**Update and upgrade Raspian installation**

sudo apt-get update

sudo apt-get upgrade

**Settings Linux(Raspbian-stretch-lite)**

Set wireless Access Point

* access point software installed,
* DHCP server software to provide connecting devices with a network address

**HELP Link**:

1. <https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>
2. <https://github.com/SurferTim/documentation/blob/6bc583965254fa292a470990c40b145f553f6b34/configuration/wireless/access-point.md>

Install required software:

sudo apt-get install dnsmasq hostapd

Turn off softwares until setting the configuration files.

sudo systemctl stop dnsmasq

sudo systemctl stop hostapd

Configure static IP address for our custom wireless network

1. Edit dhcpcd configuration file:

sudo nano /etc/dhcpcd.conf

Put this at the end of the file.

interface wlan0

static ip\_address=192.168.4.1/24

nohook wpa\_supplicant

Restart Dhcpcd daemon to configure wlan0 interface.

sudo service dhcpcd restart

1. Configure DHCP server (dnsmasq)

sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig

sudo nano /etc/dnsmasq.conf

Copy this into the file:

interface=wlan0 # Use the require wireless interface - usually wlan0

dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h

1. Configure the access point software (hostapd)

Edit file: sudo nano /etc/hostapd/hostapd.conf

Se parameters for wireless network wlan0

interface=wlan0

driver=nl80211

ssid=NameOfNetwork

hw\_mode=g

channel=7

wmm\_enabled=0

macaddr\_acl=0

auth\_algs=1

ignore\_broadcast\_ssid=0

wpa=2

wpa\_passphrase=AardvarkBadgerHedgehog

wpa\_key\_mgmt=WPA-PSK

wpa\_pairwise=TKIP

rsn\_pairwise=CCMP

Tell the system where to find this *hostapd* config file:

sudo nano /etc/default/hostapd

Replace #DAEMON\_CONF with this: DAEMON\_CONF=”/etc/hostapd/hostapd.conf”

Start it up now:

sudo systemctl start hostapd

sudo systemctl start dnsmasq

ADD ROUTING AND MASQUERADE

Edit file /etc/sysctl.conf and uncomment *net.ipv4.ip\_forward=1*

Add a masquerade for outbound traffic on eth0:

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

Save iptables rule:

sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"

Edit /etc/rc.local and add this just above "exit 0" to install these rules on boot.

iptables-restore < /etc/iptables.ipv4.nat

Reboot

Sudo shutdown –r now

1. Assign the .local domain to raspberry pi

Ip address 🡪 hostname.local

* 1. Install bonjour support on raspberry pi  
     sudo apt-get install avahi-daemon
  2. On windows download and install Bonjour Print Services dor windows v2.0.2: <https://support.apple.com/kb/DL999?locale=en_US>

# **Set wireless network:**

Wireless network management:

1. Using default raspbian-stretch manager: (dhcpcd)
   1. <https://wiki.archlinux.org/index.php/dhcpcd>
   2. <https://www.raspberrypi.org/forums/viewtopic.php?t=191453>
   3. <https://www.raspberrypi.org/forums/viewtopic.php?t=191061>
   4. <https://raspberrypi.stackexchange.com/questions/73749/how-to-connect-to-wifi-without-reboot>
   5. (Switching between wifi networks) <https://www.raspberrypi.org/forums/viewtopic.php?t=160620>
   6. https://electrondust.com/2017/11/25/setting-raspberry-pi-wifi-static-ip-raspbian-stretch-lite/
2. Using apt-get install wicd-cli:   
   https://www.raspberrypi.org/forums/viewtopic.php?t=160620
3. <https://wiki.archlinux.org/index.php/Wicd>
4. <https://packages.debian.org/jessie/wicd-cli>

# **Installing Apache:**

Links:

1. <https://www.digitalocean.com/community/tutorials/how-to-configure-the-apache-web-server-on-an-ubuntu-or-debian-vps>
2. <https://superuser.com/questions/345895/how-do-i-move-apaches-www-folder-and-then-create-a-symbolic-back-link-to-it-und>
3. <https://www.digitalocean.com/community/tutorials/how-to-move-an-apache-web-root-to-a-new-location-on-ubuntu-16-04>
4. <https://www.digitalocean.com/community/questions/discussion-about-permissions-for-web-folders>
5. <https://zaclee.net/apache/errors-apache/invalid-command-header>

sudo apt-get install apache2 –y

**Enable apache modules:**

sudo a2enmod headers

sudo a2enmod wsgi

Create a directory for the web app “*/home/pi/MeterWebApp/html”,* and inside create ***index.html*** file for the start page.

Add the *pi* user to apache webserver *www-data* group:

*sudo usermod –aG www-data<group> pi<user>*

Change Folders permissions to 664 or 775 (user and group to +rw)

*chmod –Rv ug+rw MeterWebApp*

Create an apache available websites configuration file:

File: /etc/apache2/sites-available/<MeterWebApp.conf>

*<VirtualHost \*:82>*

*# The ServerName directive sets the request scheme, hostname and port that*

*# the server uses to identify itself. This is used when creating*

*# redirection URLs. In the context of virtual hosts, the ServerName*

*# specifies what hostname must appear in the request's Host: header to*

*# match this virtual host. For the default virtual host (this file) this*

*# value is not decisive as it is used as a last resort host regardless.*

*# However, you must set it for any further virtual host explicitly.*

*ServerName datalogger.mydomain.com*

*ServerAdmin webmaster@localhost*

*DocumentRoot /home/pi/RaspDataLogger/DataloggerWebApp/html*

*#Alias /static /home/pi/RaspDataLogger/DataloggerWebApp/html*

*Header set Access-Control-Allow-Origin "\*"*

*Header set Access-Control-Allow-Methods "PUT,GET,POST,DELETE,OPTIONS"*

*Header set Access-Control-Allow-Headers "x-requested-with, content-type, accept"*

*Header set Access-Control-Allow-Headers "origin, accept, content-type"*

*Header always set Access-Control-Max-Age "1000"*

*Header set Access-Control-Allow-Headers "range"*

*<Directory /home/pi/RaspDataLogger/DataloggerWebApp/html/>*

*Order allow,deny*

*Allow from all*

*DirectoryIndex index.html*

*AllowOverride all*

*Require all granted*

*</Directory>*

*WSGIScriptAlias /v1 /home/pi/RaspDataLogger/DataloggerWebApp/WebApi/v1/application.wsgi*

*WSGIDaemonProcess AppWebSrvCfg user=pi group=pi processes=1 threads=5*

*#WSGIDaemonProcess AppWebSrvCfg user=www-data group=www-data processes=1 threads=5*

*<Directory /home/pi/RaspDataLogger/DataloggerWebApp/WebApi/v1>*

*WSGIProcessGroup AppWebSrvCfg*

*WSGIApplicationGroup %{GLOBAL}*

*Order allow,deny*

*Allow from all*

*AllowOverride all*

*Require all granted*

*</Directory>*

*# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,*

*# error, crit, alert, emerg.*

*# It is also possible to configure the loglevel for particular*

*# modules, e.g.*

*#LogLevel info ssl:warn*

*ErrorLog ${APACHE\_LOG\_DIR}/error.log*

*CustomLog ${APACHE\_LOG\_DIR}/access.log combined*

*# For most configuration files from conf-available/, which are*

*# enabled or disabled at a global level, it is possible to*

*# include a line for only one particular virtual host. For example the*

*# following line enables the CGI configuration for this host only*

*# after it has been globally disabled with "a2disconf".*

*#Include conf-available/serve-cgi-bin.conf*

*</VirtualHost>*

**Enable website/webapp:**

*sudo a2ensite <virtual host file name****: MeterWebApp.conf*** *>*

**Diasable website/webapp:**

*sudo a2dissite <virtual host file name****: MeterWebApp.conf*** *>*

**Tell apache to reload configuration file:**

*Sudo service apache2 reload*

**Edit /etc/apache2/ports.conf file:**

Add to /etc/apache2/ports.conf, the ports on which server is listening

# **Linux doc for writing shell scripts:**

<https://www.shellscript.sh/>

<https://www.raspberrypi.org/documentation/linux/usage/scripting.md>

<https://www.raspberrypi.org/documentation/linux/usage/>

# **Create SYSTEMD services:**

<https://www.raspberrypi.org/documentation/linux/usage/systemd.md>

# **Installing Bottle:**

1. Install python-pip, python-dev, build-essential  
   sudo apt-get install python-pip build-essential python-dev
2. Upgrade pip for python 2.7  
   *sudo pip install –upgrade pip*
3. Install bottle: sudo pip install bottle
4. Install python module for Request logging for WSGI application  
   help page: <https://github.com/pklaus/wsgi-request-logger>  
   install cmd: *sudo pip install wsgi-request-logger*

# **Installing bottle plugins:**

1. Install SQlite3 on Raspian stretch:
   1. Sudo apt-get install sqlite3
   2. <https://iotbytes.wordpress.com/sqlite-db-on-raspberry-pi/>
   3. <http://raspberrywebserver.com/sql-databases/set-up-an-sqlite-database-on-a-raspberry-pi.html>
2. Install bottle\_CorkPlugin: sudo pip install bottle-cork
3. Install bottle\_SQLitePlugin: sudo pip install bottle-sqlite

Bottle config stuffs:

How to specify the absolute path to the templates folder in a portable way by determining it at *runtime* with code like this:

import os

abs\_app\_dir\_path = os.path.dirname(os.path.realpath(\_\_file\_\_))

abs\_views\_path = os.path.join(abs\_app\_dir\_path, 'views')

bottle.TEMPLATE\_PATH.insert(0, abs\_views\_path )

# [**How to set up Mod\_WSGI for Python on Ubuntu**](https://serverfault.com/questions/91468/how-to-set-up-mod-wsgi-for-python-on-ubuntu)

<https://serverfault.com/questions/91468/how-to-set-up-mod-wsgi-for-python-on-ubuntu>

<https://modwsgi.readthedocs.io/en/develop/user-guides/configuration-guidelines.html>

sudo apt-get install libapache2-mod-wsgi

sudo nano /etc/apache2/mods-available/wsgi.load

LoadModule wsgi\_module /usr/lib/apache2/modules/mod\_wsgi.so

sudo a2enmod wsgi

sudo service apache2 restart

# **Installing Git:** *sudo apt-get install git*