

Release Notes

Cameron Watts

December 2022

Features & Functionality

Web Server

- Logs can be filtered (through timestamp search) and saved. As of right now, only the geophone log files are downloadable. The snapshot and acoustic files need to be handled differently (see Future Expansions section)
- Audio is converted from .pcm to .wav when uploaded to the web server, so that it is compatible with the HTML5 audio player.

Raspberry Pi Data Retrieval

- The Raspberry Pi uses a Python script called "data_retrieval.py" in the ERP directory. The script has configurable variables near the top of the file.

Setup

To get the web server up and running, you must first install Docker by following the installation steps [here](#).

- Once you've installed docker, you'll need to build the docker containers. To do that, simply `cd` into the directory containing the `docker-compose.yml` file and run the command `docker compose build`. Note that, if you're using an older version of docker, you may need to include a hyphen in the command, so that it looks like `docker-compose build`. Once you build the containers, you can simply run `docker-compose up` to spin up the web containers. From there, the IP address specified in the Nginx default.conf file should be accessible through the web browser.

If you run into any problems getting the web server to deploy properly, contact me at wattscameron26@gmail.com or [Trialon#2655](#) on Discord.

Known Issues

- Audio files seem to undergo a raise in pitch when being converted from .pcm to .wav. However, I haven't truly verified if this raise in pitch happens during conversion because I haven't found a way to play the raw .pcm audio files.
- The data retrieval program relies on predefined IP addresses so that it can connect to each sensor using SFTP. If the Raspberry Pi unit which houses the data retrieval program is ever deployed, it is important that none of the sensor devices receive a new NAT IP address due to the network configuration.
- If the audio sensor is running and capturing audio data, then the audio file which it updates will only be included in one log throughout the duration of the data retrieval script's execution. This happens because the data retrieval script in the raspberry pi looks for log files which have a name consisting of (or including) a timestamp that is "greater" than the latest obtained timestamp. This can be solved by factoring in the modification date of audio files when determining whether or not to include them in the next HTTP request; however, the developers should be aware that doing this will result in an increasingly large audio file being sent to the server every time. Currently, the Nginx web server which handles HTTP requests has a maximum client request size of 25 MB, which means no more than 25 MB can be sent at a time, through an HTTP POST request. However, this setting can be changed in the nginx default.conf file.

Future Expansions

- The web server is currently public-facing, and unsecured. Since the server already runs using Nginx, obtaining a certificate through some sort of process shouldn't be too difficult of a task.
- Heat maps would make a great addition to the web server in the next version. Also, some software which can display the seismograms generated by the Raspberry Pi Shake device would make an excellent addition.
- As of now, pages can be navigated by clicking the arrow buttons at the bottom of all log groups. In the future, there can be numbers added which would allow users to jump to a specific page of log groups.
- When data is sent the web server, it is first stored on the computer (GCP Virtual Machine Instance) which hosts the server. Then, it is mounted to the Nginx Docker image, so that Nginx can serve the files to the user; however, this will eventually use up all of the space on the host machine. Some solutions:

1. Implement a log storage cutoff on the back-end. The cutoff can be based on the amount of storage left on the host machine, or the number of logs currently stored. There are caveats to both approaches.
 - The web app runs in a docker container, so instead of attempting to obtain the amount of storage left on the host machine, one should instead set a static size limit and set the log storage cutoff based on that. Docker containers can scale in size, but it's unclear what the maximum size of a single Docker image can be. By default, the host machine has 40 GB available on the boot disk.
 - Using a predefined number of logs to determine the cutoff sounds good on paper, but depending on how big each log group is, it could cause storage issues.
2. Use an external storage service to house media data, such as Amazon S3.