ANALYSIS AND REPORTING

Document your hardware configuration (CPU speed, number of cores, RAM, Disk etc.) and software stack (RDB and version, Redis and version, programming language, libraries used, etc.)

The hardware configurations are 2.3 GHz 8-Core Intel Core i9, 16 GB of memory using MySQL verison 8.0.25 (MySQL Community Server - GPL)

Report TWO numbers in units of API calls per second:

API Method	**API Calls/Second**
postTweet	3262.4
getHomeTimeline	6.1

Describe any factors you think might have impacted your results.

1. SQL Statements: Retrieving the timelines of the user with SQL JOIN statements is pretty expensive and becomes slower as the size of the CSV files increases. The current implementation can only handle retrieving four (4) timelines per second. As the database might become more complex with larger tables and relationships, this will not be able to keep up with Twitter users in real time loading three hundred thousand tweets per second.

2. Generating unique tweet ID: Our current implementation of generating the tweet_id is to have a counter that increments the tweet_ids by one every time a Tweet object is added to the Tweets table. This process runs in linear time and slows down how long it takes to post the tweets because the tweet_id generation is very dependent on the values of the other tweet_ids.