Running a Web Server from Raspberry Pi

Jon Flanders
@jonflanders





This module is about...





You might want to get information	held on the Pi off in an easy way
---	-----------------------------------

- You might want to get information held on the Pi off in an easy way
 - □ Might want to share pictures, videos, or other data (i.e. hardware sensors etc.)

- You might want to get information held on the Pi off in an easy way
 - □ Might want to share pictures, videos, or other data (i.e. hardware sensors etc.)
- Exposing an HTTP endpoint is the easiest way to enable ubiquitous access to data from a Pi

- You might want to get information held on the Pi off in an easy way
 - Might want to share pictures, videos, or other data (i.e. hardware sensors etc.)
- Exposing an HTTP endpoint is the easiest way to enable ubiquitous access to data from a Pi
 - Or any device for that matter

Serving up static files

- nginx is currently the favorite lightweight web server
 - sudo apt-get install nginx
- Cool feature of nginx is that can serve as a remote proxy to other HTTP servers
 - Often used as a front-end to dynamic content servers running on same server on different Port

Serving up static files

- nginx is currently the favorite lightweight web server
 - sudo apt-get install nginx
- Cool feature of nginx is that can serve as a remote proxy to other HTTP servers
 - Often used as a front-end to dynamic content servers running on same server on different Port

```
upstream node{
    server 127.0.0.1:8080;
server{
   listen 80;
    ssl off;
    ssl_certificate /etc/nginx/public.crt;
    ssl_certificate_key /etc/nginx/private.rsa;
    root /usr/share/nginx/aym;
    try_files $uri @node;
        location @node{
         proxy_set_header X-Real-IP
                                            $remote addr;
          proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
          proxy_set_header Host
                                             $http_host;
          proxy_redirect
                           off;
        proxy_pass
                           http://node;
```



Dynamic content

- For dynamic content or web services there are many popular choices that can run on the Pi
 - PHP
 - Java (you need a special build of Raspbian that handles floating point differently)
 - Ruby on Rails
 - Python
 - Node.js seems to be the new hotness and JavaScript skills are fairly ubiquitous

Running Python through HTTP

- python -m CGIHTTPServer 8000
 - □ This is python 2.7 for 3.3 you need to write a bit of code using the http.server module
- Will use static source files in the current directory
 - □ put python code into an executable file named *.cgi into the cgi-bin subfolder

Running Node.js

Get dependencies

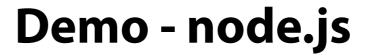
sudo apt-get install git-core build-essential

Get the node source

- git clone https://github.com/joyent/node.git
- cd node
- □ git checkout v0.10.4
- ./configure
- make (this one can take along time run right before you go to sleep)
- sudo make install (this will create and install the node package manager npm)

Using node's child_process module may become important

No node.js libraries to access the Raspberry Pi's functionality



Summary

 Running a web server on Raspberry Pi can be useful to both cause code to execute remotely or to serve up data the Pi is holding onto