# Question 1

1. **How many lines does the RDD contain?**
   **Answer:-** 9669739
   **Method:-** We create an RDD that is a collection of the lines in the file. Then we count the number of items in it.

2. **Count the number of "WARN" messages.**
   **Answer:-** 132158
   **Method:-** We create an RDD that is a collection of *case classes* that represent a line in the log file. Then filter those messages whose *debug_level* is WARN.

3. **How many repositories were processed in total when the retrieval_stage is "api_client" ?**
   **Answer:-** 71981
   **Method:-** We create an RDD that is a collection of *case classes* that represent a line in the log file. Then filter those messages whose *retrieval_stage* is api_client.rb. Then we process the repository name for each entry from the *rest* field. After that, we use MapReduce with key as the repository name to find the number of unique repositories processed.

4. **Using retrieval_stage as "api_client", find which clients did the most HTTP requests and FAILED HTTP requests from the download_id field.**
   **Answer:-** ghtorrent-13*(85528)***,** ghtorrent-13*(79623)*
   **Method:-** We create an RDD that is a collection of *case classes* that represent a line in the log file. Then filter those messages whose *retrieval_stage* is api_client.rb and check whether the request is failed or not from the *rest* field. Then we count the number of items in the resultant RDD.

5. **Find the most active hour of the day and most active repository.**
   **Answer:-** 10*(2662487)***,** greatfakeman/Tabchi*(79524)*
   **Method:-** We create an RDD that is a collection of *case classes* that represent a line in the log file. After that, we use MapReduce, with the key as the hour (from *timestamp*) or repository name (from *rest*) and count the number of entries for each hour/repo. After that we find the key with the maximum value.

6. **Which access key is failing most often?**
   **Answer:-** ac6168f8776 *(79623)*
   **Method:-** We create an RDD that is a collection of *case classes* that represent a line in the log file. After that, we use MapReduce, with the key as the access key (from *rest*) and count the number of entries for each key. After that we find the key with the maximum value.