**Data Engineer Technical Round Interview:2 hours**

* **Introduction (10 minutes)**
* **Team dynamics /also cover Agile mindset (15 minutes)**
* Working in remote environment
* Working with team members
* Pair programming/collaboration
* **Previous experience/also cover Agile mindset (20-25 minutes)**
  + Pick your own project and talk about designs, accomplishments and learnings etc...
* **Break (5 minutes)**
* **Coding exercise (20-25 minutes)**

Mike is going to prepare the coding challenge on ECW typeahead problem and below are some must haves and nice to haves that he will include in the test.

* + - Architectural diagram (senior and above)
    - Languages – any of below language. Python and PySpark are preferred.
* Python
* PySpark
* Java
* Scala
  + - Requirements

1. Must include readme
2. Must include unit tests (senior and above)
3. test submission modes- any one of below method

* GitHub or any source control system is preferred.
* Zipped file.

1. Usage of cloud resources (senior and above).

* AWS (preferred)
* Azure
* GCP

1. Bonus: use of any below orchestration service to schedule the job.

* Airflow is preferred (senior and above)
* Step function
* **Refactor the codebase (10-15 minutes)**

Mike already has idea on this for ecw problem that we are going to use as coding challenge.

* How do you change the code base?
* **CI/CD infra (5 minutes)**
  + Cover any infra as code languages like terraform or CloudFormation stack
  + Cover any of CI/CD software’s like TFS, GitHub, Jenkins etc.
* **Questions (10 mins)**
* Give time for candidate to ask any questions
* Any other questions from IMO team
* Wrap up

# Coding Exercise

IMO’s Search Product Team would like to know how often people are making different requests to their search engine. They would like to know how many different searches are being requested. The catch to this ask is that autosuggest searching may be enabled for some users; this means that, for some users, a search is executed every time they make a keystroke. IMO’s product team does not want to count these “suggestions” separately and would like them to only represent 1 search.

Data Set details

* Every request has a user ID and a timestamp associated to it
* There are approximately 1,000,000 requests per day
* Not all users have autosuggest enabled. Sadly, the product team forgot to log when this feature is turned on
* Data set lives in a flat file store and in a database (you may act as if this is an RDBMS or a NoSQL database)

The ask

* Create a process that identifies the total number of “searches”
  + Each set of autosuggest searches should be counted as one search
  + Each individual non-autosuggest search should be counted as one search
* Autosuggest searches should be identified based on similarity to the previous search
* The product team would like to see
  + Total overall count
  + Percentage of searches that use autosuggest, so that they can determine if the functionality is something to be supported.
* Include unit test coverage in your language and framework of choice
* The product team would like this process run every day for the next 60 days