CatchScan Backend Development Task

This test consists of a small project that should take no longer than 6 hours. The technical interview will be based on the implemented solution. Questions will be asked about implementation designs and considerations. An implementation need not meet all requirements stated in the specification. Focus should be on design, implementation, and code quality. Other areas may also come up in the interview, such as development velocity.

This test is not a pass/fail, but serves as a springboard for the technical discussion.

Problem Domain

Common crawl is an initiative that crawls the internet and makes the resulting data available to the public. CatchScan would like to introduce image links from this data to our own image database. The problem consists of accessing and parsing the common crawl data and extracting relevant links from the bulk data.

The data is available via a public index, which exposes a path file. This file holds a list of partial URLs which point to actual data chunks. The partial URLs need to prefixed with `https://data.commoncrawl.org` or `s3://commoncrawl/` in order to be valid.

The path file will is explained here and a snippet can be seen below.

```
crawl-data/CC-MAIN-2022-40/segments/1664030331677.90/wat/CC-MAIN-20220924151538-20220924181538-00000.warc.wat.gz
crawl-data/CC-MAIN-2022-40/segments/1664030331677.90/wat/CC-MAIN-20220924151538-20220924181538-00001.warc.wat.gz
crawl-data/CC-MAIN-2022-40/segments/1664030331677.90/wat/CC-MAIN-20220924151538-20220924181538-00002.warc.wat.gz
crawl-data/CC-MAIN-2022-40/segments/1664030331677.90/wat/CC-MAIN-20220924151538-20220924181538-00003.warc.wat.gz
crawl-data/CC-MAIN-2022-40/segments/1664030331677.90/wat/CC-MAIN-20220924151538-20220924181538-00004.warc.wat.gz
```

Each data chunk can be retrieved in WAT, WET, and WARC formats. Extracts of these data formats, their use cases, and more information can be found here.

Specification

A small CLI tool should be implemented to retrieve the latest data batch from common crawls public index and extract links from the data. The public index for november/december 2022 can be found here.

Requirements

1. Implement a simple CLI for the tool
   - Include options for different stages of the process
2. Extract path file and locate data files
3. Acquire data files and parse them into a relevant format for the tool
4. Search the data for image links (see the examples for more information)
5. Produce a CSV file with each link
6. Include the origin link (the website the image was linked to from)
7. Test if the link is still alive, exclude dead links
   - This should be done efficiently/concurrently

8. Automatically find latest path file and download it
9. Setup a docker container for the tool
    - Either this requires point 7. or the container should accept arguments via environment variables
10. Implement unit tests for the tool
    - Does not need to achieve full coverage
11. Implement integration tests for the tool
    - Does not need to achieve full coverage

The tool can be implemented in your language of choice, but you should be able to discuss reasons why you chose the language. Keep in mind that other tools maintained by CatchScan are written in Python 3. The tool must run on Linux, since any deployment target would be running Linux. The implementation should be stored in a public git repository, which we can access