Product/Systems Engineer: URL Shortener

Please note:

- You can use the language you are most comfortable with to write the code for assignment.
- Ideally assignment should not take more than ~2 hours at the max.
- Some areas of assignment might need you to understand Docker and if you have not worked on it before - don't worry. You can pick it up fairly easy with documentation or in the worst case scenario - skip the last parts and send us working code.

- **Apart from a working application, we care a lot about**:
  - Readability of code
  - Tests - Unit tests definitely and more if you can think of
  - A good structure to your code and well written file & variable names etc.
- Points marked with [BONUS] are not required - but good to have, so don't focus on them in the first iteration. Once you have things working and you think you have time, then try to get BONUS items working
- If you need more time or are stuck at some point - don't hesitate to reach out to us.

***One important thing**: Don't commit all of your code as a single commit - do commit logical units of work so we can see the work as you built it up. **Intermediate/working commits won't affect the way we judge the end result**. Please add anju-infracloud as a collaborator to the project and push to a PRIVATE github repository.*

Assignment

1. Build a simple URL shortener service that will accept a URL as an argument over a REST API and return a shortened URL as a result.
   a. If you have not used or seen a URL shortener before - please visit https://bitly.com/ and try to SHORTEN a URL. The goal in this assignment is not to build a fancy UI but an API only version of that.
   b. Don't use a shortening API - you are supposed to write that part of code.
2. The URL and shortened URL should be stored in memory by application.
   a. [BONUS] Instead of in memory, store these things in a caching solution such as Redis, Memcached etc (Or file if you really want).
3. If I again ask for the same URL, it should give me the same URL as it gave before instead of generating a new one.
4. [BONUS] Put this application in a Docker image by writing a Dockerfile and provide the docker image link along with the source code link.