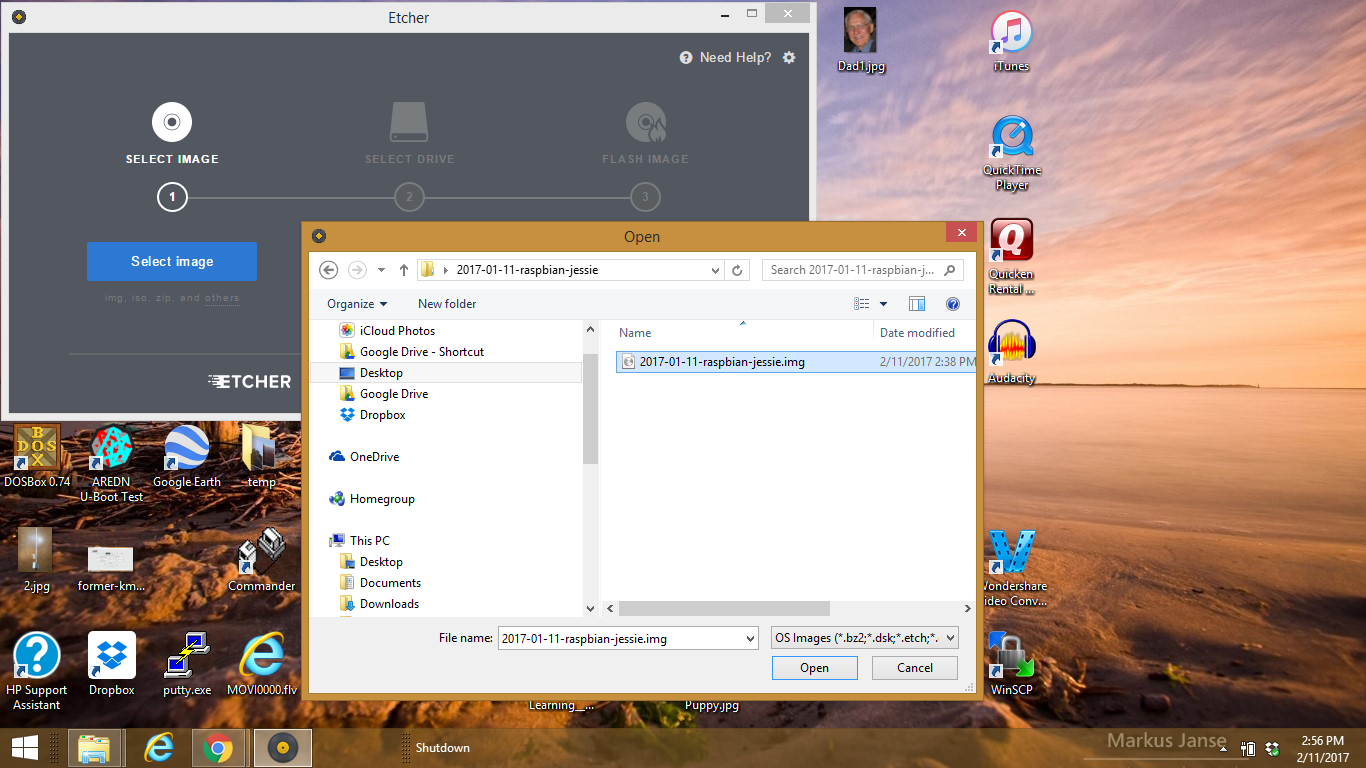
### “Burn” the Operating System to your SD card, creating a “boot image”

You can’t just copy the operating system to the SD card, you need to prepare the SD card and load it in such a way that the card is recognized as something that is booted from. You have to convert the file to an image file and burn that to the SD Card. There are several tools to do that, one used here is called “etcher”

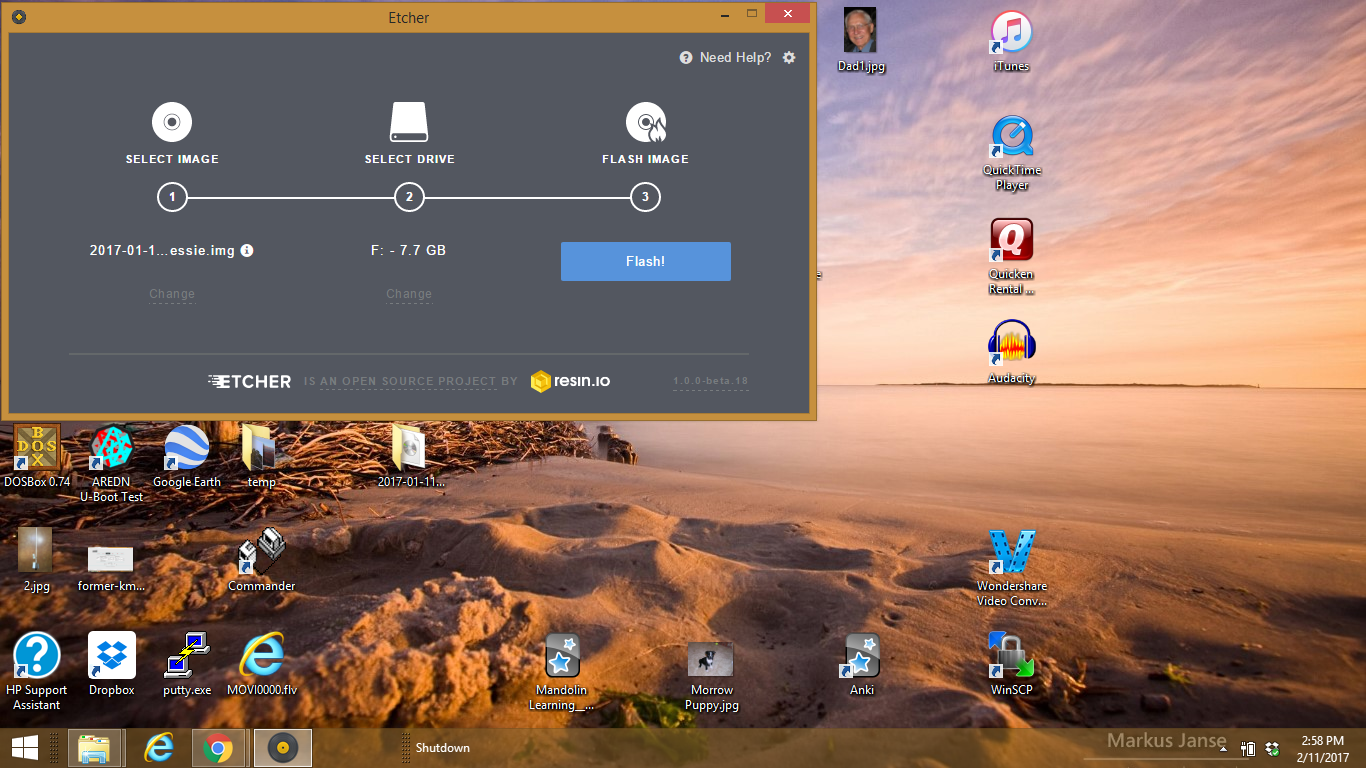
Download etcher and put it on your desktop: <https://etcher.io/>

Launch etcher.

Click “Select image” and find the Raspbian OS Image you put on the desktop, click “Open”.



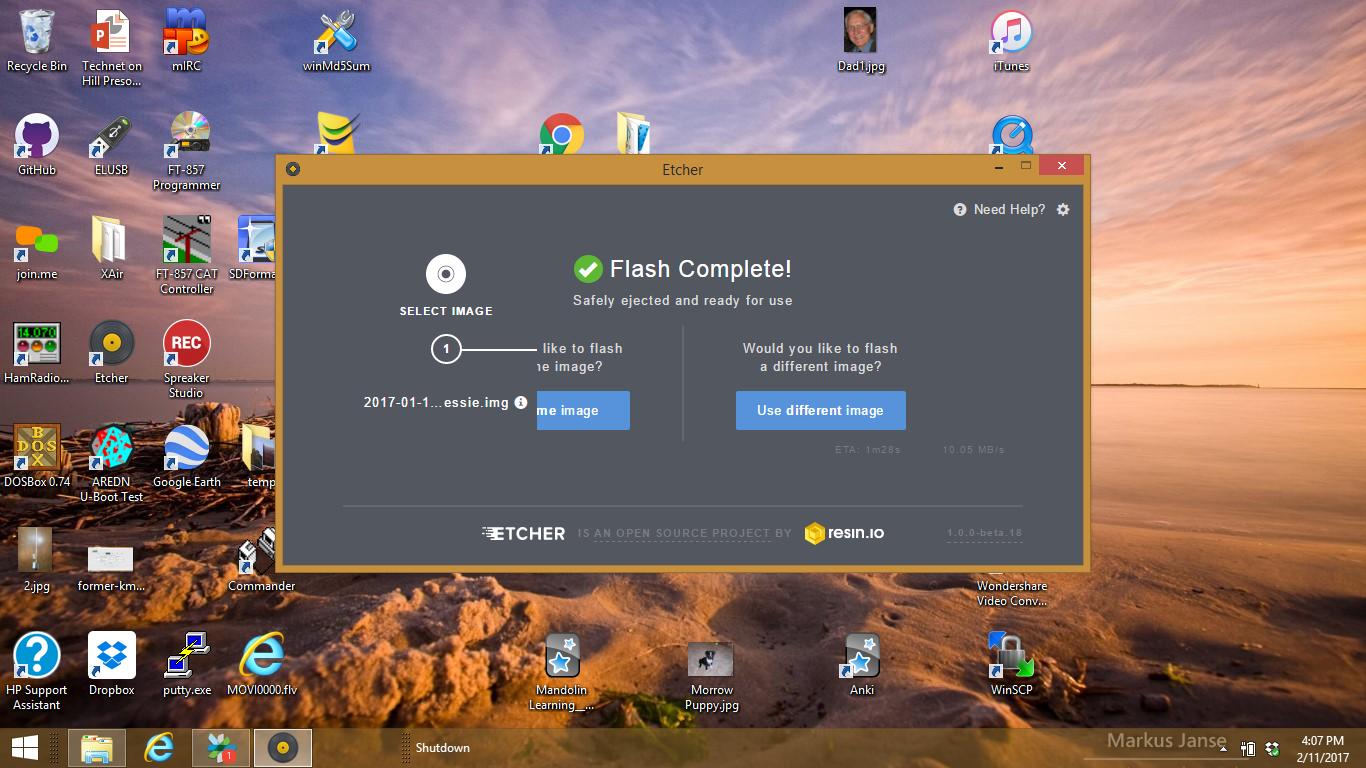
Make sure the destination (the “SELECT DRIVE, 2” is the correct drive, where the SD card is located. In this case it’s the “F” drive.



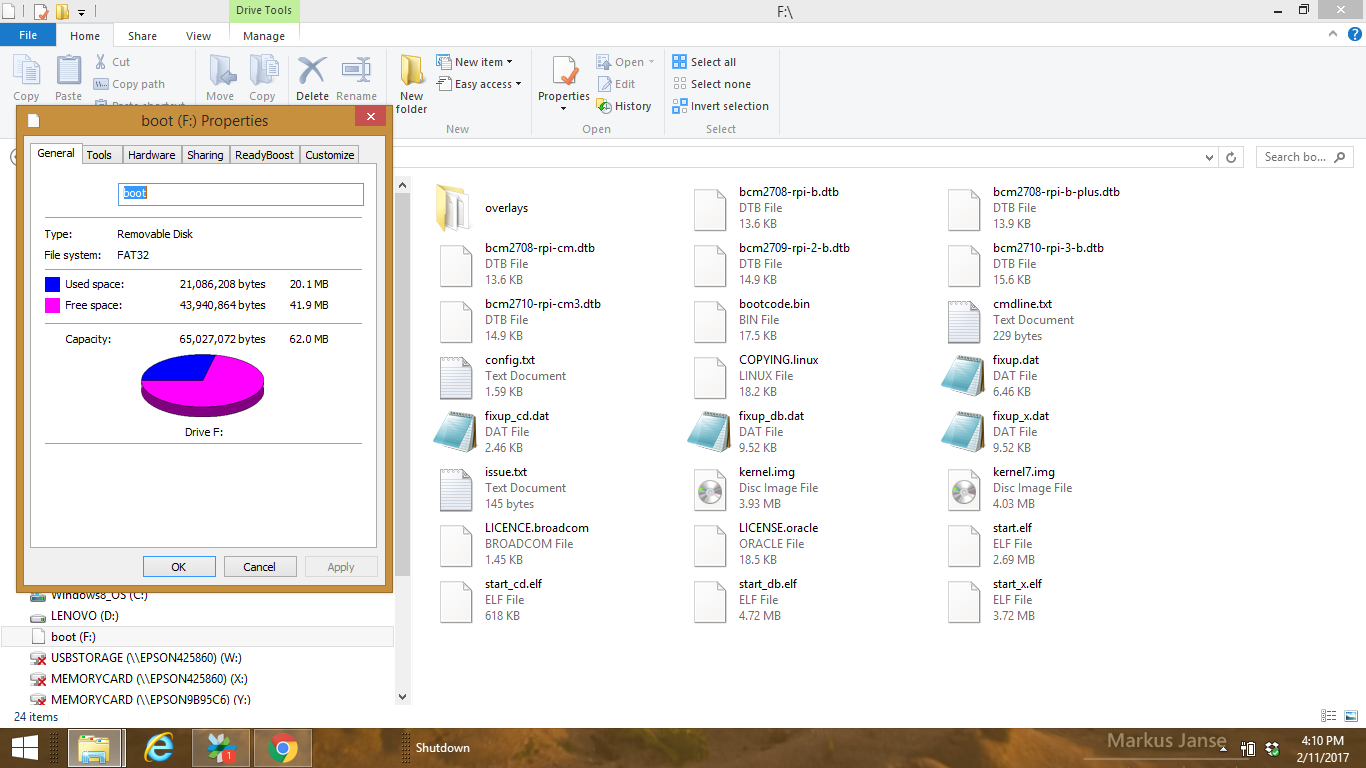
Click “Flash!”

You will see it takes some time, then it will take a long time to “verify” the image - but this is a worth step!

Once complete you’ll see “Flash Complete!”



Let’s look at the image on your SD card.



Notice all the files, included the main image boot file called “kernal.img”. This file is what the pi will boot to.

### Prepare your Raspberry pi, and boot to the MicroSD card.

Ok, take out the microSD card, make sure you’ve plugged in a monitor and also a wireless network USB fob.

Plug in the pi to your 5 volt DC source and watch it boot up.

It should say “Welcome to PIXEL” while it boots up.

You’re in!

### Initial pi Configurations

First, let’s connect the pi to the internet so we can download software we’ll need.

Either plug in a network cable or configure the pi for a wireless connection.

It is suggested to make some setting changes first, so let’s do that.

First, open up a terminal (the small icon of a screen with an arrow at the top left of your pi screen).

At the prompt you should type:

$ raspi-config

A menu comes up:

Scroll down to option 2 to change the user password. Do that, but note that when you type it looks like nothing is happening, but it is! Keep typing and then enter.

Nowdays the pi defaults to booting to the graphical user interface, let’s change that, we’ll see why later. Select “3 Boot Options”, then select “B1 Desktop / CLI”, and then select “B2 Console Autologin”. From now on it will boot to the command line.

Now choose option 7, Advanced options.

On that menu, scroll down to “A4 SSH”, and hit enter, answer “Yes” to “Would you like the SSH server to be enabled?” It then confirms it’s enabled.

Choose Advanced options again, and then select “A8 Serial”, or “Enable/Disable shell and kernel mode”, hit enter, and then answer “Yes” to “Would you like a login shell to be accessible over serial”. When you do that it confirms by saying it’s enabled. Earlier version of Raspbian had it on as default, but security concerns changed that recently. To do the next stage we need it on.

Once finished, go ahead and let it reboot, which will now be at the command line.

Now at this stage you can do whatever you want, but if you would like to utilize your windows desktop as a screen and keyboard to do the next sub projects then I highly recommend you do the next series called “Use your PC to program and configure your Raspberry Pi”. Otherwise, skip to the next sub project.

## Use your PC to program and configure your Raspberry Pi

(from the internet, or your local network).

#### Download Putty to your PC

<http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Install it on your PC desktop

There are a few steps to this, but it’s easy enough if you follow instructions. Why would you want to do this anyway? Because most of the time you are surfing on your PC for instructions on how to program your pi, why not be able to do simple “cut and paste” from your PC to the pi?” - and it will save a lot of time!

If you haven’t already enabled shell access from the serial port from above, go back and do that now!

The following instructions are pulled from this guide:

<http://www.instructables.com/id/Use-ssh-to-talk-with-your-Raspberry-Pi/>

If you need more detail go to that, otherwise, here’s the fast-track.

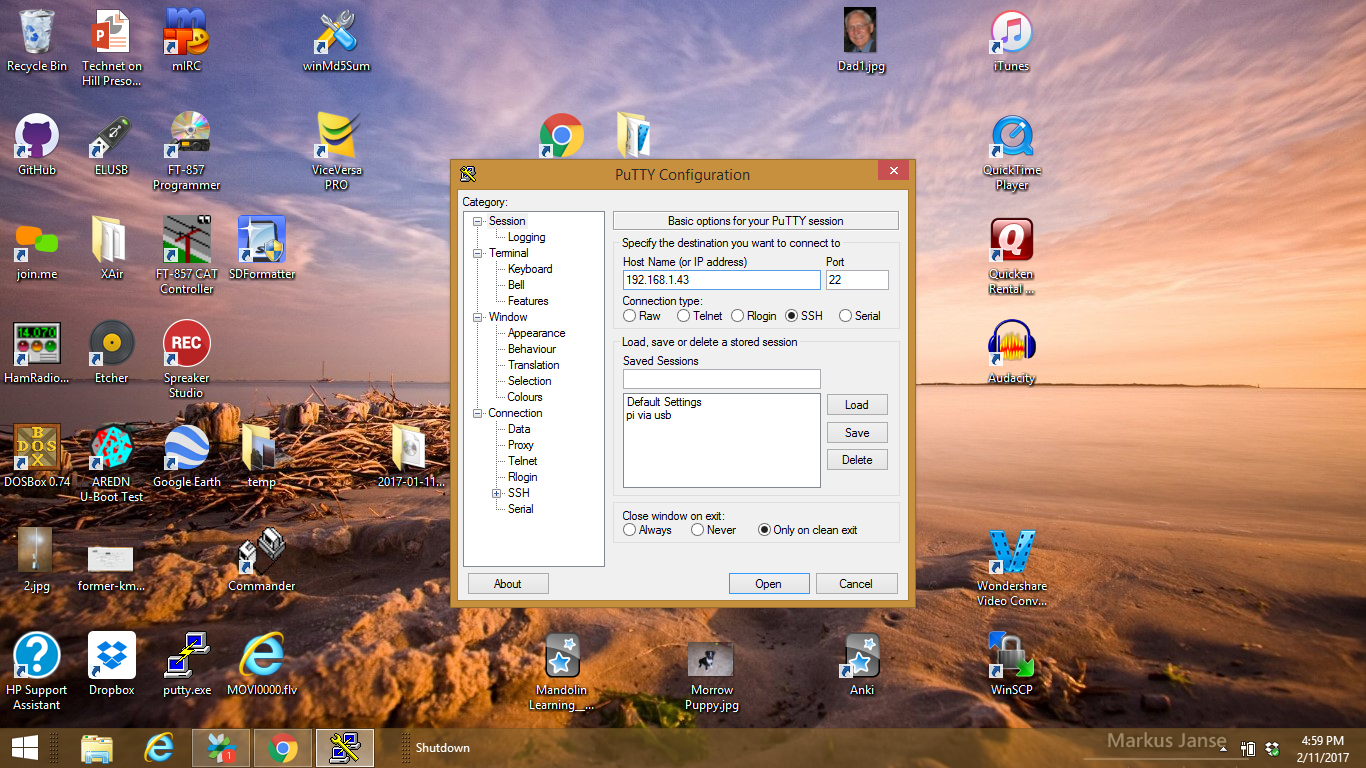
You should already be connected to the internet, so…

Type away:

$ ifconfig

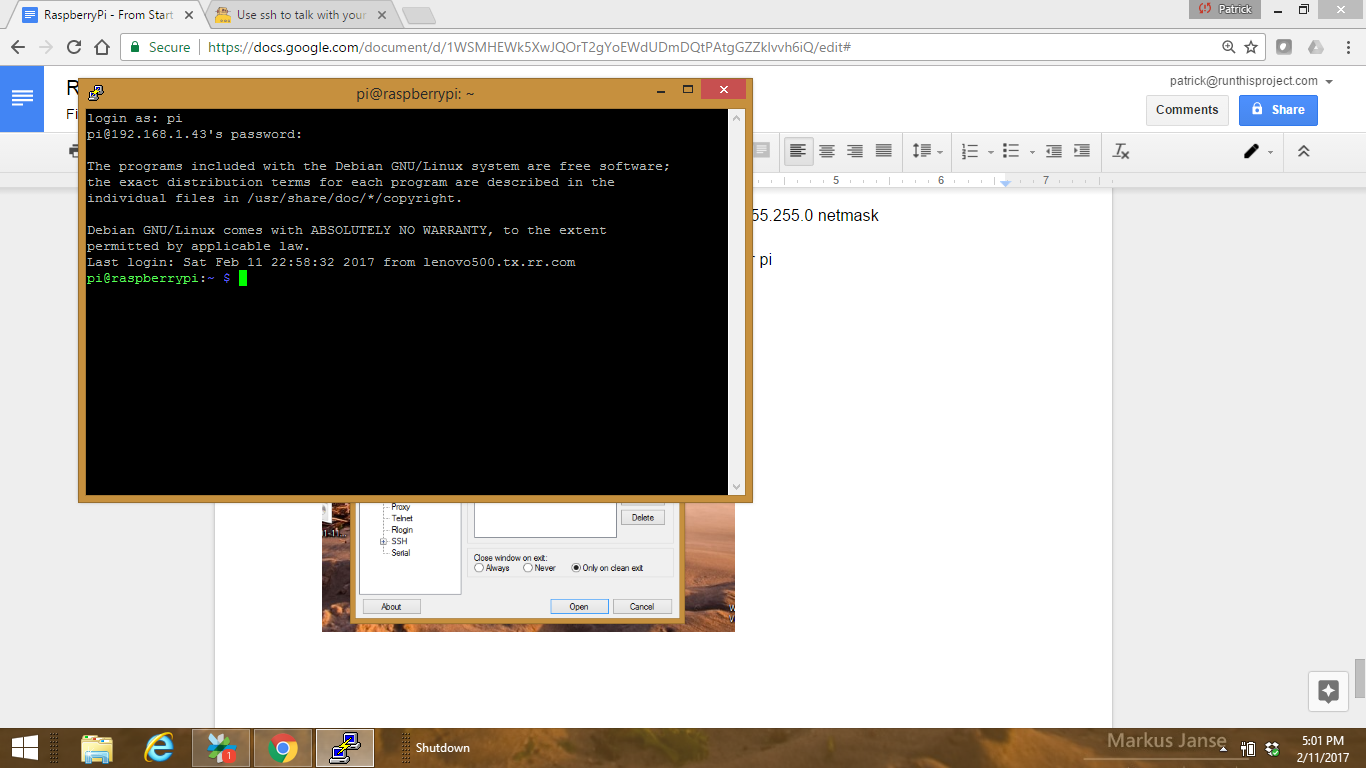
(note the “inet addr”), for example 192.168.1.43 with a 255.255.255.0 netmask

Now open putty on your pc and type in the ip address of your pi



Then click “Open” at the bottom.

You should see a warning message, just accept and proceed.

Login with username “pi”, and then with password you just changed to:

## Install Apache Web Server

Most of what is below was gleaned from the following original source: <http://www.andrew.cmu.edu/course/95-799/LAMP-Howto/lamp-linux-apache-mysql-php.html>

Run these commands to prepare for updates:

sudo gpg --keyserver pgpkeys.mit.edu --recv-key CCD91D6111A06851

sudo gpg --armor --export CCD91D6111A06851 | sudo apt-key add -

Now, install any updates for the pi:

$ sudo apt-get update

Now install Apache web server and utilites:

sudo apt-get install apache2 -y

Answer “Y” to the question of continuing.

Once this is done you should be able to browse to the server, so type in a browser:

<http://192.168.1.43/> (or the address of the pi).

#### If you want to run a web page from user directories

$ sudo a2enmod userdir

Then:

$ service apache2 restart

Now you should be able to browse to <http://192.168.1.x/~pi/>

## INSTALL PHP (for the scripting)

To allow your Apache server to process PHP files, you'll need to install PHP5 and the PHP5 module for Apache. Type the following command to install these:

sudo apt-get install php5 libapache2-mod-php5 -y

Now remove the index.html file:

sudo rm index.html

and create the file index.php:

sudo leafpad index.php

*Note: Leafpad is a graphical editor. Alternatively, use nano if you're restricted to the command line*

Put some PHP content in it:

<?php echo "hello world"; ?>

Now save and refresh your browser. You should see "hello world". This is not dynamic but still served by PHP. Try something dynamic:

<?php echo date('Y-m-d H:i:s'); ?>

or show your PHP info:

<?php phpinfo(); ?>

Create a homepage as a user  
  
If you are still root (if you have a # in your prompt), exit.  
  
Create a directory for homepages. Home directory, public\_html and all directories under public\_html must be executable (x) by all, so that web server can access files under them if it knows their name. Files must be readable by the web server.

cd $HOME  
mkdir public\_html  
echo "my homepage" > public\_html/index.html  
chmod a+x $HOME $HOME/public\_html  
chmod a+r $HOME/public\_html/index.html

## Install MySQL Server

$ sudo apt-get install mysql-server

Yes to continue.

You will be prompted to enter a password for the root user. Make sure you write this down as we will need to use this to access the MYSQL server and connect PHPMyAdmin to it.

## Sample MySQL queries using PHP

### Connect to a MySQL db via PHP

Test that MySQL is working OK with PHP.

First, create a new MySQL user and give it access to a test database so that I'm not using MySQL's root user. Heading back to the MySQL command line with:

# **mysql -uroot -p*password***

Create a new database:

mysql> **create database final;**

Create a new user with local privileges:

mysql> **create user 'pi9'@'localhost' identified by 'raspberry';**

Allow full access to the 'final’' database:

mysql> **grant all privileges on final.\* to 'pi9'@'localhost';**

Clean up:

mysql> **flush privileges;**

mysql> **quit;**

Now to create a PHP script to test the connection to the test database using the new user:

$ pico testmysql.php

# **cd /var/www**

# **nano testmysql.php**

# cd /var/www

# nano testmysql.php

d

<?php

//Define connetion variables

$servername = "localhost";

$username = "root";

$password = "xxxxxxxx";

$dbname = "final";

// Create Connection

$conn = new mysqli ($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT article, dealer, price FROM shop";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

// output data of each row

while($row = $result->fetch\_assoc()) {

Echo "Article : " . $row["article"]. " - Dealer: " . $row["dealer"]. " " .

"Price : " . $row["price"]. "<br>";

}

} else {

echo "0 results";

}

// Clean up

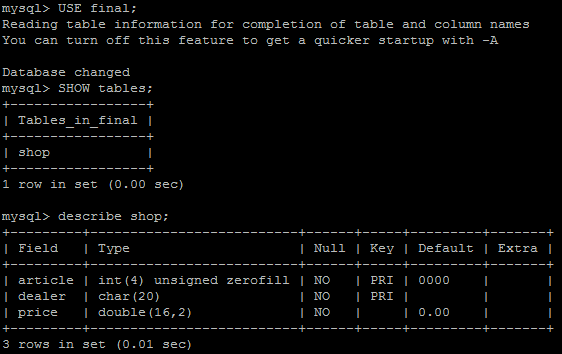
$conn->close();

?>

(then Ctrl-X, Yes)

**May have to $ mv testmysql.php /var/www/html/**

**Example:**



Common SQL commands:

mysql> SHOW DATABASES;

mysql> use DatabaseName;

mysql> SHOW TABLES;

SQL> DESCRIBE TABLENAME; (TABLE, NOT THE DATABASE!)

CREATE TABLE table\_name(

You can now enter mySQL commands to create, alter or delete databases.

Leave SQL by entering “quit”

### Write and Read to the db via PHP

## Additional References:

<http://raspberrywebserver.com/sql-databases/using-mysql-on-a-raspberry-pi.html>

<https://www.stewright.me/2014/06/tutorial-install-mysql-server-on-raspberry-pi/>

<https://www.raspberrypi.org/learning/lamp-web-server-with-wordpress/worksheet/>

<http://www.php5dp.com/get-mysql-and-php-to-work-together-in-raspberry-pi/>

## Other Help:

Code to help with a connection string:

<?php  
//Filename: IConnectInfo.php  
interface IConnectInfo  
{  
 const HOST ="localhost";  
 const UNAME ="root";  
 const PW ="yourPassword";  
 const DBNAME = "test";  
   
 public static function doConnect();  
}  
?>