**Instructions for setting up Google Assistant-Relay on a Raspberry Pi 3b+**

- Use a fresh install of Raspian Desktop (2018-06-27-raspbian-stretch.zip) – follow steps found online at https://www.raspberrypi.org/documentation/installation/installing-images/README.md

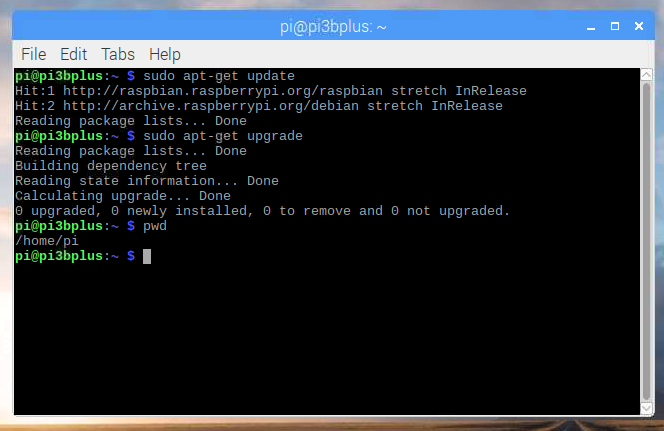
Log into the RPi desktop and open a terminal window.

**Update the Raspian Operating System**

sudo apt-get update

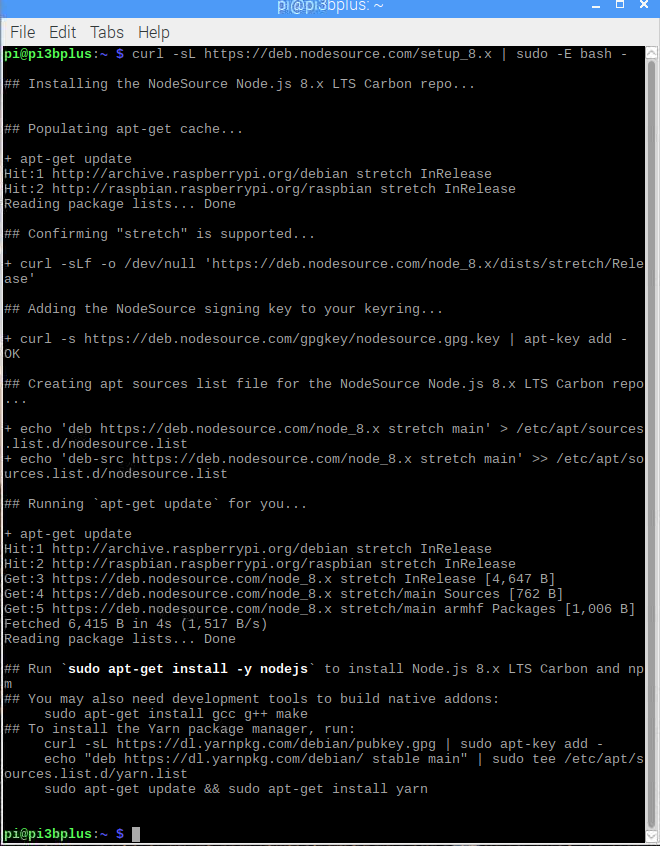
sudo apt-get upgrade

Reboot your RPi to complete installation of any upgrades.

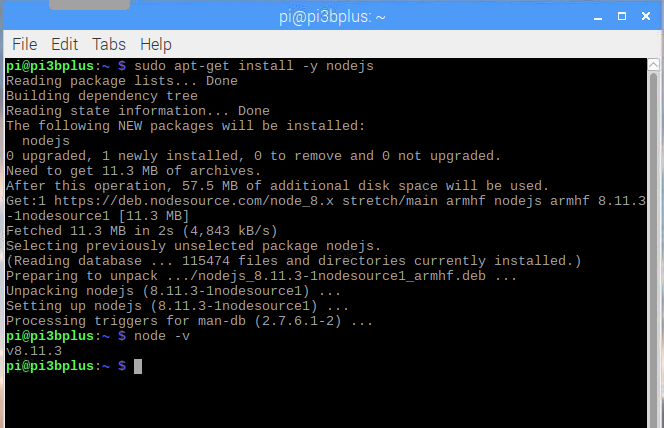


**Commands to install Node JS** (don’t use v10.x!)

curl -sL https://deb.nodesource.com/setup\_8.x | sudo -E bash -

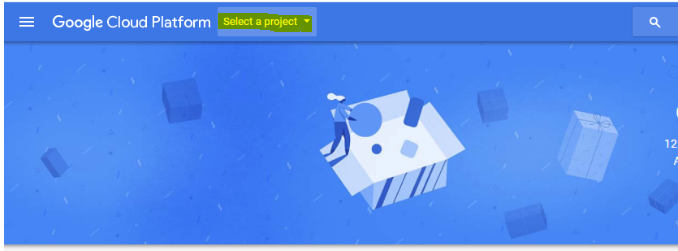


sudo apt-get install -y nodejs

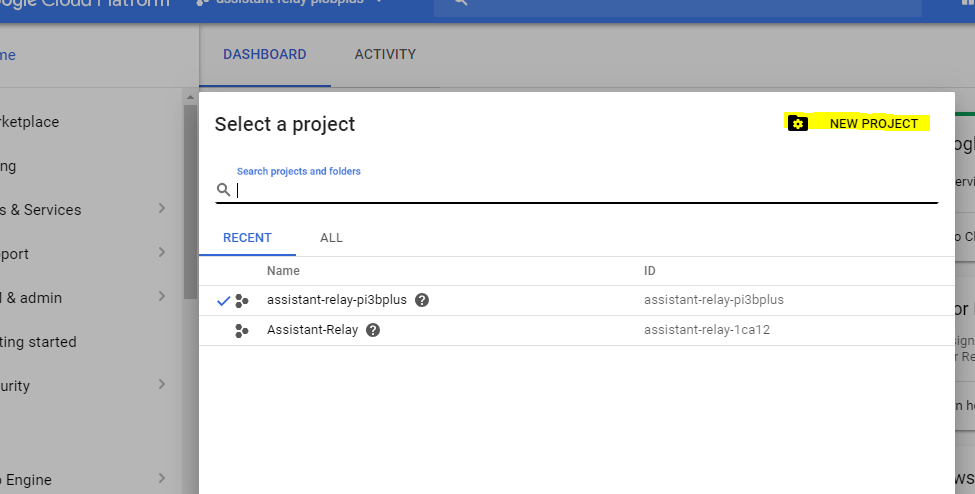


**Google Project Setup** (you can use any browser/computer for these steps. You do not need to use the RPi to create your Google Project.)

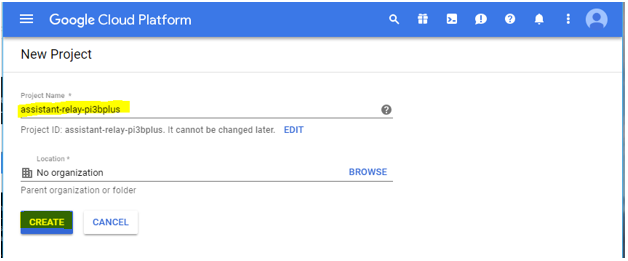
1. Go to [http://console.cloud.google.com](http://console.cloud.google.com/)
2. Click on “Select a Project”



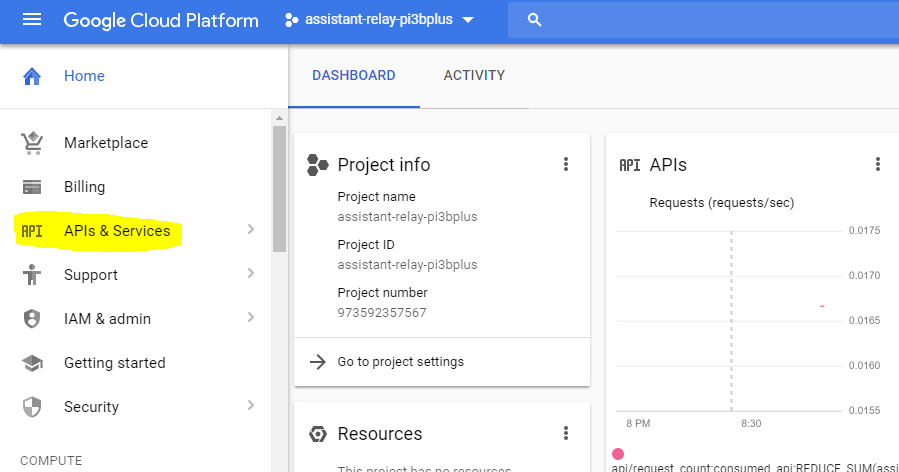
1. Click on NEW PROJECT



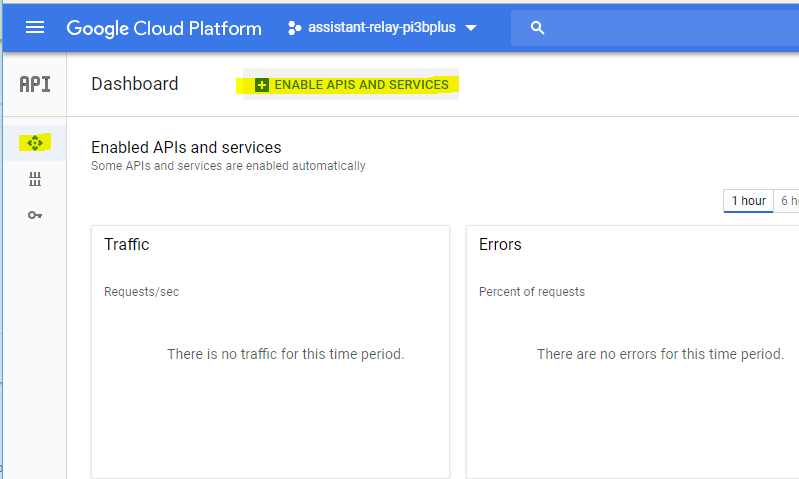
1. Type in the name of your project and click CREATE



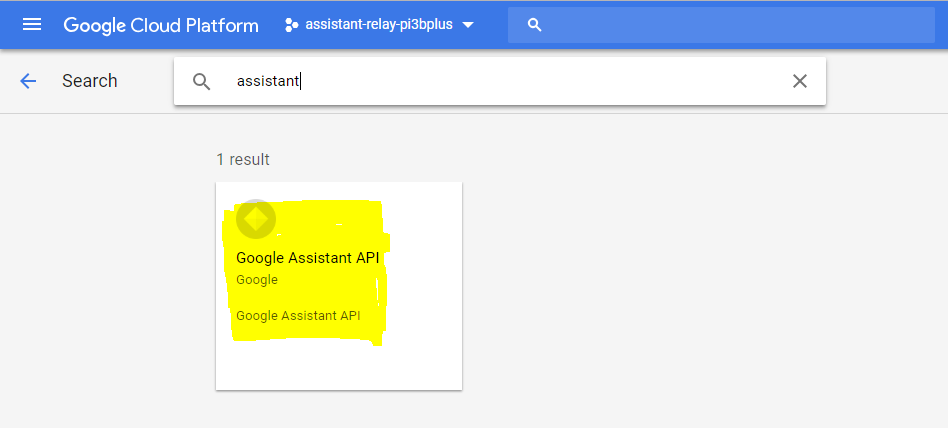
1. Wait until it is complete and then select your project from the same drop-down in step #2
2. Click on APIs and Services.



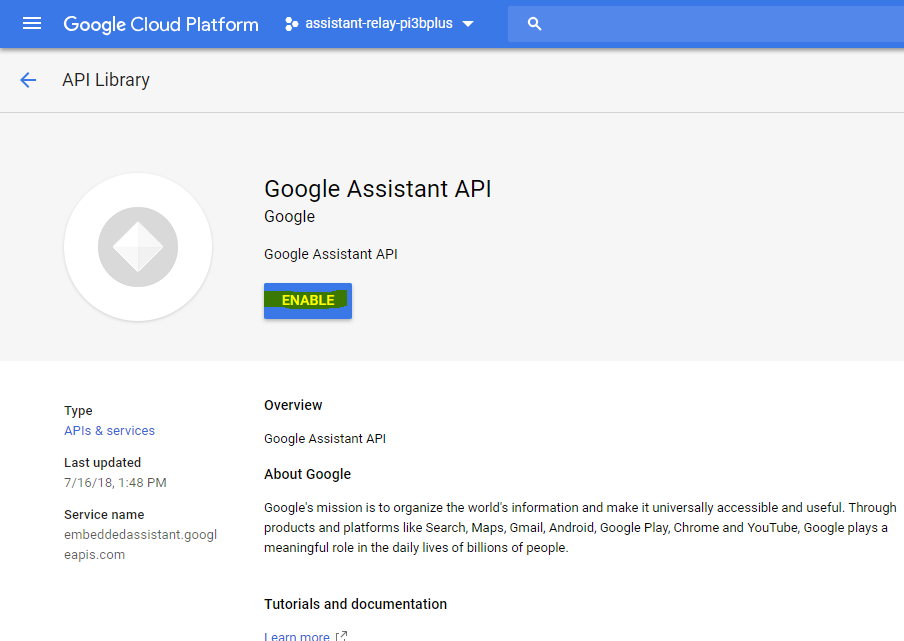
1. Click Enable APIs and Services.



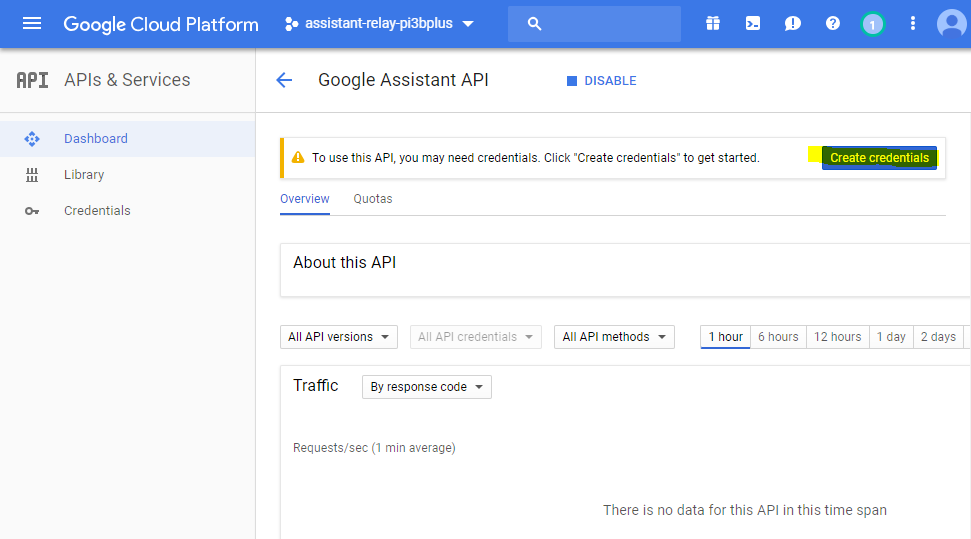
1. Start typing assistant in the search box until ‘Google Assistant API’ shows up, then click on it.



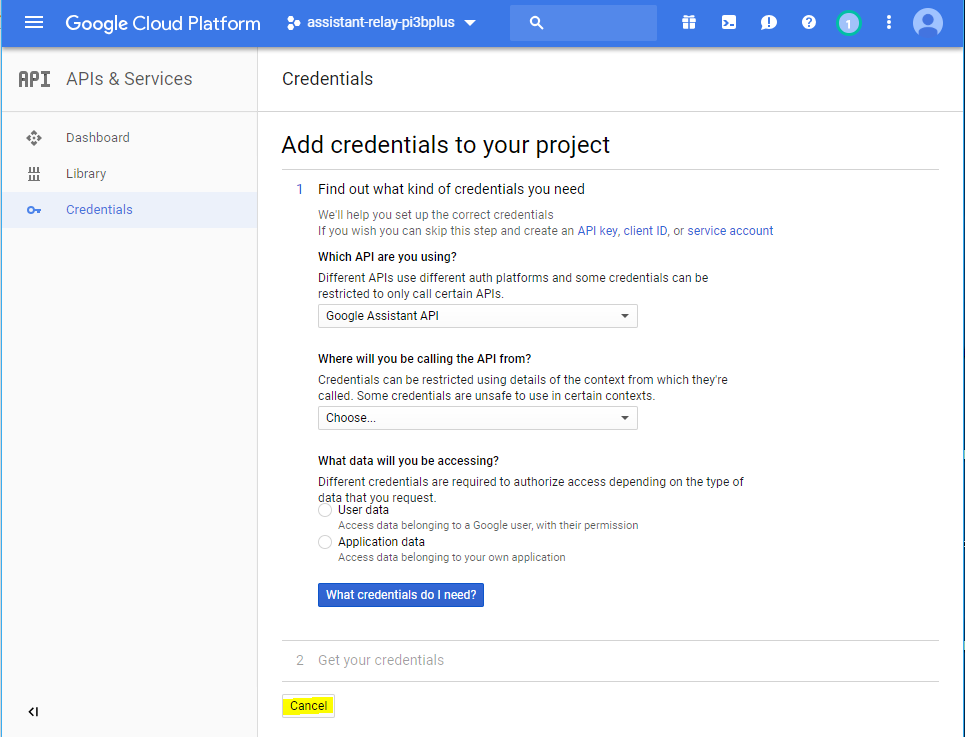
1. Click enable



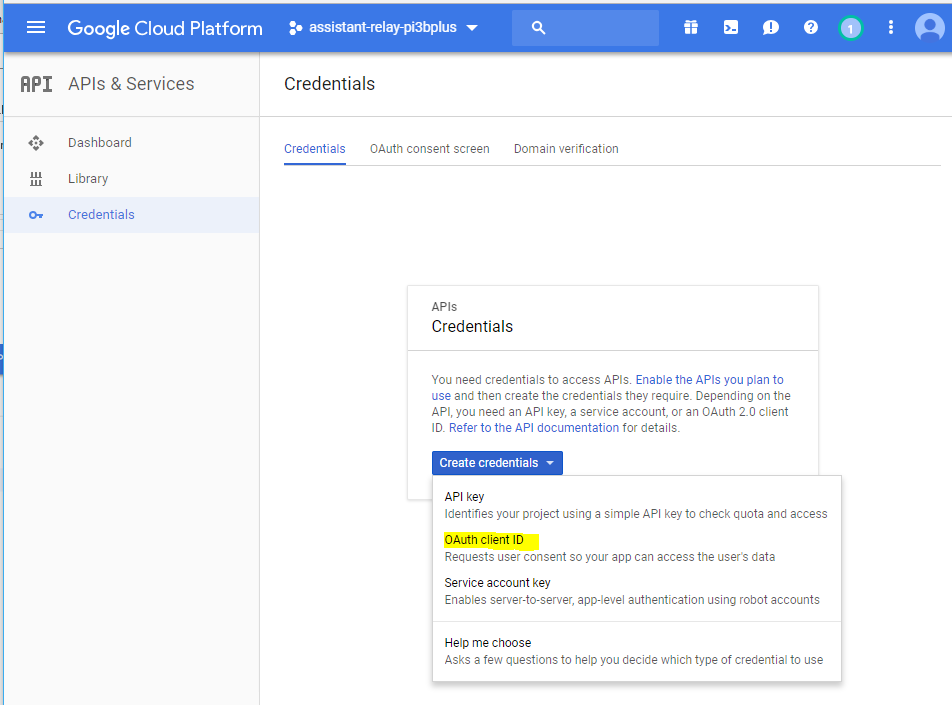
1. Click on Create Credentials



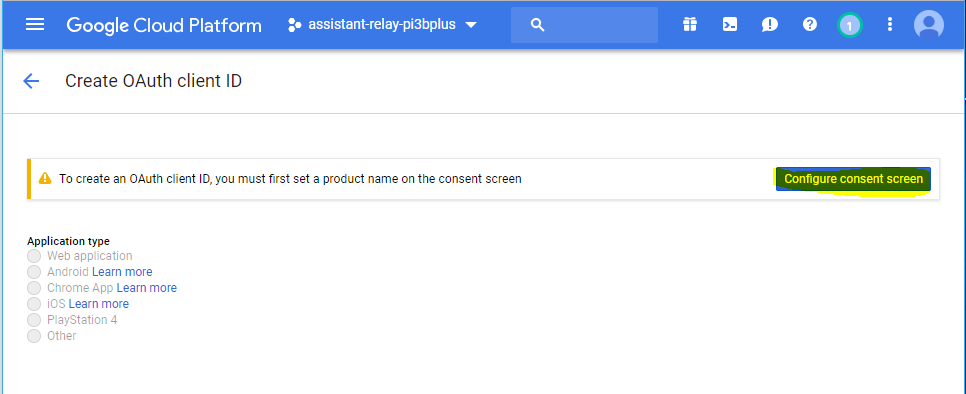
1. Click on CANCEL (I know… it seems counter-intuitive to me as well!)



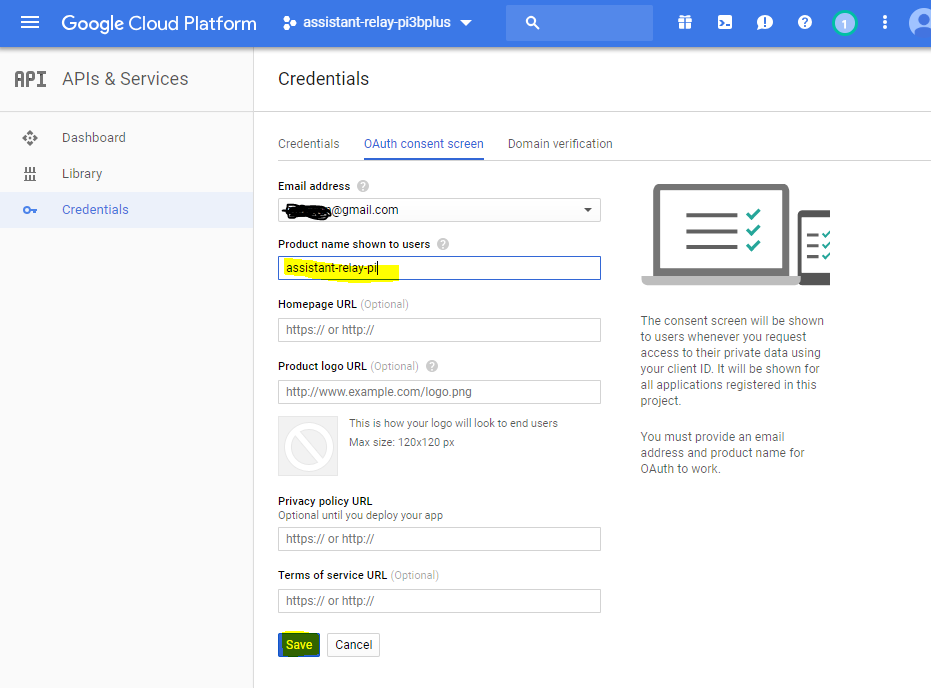
1. Click on Create Credentials then Oauth Client ID



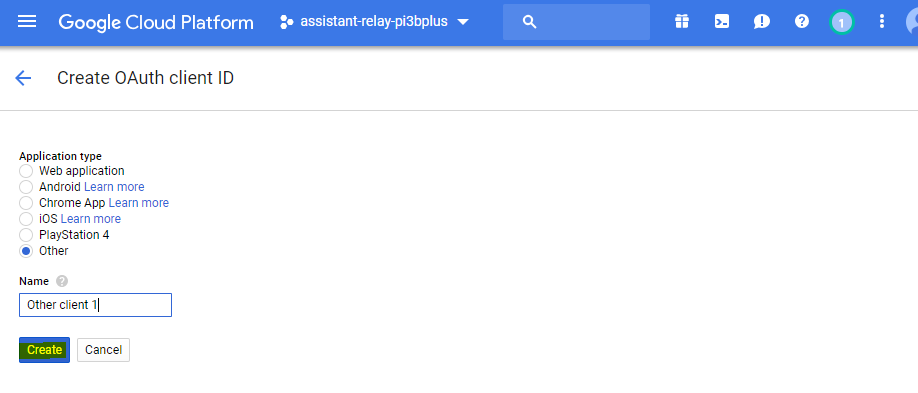
1. Click on Configure Consent Screen.



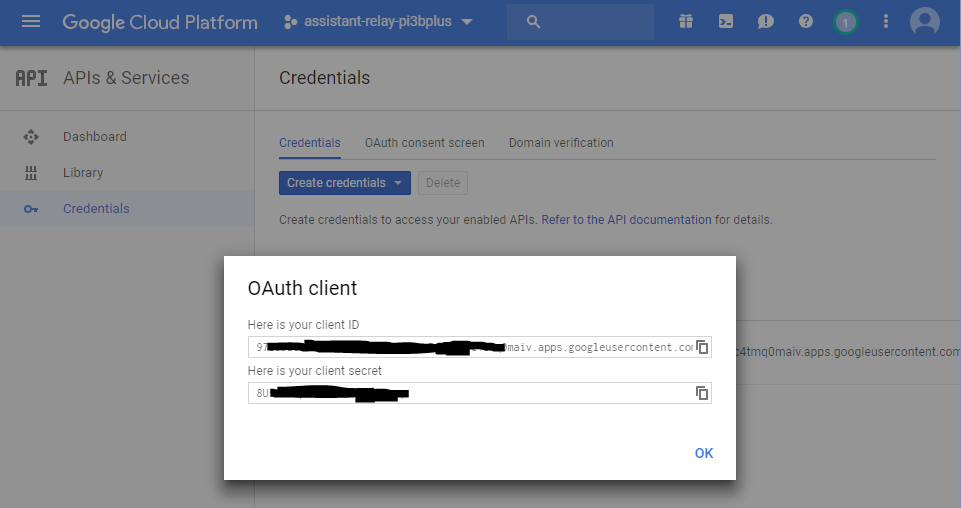
1. Fill in only the Name shown to users and click save. The rest is unused.



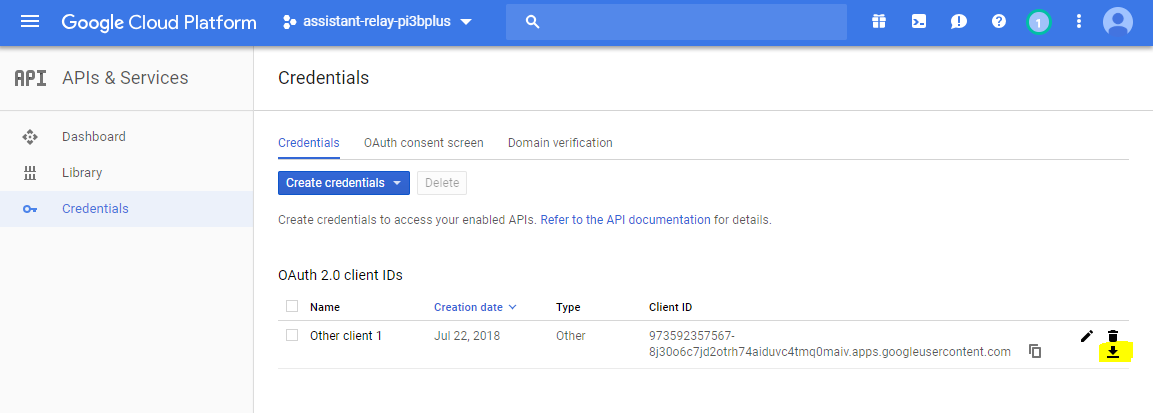
1. Click the radio button next to “Other” then click “Create”



1. The client ID and secret will pop up. You may copy them down if you’d like, however doing so is not required to use ‘assistant-relay’.



1. Click the download button.

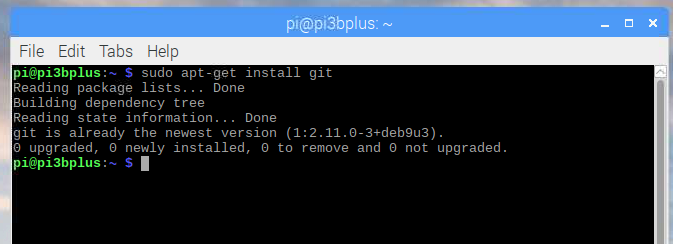


1. Once downloaded, rename the file to <your\_name>.json. For example, “ogiewon.json” NOTE: This file will be required later on during the configuration of assistant-relay on the RPi.

**Commands to install ‘assistant-relay’**

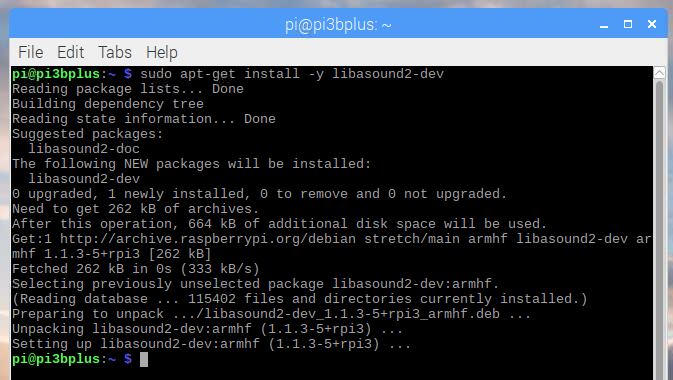
Install ‘git’ (in case it is not already installed)

sudo apt-get install git



Install ‘libasound2-dev’ (in case it is not already installed)

sudo apt-get install -y libasound2-dev

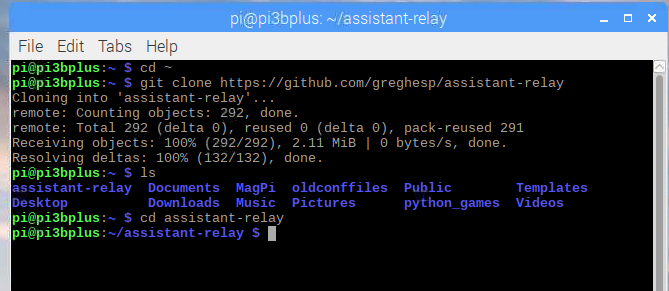


Download a copy of the ‘assistant-relay’ GitHub Repository

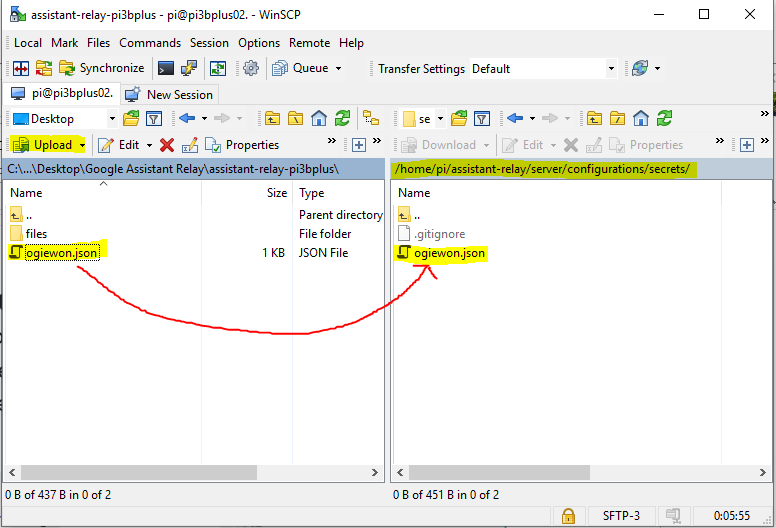
cd ~

git clone https://github.com/greghesp/assistant-relay.git

cd assistant-relay



Copy your .json file downloaded when you create your Google Project above into the ‘/home/pi/assistant-relay/server/configurations/secrets’ folder on your pi. I used WinSCP to do this as shown below, but however you choose to do it, you have to get it into the ‘/home/pi/assistant-relay/server/configurations/secrets’ folder created above by git.

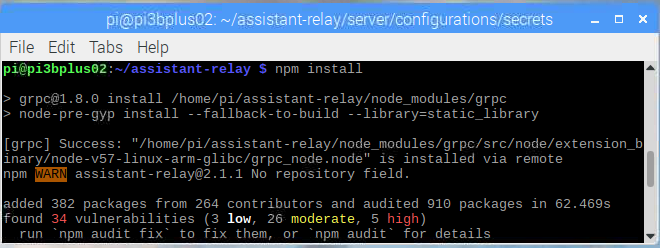


Verify the file is on your Raspberry Pi, in the ‘/home/pi/assistant-relay/server/configurations/secrets’ folder



Install assistant-relay using npm (you should still be in the \home\pi\assistant-relay folder)

npm install

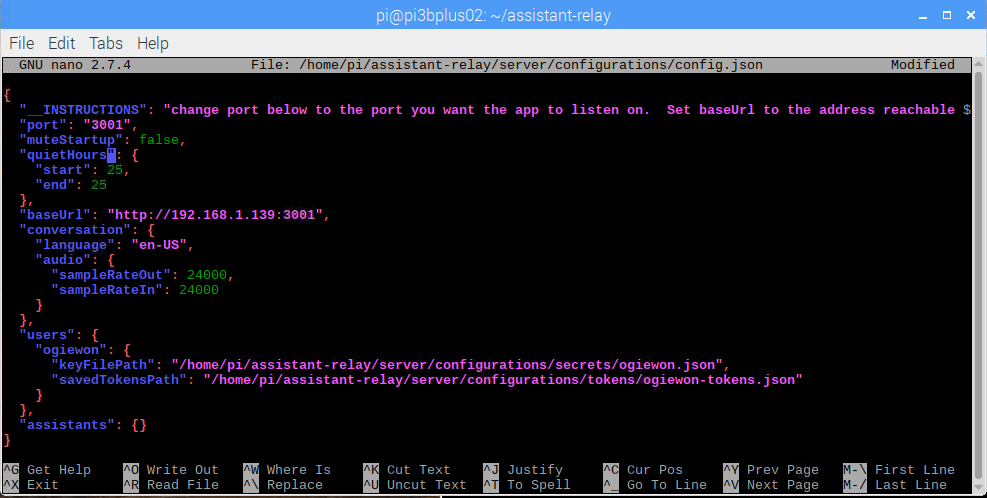


You can manually edit the config.json file found in /server/configurations to set this up.

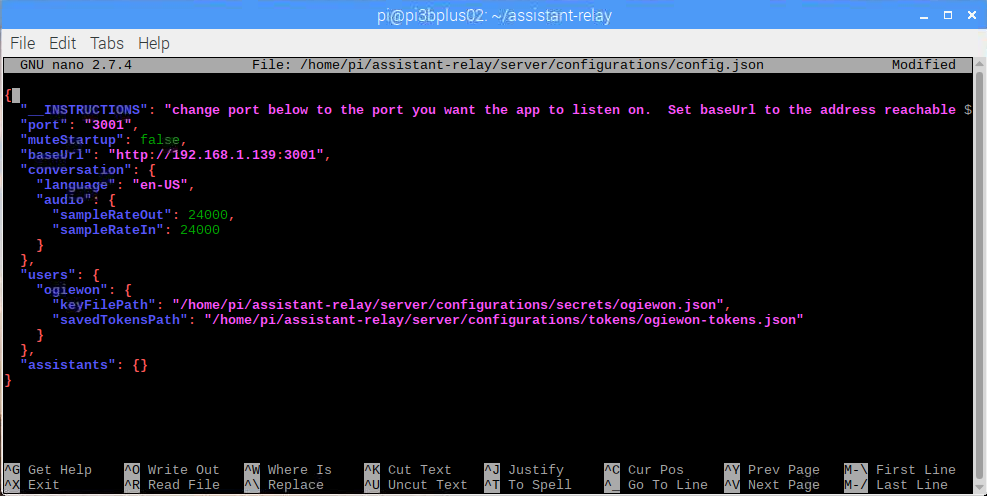
**Only edit the fields below**

| **Field** | **Decription** |
| --- | --- |
| port | What local port do you want Assistant Relay to listen on? |
| muteStartup | true or false - Will mute the startup announcement |
| quietHours | Whole numbers only. Sets the hours between which Assistant Relay will ignore commands |
| baseUrl | The port and URL to access your Assistant Relay instance on. This will normally be the machines local IP address, followed by the port set above. i.e. [http://192.168.1.102:3000](http://192.168.1.102:3000/) You can set this to an external URL using dyndns or similar if you wish to access from outside your network |

nano assistant-relay/server/configurations/config.json



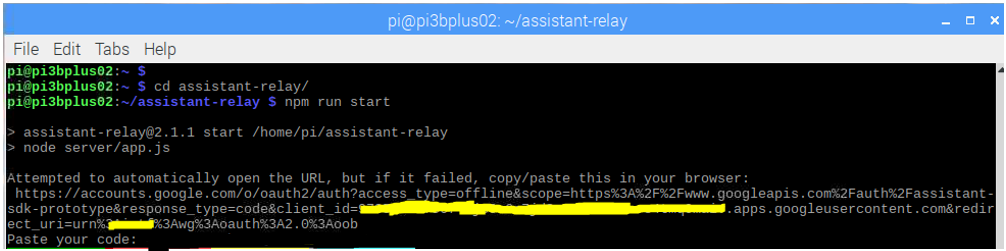
NOTE: I could never get the system to be out of “quiet hours” so I simply removed that section from the config.json file as shown below. The assistant-relay now works as expected (except for quiet hours, of course!)



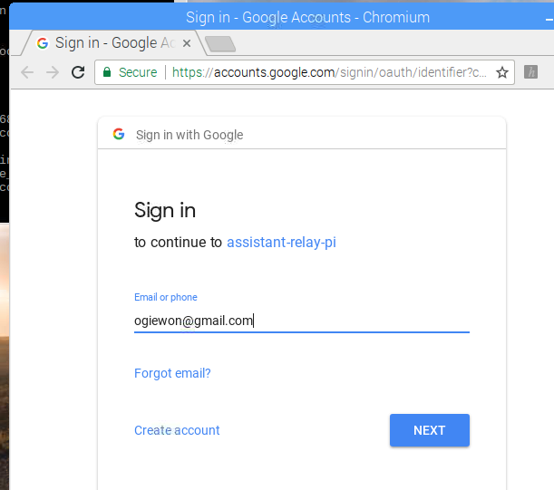
Note: The next step should be done form the RPi desktop, either using VNC or through a direct connection with a monitor and keyboard. You will need to copy and paste from a browser window to the terminal window a very long string.

Startup assistant-relay using the following command. This should cause a browser to be launched, which you will need to log into your Google account with to retrieve your code.

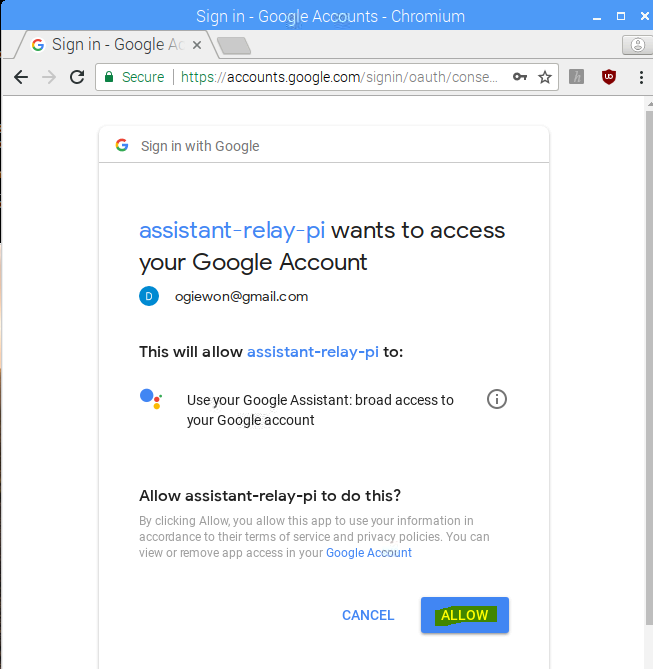
npm run start



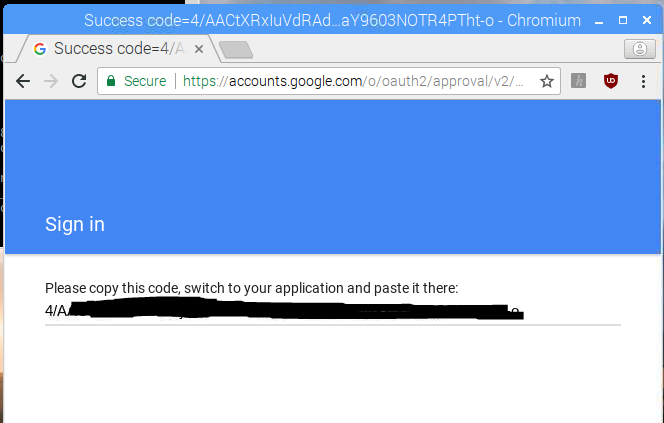
When your browser pops up, it will ask you to log into your google account. Do so.



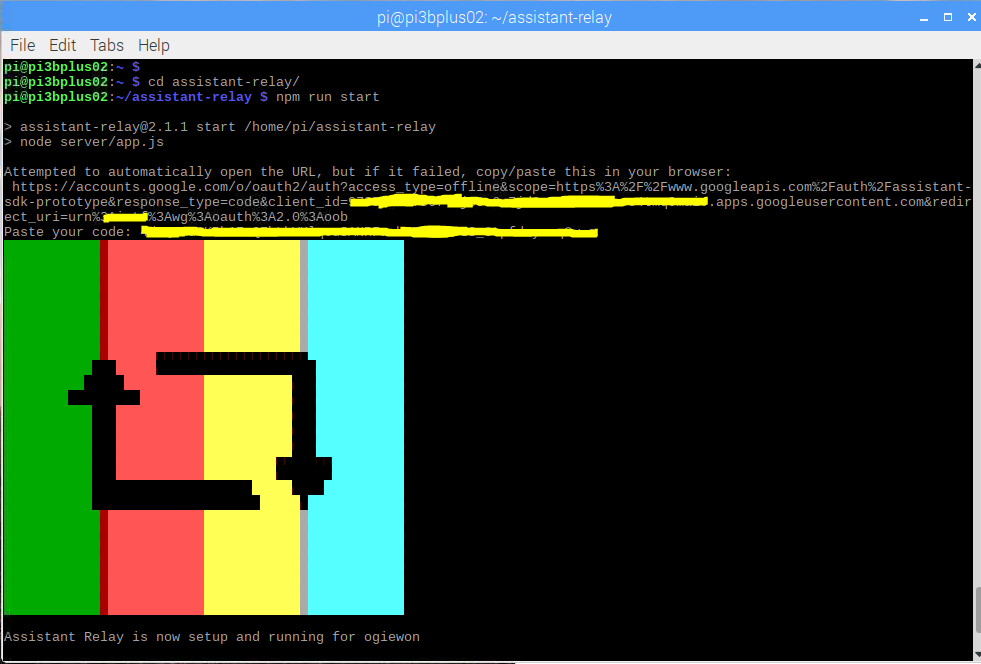
Click Allow on the following screen



The browser will display a key as shown below. Copy that key and paste it into your terminal window (Note: I simply used my mouse to Right-Click COPY from the browser and then Right-Click PASTE in the terminal window.)



After retrieving your code, paste it into the terminal window and hit <return>. You should see the following which means you’re ready!

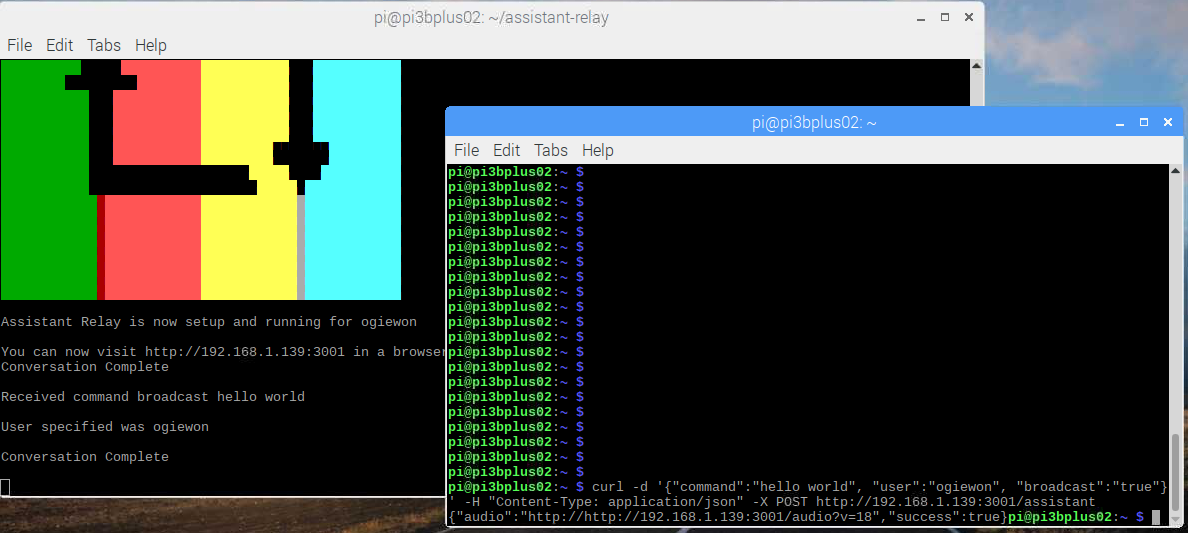


Assistant-Relay should start and you should get a welcome broadcast saying “Assistant relay is now setup and running for (your name here).”

Open a second Terminal Window and type in a test command such as

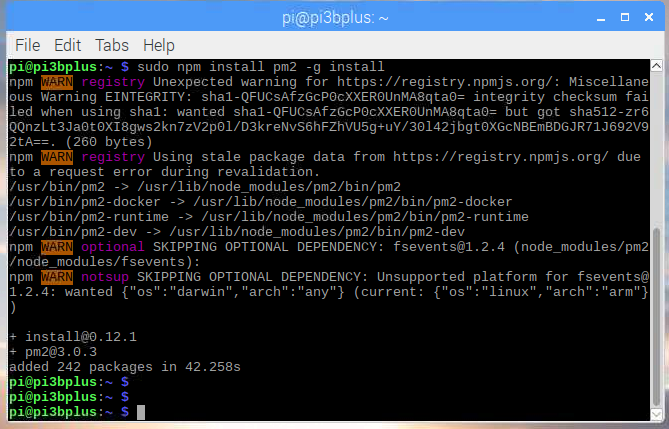
curl -d '{"command":"hello world", "user":"ogiewon", "broadcast":"true"}' -H "Content-Type: application/json" -X POST http://192.168.1.139:3001/assistant

Your Google Home devices should all say “Hello World” aloud. In the original terminal window, you will see the output of the **assistant-relay** server.



**Configure assistant-relay to start up every time your Raspberry Pi boots**

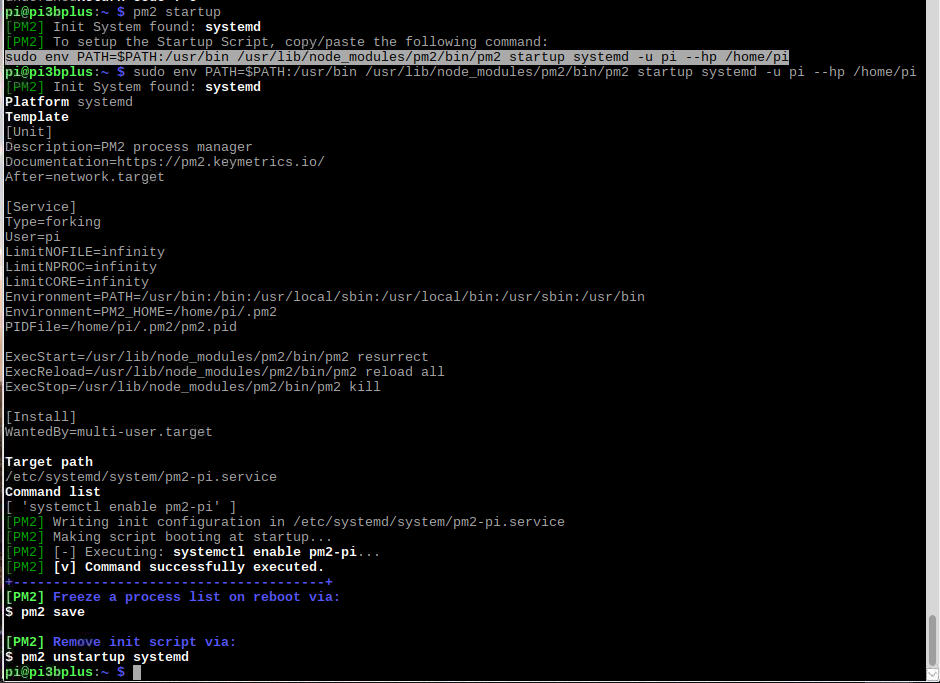
sudo npm install pm2 -g install



pm2 startup

This will generate a command to use to install pm2 to auto start as follows

sudo env PATH=$PATH:/usr/bin /usr/lib/node\_modules/pm2/bin/pm2 startup systemd -u pi --hp /home/pi

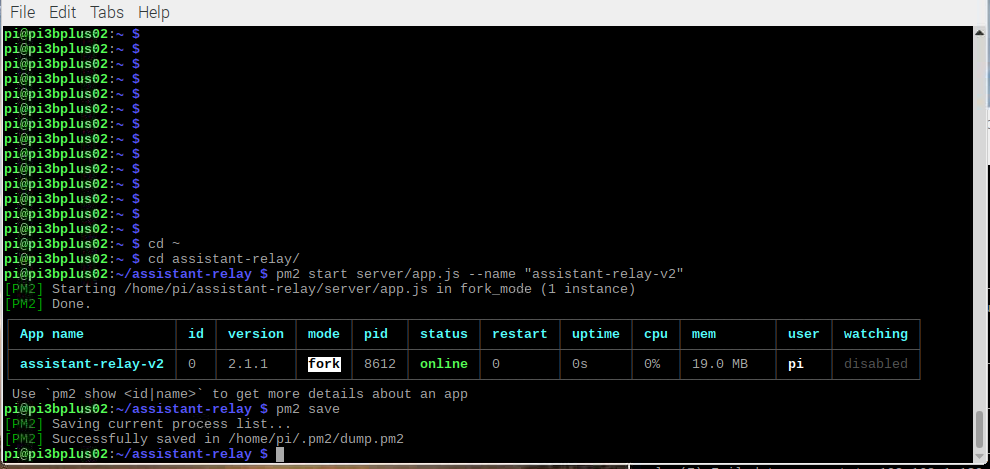


cd ~

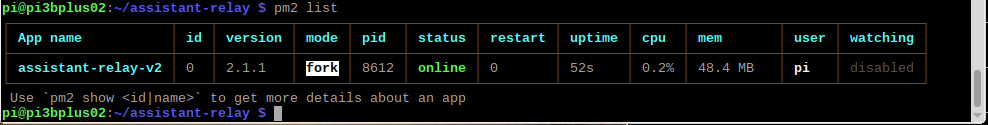
cd assistant-relay

pm2 start server/app.js --name "assistant-relay-v2"

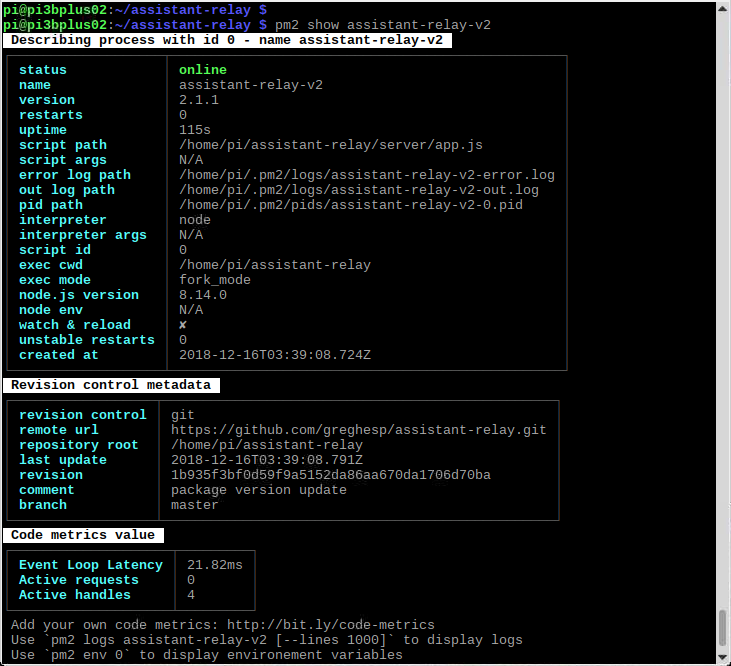
pm2 save



pm2 list



pm2 show assistant-relay-v2



At this point, you should be able to reboot your Raspberry Pi and have ‘assistant-relay’ automatically start up. To view the console log data from ‘assistant-relay’ simply type the following command.

more /home/pi/.pm2/logs/assistant-relay-v2-out.log

To stop ‘assistant-relay’, simply run

pm2 stop assistant-relay-v2

To monitor the pm2 processes, simply run

pm2 monit