The following instructions are for setting up the local server Raspberry Pi. This guide is assuming that the Raspberry Pi is a model with onboard wifi.

The first step is to ensure that the Server Pi is up do date (it would be wise to go ahead and update the detector pi(s) as well).
Copy and paste the following commands in the terminal:

```
sudo apt-get update
sudo apt-get upgrade
```

*Note: Guides to do this online always say to update and then upgrade the raspberry pi. However, it was found that the raspberry pi need to be updated again after the upgrade in certain circumstances. So, to be safe run the update command again.

```
sudo apt-get update
```

Next, the server pi needs to have the DNS Masquerade and the hostapd software installed.
CAUTION Pay attention to the terminal and make sure that both pieces of software are installed. Sometimes, the operating system runs into issues and can't install the hostapd software on the first go. If this happens, re-run the update command and try again. This is a known issue with no plausible fix.

```
sudo apt-get install dnsmasq
sudo apt-get install hostapd
```

Next, stop the services since they have not been configured.

```
sudo systemctl stop dnsmasq
sudo systemctl stop hostapd
```

Now, we need to assign a static IP to the server pi if it is to act as an access point.
To do so, type the following command into the terminal:

```
sudo nano /etc/dhcpcd.conf
```

Then, add the following to the end of the file and save it with 'Ctrl o' and exit with 'Ctrl x':

```
interface wlan0
   static ip_address=192.168.4.1/24
```

The dhcpcd daemon service now needs to be restarted, copy the following command into the terminal:

sudo service dhcpcd restart

Now, the DHCP (Dynamic Host Configuration Protocol) server needs to be configured (This is the dnsmasq). To do this, a new config file will be created and the old one will be saved. Type the following into the terminal:

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

Then, type the following into the new config file:

Interface=wlan0 #Use the required wireless interface - usually wlan0
dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h

The above code assigns IP addresses between 192.168.4.x where x ranges from 2 - 20.

Now it is time to configure the host software (hostapd). To do this, a new hostapd.conf file needs to be created and edited. To do so, type the following into the terminal:

sudo nano /etc/hostapd/hostapd.conf

Now, paste the following into the file and edit the highlighted areas:

interface =wlan0
driver=nl80211
ssid=NAMEOFYOURNETWORK
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=YOURNETWORKPASSWORD
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

Save the file with Ctrl o and exit with Ctrl x.
Next, the system needs to be updated so that it knows where the hostapd.conf file is located. To do this, type the following in the terminal:

sudo nano /etc/default/hostapd

Next, find this line: #DAEMON_CONF=" " and replace it with the following:

DAEMON_CONF="/etc/hostapd/hostapd.conf"

*NOTE: Be sure to that when you are finished that the # has been removed from the front of the line otherwise the wireless access point will not function.

Now, the services need to be started again. Type the following into the terminal:

sudo systemctl start hostapd
sudo systemctl start dnsmasq

Next, we need to add masquerading and routing. Enter the following in the terminal:

sudo nano /etc/sysctl.conf

Now, look for the line: #net.ipv4.ip_forward=1 and remove the # in the beginning to uncomment the line. Save the file and exit.
Now, we will add masquerading to outbound traffic on eth0 (We will not actually be forwarding traffic to eth0, but the server pi will connect to the internet).
Paste the following in the terminal:

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

*Note: The previous step will not work if the hostapd software was not installed correctly.

Save the IP-table rule with the following command in the terminal:

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

Now, edit the rc.local file with the following command in the terminal:

sudo nano /etc/rc.local

And add the following line just below “exit 0”:

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

This will load the iptables rule at boot. Finally, restart the pi with the follow command:

sudo reboot

Your network SSID should now be visible via other devices. Once you join the network your pi is broadcasting, you should be able to ssh into it with the following command (if ssh was enabled in the Raspberry Pi’s configuration):

ssh pi@192.168.4.1

*Note: If you are using Windows you will need to download and install a program like Putty to ssh into the linux system.

Setting up Samba File Server

If following this document from the beginning then there is no need to upgrade and update your system. To install, type the following in the terminal:

sudo apt-get update
sudo apt-get upgrade
sudo apt-get install samba samba-common-bin

A shared filed directory needs to be created. Type the following in the terminal:

sudo mkdir -m 1777 /share

Samba’s config file needs to be edited:

sudo nano /etc/samba/smb.conf

Paste the following at the end of the file:

[share]
Comment = Shared samba file directory
Path = /share
Browseable = Yes
Writeable = Yes
only guest = No
create mask = 0777
directory mask = 0777
Public = Yes
Guest ok = Yes

Save the file with Ctrl o and exit with Ctrl x. Now, restart samba with the following command in the terminal:

sudo /etc/init.d/samba restart

Now that samba is installed and the folder is marked as shared, it should show up in the terminal with a different color (green by default) designating it's permissions. Any device connected on the network should be able to share to this folder if the device also has samba installed.