Design Documentation

For our Assignment 9, we implemented 3 features, a l**ocation handler**, which logs the rough locations of where requests are coming from, **made the server reconfigurable** while it is running**,** and **markdown handling**, which allows us to serve .md files. The documentation of the features is shown below.

Location Handler Feature

Access the feature

To access this feature you must go to
http://ec2-52-37-251-244.us-west-2.compute.amazonaws.com/location

Example of the feature

This feature shows the locations of where requests to our server originated from. An example is shown below:

```
Florida: *
Moscow: *
New York: ***
California: ********************************************
```

The asterisks show the amount of requests from location. We used proxies in Florida and New York to test the functionality. What is interesting is that the request from Moscow did not originate from one of our teammates (which was kind of cool and scary at the same time).

How to test the feature

If you would like to test the functionality, you can use the following proxy websites (New York and Florida), with the link http://ec2-52-37-251-244.us-west-2.compute.amazonaws.com/echo

1. https://www.missionimproxyble.com/
2. http://www.covertbrowsing.com/

Then, after making those requests, if you hit our location handler at http://ec2-52-37-251-244.us-west-2.compute.amazonaws.com/location, you will see the requests logged from those locations.

Methodology

We hit the endpoint ipinfo.io to get the location based on IP address in a JSON format and parsed the location and logged it. The handler stores the location and its frequency in a map, making it easy to output to the requester.

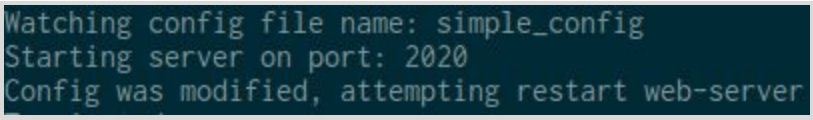
Reconfigurable Web Server Feature

Access the feature

In order to access the feature, one must have the ability to change the simple_config file that the webserver is running on. In our case, it should be demo'd on a local version of our web-server running. NOTE: As we had implemented three features, we did not completely finish this one by making it accessible via a handler. Our next step, if we had more time, would be to create a handler that could allow the user to switch config files while it is running on AWS.

Example of the feature

This feature reloads the server if the config file has been modified and is still in a valid format. An example terminal output is shown below:



It is hard to show through this documentation the demonstration of the feature. However, it can be seen that during one execution of the program, we were presumably able to detect file change events and act on them. For a real example, one must try the live demo, which we explain below.

How to test the feature

In order to test the feature, one must have two terminal windows open, and must be using a local version of our web server. Here are the steps to show that the feature works:

1. Run the command **make run.** This runs our web-server with config file demo_config.

2. Then query our server on the browser, using the command localhost:8009/static/index.html
3. Use the second terminal window to run **emacs demo_config**, and change root of the static file handling to "./pub" (which does not exist), and save.
4. Now, when you query localhost:8009/static/index.html, 404 Not Found is returned to the user

Methodology

We used a supervisor script that forks a child process to run the webserver in after doing some preliminary checks on the simple_config. Then the parent process listens for file modification events on the config file. If the file is ever modified, we check to see that it is a valid config. If the config is valid, then we kill the child process and start the server in a new process and load in the new config.

Markdown Feature

Access the feature

To access this feature you must go to
http://ec2-52-37-251-244.us-west-2.compute.amazonaws.com/static/README.md

Example of the feature/How to test the feature

The feature allows the serving of markdown rendering If you go to the link above, you can see the markdown rendering working. To see the raw version of the file, go to

http://ec2-52-37-251-244.us-west-2.compute.amazonaws.com/static/README.txt

Methodology

We used a mixture of our existing static file handler and the library https://github.com/sevenjay/cpp-markdown that is converts the markdown file to html. We first recognize a request as a request for a .md file, and then handle it using the the library, and serve the html back using the static file handler.