# CHANGING HUE WITH THE RASPBERRY PI

# What are we going to do today?

We are going to take a completely fresh Raspberry Pi, install a few necessary things and then control the Hue Lightbulb!

**Raspberry Pi**

Raspberry Pi Zero W £9.60

**Philips Hue**

Philips Hue Go : £69.95

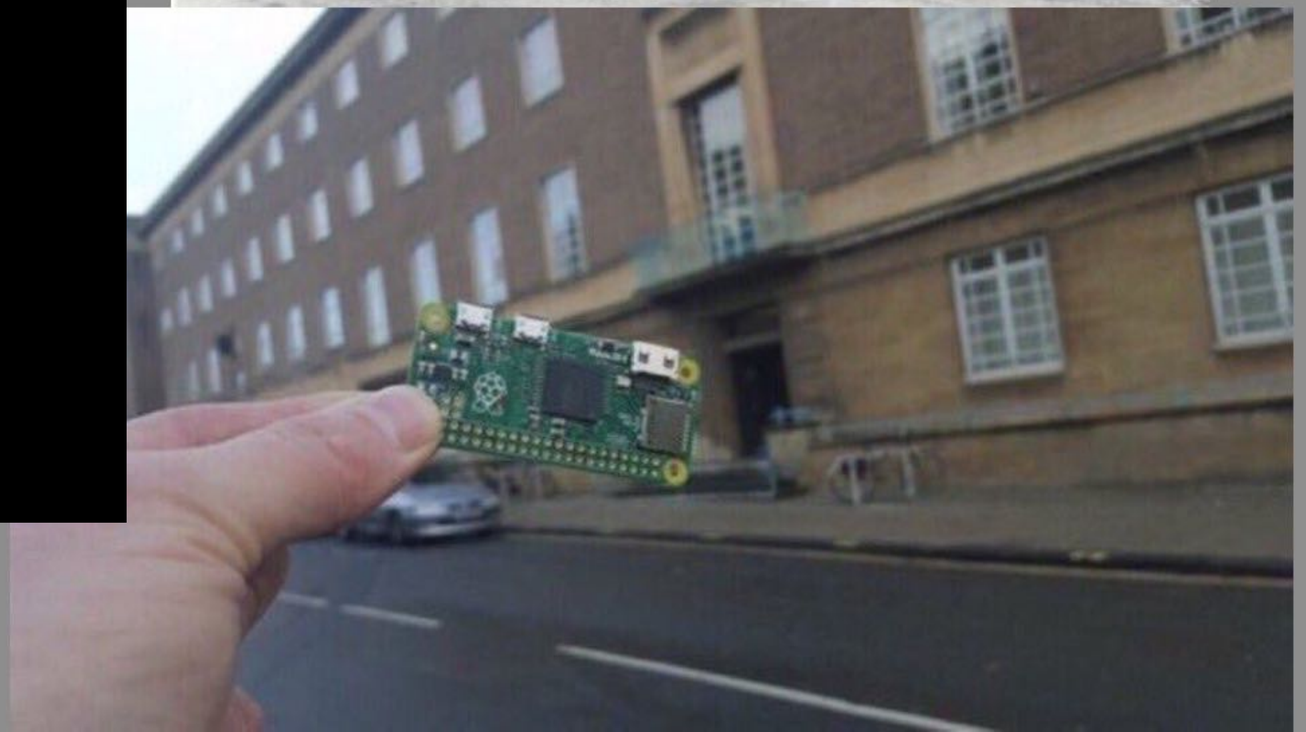
This is a different demonstration than normal and using the Pi gives us the opportunity to do things that you would never consider doing with a ‘proper’ server.

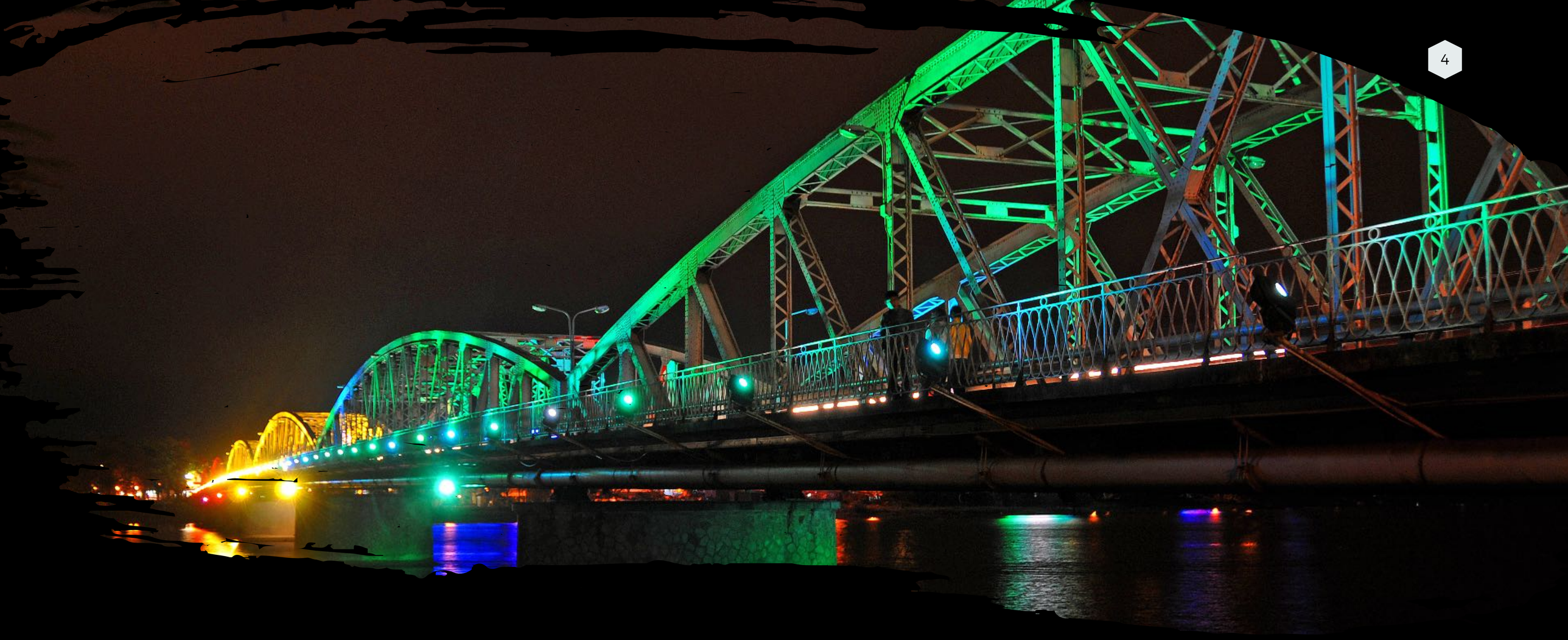
**@ValaAfshar**

1957: 13 people deliver a computer

2017: person can hold 13 computers in one hand

[both photos at same location]
https://twitter.com/ValaAfshar/status/849115595207393280



# HUE BRIDGE

Lets take a moment or two and explain how the Hue bridge works and how we can connect to it.

# Hue Bridge

Controls all aspects of all Hue devices

## Zigbee

Uses the open Zigbee radio standard to talk to every device.

## Controls Devices

The hub is used to add new devices (bulbs/switches/sensors) and controls which apps can be used.

## Well Documented API

The Hue hub has great well documented API and resources, others have created wrappers for different languages such as PHP (which is what we will be using!).

## Homekit/AVS enabled

The newer square hub works out of the box with Apple HomeKit and can work with Amazon Voice services and everything else cool!

# So now what?

So we have our Hue light connected to the internet and a Pi with Apache and PHP installed on it.  What could we possibly do with the Hue bulb?


## Hue IOS/Android Apps

Allows setup/control of the lights and also ‘Hey Siri’ and google voice alternative


## Attach sensors or switches

Having voice control is great but a light switch is easer for your grandmother to use


## Connect to external services

Lots of services like IFTTT or Zapier can connect and control your lights


## Code your own API interactions!

This is what we are going to do, because we can!

# Setting up the Raspberry Pi

Before we can do anything we need to install some essential packages

### Update the Pi

It's always a good idea to at least start a new project with everything up to date

### Apache

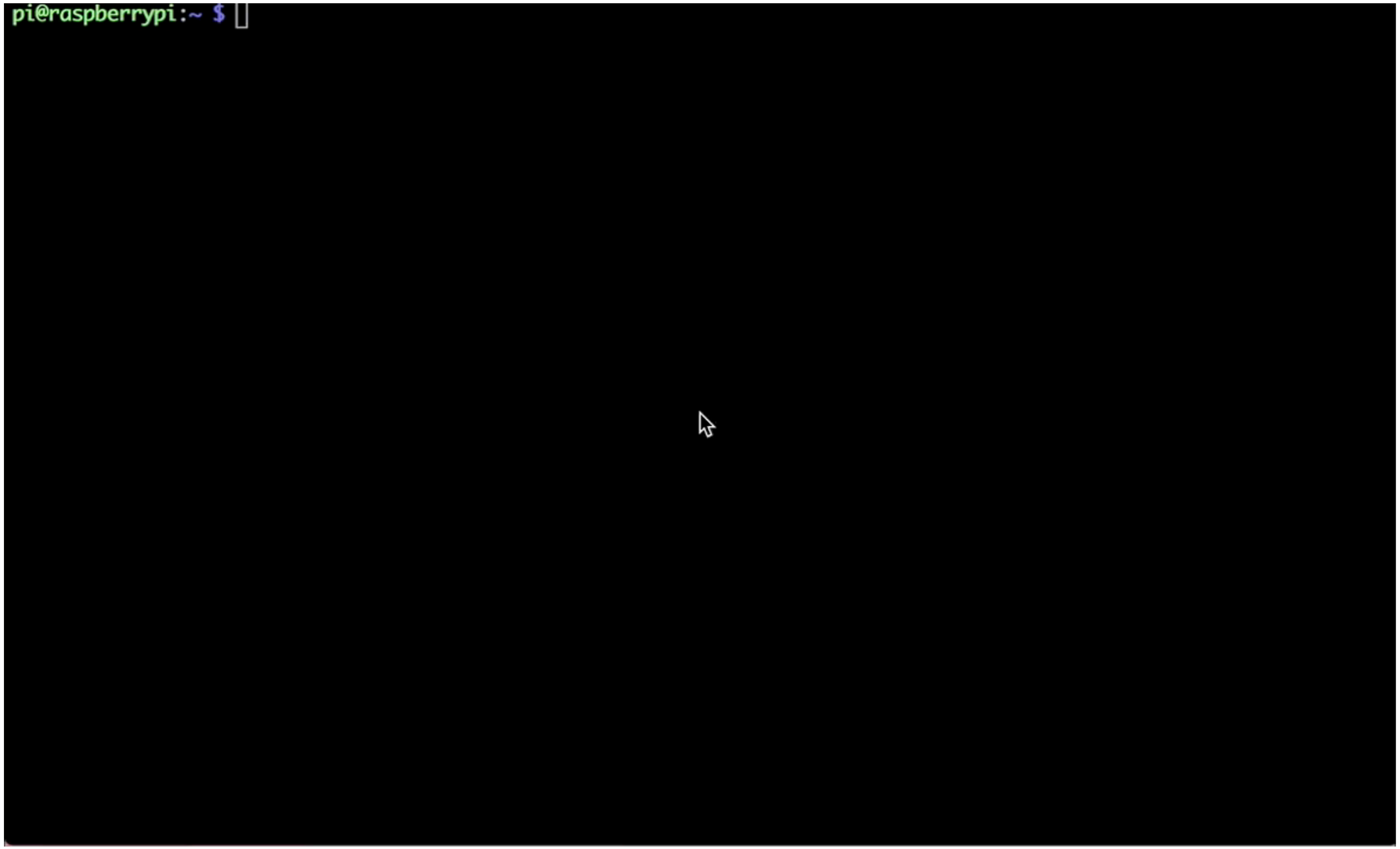
We will be using PHP so err.. Apache!

### PHP

We could use ruby or python but who am I kidding, I can't even code PHP correctly so what chance do I have using something else.

### Composer

Who uses PHP without composer (apart from me 12 months ago?)




# Update the Pi

- **sudo apt-get update && sudo apt-get upgrade**
- Why?: This makes sure that everything is up to date and fresh, when setting up something new there is no reason why not to do this.

🕑 Real time taken: 19 minutes

# Install Apache

- **sudo apt-get install apache2 apache2-utils**
- Why?: PHP has a built in server, but Apache is more flexible and it’s SO easy to install.

🕑 Real time taken: 2 minutes


# Install PHP

- **sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache php5-mysql php5-curl php5-gd**

**Restart Apache**

- **sudo service apache2 restart**
  **Why?: Because we want to write some PHP**

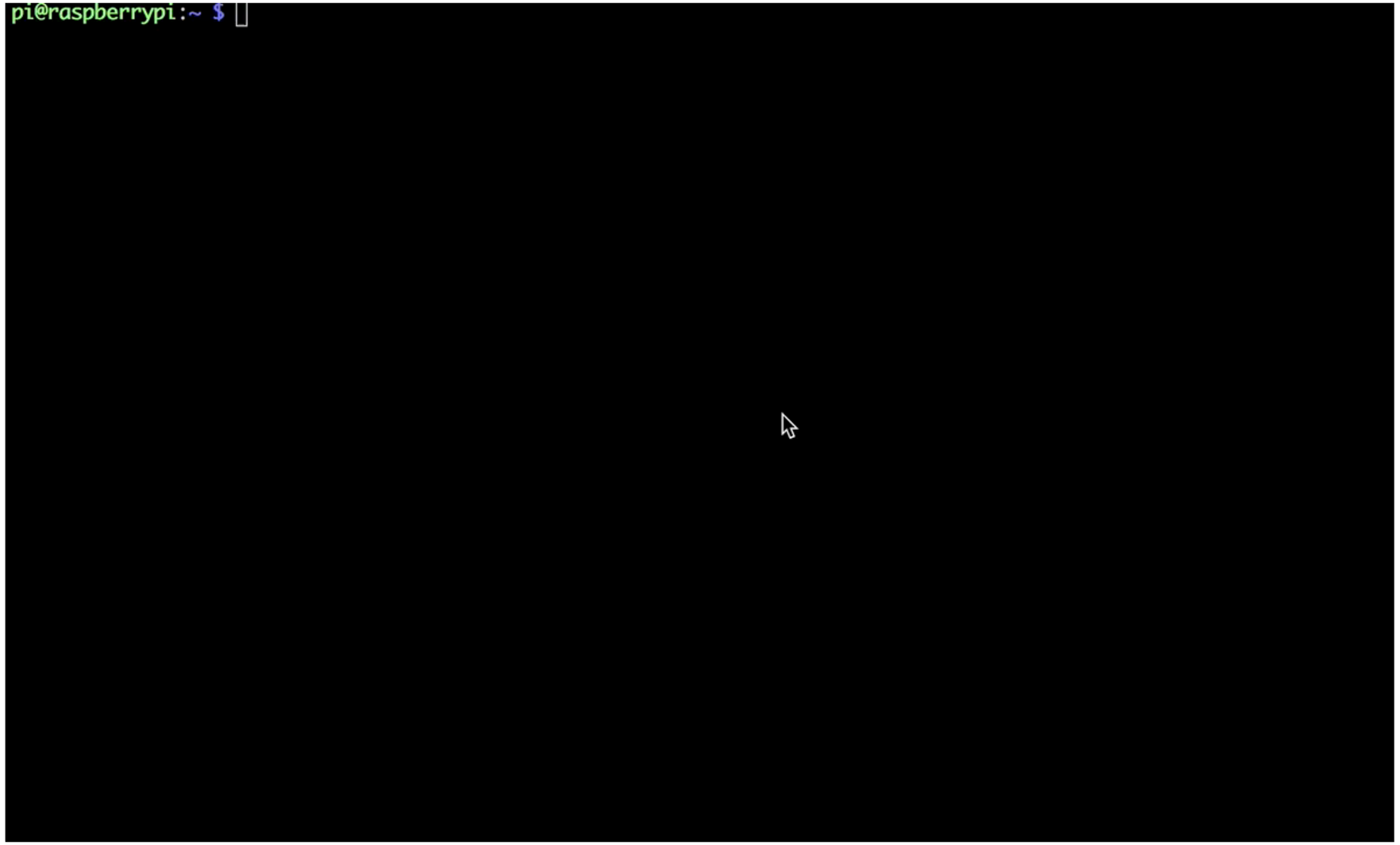
🕑 Real time taken: 4 minutes

“ The "S" in IoT stands for Security”

— Someone on the internet

# Permissions

- Don’t do this, setup a proper virtual host. Doing this is outside of my talk and so chmod is just a quick hack
- sudo chmod -R 777 /var/www/html





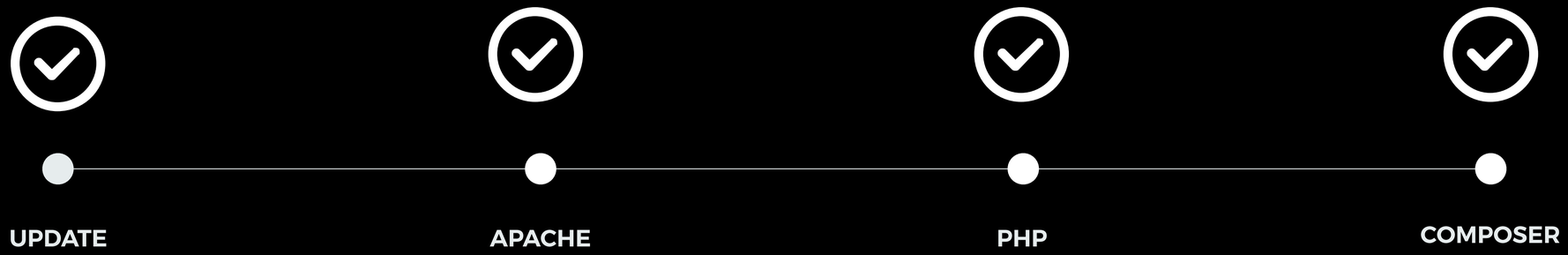
# Install Composer

- **wget https://getcomposer.org/installer**
- **php installer**
- **sudo mv composer.phar /usr/local/bin/composer**
- **rm installer**

- Why?: Composer is an amazing way to manage 3rd party libraries

🕑 Real time taken: 4 minutes

# Now we have everything setup and ready to go!




We now have our Pi setup with everything updated and installed ready for us to start development on. Apart from a firewall there isn't much difference between this and a normal a production scale web server ( as long as you setup a virtual host! )

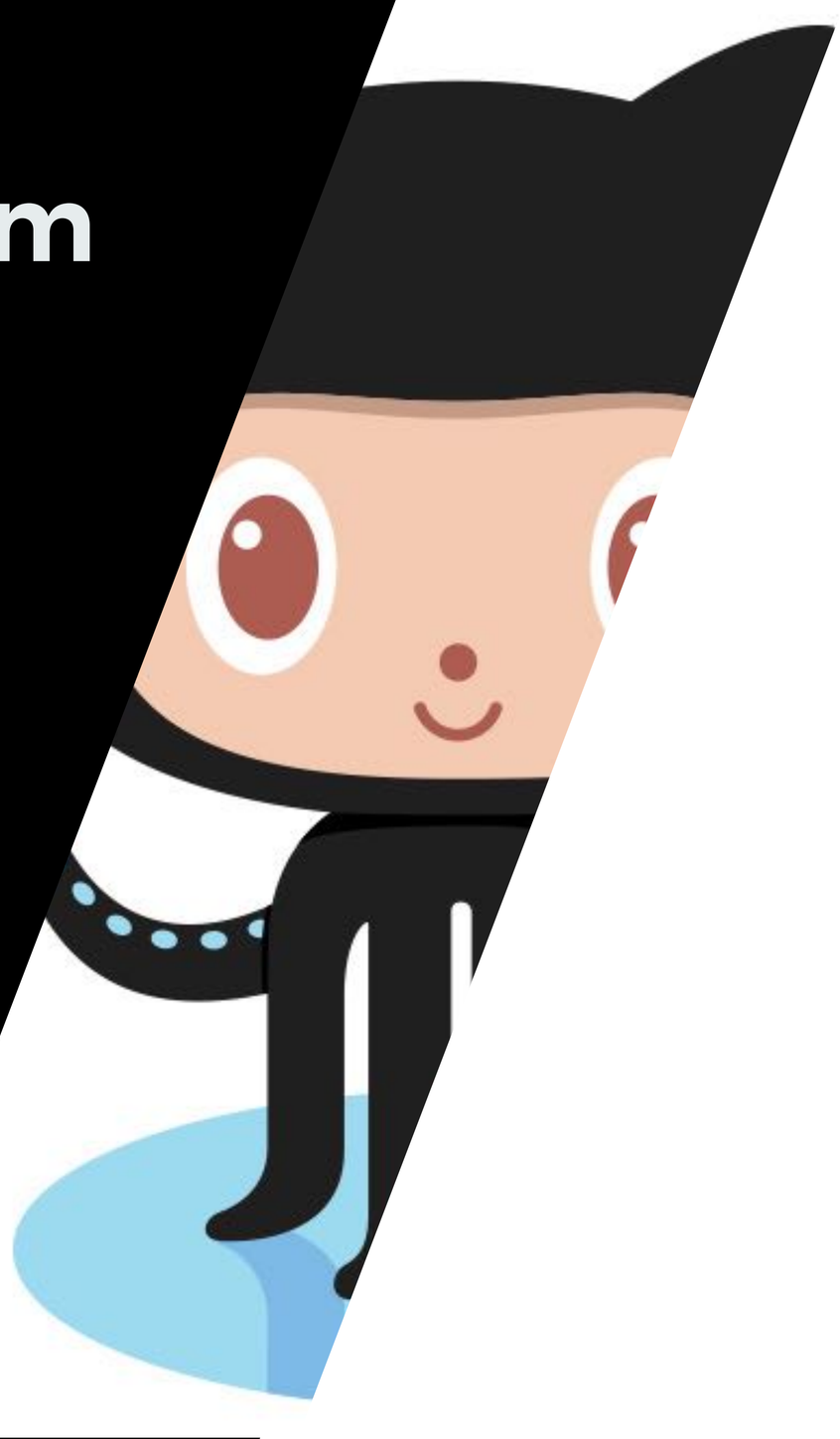
We are going to be doing some things that you would not normally do or even consider on a production web server. But because this is a £10 computer running on a local network we can get away with doing some crazy things with PHP.

# Cloning my code from Github

We want to get the code I have pre-written off of Github and onto the Raspberry Pi

- cd /var/www/html
- git clone https://github.com/thickey256/ThickeyHue.git

This will now put everything we need into a newly created directory which is located at /var/www/html/ThickeyHue




# Composer Install

This will download both the required packages that we need to use and also sort out the autoloading of classes etc.

- composer install

The two packages that we are using are as follows:

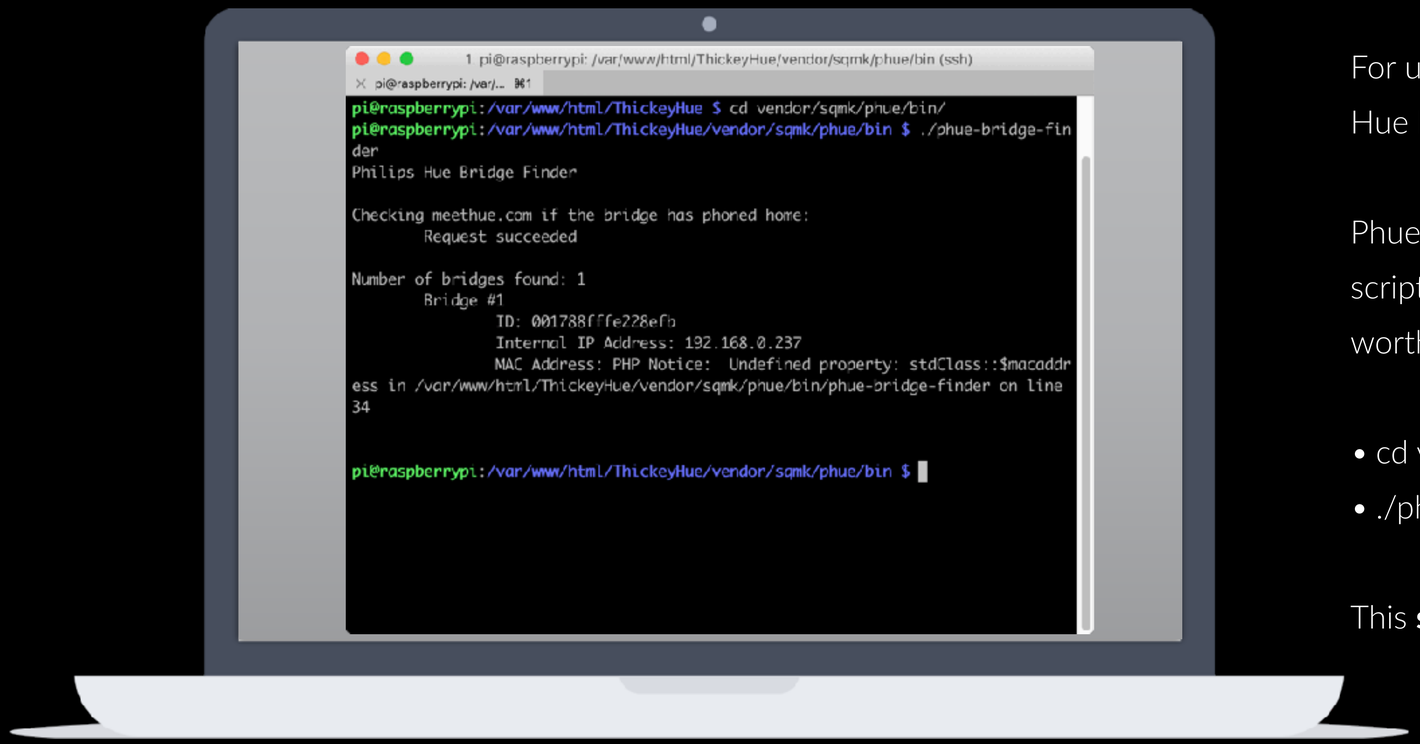
## Phue

Does all the hard work and offers a friendly interface for us to talk to our Hue bridge and build
More info here >

## Color

It may be spelt incorrectly but this is a very simple script to convert hex colour codes to RGB values. I was too lazy to code this myself.
More info here >

# Get Hue bridge IP address




For us to talk to our lightbulbs we need to firstly get the IP address of our Hue Bridge.

Phue a few handy scripts that can help us with this, it's actually just a PHP script that uses Phue functions, as we just need to do this once it's not worth coding our own.

- cd vendor/sqmk/phue/bin/
- ./phue-bridge-finder

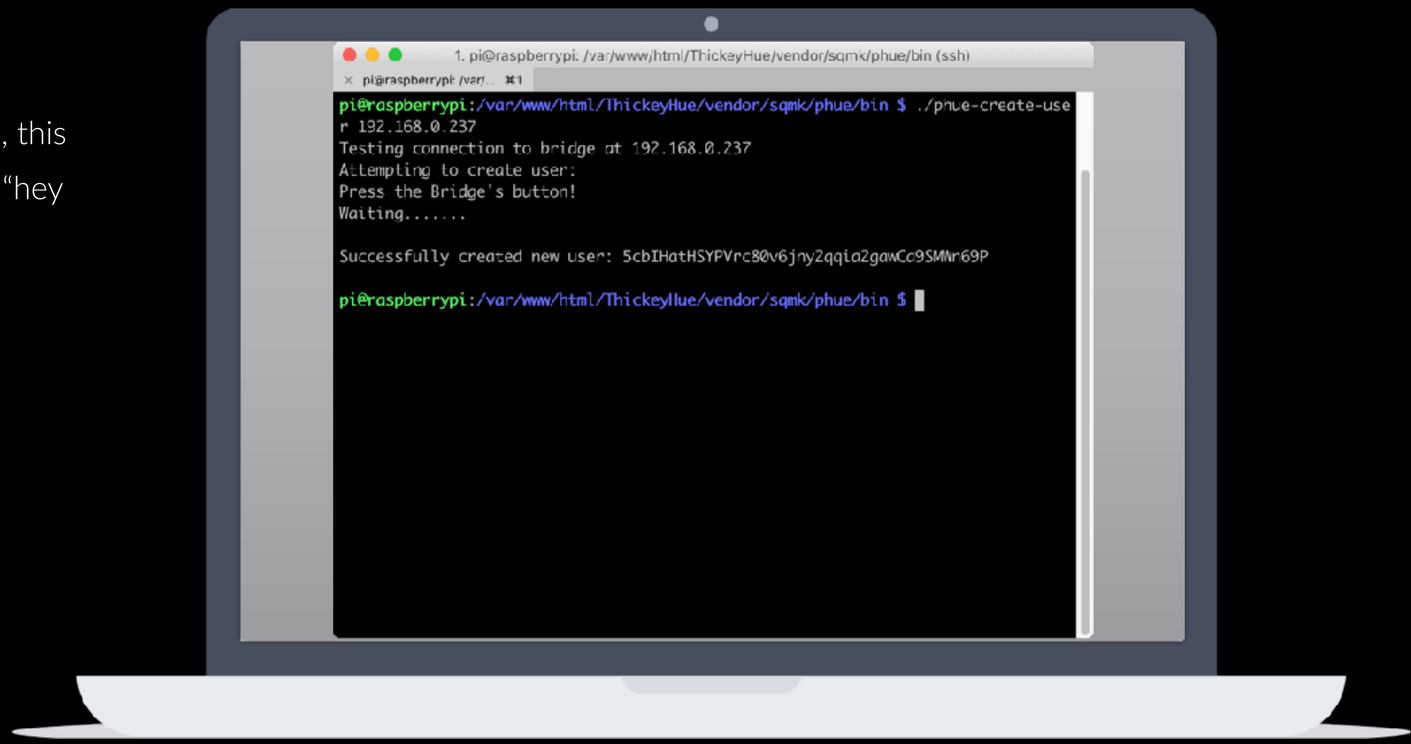
This **should** give us the IP address that we need!

# Generate a Hue bridge user token

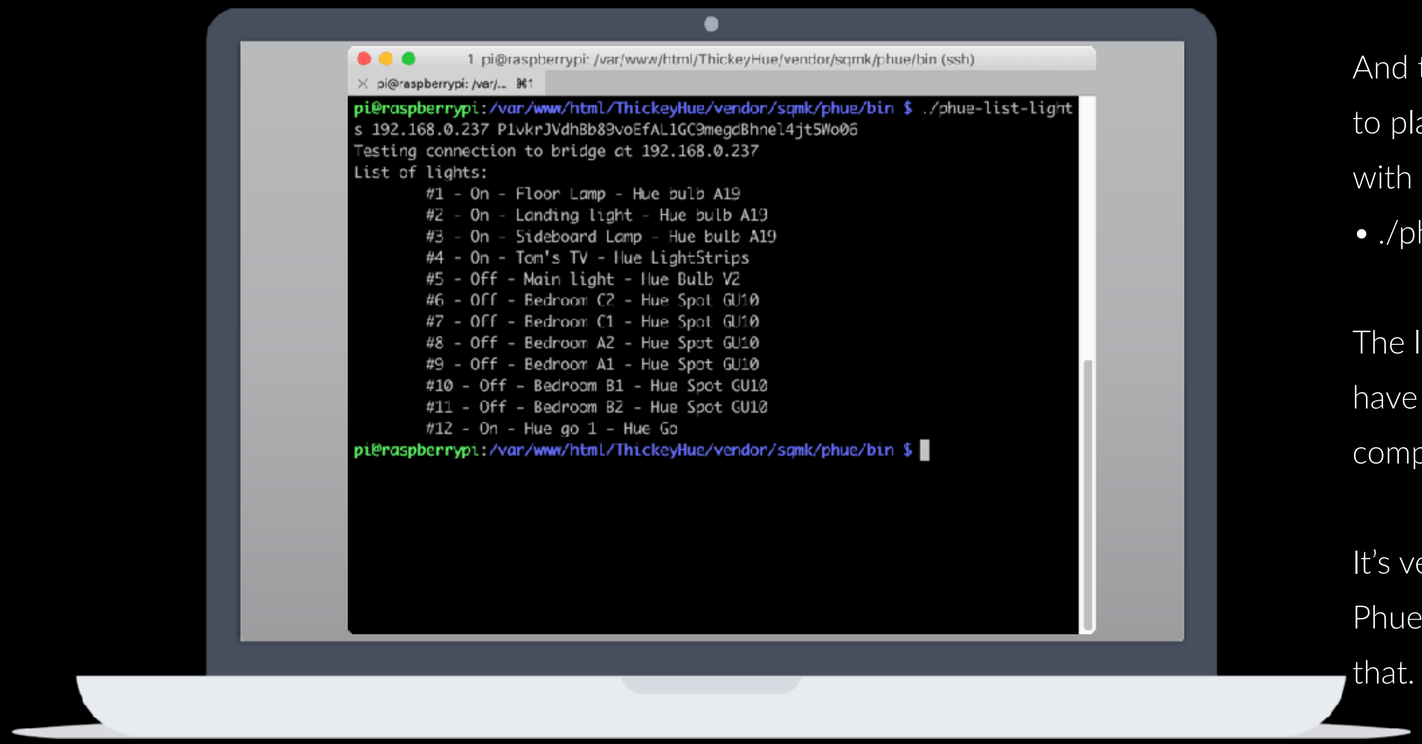
Finally some (rudimentary) security!

We have the IP address for our Bridge but we also need a user token, this involves making a request to the bridge and pressing a button to say “hey this request is ok, give them a user token”.

./phue-create-user YOUR-BRIDGE-IP-HERE



# Find out which light to control




And the final step in this nightmare is to work out which light it is we want to play with.  I would strongly recommend you setup the bridge and lights with a phone app first but you *could* do this with Phue if you were crazy.

- ./phue-list-lights YOUR-BRIDGE-IP-HERE YOUR-USER-TOKEN-HERE

The light we want to play with in this example is number 12. If you just have one bulb **you should not assume** it will be number 1.. because computers!

It's very easy and possible to control more than one bulb at once with Phue, but it's simply looping through numbers so not much merit in demo that. and I only have one bulb here with me today.

# Update our config

So we want to replace our newly found information into our config file, it's located at /var/www/html/ThickeyHue/lib/Service/PhueConfig.php

Simply replace the hueBridgeIp, hueBridgeUser and lightId values with the values we identified/generated.

If the values are the same as the values already in the config file then welcome to my house and wifi network!

# Demo Time

# Disco Time

# What did we learn?

- How To setup Apache and PHP on a Raspberry Pi
- How not to give global permissions on /var/www/html
- How to setup a VERY basic API to control Hue bulbs
- How to control a light with a banana 🍌
- The Raspberry Pi is a great IOT control device

# Questions?

You can find all my code and these slides on GitHub at

https://github.com/thickey256/ThickeyHue

Twiter: @thickey

Email: thickey@thickey.com