Install Windows 10 software

Friday, February 10, 2017 5:38 PM

Windows Install for raspberry pi installation disk

Install 7zip
Install Win32DiskImager

Download Latest Rasperian Headless Image

Tuesday, June 9, 2015 19:29

Use 7zip to unzip it. Cannot be in directory such as Google Drive. I put it in the C:\users \<name>\directory

Run Win32DiskImager in Adminstrator mode. Left Click and Select More Option.

"ssh" back in and create to three directories in /home/pi

A. "application"
B. "python"
C. "redis"

Adding SSH capability to boot image

Monday, February 6, 2017 6:18 PM

Use text editor to store ssh in top directory of the boot partition
Use cmd to rename ssh.txt to ssh -- This enables ssh during boot up.

Boot Machine and SSH to PI

Friday, February 10, 2017 5:41 PM

Start up raspberry pi with sd card

Look on router to find ip address of the raspberry pi.

Now ssh into the pi by ssh pi@<ipaddress>
Password is raspberry.

Enter the command
raspi-config

To configure the pi.

Select the following options

1. Expand File System
2. Change password for pi account. I use the pi user name for my stuff
3. Change Host Name
4. Enable SSH
5. Enable SPI, I2C just in case

Save changes and system will reboot.

Change Password

Friday, February 10, 2017 5:44 PM

ssh pi@<< address>>
Use password raspberry

sudo passwd
Change password to desired password
I use ready2go

Add application directories in ~

Friday, February 10, 2017 5:46 PM

Setup the following directories

"redis","python", "new_python",

Changing time zone

Friday, February 17, 2017 11:51 AM

dpkg-reconfigure tzdata

Formatting usb drive to ext4

Saturday, March 25, 2017 7:47 PM

To identify usb drive do ls /dev/sd* and see which drive it is.

Remove drive to verify chosen drive

I was having a similar problem. I resolved this by first unmounting the partition via cli:
`sudo umount /dev/sdb1 (or whatever your usb is)`
Then formatting the partition for ext4 (it cannot be formatted while mounted/in use):
`sudo mkfs.ext4 /dev/sdb1`
That should do the trick, worked for me.

From <http://askubuntu.com/questions/149984/formatting-usb-flash-memory-for-ubuntu-ext4>

Restore to vfat

```
sudo mkdosfs -F 32 -I /dev/sdc1
```

From <https://www.garron.me/en/go2linux/format-usb-drive-fat32-file-system-ubuntu-linux.html>

Permanently mounting usb drive

Saturday, March 25, 2017 8:12 PM

Use
sudo blkid to identify UUID of usb drive

From <http://posts.danharper.me/raspberry-pi-2-auto-mount-usb/>

Sudo nano /etc/fstab
Add following line

```
UUID="fb840100-a8f0-4b55-b821-6d97e4221363"  /home/pi/usb_drv   ext4    defaults  0  0
```

Setting up LAN to Static address

Friday, February 10, 2017 11:41 AM

Execute sudo nano /etc/dhcpcd.conf and change the appropriate settings

pi@irrigation_controller_2:/etc $ cat dhcpcd.conf
A sample configuration for dhcpcd.
See dhcpcd.conf(5) for details.

Allow users of this group to interact with dhcpcd via the control socket.
#controlgroup wheel

Inform the DHCP server of our hostname for DDNS.
hostname

Use the hardware address of the interface for the Client ID.
clientid
or
Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per RFC4361.
#duid

Persist interface configuration when dhcpcd exits.
persistent

Rapid commit support.
Safe to enable by default because it requires the equivalent option set
on the server to actually work.
option rapid_commit

A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
Most distributions have NTP support.
option ntp_servers
Respect the network MTU.
Some interface drivers reset when changing the MTU so disabled by default.
#option interface_mtu

A ServerID is required by RFC2131.
require dhcp_server_identifier

Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

A hook script is provided to lookup the hostname if not set by the DHCP
server, but it should not be run by default.
nohook lookup-hostname
interface eth0

```
static ip_address=192.168.1.84/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1

interface wlan0

static ip_address=192.168.1.85/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
pi@irrigation_controller_2:/etc $
```

Setting up WIFI

Friday, February 10, 2017 11:47 AM

Instructions taken from https://www.maketecheasier.com/setup-wifi-on-raspberry-pi/

command line set up

If you aren't using the desktop then the WiFi can be configured using the command line. Raspbian should come with all the correct packages pre-installed but if any of the commands or files mentioned below aren't available, then run this command to install them:

```
sudo apt-get install wpasupplicant wireless-tools
```

The general network settings are configured in "/etc/network/interfaces" while the Wi-Fi details are set in the "/etc/wpa_supplicant/wpa_supplicant.conf" file. First edit the "interfaces" file:

```
sudo nano /etc/network/interfaces
```

Ensure that the section about `wlan0` (typically found at the end of the file) reads as follows:

```
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

If there are difference then change them to accordingly. Don't alter any of the lines about the `lo` adapter or the `eth0` adapter. Press "CTRL + X" to exit nano (press Y and then press ENTER when prompted).

To get a list of the currently available wireless networks, use the `iwlist` command:

```
sudo iwlist wlan0 scan
```

If there is too much information, use `grep` to find the fields you need. For example to see just the ESSIDs, use:

```
sudo iwlist wlan0 scan | grep ESSID
```

```
pi@raspberrypi ~ $ sudo iwlist wlan0 scan | grep ESSID
                    ESSID:"Gary"
                    ESSID:"Gary2"
                    ESSID:"Gary3"
                    ESSID:"SRX-WR150WH"
```

Pick a network and add the network authentication information in the "wpa_supplicant.conf" file:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

The first two lines should already read:

<<<< On my WIFI setup the country code was GB, Change it to US>>>>

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
```

Now add the following:

```
network={
    ssid="YourSSID"
    psk="password"
    key_mgmt=WPA-PSK
```

```
}
```

If your router is configured using WEP for encryption then the network information will look like this:

```
network={
    ssid="YourSSID"
    wep_key0="password12345"
    key_mgmt=NONE
}
```

For those of you familiar with advanced WiFi configurations, the network information can also include the following fields:

- proto – Protocol type can be: RSN (for WP2) and WPA (for WPA1).
- pairwise – CCMP or TKIP (for WPA2 or WPA1).
- auth_alg – authentication algorithm, can be OPEN for both WPA1/WPA2 and less commonly SHARED or LEAP.

Press “CTRL + X” to exit nano and save the file, press Y and then press ENTER when prompted. Finally reboot your Pi:

```
sudo reboot
```

You can check the status of the wireless connection using `ifconfig` (to see if `wlan0` has acquired an IP address) and `iwconfig` to check which network the wireless adapter is using.

```
pi@raspberrypi ~ $ iwconfig
wlan0     IEEE 802.11bgn  ESSID:"Gary"  Nickname:"<WIFI@REALTEK>"
          Mode:Managed  Frequency:2.412 GHz  Access Point: 72:4C:A9:A2:A0:E8
          Bit Rate:135 Mb/s   Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Power Management:off
          Link Quality=98/100  Signal level=100/100  Noise level=0/100
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

lo        no wireless extensions.

eth0      no wireless extensions.
```

If you have any questions about wireless on the Raspberry Pi, please ask them in the comments and we will see if we can help.

From <https://www.maketecheasier.com/setup-wifi-on-raspberry-pi/>

From <https://www.maketecheasier.com/setup-wifi-on-raspberry-pi/>

Misc System Setup

Friday, February 10, 2017 5:27 PM

I find this pi command useful
ps -aux | grep python

I use if to find processes which are running python
Can be generalized for other strings

Installing NTPD Service

Wednesday, June 10, 2015
19:13

sudo apt-get install ntpdate

Friday, February 10, 2017 5:29 PM

How to recover a lost password

Thursday, June 4, 2015
14:46

Step 1 – Grab The SD Card

Power down the Pi and remove the SD card. Insert it into your PC.

Step 2 – Edit cmdline.txt

The boot partition should be visible and contain a file named “cmdline.txt”. Edit this file in a text editor and add the following to the end of the existing text :

```
init=/bin/sh
```

If the original content was :

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

it should now look like :

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait init=/bin/sh
```

Make sure it is all one line! Save the text file and eject the SD card from the PC.

Step 3 – Reset the Password

Insert the card into a Pi that is connected to a monitor and keyboard. Power up the Pi. There may be a delay but you should be presented with a flashing cursor.

At the prompt type the following command :

```
passwd pi
```

You will then be prompted for a new password. Enter it carefully and press the [Return] key. It will now

ask you to retype the password.

```
passwd pi
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

The password has been changed.

Now type the following commands :

`sync exec /sbin/init`

The Pi will continue to boot and return you to the normal command line prompt.

Shutdown the Pi and power it off.

`sudo halt`

Step 4 – Edit cmdline.txt

Using the PC edit the “cmdline.txt” file again and remove the “init=/bin/sh” text you added in Step 2.

You can now return the SD card to your Pi, reboot and use the new password.

internal temperature of Raspberry PI 2

Tuesday, June 9, 2015
19:26

```
#!/bin/bash cpuTemp0=$(cat /sys/class/thermal/thermal_zone0/temp) cpuTemp1=
$(($cpuTemp0/1000)) cpuTemp2=$(($cpuTemp0/100)) cpuTempM=$(($cpuTemp2 % $cpuTemp1)) echo
CPU temp"="$cpuTemp1"."$cpuTempM"'C" echo GPU $(/opt/vc/bin/vcgencmd measure_temp)
```

RPI vcgencmd usage

Verified commands

Command outputs are from firmware version 362371.

`vcgencmd commands`

Shows a list of possible commands.

```
root@raspberrypi:~# vcgencmd commands commands="vcos, ap_output_control,
ap_output_post_processing, vchi_test_init, vchi_test_exit, pm_set_policy, pm_get_status,
pm_show_stats, pm_start_logging, pm_stop_logging, version, commands, set_vll_dir,
led_control, set_backlight, set_logging, get_lcd_info, set_bus_arbiter_mode, cache_flush,
otp_dump, codec_enabled, measure_clock, measure_volts, measure_temp, get_config,
hdmi_ntsc_freqs, render_bar, disk_notify, inuse_notify, sus_suspend, sus_status,
sus_is_enabled, sus_stop_test_thread, egl_platform_switch, mem_validate, mem_oom,
mem_reloc_stats, file, vctest_memmap, vctest_start, vctest_stop, vctest_set, vctest_get"
```

(I've just checked on a more recent firmware version and there's now also `get_camera`, `get_mem` and `hdmi_status_show` commands)

`vcgencmd measure_clock <clock>`

Shows clock frequency, clock can be one of arm, core, h264, isp, v3d, uart, pwm, emmc, pixel, vec, hdmi, dpi.

```
root@raspberrypi:~# \ > for src in arm core h264 isp v3d uart pwm emmc pixel vec hdmi dpi ; do
\ > echo -e "$src:\t$(vcgencmd measure_clock $src)" ; \ > done arm: frequency(45)=700000000
core: frequency(1)=250000000 h264: frequency(28)=0 isp: frequency(42)=250000000 v3d:
frequency(43)=250000000 uart: frequency(22)=3000000 pwm: frequency(25)=0 emmc:
frequency(47)=100000000 pixel: frequency(29)=154000000 vec: frequency(10)=0 hdmi:
```

frequency(9)=163682000 dpi: frequency(4)=0

vcgencmd measure_volts <*id*>

Shows voltage. id can be one of core, sdram_c, sdram_i, sdram_p, and defaults to core if not specified.

root@raspberrypi:~# \ > for id in core sdram_c sdram_i sdram_p ; do \ > echo -e "$id:\t $(vcgencmd measure_volts $id)" ; \ > done core: volt=1.20V sdram_c: volt=1.20V sdram_i: volt= 1.20V sdram_p: volt=1.23V

vcgencmd measure_temp

Shows core temperature of BCM2835 SoC.

root@raspberrypi:~# vcgencmd measure_temp temp=42.8'C

vcgencmd codec_enabled <*codec*>

Shows if the specified codec is enabled, codec can be one of H264, MPG2, WVC1, MPG4, MJPG, WMV9. Please note this was run on a Pi with the MPG2 and VC1 licences enabled.

root@raspberrypi:~# \ > for codec in H264 MPG2 WVC1 MPG4 MJPG WMV9 ; do \ > echo -e "$codec:\t$(vcgencmd codec_enabled $codec)" ; \ > done H264: H264=enabled MPG2: MPG2 =enabled WVC1: WVC1=enabled MPG4: MPG4=enabled MJPG: MJPG=enabled WMV9: WMV9=enabled

If you also follow the instructions in this Forum post then the codec options for VP6 and VP8 (WEBM) also work, this is experimental and unsupported at present so any changes are at your own risk.

vcgencmd get_config *[config|int|str]* Will print the configurations you have set. Argument can ether be a specific option or int, showing all configs with number-datatype, or str showing all configurations with datatype sting (aka text).

root@raspberrypi:~# vcgencmd get_config int arm_freq=1000 core_freq=500 sdram_freq=600 over_voltage=6 disable_overscan=1 force_pwm_open=1

vcgencmd get_mem *arm/gpu*

Shows how much memory is split between the CPU (arm) and GPU.

```
root@raspberrypi:~# vcgencmd get_mem arm && vcgencmd get_mem gpu arm=448M gpu=
64M
```

`vcgencmd version`

Shows the firmware version

```
root@raspberrypi:~# vcgencmd version Jan 13 2013 16:24:29 Copyright (c) 2012 Broadcom
version 362371 (release)
```

`vcgencmd otp_dump`

Displays the contents of the OTP (One Time Programmable) memory embedded inside the SoC.

```
root@raspberrypi:~# vcgencmd otp_dump 08:00000000 09:00000000 10:00000000
11:00000000 12:00000000 13:00000000 14:00000000 15:00000000 16:00280000 17:1020000a
18:1020000a 19:ffffffff 20:ffffffff 21:ffffffff 22:ffffffff 23:ffffffff 24:ffffffff 25:ffffffff 26:ffffffff
27:0000c2c2 28:6c14ba0d 29:93eb45f2 30:0000000e 31:00000000 32:00000000 33:00000000
34:00000000 35:00000000 36:00000000 37:00000000 38:00000000 39:00000000 40:00000000
41:00000000 42:00000000 43:00000000 44:00000000 45:00000000 46:00000000 47:00000000
48:00000000 49:00000000 50:00000000 51:00000000 52:00000000 53:00000000 54:00000000
55:00000000 56:00000000 57:00000000 58:00000000 59:00000000 60:00000000 61:00000000
62:00000000 63:00000000 64:00000000
```

Locations 28 and 30 store the *Serial* and *Revision* values that get displayed by /proc/cpuinfo (the *Serial* is also used to determine the Ethernet MAC address on Model B boards), and location 32 stores the value of the warranty bit. Purpose of values in other locations is unknown.

`vcgencmd set_backlight`

Currently unusable, might be used in the future to control the backlight of LCD displays

`vcgencmd render_bar`

Debug function created by Dom, used in OMXPlayer

Parameters and function of other vcgencmd commands are not known.

Changing Timezone

Friday, February 10, 2017 5:35 PM

Wednesday, June 10, 2015
19:14

sudo tzselect

or

much better

dpkg-reconfigure tzdata

Tensor flow required packages

Friday, February 10, 2017 5:36 PM

sudo apt-get install -y autoconf

Linker optimiumization flags
OPTFLAGS="-Os
mfpu= neon-vfpv4
-funsafe-math-optimization
-ftree-vectorize"

Install GIT

Friday, February 10, 2017 11:53 AM

"sudo apt-get install git"

Installing pip

Thursday, October 15, 2015 13:51

PIP

Not all Python packages are available in the Raspbian archives, and those that are can sometimes be out-of-date. If you can't find a suitable version in the Raspbian archives, you can install packages from the Python Package Index (PyPI). To do so, use the `pip` tool.

`pip` is installed by default in Raspbian Jessie (but not Raspbian Wheezy or Jessie Lite). You can install it with `apt`:

```
sudo apt-get install python3-pip
```

To get the Python 2 version:

```
sudo apt-get install python-pip
```

`pip3` installs modules for Python 3, and `pip` installs modules for Python 2.

Uninstall Python modules with `pip3 uninstall` or `pip uninstall`.

From <https://www.raspberrypi.org/documentation/linux/software/python.md>

From <https://www.raspberrypi.org/documentation/linux/software/python.md>

Install Python Basics

Friday, February 10, 2017 11:57 AM

As we saw in the previous step's file output, the Raspberry Pi comes with several Python packages already installed. We need to supplement it with a few more prerequisites. In the terminal window, run this command:

```
1 sudo apt-get install build-essential python-dev python-distlib python-setuptools python-pip
  python-wheel libzmq-dev libgdal-dev
```

The build-essential package is required for building Debian packages; python-dev, python-distlib, and python-setuptools provide several Python development and packaging tools; python-pip and python-wheel are useful for installing Python packages; libzmq-dev is needed for Jupyter notebooks; libgdal-dev is needed for geospatial analysis with geopandas.

Install Pandas dependencies

Wednesday, June 10, 2015 19:12

Pandas has several recommended and optional dependencies that unlock functionality or provide significant performance enhancements. To install them all, run the following two commands:

```
1 sudo apt-get install xsel xclip libxml2-dev libxslt-dev python-lxml python-h5py python-numexpr
  python-dateutil python-six python-tz python-bs4 python-html5lib python-openpyxl python-tables
  python-xlrd python-xlwt cython python-sqlalchemy python-xlsxwriter python-jinja2 python-boto
  python-gflags python-googleapi python-httplib2 python-zmq libspatialindex-dev
  sudo pip install bottleneck rtree
```

Install Scientific Stack

Friday, February 10, 2017 12:05 PM

Step 6: Install the scientific Python stack

Fortunately we can use apt-get to install all the massive, complex packages that make up the Python scientific stack without having to compile everything. This makes the process much, much faster.

```
1 sudo apt-get install python-numpy python-matplotlib python-mpltoolkits.basemap python-scipy
  python-sklearn python-statsmodels python-pandas
```

If you need a specific version of these packages or want a more up-to-date version than exists in the Debian repositories, you can use pip to install it, but be prepared for a slow compilation process.

From <http://geoffboeing.com/2016/03/scientific-python-raspberry-pi/>

Install Python Redis Client

Friday, February 10, 2017 12:06 PM

Install Redis Python Client

"sudo apt-get install python-redis"

Install Rabbitmq Client

Friday, February 10, 2017 12:07 PM

Install rabbitmq python client
"sudo apt-get install python-pika"

Installing Flask -- web server

Friday, February 10, 2017 12:15 PM

```
sudo pip install Flask
```

From <http://flask.pocoo.org/docs/0.12/installation/>

`sudo pip install Jinja2`

sudo pip install Werkzeug

For voice recognition install this module

```
sudo pip install flask-ask
```

From <https://alexatutorial.com/flask-ask/getting_started.html>

Download and Extract the Source Code

Friday, February 10, 2017 12:31 PM

Since we won't need to keep the source code that we'll compile long term (we can always re-download it), we will build in the ~/redis directory. Let's move there now:

- cd ~/redis

Now, download the latest stable version of Redis. This is always available at a stable download URL:

- curl -O http://download.redis.io/redis-stable.tar.gz

Unpack the tarball by typing:

- tar xzvf redis-stable.tar.gz

Move into the Redis source directory structure that was just extracted:

- cd redis-stable

Build and Install Redis

Friday, February 10, 2017 6:15 PM

Now, we can compile the Redis binaries by typing:

- make

After the binaries are compiled, run the test suite to make sure everything was built correctly. You can do this by typing:

Tcl needs to be installed first
sudo apt-get tcl
sudo apt-get install tk8.5

-edit : may also need to

sudo apt-get remove tk8.4 tcl8.4

From <https://ubuntuforums.org/showthread.php?t=1592317>

- make test

This will typically take a few minutes to run. Once it is complete, you can install the binaries onto the system by typing:

- sudo make install

>

Configure Redis

Friday, February 10, 2017 6:15 PM

Now that Redis is installed, we can begin to configure it.

To start off, we need to create a configuration directory. We will use the conventional /etc/redisdirectory, which can be created by typing:

- sudo mkdir /etc/redis

Now, copy over the sample Redis configuration file included in the Redis source archive:

- sudo nano /etc/redis/redis.conf

Configure the redis file in the following manner

```
pi@irrigation_controller_2:/etc/redis $ cat redis.conf
# Redis configuration file example.
#
# Note that in order to read the configuration file, Redis must be
# started with the file path as first argument:
#
# ./redis-server /path/to/redis.conf

# Note on units: when memory size is needed, it is possible to specify
# it in the usual form of 1k 5GB 4M and so forth:
#
# 1k => 1000 bytes
# 1kb => 1024 bytes
# 1m => 1000000 bytes
# 1mb => 1024*1024 bytes
# 1g => 1000000000 bytes
# 1gb => 1024*1024*1024 bytes
#
# units are case insensitive so 1GB 1Gb 1gB are all the same.

################################## INCLUDES ###################################

# Include one or more other config files here.  This is useful if you
# have a standard template that goes to all Redis servers but also need
# to customize a few per-server settings.  Include files can include
# other files, so use this wisely.
#
# Notice option "include" won't be rewritten by command "CONFIG REWRITE"
```

```
# from admin or Redis Sentinel. Since Redis always uses the last processed
# line as value of a configuration directive, you'd better put includes
# at the beginning of this file to avoid overwriting config change at runtime.
#
# If instead you are interested in using includes to override configuration
# options, it is better to use include as the last line.
#
# include /path/to/local.conf
# include /path/to/other.conf

################################## NETWORK #####################################

# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all the network interfaces available on the server.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
#
# Examples:
#
# right now we are not restricting ip address
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
#
# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only into
# the IPv4 lookback interface address (this means Redis will be able to
# accept connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#bind 127.0.0.1

# Protected mode is a layer of security protection, in order to avoid that
# Redis instances left open on the internet are accessed and exploited.
#
# When protected mode is on and if:
#
# 1) The server is not binding explicitly to a set of addresses using the
#    "bind" directive.
# 2) No password is configured.
#
# The server only accepts connections from clients connecting from the
# IPv4 and IPv6 loopback addresses 127.0.0.1 and ::1, and from Unix domain
# sockets.
#
# By default protected mode is enabled. You should disable it only if
# you are sure you want clients from other hosts to connect to Redis
# even if no authentication is configured, nor a specific set of interfaces
# are explicitly listed using the "bind" directive.
```

```
protected-mode yes

# Accept connections on the specified port, default is 6379 (IANA #815344).
# If port 0 is specified Redis will not listen on a TCP socket.
port 6379

# TCP listen() backlog.
#
# In high requests-per-second environments you need an high backlog in order
# to avoid slow clients connections issues. Note that the Linux kernel
# will silently truncate it to the value of /proc/sys/net/core/somaxconn so
# make sure to raise both the value of somaxconn and tcp_max_syn_backlog
# in order to get the desired effect.
tcp-backlog 511

# Unix socket.
#
# Specify the path for the Unix socket that will be used to listen for
# incoming connections. There is no default, so Redis will not listen
# on a unix socket when not specified.
#
# unixsocket /tmp/redis.sock
# unixsocketperm 700

# Close the connection after a client is idle for N seconds (0 to disable)
timeout 0

# TCP keepalive.
#
# If non-zero, use SO_KEEPALIVE to send TCP ACKs to clients in absence
# of communication. This is useful for two reasons:
#
# 1) Detect dead peers.
# 2) Take the connection alive from the point of view of network
#    equipment in the middle.
#
# On Linux, the specified value (in seconds) is the period used to send ACKs.
# Note that to close the connection the double of the time is needed.
# On other kernels the period depends on the kernel configuration.
#
# A reasonable value for this option is 300 seconds, which is the new
# Redis default starting with Redis 3.2.1.
tcp-keepalive 300

################################# GENERAL #####################################

# By default Redis does not run as a daemon. Use 'yes' if you need it.
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.
daemonize no

# If you run Redis from upstart or systemd, Redis can interact with your
# supervision tree. Options:
#   supervised no      - no supervision interaction
```

```
#   supervised upstart - signal upstart by putting Redis into SIGSTOP mode
#   supervised systemd - signal systemd by writing READY=1 to $NOTIFY_SOCKET
#   supervised auto    - detect upstart or systemd method based on
#                        UPSTART_JOB or NOTIFY_SOCKET environment variables
# Note: these supervision methods only signal "process is ready."
#       They do not enable continuous liveness pings back to your supervisor.
supervised no

# If a pid file is specified, Redis writes it where specified at startup
# and removes it at exit.
#
# When the server runs non daemonized, no pid file is created if none is
# specified in the configuration. When the server is daemonized, the pid file
# is used even if not specified, defaulting to "/var/run/redis.pid".
#
# Creating a pid file is best effort: if Redis is not able to create it
# nothing bad happens, the server will start and run normally.
pidfile /var/run/redis_6379.pid

# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel notice

# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
# output for logging but daemonize, logs will be sent to /dev/null
logfile ""

# To enable logging to the system logger, just set 'syslog-enabled' to yes,
# and optionally update the other syslog parameters to suit your needs.
# syslog-enabled no

# Specify the syslog identity.
# syslog-ident redis

# Specify the syslog facility. Must be USER or between LOCAL0-LOCAL7.
# syslog-facility local0

# Set the number of databases. The default database is DB 0, you can select
# a different one on a per-connection basis using SELECT <dbid> where
# dbid is a number between 0 and 'databases'-1
databases 16

################################ SNAPSHOTTING  ################################
#
# Save the DB on disk:
#
#   save <seconds> <changes>
#
```

```
#   Will save the DB if both the given number of seconds and the given
#   number of write operations against the DB occurred.
#
#   In the example below the behaviour will be to save:
#   after 900 sec (15 min) if at least 1 key changed
#   after 300 sec (5 min) if at least 10 keys changed
#   after 60 sec if at least 10000 keys changed
#
#   Note: you can disable saving completely by commenting out all "save" lines.
#
#   It is also possible to remove all the previously configured save
#   points by adding a save directive with a single empty string argument
#   like in the following example:
#
#   save ""

save 900 1
save 300 10
save 60 10000

# By default Redis will stop accepting writes if RDB snapshots are enabled
# (at least one save point) and the latest background save failed.
# This will make the user aware (in a hard way) that data is not persisting
# on disk properly, otherwise chances are that no one will notice and some
# disaster will happen.
#
# If the background saving process will start working again Redis will
# automatically allow writes again.
#
# However if you have setup your proper monitoring of the Redis server
# and persistence, you may want to disable this feature so that Redis will
# continue to work as usual even if there are problems with disk,
# permissions, and so forth.
stop-writes-on-bgsave-error yes

# Compress string objects using LZF when dump .rdb databases?
# For default that's set to 'yes' as it's almost always a win.
# If you want to save some CPU in the saving child set it to 'no' but
# the dataset will likely be bigger if you have compressible values or keys.
rdbcompression yes

# Since version 5 of RDB a CRC64 checksum is placed at the end of the file.
# This makes the format more resistant to corruption but there is a performance
# hit to pay (around 10%) when saving and loading RDB files, so you can disable it
# for maximum performances.
#
# RDB files created with checksum disabled have a checksum of zero that will
# tell the loading code to skip the check.
rdbchecksum yes

# The filename where to dump the DB
dbfilename dump.rdb
```

```
# The working directory.
#
# The DB will be written inside this directory, with the filename specified
# above using the 'dbfilename' configuration directive.
#
# The Append Only File will also be created inside this directory.
#
# Note that you must specify a directory here, not a file name.
dir /home/pi/redis

################################## REPLICATION #################################

# Master-Slave replication. Use slaveof to make a Redis instance a copy of
# another Redis server. A few things to understand ASAP about Redis replication.
#
# 1) Redis replication is asynchronous, but you can configure a master to
#    stop accepting writes if it appears to be not connected with at least
#    a given number of slaves.
# 2) Redis slaves are able to perform a partial resynchronization with the
#    master if the replication link is lost for a relatively small amount of
#    time. You may want to configure the replication backlog size (see the next
#    sections of this file) with a sensible value depending on your needs.
# 3) Replication is automatic and does not need user intervention. After a
#    network partition slaves automatically try to reconnect to masters
#    and resynchronize with them.
#
# slaveof <masterip> <masterport>

# If the master is password protected (using the "requirepass" configuration
# directive below) it is possible to tell the slave to authenticate before
# starting the replication synchronization process, otherwise the master will
# refuse the slave request.
#
# masterauth <master-password>

# When a slave loses its connection with the master, or when the replication
# is still in progress, the slave can act in two different ways:
#
# 1) if slave-serve-stale-data is set to 'yes' (the default) the slave will
#    still reply to client requests, possibly with out of date data, or the
#    data set may just be empty if this is the first synchronization.
#
# 2) if slave-serve-stale-data is set to 'no' the slave will reply with
#    an error "SYNC with master in progress" to all the kind of commands
#    but to INFO and SLAVEOF.
#
slave-serve-stale-data yes

# You can configure a slave instance to accept writes or not. Writing against
# a slave instance may be useful to store some ephemeral data (because data
# written on a slave will be easily deleted after resync with the master) but
# may also cause problems if clients are writing to it because of a
# misconfiguration.
```

```
#
# Since Redis 2.6 by default slaves are read-only.
#
# Note: read only slaves are not designed to be exposed to untrusted clients
# on the internet. It's just a protection layer against misuse of the instance.
# Still a read only slave exports by default all the administrative commands
# such as CONFIG, DEBUG, and so forth. To a limited extent you can improve
# security of read only slaves using 'rename-command' to shadow all the
# administrative / dangerous commands.
slave-read-only yes

# Replication SYNC strategy: disk or socket.
#
# -------------------------------------------------------
# WARNING: DISKLESS REPLICATION IS EXPERIMENTAL CURRENTLY
# -------------------------------------------------------
#
# New slaves and reconnecting slaves that are not able to continue the replication
# process just receiving differences, need to do what is called a "full
# synchronization". An RDB file is transmitted from the master to the slaves.
# The transmission can happen in two different ways:
#
# 1) Disk-backed: The Redis master creates a new process that writes the RDB
#                 file on disk. Later the file is transferred by the parent
#                 process to the slaves incrementally.
# 2) Diskless: The Redis master creates a new process that directly writes the
#              RDB file to slave sockets, without touching the disk at all.
#
# With disk-backed replication, while the RDB file is generated, more slaves
# can be queued and served with the RDB file as soon as the current child producing
# the RDB file finishes its work. With diskless replication instead once
# the transfer starts, new slaves arriving will be queued and a new transfer
# will start when the current one terminates.
#
# When diskless replication is used, the master waits a configurable amount of
# time (in seconds) before starting the transfer in the hope that multiple slaves
# will arrive and the transfer can be parallelized.
#
# With slow disks and fast (large bandwidth) networks, diskless replication
# works better.
repl-diskless-sync no

# When diskless replication is enabled, it is possible to configure the delay
# the server waits in order to spawn the child that transfers the RDB via socket
# to the slaves.
#
# This is important since once the transfer starts, it is not possible to serve
# new slaves arriving, that will be queued for the next RDB transfer, so the server
# waits a delay in order to let more slaves arrive.
#
# The delay is specified in seconds, and by default is 5 seconds. To disable
# it entirely just set it to 0 seconds and the transfer will start ASAP.
repl-diskless-sync-delay 5
```

```
# Slaves send PINGs to server in a predefined interval. It's possible to change
# this interval with the repl_ping_slave_period option. The default value is 10
# seconds.
#
# repl-ping-slave-period 10

# The following option sets the replication timeout for:
#
# 1) Bulk transfer I/O during SYNC, from the point of view of slave.
# 2) Master timeout from the point of view of slaves (data, pings).
# 3) Slave timeout from the point of view of masters (REPLCONF ACK pings).
#
# It is important to make sure that this value is greater than the value
# specified for repl-ping-slave-period otherwise a timeout will be detected
# every time there is low traffic between the master and the slave.
#
# repl-timeout 60

# Disable TCP_NODELAY on the slave socket after SYNC?
#
# If you select "yes" Redis will use a smaller number of TCP packets and
# less bandwidth to send data to slaves. But this can add a delay for
# the data to appear on the slave side, up to 40 milliseconds with
# Linux kernels using a default configuration.
#
# If you select "no" the delay for data to appear on the slave side will
# be reduced but more bandwidth will be used for replication.
#
# By default we optimize for low latency, but in very high traffic conditions
# or when the master and slaves are many hops away, turning this to "yes" may
# be a good idea.
repl-disable-tcp-nodelay no

# Set the replication backlog size. The backlog is a buffer that accumulates
# slave data when slaves are disconnected for some time, so that when a slave
# wants to reconnect again, often a full resync is not needed, but a partial
# resync is enough, just passing the portion of data the slave missed while
# disconnected.
#
# The bigger the replication backlog, the longer the time the slave can be
# disconnected and later be able to perform a partial resynchronization.
#
# The backlog is only allocated once there is at least a slave connected.
#
# repl-backlog-size 1mb

# After a master has no longer connected slaves for some time, the backlog
# will be freed. The following option configures the amount of seconds that
# need to elapse, starting from the time the last slave disconnected, for
# the backlog buffer to be freed.
#
# A value of 0 means to never release the backlog.
```

```
#
# repl-backlog-ttl 3600

# The slave priority is an integer number published by Redis in the INFO output.
# It is used by Redis Sentinel in order to select a slave to promote into a
# master if the master is no longer working correctly.
#
# A slave with a low priority number is considered better for promotion, so
# for instance if there are three slaves with priority 10, 100, 25 Sentinel will
# pick the one with priority 10, that is the lowest.
#
# However a special priority of 0 marks the slave as not able to perform the
# role of master, so a slave with priority of 0 will never be selected by
# Redis Sentinel for promotion.
#
# By default the priority is 100.
slave-priority 100

# It is possible for a master to stop accepting writes if there are less than
# N slaves connected, having a lag less or equal than M seconds.
#
# The N slaves need to be in "online" state.
#
# The lag in seconds, that must be <= the specified value, is calculated from
# the last ping received from the slave, that is usually sent every second.
#
# This option does not GUARANTEE that N replicas will accept the write, but
# will limit the window of exposure for lost writes in case not enough slaves
# are available, to the specified number of seconds.
#
# For example to require at least 3 slaves with a lag <= 10 seconds use:
#
# min-slaves-to-write 3
# min-slaves-max-lag 10
#
# Setting one or the other to 0 disables the feature.
#
# By default min-slaves-to-write is set to 0 (feature disabled) and
# min-slaves-max-lag is set to 10.

# A Redis master is able to list the address and port of the attached
# slaves in different ways. For example the "INFO replication" section
# offers this information, which is used, among other tools, by
# Redis Sentinel in order to discover slave instances.
# Another place where this info is available is in the output of the
# "ROLE" command of a masteer.
#
# The listed IP and address normally reported by a slave is obtained
# in the following way:
#
#   IP: The address is auto detected by checking the peer address
#   of the socket used by the slave to connect with the master.
#
```

```
#   Port: The port is communicated by the slave during the replication
#   handshake, and is normally the port that the slave is using to
#   list for connections.
#
# However when port forwarding or Network Address Translation (NAT) is
# used, the slave may be actually reachable via different IP and port
# pairs. The following two options can be used by a slave in order to
# report to its master a specific set of IP and port, so that both INFO
# and ROLE will report those values.
#
# There is no need to use both the options if you need to override just
# the port or the IP address.
#
# slave-announce-ip 5.5.5.5
# slave-announce-port 1234

################################## SECURITY ###################################

# Require clients to issue AUTH <PASSWORD> before processing any other
# commands.  This might be useful in environments in which you do not trust
# others with access to the host running redis-server.
#
# This should stay commented out for backward compatibility and because most
# people do not need auth (e.g. they run their own servers).
#
# Warning: since Redis is pretty fast an outside user can try up to
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
#
# requirepass foobared

# Command renaming.
#
# It is possible to change the name of dangerous commands in a shared
# environment. For instance the CONFIG command may be renamed into something
# hard to guess so that it will still be available for internal-use tools
# but not available for general clients.
#
# Example:
#
# rename-command CONFIG b840fc02d524045429941cc15f59e41cb7be6c52
#
# It is also possible to completely kill a command by renaming it into
# an empty string:
#
# rename-command CONFIG ""
#
# Please note that changing the name of commands that are logged into the
# AOF file or transmitted to slaves may cause problems.

################################### LIMITS ####################################

# Set the max number of connected clients at the same time. By default
```

```
# this limit is set to 10000 clients, however if the Redis server is not
# able to configure the process file limit to allow for the specified limit
# the max number of allowed clients is set to the current file limit
# minus 32 (as Redis reserves a few file descriptors for internal uses).
#
# Once the limit is reached Redis will close all the new connections sending
# an error 'max number of clients reached'.
#
# maxclients 10000

# Don't use more memory than the specified amount of bytes.
# When the memory limit is reached Redis will try to remove keys
# according to the eviction policy selected (see maxmemory-policy).
#
# If Redis can't remove keys according to the policy, or if the policy is
# set to 'noeviction', Redis will start to reply with errors to commands
# that would use more memory, like SET, LPUSH, and so on, and will continue
# to reply to read-only commands like GET.
#
# This option is usually useful when using Redis as an LRU cache, or to set
# a hard memory limit for an instance (using the 'noeviction' policy).
#
# WARNING: If you have slaves attached to an instance with maxmemory on,
# the size of the output buffers needed to feed the slaves are subtracted
# from the used memory count, so that network problems / resyncs will
# not trigger a loop where keys are evicted, and in turn the output
# buffer of slaves is full with DELs of keys evicted triggering the deletion
# of more keys, and so forth until the database is completely emptied.
#
# In short... if you have slaves attached it is suggested that you set a lower
# limit for maxmemory so that there is some free RAM on the system for slave
# output buffers (but this is not needed if the policy is 'noeviction').
#
#Limit db size to 1/2 of raspberry pi memory
 maxmemory 500mb

# MAXMEMORY POLICY: how Redis will select what to remove when maxmemory
# is reached. You can select among five behaviors:
#
# volatile-lru -> remove the key with an expire set using an LRU algorithm
# allkeys-lru -> remove any key according to the LRU algorithm
# volatile-random -> remove a random key with an expire set
# allkeys-random -> remove a random key, any key
# volatile-ttl -> remove the key with the nearest expire time (minor TTL)
# noeviction -> don't expire at all, just return an error on write operations
#
# Note: with any of the above policies, Redis will return an error on write
#       operations, when there are no suitable keys for eviction.
#
#       At the date of writing these commands are: set setnx setex append
#       incr decr rpush lpush rpushx lpushx linsert lset rpoplpush sadd
#       sinter sinterstore sunion sunionstore sdiff sdiffstore zadd zincrby
#       zunionstore zinterstore hset hsetnx hmset hincrby incrby decrby
```

```
#      getset mset msetnx exec sort
#
# The default is:
#
# maxmemory-policy noeviction

# LRU and minimal TTL algorithms are not precise algorithms but approximated
# algorithms (in order to save memory), so you can tune it for speed or
# accuracy. For default Redis will check five keys and pick the one that was
# used less recently, you can change the sample size using the following
# configuration directive.
#
# The default of 5 produces good enough results. 10 Approximates very closely
# true LRU but costs a bit more CPU. 3 is very fast but not very accurate.
#
# maxmemory-samples 5

############################## APPEND ONLY MODE ###############################

# By default Redis asynchronously dumps the dataset on disk. This mode is
# good enough in many applications, but an issue with the Redis process or
# a power outage may result into a few minutes of writes lost (depending on
# the configured save points).
#
# The Append Only File is an alternative persistence mode that provides
# much better durability. For instance using the default data fsync policy
# (see later in the config file) Redis can lose just one second of writes in a
# dramatic event like a server power outage, or a single write if something
# wrong with the Redis process itself happens, but the operating system is
# still running correctly.
#
# AOF and RDB persistence can be enabled at the same time without problems.
# If the AOF is enabled on startup Redis will load the AOF, that is the file
# with the better durability guarantees.
#
# Please check http://redis.io/topics/persistence for more information.

appendonly no

# The name of the append only file (default: "appendonly.aof")

appendfilename "appendonly.aof"

# The fsync() call tells the Operating System to actually write data on disk
# instead of waiting for more data in the output buffer. Some OS will really flush
# data on disk, some other OS will just try to do it ASAP.
#
# Redis supports three different modes:
#
# no: don't fsync, just let the OS flush the data when it wants. Faster.
# always: fsync after every write to the append only log. Slow, Safest.
# everysec: fsync only one time every second. Compromise.
#
```

```
# The default is "everysec", as that's usually the right compromise between
# speed and data safety. It's up to you to understand if you can relax this to
# "no" that will let the operating system flush the output buffer when
# it wants, for better performances (but if you can live with the idea of
# some data loss consider the default persistence mode that's snapshotting),
# or on the contrary, use "always" that's very slow but a bit safer than
# everysec.
#
# More details please check the following article:
# http://antirez.com/post/redis-persistence-demystified.html
#
# If unsure, use "everysec".

# appendfsync always
appendfsync everysec
# appendfsync no

# When the AOF fsync policy is set to always or everysec, and a background
# saving process (a background save or AOF log background rewriting) is
# performing a lot of I/O against the disk, in some Linux configurations
# Redis may block too long on the fsync() call. Note that there is no fix for
# this currently, as even performing fsync in a different thread will block
# our synchronous write(2) call.
#
# In order to mitigate this problem it's possible to use the following option
# that will prevent fsync() from being called in the main process while a
# BGSAVE or BGREWRITEAOF is in progress.
#
# This means that while another child is saving, the durability of Redis is
# the same as "appendfsync none". In practical terms, this means that it is
# possible to lose up to 30 seconds of log in the worst scenario (with the
# default Linux settings).
#
# If you have latency problems turn this to "yes". Otherwise leave it as
# "no" that is the safest pick from the point of view of durability.

no-appendfsync-on-rewrite no

# Automatic rewrite of the append only file.
# Redis is able to automatically rewrite the log file implicitly calling
# BGREWRITEAOF when the AOF log size grows by the specified percentage.
#
# This is how it works: Redis remembers the size of the AOF file after the
# latest rewrite (if no rewrite has happened since the restart, the size of
# the AOF at startup is used).
#
# This base size is compared to the current size. If the current size is
# bigger than the specified percentage, the rewrite is triggered. Also
# you need to specify a minimal size for the AOF file to be rewritten, this
# is useful to avoid rewriting the AOF file even if the percentage increase
# is reached but it is still pretty small.
#
# Specify a percentage of zero in order to disable the automatic AOF
```

```
# rewrite feature.

auto-aof-rewrite-percentage 100
auto-aof-rewrite-min-size 64mb

# An AOF file may be found to be truncated at the end during the Redis
# startup process, when the AOF data gets loaded back into memory.
# This may happen when the system where Redis is running
# crashes, especially when an ext4 filesystem is mounted without the
# data=ordered option (however this can't happen when Redis itself
# crashes or aborts but the operating system still works correctly).
#
# Redis can either exit with an error when this happens, or load as much
# data as possible (the default now) and start if the AOF file is found
# to be truncated at the end. The following option controls this behavior.
#
# If aof-load-truncated is set to yes, a truncated AOF file is loaded and
# the Redis server starts emitting a log to inform the user of the event.
# Otherwise if the option is set to no, the server aborts with an error
# and refuses to start. When the option is set to no, the user requires
# to fix the AOF file using the "redis-check-aof" utility before to restart
# the server.
#
# Note that if the AOF file will be found to be corrupted in the middle
# the server will still exit with an error. This option only applies when
# Redis will try to read more data from the AOF file but not enough bytes
# will be found.
aof-load-truncated yes

################################ LUA SCRIPTING  ###############################

# Max execution time of a Lua script in milliseconds.
#
# If the maximum execution time is reached Redis will log that a script is
# still in execution after the maximum allowed time and will start to
# reply to queries with an error.
#
# When a long running script exceeds the maximum execution time only the
# SCRIPT KILL and SHUTDOWN NOSAVE commands are available. The first can be
# used to stop a script that did not yet called write commands. The second
# is the only way to shut down the server in the case a write command was
# already issued by the script but the user doesn't want to wait for the natural
# termination of the script.
#
# Set it to 0 or a negative value for unlimited execution without warnings.
lua-time-limit 5000

################################ REDIS CLUSTER  ###############################
#
# ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
# WARNING EXPERIMENTAL: Redis Cluster is considered to be stable code, however
# in order to mark it as "mature" we need to wait for a non trivial percentage
# of users to deploy it in production.
```

```
# ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#
# Normal Redis instances can't be part of a Redis Cluster; only nodes that are
# started as cluster nodes can. In order to start a Redis instance as a
# cluster node enable the cluster support uncommenting the following:
#
# cluster-enabled yes

# Every cluster node has a cluster configuration file. This file is not
# intended to be edited by hand. It is created and updated by Redis nodes.
# Every Redis Cluster node requires a different cluster configuration file.
# Make sure that instances running in the same system do not have
# overlapping cluster configuration file names.
#
# cluster-config-file nodes-6379.conf

# Cluster node timeout is the amount of milliseconds a node must be unreachable
# for it to be considered in failure state.
# Most other internal time limits are multiple of the node timeout.
#
# cluster-node-timeout 15000

# A slave of a failing master will avoid to start a failover if its data
# looks too old.
#
# There is no simple way for a slave to actually have a exact measure of
# its "data age", so the following two checks are performed:
#
# 1) If there are multiple slaves able to failover, they exchange messages
#    in order to try to give an advantage to the slave with the best
#    replication offset (more data from the master processed).
#    Slaves will try to get their rank by offset, and apply to the start
#    of the failover a delay proportional to their rank.
#
# 2) Every single slave computes the time of the last interaction with
#    its master. This can be the last ping or command received (if the master
#    is still in the "connected" state), or the time that elapsed since the
#    disconnection with the master (if the replication link is currently down).
#    If the last interaction is too old, the slave will not try to failover
#    at all.
#
# The point "2" can be tuned by user. Specifically a slave will not perform
# the failover if, since the last interaction with the master, the time
# elapsed is greater than:
#
#   (node-timeout * slave-validity-factor) + repl-ping-slave-period
#
# So for example if node-timeout is 30 seconds, and the slave-validity-factor
# is 10, and assuming a default repl-ping-slave-period of 10 seconds, the
# slave will not try to failover if it was not able to talk with the master
# for longer than 310 seconds.
#
# A large slave-validity-factor may allow slaves with too old data to failover
```

```
# a master, while a too small value may prevent the cluster from being able to
# elect a slave at all.
#
# For maximum availability, it is possible to set the slave-validity-factor
# to a value of 0, which means, that slaves will always try to failover the
# master regardless of the last time they interacted with the master.
# (However they'll always try to apply a delay proportional to their
# offset rank).
#
# Zero is the only value able to guarantee that when all the partitions heal
# the cluster will always be able to continue.
#
# cluster-slave-validity-factor 10

# Cluster slaves are able to migrate to orphaned masters, that are masters
# that are left without working slaves. This improves the cluster ability
# to resist to failures as otherwise an orphaned master can't be failed over
# in case of failure if it has no working slaves.
#
# Slaves migrate to orphaned masters only if there are still at least a
# given number of other working slaves for their old master. This number
# is the "migration barrier". A migration barrier of 1 means that a slave
# will migrate only if there is at least 1 other working slave for its master
# and so forth. It usually reflects the number of slaves you want for every
# master in your cluster.
#
# Default is 1 (slaves migrate only if their masters remain with at least
# one slave). To disable migration just set it to a very large value.
# A value of 0 can be set but is useful only for debugging and dangerous
# in production.
#
# cluster-migration-barrier 1

# By default Redis Cluster nodes stop accepting queries if they detect there
# is at least an hash slot uncovered (no available node is serving it).
# This way if the cluster is partially down (for example a range of hash slots
# are no longer covered) all the cluster becomes, eventually, unavailable.
# It automatically returns available as soon as all the slots are covered again.
#
# However sometimes you want the subset of the cluster which is working,
# to continue to accept queries for the part of the key space that is still
# covered. In order to do so, just set the cluster-require-full-coverage
# option to no.
#
# cluster-require-full-coverage yes

# In order to setup your cluster make sure to read the documentation
# available at http://redis.io web site.

################################## SLOW LOG ###################################

# The Redis Slow Log is a system to log queries that exceeded a specified
# execution time. The execution time does not include the I/O operations
```

```
# like talking with the client, sending the reply and so forth,
# but just the time needed to actually execute the command (this is the only
# stage of command execution where the thread is blocked and can not serve
# other requests in the meantime).
#
# You can configure the slow log with two parameters: one tells Redis
# what is the execution time, in microseconds, to exceed in order for the
# command to get logged, and the other parameter is the length of the
# slow log. When a new command is logged the oldest one is removed from the
# queue of logged commands.

# The following time is expressed in microseconds, so 1000000 is equivalent
# to one second. Note that a negative number disables the slow log, while
# a value of zero forces the logging of every command.
slowlog-log-slower-than 10000

# There is no limit to this length. Just be aware that it will consume memory.
# You can reclaim memory used by the slow log with SLOWLOG RESET.
slowlog-max-len 128

################################ LATENCY MONITOR ##############################

# The Redis latency monitoring subsystem samples different operations
# at runtime in order to collect data related to possible sources of
# latency of a Redis instance.
#
# Via the LATENCY command this information is available to the user that can
# print graphs and obtain reports.
#
# The system only logs operations that were performed in a time equal or
# greater than the amount of milliseconds specified via the
# latency-monitor-threshold configuration directive. When its value is set
# to zero, the latency monitor is turned off.
#
# By default latency monitoring is disabled since it is mostly not needed
# if you don't have latency issues, and collecting data has a performance
# impact, that while very small, can be measured under big load. Latency
# monitoring can easily be enabled at runtime using the command
# "CONFIG SET latency-monitor-threshold <milliseconds>" if needed.
latency-monitor-threshold 0

############################# EVENT NOTIFICATION ##############################

# Redis can notify Pub/Sub clients about events happening in the key space.
# This feature is documented at http://redis.io/topics/notifications
#
# For instance if keyspace events notification is enabled, and a client
# performs a DEL operation on key "foo" stored in the Database 0, two
# messages will be published via Pub/Sub:
#
# PUBLISH __keyspace@0__:foo del
# PUBLISH __keyevent@0__:del foo
#
```

```
# It is possible to select the events that Redis will notify among a set
# of classes. Every class is identified by a single character:
#
#  K     Keyspace events, published with __keyspace@<db>__ prefix.
#  E     Keyevent events, published with __keyevent@<db>__ prefix.
#  g     Generic commands (non-type specific) like DEL, EXPIRE, RENAME, ...
#  $     String commands
#  l     List commands
#  s     Set commands
#  h     Hash commands
#  z     Sorted set commands
#  x     Expired events (events generated every time a key expires)
#  e     Evicted events (events generated when a key is evicted for maxmemory)
#  A     Alias for g$lshzxe, so that the "AKE" string means all the events.
#
#  The "notify-keyspace-events" takes as argument a string that is composed
#  of zero or multiple characters. The empty string means that notifications
#  are disabled.
#
#  Example: to enable list and generic events, from the point of view of the
#           event name, use:
#
#  notify-keyspace-events Elg
#
#  Example 2: to get the stream of the expired keys subscribing to channel
#             name __keyevent@0__:expired use:
#
#  notify-keyspace-events Ex
#
#  By default all notifications are disabled because most users don't need
#  this feature and the feature has some overhead. Note that if you don't
#  specify at least one of K or E, no events will be delivered.
notify-keyspace-events ""

################################ ADVANCED CONFIG ###############################

# Hashes are encoded using a memory efficient data structure when they have a
# small number of entries, and the biggest entry does not exceed a given
# threshold. These thresholds can be configured using the following directives.
hash-max-ziplist-entries 1000
hash-max-ziplist-value 1000

# Lists are also encoded in a special way to save a lot of space.
# The number of entries allowed per internal list node can be specified
# as a fixed maximum size or a maximum number of elements.
# For a fixed maximum size, use -5 through -1, meaning:
# -5: max size: 64 Kb  <-- not recommended for normal workloads
# -4: max size: 32 Kb  <-- not recommended
# -3: max size: 16 Kb  <-- probably not recommended
# -2: max size: 8 Kb   <-- good
# -1: max size: 4 Kb   <-- good
# Positive numbers mean store up to _exactly_ that number of elements
# per list node.
```

The highest performing option is usually -2 (8 Kb size) or -1 (4 Kb size),
but if your use case is unique, adjust the settings as necessary.
list-max-ziplist-size -3

Lists may also be compressed.
Compress depth is the number of quicklist ziplist nodes from *each* side of
the list to *exclude* from compression. The head and tail of the list
are always uncompressed for fast push/pop operations. Settings are:
0: disable all list compression
1: depth 1 means "don't start compressing until after 1 node into the list,
going from either the head or tail"
So: [head]->node->node->...->node->[tail]
[head], [tail] will always be uncompressed; inner nodes will compress.
2: [head]->[next]->node->node->...->node->[prev]->[tail]
2 here means: don't compress head or head->next or tail->prev or tail,
but compress all nodes between them.
3: [head]->[next]->[next]->node->node->...->node->[prev]->[prev]->[tail]
etc.
list-compress-depth 0

Sets have a special encoding in just one case: when a set is composed
of just strings that happen to be integers in radix 10 in the range
# of 64 bit signed integers.
The following configuration setting sets the limit in the size of the
set in order to use this special memory saving encoding.
set-max-intset-entries 1000

Similarly to hashes and lists, sorted sets are also specially encoded in
order to save a lot of space. This encoding is only used when the length and
elements of a sorted set are below the following limits:
zset-max-ziplist-entries 1000
zset-max-ziplist-value 1000

HyperLogLog sparse representation bytes limit. The limit includes the
16 bytes header. When an HyperLogLog using the sparse representation crosses
this limit, it is converted into the dense representation.
#
A value greater than 16000 is totally useless, since at that point the
dense representation is more memory efficient.
#
The suggested value is ~ 3000 in order to have the benefits of
the space efficient encoding without slowing down too much PFADD,
which is O(N) with the sparse encoding. The value can be raised to
~ 10000 when CPU is not a concern, but space is, and the data set is
composed of many HyperLogLogs with cardinality in the 0 - 15000 range.
hll-sparse-max-bytes 3000

Active rehashing uses 1 millisecond every 100 milliseconds of CPU time in
order to help rehashing the main Redis hash table (the one mapping top-level
keys to values). The hash table implementation Redis uses (see dict.c)
performs a lazy rehashing: the more operation you run into a hash table
that is rehashing, the more rehashing "steps" are performed, so if the
server is idle the rehashing is never complete and some more memory is used

```
# by the hash table.
#
# The default is to use this millisecond 10 times every second in order to
# actively rehash the main dictionaries, freeing memory when possible.
#
# If unsure:
# use "activerehashing no" if you have hard latency requirements and it is
# not a good thing in your environment that Redis can reply from time to time
# to queries with 2 milliseconds delay.
#
# use "activerehashing yes" if you don't have such hard requirements but
# want to free memory asap when possible.
activerehashing yes

# The client output buffer limits can be used to force disconnection of clients
# that are not reading data from the server fast enough for some reason (a
# common reason is that a Pub/Sub client can't consume messages as fast as the
# publisher can produce them).
#
# The limit can be set differently for the three different classes of clients:
#
# normal -> normal clients including MONITOR clients
# slave  -> slave clients
# pubsub -> clients subscribed to at least one pubsub channel or pattern
#
# The syntax of every client-output-buffer-limit directive is the following:
#
# client-output-buffer-limit <class> <hard limit> <soft limit> <soft seconds>
#
# A client is immediately disconnected once the hard limit is reached, or if
# the soft limit is reached and remains reached for the specified number of
# seconds (continuously).
# So for instance if the hard limit is 32 megabytes and the soft limit is
# 16 megabytes / 10 seconds, the client will get disconnected immediately
# if the size of the output buffers reach 32 megabytes, but will also get
# disconnected if the client reaches 16 megabytes and continuously overcomes
# the limit for 10 seconds.
#
# By default normal clients are not limited because they don't receive data
# without asking (in a push way), but just after a request, so only
# asynchronous clients may create a scenario where data is requested faster
# than it can read.
#
# Instead there is a default limit for pubsub and slave clients, since
# subscribers and slaves receive data in a push fashion.
#
# Both the hard or the soft limit can be disabled by setting them to zero.
client-output-buffer-limit normal 0 0 0
client-output-buffer-limit slave 256mb 64mb 60
client-output-buffer-limit pubsub 32mb 8mb 60

# Redis calls an internal function to perform many background tasks, like
# closing connections of clients in timeout, purging expired keys that are
```

```
# never requested, and so forth.
#
# Not all tasks are performed with the same frequency, but Redis checks for
# tasks to perform according to the specified "hz" value.
#
# By default "hz" is set to 10. Raising the value will use more CPU when
# Redis is idle, but at the same time will make Redis more responsive when
# there are many keys expiring at the same time, and timeouts may be
# handled with more precision.
#
# The range is between 1 and 500, however a value over 100 is usually not
# a good idea. Most users should use the default of 10 and raise this up to
# 100 only in environments where very low latency is required.
hz 10

# When a child rewrites the AOF file, if the following option is enabled
# the file will be fsync-ed every 32 MB of data generated. This is useful
# in order to commit the file to the disk more incrementally and avoid
# big latency spikes.
aof-rewrite-incremental-fsync yes
```

Install Redis as a service

Friday, February 10, 2017 6:15 PM

pi@irr1 /etc/init.d $ ls
alsa-utils dbus keyboard-setup mountnfs.sh rc.local sendsigs umountnfs.sh
avahi-daemon dhcpcd killprocs mtab.sh rcS single umountroot
bootlogs dphys-swapfile kmod networking README skeleton urandom
bootmisc.sh fake-hwclock lightdm nfs-common reboot smb.conf x11-common
cgroup-bin halt motd ntp ***redis*** ssh
checkfs.sh hostname.sh mountall-bootclean.sh plymouth rmnologin sudo
checkroot-bootclean.sh hwclock.sh mountall.sh plymouth-log rpcbind triggerhappy
checkroot.sh ifplugd mountdevsubfs.sh procps rsync udev
console-setup irrigation mountkernfs.sh raspi-config rsyslog udev-mtab
cron kbd mountnfs-bootclean.sh rc samba umountfs

We will add redis as a startup file

The Startup file has the following script

```
#! /bin/sh
# /etc/init.d/noip

### BEGIN INIT INFO
# Provides:          noip
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Simple script to start a program at boot
# Description:       A simple script from www.stuffaboutcode.com which will start / stop a program a boot / shutdown.
### END INIT INFO


swapoff --all



# If you want a command to always run, put it here

# Carry out specific functions when asked to by the system
case "$1" in
  start)
    echo "Starting redis"
    # run application you want to start
    /home/pi/redis/redis-start&
    ;;
  stop)
    echo "Stopping redis"
    # kill application you want to stop
```

```
    killall redis-server
    ;;
  *)
    echo "Usage: /etc/init.d/redis {start|stop}"
    exit 1
    ;;
esac
```

The file located in /home/pi/redis/redis-start is as follows
/sbin/sysctl vm.overcommit_memory=1
/usr/local/bin/redis-server /etc/redis/redis.conf

Test script by issueing the command
"sudo /etc/init.d/redis start"

Redis server should start on ssh console

If script is successful then make script startup automatically

```
sudo update-rc.d redis defaults
```

Move redis to usb data drive

Saturday, March 25, 2017 8:43 PM

In /etc/redis/redis.conf
Move store directory to /datadisk/redis

Make sure that /datadisk/redis directory exits
Before restarting redis

Install apps

Friday, February 10, 2017 8:03 PM

Generating Certificate

Friday, February 17, 2017 6:21 PM

Overview

The following is an extremely simplified view of how SSL is implemented and what part the certificate plays in the entire process.

Normal web traffic is sent unencrypted over the Internet. That is, anyone with access to the right tools can snoop all of that traffic. Obviously, this can lead to problems, especially where security and privacy is necessary, such as in credit card data and bank transactions. The Secure Socket Layer is used to encrypt the data stream between the web server and the web client (the browser).

SSL makes use of what is known as **asymmetric cryptography**, commonly referred to as **public key cryptography (PKI)**. With public key cryptography, two keys are created, one public, one private. Anything encrypted with either key can only be decrypted with its corresponding key. Thus if a message or data stream were encrypted with the server's private key, it can be decrypted only using its corresponding public key, ensuring that the data only could have come from the server.

If SSL utilizes public key cryptography to encrypt the data stream traveling over the Internet, why is a certificate necessary? The technical answer to that question is that a certificate is not really necessary - the data is secure and cannot easily be decrypted by a third party. However, certificates do serve a crucial role in the communication process. The certificate, signed by a trusted Certificate Authority (CA), ensures that the certificate holder is really who he claims to be. Without a trusted signed certificate, your data may be encrypted, however, the party you are communicating with may not be whom you think. Without certificates, impersonation attacks would be much more common.

Step 1: Generate a Private Key

The **openssl** toolkit is used to generate an **RSA Private Key** and **CSR (Certificate Signing Request)**. It can also be used to generate self-signed certificates which can be used for testing purposes or internal usage.

The first step is to create your RSA Private Key. This key is a 1024 bit RSA key which is encrypted using Triple-DES and stored in a PEM format so that it is readable as ASCII text.

```
openssl genrsa -des3 -out server.key 1024

Generating RSA private key, 1024 bit long modulus
.....................................................++++++
........++++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

Step 2: Generate a CSR (Certificate Signing Request)

Once the private key is generated a Certificate Signing Request can be generated. The CSR is then used in one of two ways. Ideally, the CSR will be sent to a Certificate Authority, such as Thawte or Verisign who will verify the identity of the requestor and issue a signed certificate. **The second option is to self-sign the CSR, which will be demonstrated in the next section**.

During the generation of the CSR, you will be prompted for several pieces of information. These are the X.509 attributes of the certificate. One of the prompts will be for "Common Name (e.g., YOUR name)". It is important that this field be filled in with the fully qualified domain name of the server to be protected by SSL. If the website to be protected will be https://public.akadia.com, then enter public.akadia.com at this prompt. The command to generate the CSR is as follows:

```
openssl req -new -key server.key -out server.csr

Country Name (2 letter code) [GB]:CH
```

```
State or Province Name (full name) [Berkshire]:Bern
Locality Name (eg, city) [Newbury]:Oberdiessbach
Organization Name (eg, company) [My Company Ltd]:Akadia AG
Organizational Unit Name (eg, section) []:Information Technology
Common Name (eg, your name or your server's hostname)
[]:public.akadia.com
Email Address []:martin dot zahn at akadia dot ch
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Step 3: Remove Passphrase from Key

One unfortunate side-effect of the pass-phrased private key is **that Apache will ask for the pass-phrase each time the web server is started**. Obviously this is not necessarily convenient as someone will not always be around to type in the pass-phrase, such as after a reboot or crash. mod_ssl includes the ability to use an external program in place of the built-in pass-phrase dialog, however, this is not necessarily the most secure option either. **It is possible to remove the Triple-DES encryption from the key**, thereby no longer needing to type in a pass-phrase. If the private key is no longer encrypted, it is critical that this file only be readable by the root user! If your system is ever compromised and a third party obtains your unencrypted private key, the corresponding certificate will need to be revoked. With that being said, use the following command to remove the pass-phrase from the key:

```
cp server.key server.key.org
openssl rsa -in server.key.org -out server.key
```

The newly created server.key file has no more passphrase in it.

```
-rw-r--r-- 1 root root 745 Jun 29 12:19 server.csr
-rw-r--r-- 1 root root 891 Jun 29 13:22 server.key
-rw-r--r-- 1 root root 963 Jun 29 13:22 server.key.org
```

Step 4: Generating a Self-Signed Certificate

At this point you will need to generate a self-signed certificate because you either don't plan on having your certificate signed by a CA, or you wish to test your new SSL implementation while the CA is signing your certificate. This temporary certificate will generate an error in the client browser to the effect that the signing certificate authority is unknown and not trusted.

To generate a temporary certificate which is good for 365 days, issue the following command:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out
server.crt
Signature ok
subject=/C=CH/ST=Bern/L=Oberdiessbach/O=Akadia AG/OU=Information
Technology/CN=public.akadia.com/Email=martin dot zahn at akadia dot ch
Getting Private key
```

Step 5: Installing the Private Key and Certificate

When Apache with mod_ssl is installed, it creates several directories in the Apache config directory. The location of this directory will differ depending on how Apache was compiled.

```
cp server.crt /usr/local/apache/conf/ssl.crt
cp server.key /usr/local/apache/conf/ssl.key
```

Step 6: Configuring SSL Enabled Virtual Hosts

```
SSLEngine on
SSLCertificateFile /usr/local/apache/conf/ssl.crt/server.crt
SSLCertificateKeyFile /usr/local/apache/conf/ssl.key/server.key
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
CustomLog logs/ssl_request_log \
   "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
```

Step 7: Restart Apache and Test

```
/etc/init.d/httpd stop
/etc/init.d/httpd stop
```

https://public.akadia.com

From <http://www.akadia.com/services/ssh_test_certificate.html>

From <https://www.digitalocean.com/community/tutorials/how-to-create-an-ssh-ca-to-validate-hosts-and-clients-with-ubuntu>

Setup flask to do https

Friday, February 17, 2017 9:41 PM

To run flask as a normal web server do

```
if __name__ == '__main__':
  print "startup_dict",startup_dict
  app.run(threaded=True , use_reloader=True, host='0.0.0.0',port=int(startup_dict["PORT"]))
```

To run flask as a https web server do

```
if __name__ == '__main__':
  print "startup_dict",startup_dict
  app.run(threaded=True , use_reloader=True,
host='0.0.0.0',port=int(startup_dict["PORT"]) ,ssl_context=(startup_dict["crt_file"],
startup_dict["key_file"]))
```

Flask Web Server Setup

Friday, February 10, 2017 12:28 PM

In the directory /home/pi/flask_web

Execute the following shell commands

sudo pip install bitstuct

Format SD CARD FOR RASPBERRY PI IMAGE

Thursday, June 4, 2015 13:35

For Ubuntu Linux Home Machine

Download gparted

RASPBIAN Debian Wheezy

Extract the Image to get a image file .img

Execute the following commands

1. Run df -h to see what devices are currently mounted
2. umount /dev/<device> dd bs=4M if=~/2012-12-16-wheezy-raspbian.img of=/dev/sdd

important sdd should be the blk0 of the drive

3. run sudo sync
4. Remove SD card from card reader, insert it in the Raspberry Pi

SSH

Thursday, June 4, 2015 14:36

SSH is enabled by default in the raspberry pi by default
The default initial user is pi and initial password is raspberry

Change the password of pi to ready2go

sudo passwd

Services can be setup of using the following commands

```
sudo raspi-config
```

adding ssh keys from a Ubuntu Client

Thursday, June 11, 2015 13:10

Generate key

ssh-keygen -t rsa -b 4096

Copy Key

ssh-copy-id <username>@<host>

--- IMPORTANT OWNERSHIP ISSUES

Strict modes, as noted in the warning, checks file and folder permissions of your home folder, the .ssh folder and the authorized keys file.

From the sshd man page:

~/.ssh/authorized_keys
Lists the public keys (RSA/DSA) that can be used for logging in as this user. The format of this file is described above. The content of the file is not highly sensitive, but the recommended permissions are read/write for the user, and not accessible by others.

If this file, [meaning the authorized_keys file], the ~/.ssh directory, or the user's home directory are writable by other users, then the file could be modified or replaced by unauthorized users. In this case, sshd will not allow it to be used unless the StrictModes option has been set to ''no''.

From what I have read, your home folder and the .ssh folder should have your name for both the owner and group and permissions must be set so that there is no access for 'others'
My home folder shows: drwxrwx---
the .ssh folder is: drwx------
and the authorized_keys file is: -rw-------

If these conditions are not met, and StrictModes is on, then key pair authentication will be blocked.

Regards

anita2R

New Pi password

Tuesday, June 9, 2015 18:33

ready2go

Install Redis

Thursday, June 4, 2015 14:46

Execute the following commands
mkdir redis
cd redis

wget http://download.redis.io/redis-stable.tar.gz tar xvzf redis-stable.tar.gz cd redis-stable make

make

- **redis-server** is the Redis Server itself.
- **redis-sentinel** is the Redis Sentinel executable (monitoring and failover).
- **redis-cli** is the command line interface utility to talk with Redis.
- **redis-benchmark** is used to check Redis performances.
- **redis-check-aof** and **redis-check-dump** are useful in the rare event of corrupted data files.

It is a good idea to copy both the Redis server and the command line interface in proper places, either manually using the following commands:

- sudo cp src/redis-server /usr/local/bin/
- sudo cp src/redis-cli /usr/local/bin/

How to recover a lost password

Thursday, June 4, 2015 14:46

Step 1 – Grab The SD Card

Power down the Pi and remove the SD card. Insert it into your PC.

Step 2 – Edit cmdline.txt

The boot partition should be visible and contain a file named “cmdline.txt”. Edit this file in a text editor and add the following to the end of the existing text :

```
init=/bin/sh
```

If the original content was :

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

it should now look like :

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait init=/bin/sh
```

Make sure it is all one line! Save the text file and eject the SD card from the PC.

Step 3 – Reset the Password

Insert the card into a Pi that is connected to a monitor and keyboard. Power up the Pi. There may be a delay but you should be presented with a flashing cursor.

At the prompt type the following command :

```
passwd pi
```

You will then be prompted for a new password. Enter it carefully and press the [Return] key. It will now ask you to retype the password.

```
passwd pi
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

The password has been changed.

Now type the following commands :

```
sync exec /sbin/init
```

The Pi will continue to boot and return you to the normal command line prompt.

Shutdown the Pi and power it off.

```
sudo halt
```

Step 4 – Edit cmdline.txt

Using the PC edit the “cmdline.txt” file again and remove the “init=/bin/sh” text you added in Step 2.

You can now return the SD card to your Pi, reboot and use the new password.

Making a Service Start at Boot Up

Thursday, June 4, 2015 16:44

Raspberry Pi - run program at start-up

Anyway, I wanted to get my Raspberry Pi to start no-ip dynamic dns service when it started-up, so I wouldn't have to remember to start it every time it was powered up. For details on how to install no-ip on the Pi, see this post.

There are loads of ways of running a command at start-up in Linux but my favoured approach is to create an initialisation script in /etc/init.d and register it using update-rc.d. This way the application is started and stopped automatically when the system boots / shutdowns.

Create script in /etc/init.d

```
sudo nano /etc/init.d/NameOfYourScript
```

The following is an example based on starting up the no-ip service [/usr/local/bin/noip], but change the name of the script and the command to start and stop it and it would work for any command.

```
#! /bin/sh
# /etc/init.d/noip

### BEGIN INIT INFO
# Provides:          noip
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Simple script to start a program at boot
# Description:       A simple script from www.stuffaboutcode.comwhich will start / stop a program a boot / shutdown.
### END INIT INFO

# If you want a command to always run, put it here

# Carry out specific functions when asked to by the system
case "$1" in
  start)
    echo "Starting noip"
    # run application you want to start
    /usr/local/bin/noip2
    ;;
  stop)
    echo "Stopping noip"
    # kill application you want to stop
```

```
    killall noip2
    ;;
  *)
    echo "Usage: /etc/init.d/noip {start|stop}"
    exit 1
    ;;
esac

exit 0
```

Warning - its important you test your script first and make sure it doesn't need a user to provide a response, press "y" or similar, because you may find it hangs the raspberry pi on boot waiting for a user (who's not there) to do something!

Make script executable

```
sudo chmod 755 /etc/init.d/NameOfYourScript
```

Test starting the program

```
sudo /etc/init.d/NameOfYourScript start
```

Test stopping the program

```
sudo /etc/init.d/NameOfYourScript stop
```

Register script to be run at start-up
To register your script to be run at start-up and shutdown, run the following command:

```
sudo update-rc.d NameOfYourScript defaults
```

Note - The header at the start is to make the script LSB compliant and provides details about the start up script and you should only need to change the name. If you want to know more about creating LSB scripts for managing services, see http://wiki.debian.org/LSBInitScripts

If you ever want to remove the script from start-up, run the following command:

```
sudo update-rc.d -f  NameOfYourScript remove
```

Current Redis Config and DB

Tuesday, June 9, 2015 19:13

Redis db and redis.conf is located in the directory

~/redis/redis-start

#! /bin/sh

/etc/init.d/noip

The Startup and shutdown redis is shown below

BEGIN INIT INFO

Provides: noip

Required-Start: $remote_fs $syslog

Required-Stop: $remote_fs $syslog

Default-Start: 2 3 4 5

Default-Stop: 0 1 6

Short-Description: Simple script to start a program at boot

Description: A simple script from www.stuffaboutcode.com which will start / stop a program a boot / shutdown.

END INIT INFO

If you want a command to always run, put it here

```
# Carry out specific functions when asked to by the system

case "$1" in

  start)

    echo "Starting redis"

    # run application you want to start

    /home/pi/redis/redis-start/redis-start&

    ;;

  stop)

    echo "Stopping redis"

    # kill application you want to stop

    killall redis-server

    ;;

  *)

    echo "Usage: /etc/init.d/redis {start|stop}"

    exit 1

    ;;

esac



exit 0
```

changing address to static address

Thursday, June 4, 2015 15:58

Then enter this command to **edit the network interfaces file**:
sudo nano /etc/network/interfaces
replacing nano with your preferred text editor if you want.

First of all replace the 'dhcp' underlined in the screenshot earlier with 'static' to give the line:
iface eth0 inet static

Then directly below enter the following using the information you noted down earlier. Type everything very carefully and double check everything before you save the modified file.

address 192.168.1.70
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.254

Installing NTPD Service

Wednesday, June 10, 2015 19:13

sudo apt-get install ntpdate

Changing Timezone

Wednesday, June 10, 2015 19:14

sudo tzselect

or

much better

dpkg-reconfigure tzdata

installing python pip

Thursday, June 4, 2015 16:45

```
sudo apt-get install python-pip
```

installing python redis api

Thursday, June 4, 2015 16:48

```
sudo pip install redis
```

Test Program

```
import redis

r = redis.StrictRedis(host='localhost', port=6379, db=0)

r.set('foo', 'bar')

r.get('foo')

r.delete('foo')  # get rid of test key
```

Installing pika

Thursday, June 4, 2015 16:58

sudo pip install pika

install Adafruit raspberry pi I/O Library

Thursday, June 4, 2015 17:00

Before you start, you'll need to have the python smbus library installed, run **apt-get install python-smbus**

Downloading the Code from Github

The easiest way to get the code onto your Pi is to hook up an Ethernet cable, and clone it directly using 'git', which is installed by default on most distros. Simply run the following commands from an appropriate location (ex. "/home/pi"):

1. $ git clone https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code.git
2. $ cd Adafruit-Raspberry-Pi-Python-Code
3. $ cd Adafruit_PWM_Servo_Driver

Original (256MB) Raspberry Pi's use I2C bus 0, while Second Revision Pi's use I2C bus 1. The code attempts to determine the version of Pi it's running on, and will set the I2C bus automatically.

Enabling The I2C Port

You now use the raspbian config tool:

```
sudo raspi-config
```

Select 'Advanced Options' and then select the relevant option.

Enabling The I2C Port On Old Raspbian Distributions

The I2C ports need to be enabled in Raspbian before they can be used.

Edit the modules file

```
sudo nano /etc/modules
```

Add these lines:

```
i2c-bcm2708 i2c-dev
```

Exit and save the file

Now edit the modules blacklist file:

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Add a '#' character to this line so it commented out:

```
#blacklist i2c-bcm2708
```

Exit and save the file.

Finally install the I2C utilities:

```
sudo apt-get install python-smbus i2c-tools
```

Enter "sudo reboot" to restart the pi and now the I2C pins will be available to use.

Checking For Connected Devices

At the command prompt type one of these depending on whether you are using the I2C0 or I2C1 port:

```
sudo i2cdetect -y 0 //or sudo i2cdetect -y 1
```

The 7 bit I2C address of all found devices will be shown (ignoring the R/W bit, so I2C address 0000 0110 is displayed as hex 03).

Internal temperature of Raspberry PI 2

Tuesday, June 9, 2015 19:26

```
#!/bin/bash cpuTemp0=$(cat /sys/class/thermal/thermal_zone0/temp) cpuTemp1=
$(($cpuTemp0/1000)) cpuTemp2=$(($cpuTemp0/100)) cpuTempM=$(($cpuTemp2 % $cpuTemp1)) echo
CPU temp"="$cpuTemp1"."$cpuTempM"'C" echo GPU $(/opt/vc/bin/vcgencmd measure_temp)
```

RPI vcgencmd usage

Verified commands

Command outputs are from firmware version 362371.

`vcgencmd commands`

Shows a list of possible commands.

```
root@raspberrypi:~# vcgencmd commands commands="vcos, ap_output_control,
ap_output_post_processing, vchi_test_init, vchi_test_exit, pm_set_policy, pm_get_status,
pm_show_stats, pm_start_logging, pm_stop_logging, version, commands, set_vll_dir,
led_control, set_backlight, set_logging, get_lcd_info, set_bus_arbiter_mode, cache_flush,
otp_dump, codec_enabled, measure_clock, measure_volts, measure_temp, get_config,
hdmi_ntsc_freqs, render_bar, disk_notify, inuse_notify, sus_suspend, sus_status,
sus_is_enabled, sus_stop_test_thread, egl_platform_switch, mem_validate, mem_oom,
mem_reloc_stats, file, vctest_memmap, vctest_start, vctest_stop, vctest_set, vctest_get"
```

(I've just checked on a more recent firmware version and there's now also `get_camera`, `get_mem` and `hdmi_status_show` commands)

`vcgencmd measure_clock <clock>`

Shows clock frequency, clock can be one of arm, core, h264, isp, v3d, uart, pwm, emmc, pixel, vec, hdmi, dpi.

```
root@raspberrypi:~# \ > for src in arm core h264 isp v3d uart pwm emmc pixel vec hdmi dpi ; do
\ > echo -e "$src:\t$(vcgencmd measure_clock $src)" ; \ > done arm: frequency(45)=700000000
core: frequency(1)=250000000 h264: frequency(28)=0 isp: frequency(42)=250000000 v3d:
frequency(43)=250000000 uart: frequency(22)=3000000 pwm: frequency(25)=0 emmc:
frequency(47)=100000000 pixel: frequency(29)=154000000 vec: frequency(10)=0 hdmi:
frequency(9)=163682000 dpi: frequency(4)=0
```

`vcgencmd measure_volts <id>`

Shows voltage. id can be one of core, sdram_c, sdram_i, sdram_p, and defaults to core if not specified.

root@raspberrypi:~# \ > for id in core sdram_c sdram_i sdram_p ; do \ > echo -e "$id:\t $(vcgencmd measure_volts $id)" ; \ > done core: volt=1.20V sdram_c: volt=1.20V sdram_i: volt= 1.20V sdram_p: volt=1.23V

vcgencmd measure_temp

Shows core temperature of BCM2835 SoC.

root@raspberrypi:~# vcgencmd measure_temp temp=42.8'C

vcgencmd codec_enabled <*codec*>

Shows if the specified codec is enabled, codec can be one of H264, MPG2, WVC1, MPG4, MJPG, WMV9. Please note this was run on a Pi with the MPG2 and VC1 licences enabled.

root@raspberrypi:~# \ > for codec in H264 MPG2 WVC1 MPG4 MJPG WMV9 ; do \ > echo -e "$codec:\t$(vcgencmd codec_enabled $codec)" ; \ > done H264: H264=enabled MPG2: MPG2 =enabled WVC1: WVC1=enabled MPG4: MPG4=enabled MJPG: MJPG=enabled WMV9: WMV9=enabled

If you also follow the instructions in this Forum post then the codec options for VP6 and VP8 (WEBM) also work, this is experimental and unsupported at present so any changes are at your own risk.

vcgencmd get_config *[config|int|str]* Will print the configurations you have set. Argument can ether be a specific option or int, showing all configs with number-datatype, or str showing all configurations with datatype sting (aka text).

root@raspberrypi:~# vcgencmd get_config int arm_freq=1000 core_freq=500 sdram_freq=600 over_voltage=6 disable_overscan=1 force_pwm_open=1

vcgencmd get_mem *arm/gpu*

Shows how much memory is split between the CPU (arm) and GPU.

root@raspberrypi:~# vcgencmd get_mem arm && vcgencmd get_mem gpu arm=448M gpu= 64M

vcgencmd version

Shows the firmware version

```
root@raspberrypi:~# vcgencmd version Jan 13 2013 16:24:29 Copyright (c) 2012 Broadcom
version 362371 (release)
```

vcgencmd otp_dump

Displays the contents of the OTP (One Time Programmable) memory embedded inside the SoC.

```
root@raspberrypi:~# vcgencmd otp_dump 08:00000000 09:00000000 10:00000000
11:00000000 12:00000000 13:00000000 14:00000000 15:00000000 16:00280000 17:1020000a
18:1020000a 19:ffffffff 20:ffffffff 21:ffffffff 22:ffffffff 23:ffffffff 24:ffffffff 25:ffffffff 26:ffffffff
27:0000c2c2 28:6c14ba0d 29:93eb45f2 30:0000000e 31:00000000 32:00000000 33:00000000
34:00000000 35:00000000 36:00000000 37:00000000 38:00000000 39:00000000 40:00000000
41:00000000 42:00000000 43:00000000 44:00000000 45:00000000 46:00000000 47:00000000
48:00000000 49:00000000 50:00000000 51:00000000 52:00000000 53:00000000 54:00000000
55:00000000 56:00000000 57:00000000 58:00000000 59:00000000 60:00000000 61:00000000
62:00000000 63:00000000 64:00000000
```

Locations 28 and 30 store the *Serial* and *Revision* values that get displayed by /proc/cpuinfo (the *Serial* is also used to determine the Ethernet MAC address on Model B boards), and location 32 stores the value of the warranty bit. Purpose of values in other locations is unknown.

vcgencmd set_backlight

Currently unusable, might be used in the future to control the backlight of LCD displays

vcgencmd render_bar

Debug function created by Dom, used in OMXPlayer

Parameters and function of other vcgencmd commands are not known.

Set Up Flask --- Python based web server

Friday, June 5, 2015 10:55

Install Flask

```
sudo pip install Flask
```

Install Template Engine

```
pip install Jinja2
```

Install Support Library

```
pip install Werkzeug
```

Installing xmltodict

Wednesday, June 10, 2015 19:20

pip install xmltodict

redis slave

Thursday, June 11, 2015 13:41

SLAVEOF address port # follows default port is 6379

to cancel

SLAVEOF NO ONE

limited root file system size

Wednesday, June 17, 2015 17:18

Card image was sized for 2G
Use raspi-config to expand file size

setting up Samba

Thursday, June 18, 2015 09:30

Set up is taken from

http://raspberrypihq.com/how-to-share-a-folder-with-a-windows-computer-from-a-raspberry-pi/

Install and configure required software

To share network folders to a Windows computer we need to install some special software on the Raspberry Pi. The software providing the secret sauce this time is called *Samba.* The *Samba* software package implements the SMB protocol and provides support for the Windows naming service (WINS) and for joining a Windows Workgroup.

Installing the software is easy – login to your Raspberry Pi and run:

```
sudo apt-get install samba samba-common-bin
```

After installation configure the software by opening the file */etc/samba/smb.conf* using the command:

```
sudo nano /etc/samba/smb.conf
```

Read through the file and make sure you have the following parameters set:

```
workgroup = WORKGROUP wins support = yes
```

You can use anything as your workgroup name as long as it is alphanumerical and matches the workgroup you would like to join. The default workgroup in Windows 7 is WORKGROUP.

Setup folder to share

Next step is to create the folder you would like to share. To create a folder called “share” in your home directory do the following:

```
mkdir ~/share
```

With the folder created we can now tell the Samba software to share it on the network. Open the file */etc/samba/smb.conf* using the command:

```
sudo nano /etc/samba/smb.conf
```

```
add
```

```
 workgroup = WORKGROUP
 workgroup = WORKGROUP
```

Scroll to the bottom and add the following:

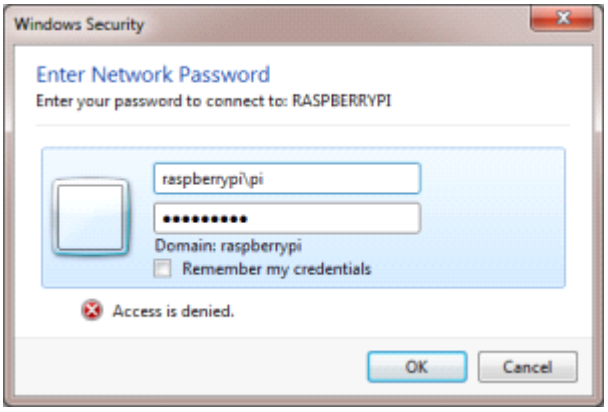
Notice how we tell Samba that public access is not allowed via “public=no” – this means that anyone wanting to access the shared folder must login with a valid user.

In this case the valid user is the user called “pi”. To let Samba know that “pi” is a network user run the command:

```
sudo smbpasswd -a pi
```

And enter pi’s password twice (default: raspberry).

At this point we can now login to the share from our Windows computer – use Domain: raspberrypi, User: pi and Password: raspberry (unless you changed the password) as you can see below:



If you do not want to deal with logging in you can always make the share publicly available by changing the config file to say:

```
public=yes
```

However please note that this is *extremely dangerous* since anyone will be able to access, modify and delete your files.

The reference smb.conf file is

```
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Some options that are often worth tuning have been included as
# commented-out examples in this file.
#  - When such options are commented with ";", the proposed setting
#    differs from the default Samba behaviour
#  - When commented with "#", the proposed setting is the default
#    behaviour of Samba but the option is considered important
#    enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
```

"testparm" to check that you have not made any basic syntactic

errors.

A well-established practice is to name the original file

"smb.conf.master" and create the "real" config file with

testparm -s smb.conf.master >smb.conf

This minimizes the size of the really used smb.conf file

which, according to the Samba Team, impacts performance

However, use this with caution if your smb.conf file contains nested

"include" statements. See Debian bug #483187 for a case

where using a master file is not a good idea.

#

#======================= Global Settings =======================

[global]

Browsing/Identification

Change this to the workgroup/NT-domain name your Samba server will part of

```
   workgroup = WORKGROUP



# server string is the equivalent of the NT Description field

   server string = %h server



# Windows Internet Name Serving Support Section:

# WINS Support - Tells the NMBD component of Samba to enable its WINS Server

   wins support = yes



# WINS Server - Tells the NMBD components of Samba to be a WINS Client

# Note: Samba can be either a WINS Server, or a WINS Client, but NOT both

;   wins server = w.x.y.z



# This will prevent nmbd to search for NetBIOS names through DNS.

   dns proxy = no



# What naming service and in what order should we use to resolve host names

# to IP addresses
```

;   name resolve order = lmhosts host wins bcast

Networking

The specific set of interfaces / networks to bind to

This can be either the interface name or an IP address/netmask;

interface names are normally preferred

;   interfaces = 127.0.0.0/8 eth0

Only bind to the named interfaces and/or networks; you must use the

'interfaces' option above to use this.

It is recommended that you enable this feature if your Samba machine is

not protected by a firewall or is a firewall itself. However, this

option cannot handle dynamic or non-broadcast interfaces correctly.

;   bind interfaces only = yes

Debugging/Accounting

This tells Samba to use a separate log file for each machine

that connects

log file = /var/log/samba/log.%m

Cap the size of the individual log files (in KiB).

max log size = 1000

If you want Samba to only log through syslog then set the following

parameter to 'yes'.

syslog only = no

We want Samba to log a minimum amount of information to syslog. Everything

should go to /var/log/samba/log.{smbd,nmbd} instead. If you want to log

through syslog you should set the following parameter to something higher.

syslog = 0

Do something sensible when Samba crashes: mail the admin a backtrace

```
   panic action = /usr/share/samba/panic-action %d




######## Authentication #######



# "security = user" is always a good idea. This will require a Unix account

# in this server for every user accessing the server. See

# /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/ServerType.html

# in the samba-doc package for details.

#   security = user



# You may wish to use password encryption.  See the section on

# 'encrypt passwords' in the smb.conf(5) manpage before enabling.

   encrypt passwords = true



# If you are using encrypted passwords, Samba will need to know what

# password database type you are using.

   passdb backend = tdbsam
```

obey pam restrictions = yes

This boolean parameter controls whether Samba attempts to sync the Unix

password with the SMB password when the encrypted SMB password in the

passdb is changed.

unix password sync = yes

For Unix password sync to work on a Debian GNU/Linux system, the following

parameters must be set (thanks to Ian Kahan <<kahan@informatik.tu-muenchen.de> for

sending the correct chat script for the passwd program in Debian Sarge).

passwd program = /usr/bin/passwd %u

passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n *password \supdated\ssuccessfully* .

This boolean controls whether PAM will be used for password changes

when requested by an SMB client instead of the program listed in

'passwd program'. The default is 'no'.

pam password change = yes

This option controls how unsuccessful authentication attempts are mapped

to anonymous connections

  map to guest = bad user

########## Domains ############

Is this machine able to authenticate users. Both PDC and BDC

must have this setting enabled. If you are the BDC you must

change the 'domain master' setting to no

#

;   domain logons = yes

#

The following setting only takes effect if 'domain logons' is set

It specifies the location of the user's profile directory

from the client point of view)

The following required a [profiles] share to be setup on the

samba server (see below)

;   logon path = \\%N\profiles\%U

Another common choice is storing the profile in the user's home directory

(this is Samba's default)

logon path = \\%N\%U\profile

The following setting only takes effect if 'domain logons' is set

It specifies the location of a user's home directory (from the client

point of view)

;   logon drive = H:

logon home = \\%N\%U

The following setting only takes effect if 'domain logons' is set

It specifies the script to run during logon. The script must be stored

in the [netlogon] share

NOTE: Must be store in 'DOS' file format convention

;   logon script = logon.cmd

This allows Unix users to be created on the domain controller via the SAMR

RPC pipe. The example command creates a user account with a disabled Unix

password; please adapt to your needs

; add user script = /usr/sbin/adduser --quiet --disabled-password --gecos "" %u

This allows machine accounts to be created on the domain controller via the

SAMR RPC pipe.

The following assumes a "machines" group exists on the system

; add machine script = /usr/sbin/useradd -g machines -c "%u machine account" -d /var/lib/samba -s /bin/false %u

This allows Unix groups to be created on the domain controller via the SAMR

RPC pipe.

; add group script = /usr/sbin/addgroup --force-badname %g

########## Printing ##########

If you want to automatically load your printer list rather

than setting them up individually then you'll need this

load printers = yes

lpr(ng) printing. You may wish to override the location of the

```
# printcap file
;   printing = bsd
;   printcap name = /etc/printcap


# CUPS printing.  See also the cupsaddsmb(8) manpage in the
# cupsys-client package.
;   printing = cups
;   printcap name = cups


############ Misc ############


# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
;   include = /home/samba/etc/smb.conf.%m


# Most people will find that this option gives better performance.
# See smb.conf(5) and /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/speed.html
# for details
```

```
# You may want to add the following on a Linux system:

#        SO_RCVBUF=8192 SO_SNDBUF=8192

#   socket options = TCP_NODELAY



# The following parameter is useful only if you have the linpopup package

# installed. The samba maintainer and the linpopup maintainer are

# working to ease installation and configuration of linpopup and samba.

;   message command = /bin/sh -c '/usr/bin/linpopup "%f" "%m" %s; rm %s' &



# Domain Master specifies Samba to be the Domain Master Browser. If this

# machine will be configured as a BDC (a secondary logon server), you

# must set this to 'no'; otherwise, the default behavior is recommended.

#   domain master = auto



# Some defaults for winbind (make sure you're not using the ranges

# for something else.)

;   idmap uid = 10000-20000

;   idmap gid = 10000-20000

;   template shell = /bin/bash
```

```
# The following was the default behaviour in sarge,

# but samba upstream reverted the default because it might induce

# performance issues in large organizations.

# See Debian bug #368251 for some of the consequences of *not*

# having this setting and smb.conf(5) for details.

;   winbind enum groups = yes

;   winbind enum users = yes



# Setup usershare options to enable non-root users to share folders

# with the net usershare command.



# Maximum number of usershare. 0 (default) means that usershare is disabled.

;   usershare max shares = 100



# Allow users who've been granted usershare privileges to create

# public shares, not just authenticated ones

   usershare allow guests = yes
```

```
#======================= Share Definitions =======================

[pihome]

  comment= Pi Home

  path=/home/pi

  browseable=Yes

  writeable=Yes

  only guest=no

  create mask=0777

  directory mask=0777

  public=no



[homes]

  comment = Home Directories

  browseable = no



# By default, the home directories are exported read-only. Change the

# next parameter to 'no' if you want to be able to write to them.

  read only = yes
```

File creation mask is set to 0700 for security reasons. If you want to

create files with group=rw permissions, set next parameter to 0775.

  create mask = 0700

Directory creation mask is set to 0700 for security reasons. If you want to

create dirs. with group=rw permissions, set next parameter to 0775.

  directory mask = 0700

By default, \\server\username shares can be connected to by anyone

with access to the samba server.

The following parameter makes sure that only "username" can connect

to \\server\username

This might need tweaking when using external authentication schemes

  valid users = %S

Un-comment the following and create the netlogon directory for Domain Logons

(you need to configure Samba to act as a domain controller too.)

;[netlogon]

; comment = Network Logon Service

```
;   path = /home/samba/netlogon
;   guest ok = yes
;   read only = yes


# Un-comment the following and create the profiles directory to store
# users profiles (see the "logon path" option above)
# (you need to configure Samba to act as a domain controller too.)
# The path below should be writable by all users so that their
# profile directory may be created the first time they log on
;[profiles]
;   comment = Users profiles
;   path = /home/samba/profiles
;   guest ok = no
;   browseable = no
;   create mask = 0600
;   directory mask = 0700


[printers]
   comment = All Printers
   browseable = no
```

```
   path = /var/spool/samba

   printable = yes

   guest ok = no

   read only = yes

   create mask = 0700



# Windows clients look for this share name as a source of downloadable

# printer drivers

[print$]

   comment = Printer Drivers

   path = /var/lib/samba/printers

   browseable = yes

   read only = yes

   guest ok = no

# Uncomment to allow remote administration of Windows print drivers.

# You may need to replace 'lpadmin' with the name of the group your

# admin users are members of.

# Please note that you also need to set appropriate Unix permissions

# to the drivers directory for these users to have write rights in it
```

```
;   write list = root, @lpadmin



# A sample share for sharing your CD-ROM with others.

;[cdrom]

;   comment = Samba server's CD-ROM

;   read only = yes

;   locking = no

;   path = /cdrom

;   guest ok = yes



# The next two parameters show how to auto-mount a CD-ROM when the

# cdrom share is accesed. For this to work /etc/fstab must contain

# an entry like this:

#

#       /dev/scd0   /cdrom  iso9660 defaults,noauto,ro,user   0 0

#

# The CD-ROM gets unmounted automatically after the connection to the

#

# If you don't want to use auto-mounting/unmounting make sure the CD

# is mounted on /cdrom
```

#

;   preexec = /bin/mount /cdrom

;   postexec = /bin/umount /cdrom

[PiShare]
   comment=Raspberry Pi Share
   path=/home/pi
   browseable=Yes
   writeable=Yes
   guest ok=yes
   create mask = 0777
   directory mask = 0777
   public = yes

****
Also change permissions on /home/pi to 0777
chmod -R 0777
Remove this for production environment

setting up Cherry Py

Monday, July 6, 2015 09:45

sudo pip install cherrypy

auto mounting a usb drive

Monday, September 21, 2015 17:43

Identify the drive, in this case it's `/dev/sda1` with a UUID of `E033-1109` and type FAT:

`sudo blkid # /dev/sda1: LABEL="ELEMENTS" UUID="E033-1109" TYPE="vfat"`

Create a dir it will be mounted at:

`sudo mkdir /mnt/usbel`

Own it:

`sudo chown -R pi:pi /mnt/usbel`

You could manually mount it

`sudo mount -o uid=pi -o gid=pi -t vfat /dev/sda1 /mnt/usbel`

But let's auto-mount it instead:

`sudo vim /etc/fstab`

Add add the following line to the bottom, **it should all be on one line**:

`UUID=E033-1109 /mnt/usbel vfat auto,users,rw,uid=1000,gid=100,umask=0002 0 0`

We're mounting the drive by its UUID to ensure we always get the correct drive. We're setting `rw` to allow read-write permissions, `uid=1000` sets the user to "pi" (see the `id` command), `gid=100` sets the group to "users" (again, see the `id` command) and `umask=0002` sets `rwxrwxr-x` permissions on the mount.

For more details on the possible permissions, see What is umask and fstab permissions explained.

We can reapply our mounts with `sudo mount -a` before we `sudo reboot`.

I had issues where this just wouldn't mount on reboot. I'd have to `sudo mount -a` manually. I found a few different threads where Raspberry Pi 2 users were encountering the same issues, which seems to be caused by a bug in Linux/Debian. A permanent fix will apparently be in Debian's "Jessie" release, but until then adding a *delay* works to give the USB drive to initialise:

`sudo vi /mount/cmdline.txt`

Add to the end (others report `5` works, but I needed to up it to `10`):

`rootdelay=10`

used this instead
UUID="fb840100-a8f0-4b55-b821-6d97e4221363" /home/pi/usb_drv ext4 defaults 0 0

from following link
http://raspberrypi.stackexchange.com/questions/28451/external-hd-wont-automount

turning swap off

Tuesday, October 20, 2015 1:12 PM

Has to be done every time at reboot
swapoff --all

Pep8 formating

Wednesday, June 7, 2017 12:55 PM

To install
`pip install --upgrade autopep8`

From <https://github.com/hhatto/autopep8#installation>

To run
`autopep8 --in-place --aggressive --aggressive <filename>`

From <https://github.com/hhatto/autopep8#installation>

Scipy python 3 do

Monday, September 11, 2017 11:05 AM

sudo apt-get install python3-scipy

From <https://www.raspberrypi.org/forums/viewtopic.php?f=32&t=112308>