# Cloudability Technical Exercise

The technical exercise has two components, one to exercise programming skills and one to exercise design skills. In both cases, you can use whatever tools you think would best get the job done and/or are most comfortable with.

We'll evaluate your solutions along the following dimensions.

1. **Thought process.** That is, how you approached the problem and organized your solution.
2. **Tool competency.** You get to choose your own tools, so you're claiming that these are the tools you're most proficient with.
3. **Depth of knowledge.** That is, how well do you understand Computer Science and software engineering.

## Problem #1: Summarize a Large CSV File (Programming)

Attached are three CSV files, issued by Amazon Web Services to Cloudability. Each file describes, on an hour-by-hour basis, what resources you used in a month. The data is broken down by account, by service, by availability zone, by resource, by tags. Unzipped, each file is about 100MB in size.

A few notes on the format of the files.

1. The first line of the file are headers.
2. The headers beginning with "user:" are tags assigned to the resource, and any value in those columns is considered the "tag value."
    1. Amazon tags are key-pair tags, i.e. `"Environment = Production"` would be `"user:Environment"` in the header, and `"Production"` in the relevant row within the file.
3. It's possible for these files to be many GB in size.

### The Problem

Using any tool, technology, and technique that you find relevant and are comfortable with, develop an app (or apps) that does the following:

- Count the unique types of EC2 instances within the file.
- Generate a report of total cost by day, by tag. (i.e. on March 3rd, Environment = Production cost $100.0)
- **Bonus points:** Find instances that changed tags during the month, and the timestamp on which they changed.

### The Solution

Create a Github repo with your solution in it. If you can't use Github due to sensitivity constraints, it's OK to create a tar ball of your source and share it with us either via email or Dropbox or…

With each solution, please include a README file that has the following info.

- How to run the application against a file.
- Any notes about approach that you think are relevant and tradeoffs you made in your solution.
- For the algorithms you implement, the time and space complexity in Big-O notation
- Which constraint (time v. space) you optimized for and why.
- **Bonus points:** For the algorithms you implement, the best- and worst-case runtime complexity and the scenarios under which they occur.

## Problem #2: Create a Visitor Analytics Tool (Design)

Assume that you've been given access to the logs for one of the Top 5 websites in the world (e.g. Google), and that user IDs have been embedded in the logs by the web servers. Thus, the log format is as follows:

```
[2013-05-04 01:03:31] WEB WORKER GET /hello-world (75.23.54.234) (User ID: 6)
```

### The Problem

Design a system that does the following things:

- Updates a data store (of your choosing) with the number of unique visitors per hour.
- Updates a data store (of your choosing) with unique user IDs who returned more than once in a day.
- Delivers a daily report on the above.

The system that you design should be able to efficiently ingest a day's worth of log entries (e.g. billions of records daily)

### The Solution

What we're looking for here is only a description, but it must be detailed. Include the following:

1. Detailed description of the data flow.
2. Technologies used, and rationale for using them.
3. Any operational concerns or tradeoffs made.
4. **Bonus points:** A diagram of the system you've designed.

This can be sent in an email, a text file, or a Google doc.