**Overview:**

You are to develop a microservice-based application with a minimum of 4 microservices and use either   
Docker-Compose or Kubernetes to orchestrate the application. You must also use Docker Hub to put   
your microservices into repositories and then use those repositories in your compose or kubernetes   
YML file(s). You can use either grpc (ideally with streaming) or REST.   
   
The file PARIS\_AGREEMENT.txt contains the text from the 2015 Paris Agreement on Climate   
Change. One of your microservices must read from that file and send each word to another   
microservice for analysis; that analysis service must determine the average word and sentence length   
(excluding digits, punctuation, etc.) and the frequency of each of the following words (case   
insensitive):   
   
Food   
Technology   
Deforestation   
Socioeconomic   
Ecological   
Diversification (or diversify, diversifying, etc.)   
Sustainability (or sustain, sustainable, etc.)   
2 other words or your choice   
   
You will need a microservice to store the statistics using keys, e.g. with Redis. You will need another   
microservice to display the results on a web-page, e.g. using Python Flask. The stats to be displayed   
are: average word length; average sentence length, number of occurrences of each of the list words   
above.   
   
Set up 1 test (e.g. gatling or postman) and 1 monitor (e.g. python metrics or prometheus).

A document containing:   
1. instructions on how to run the app with screenshots showing it running   
2. a diagram showing the architecture of your application   
3. 200 words on why you chose the various integration and storage mechanisms   
4. 200 words on how you would test such a complex architecture (compared with a more   
traditional monolith approach)   
5. and ditto for monitoring – 200 words on how you would monitor something complex like this   
(particularly if it was scaled out more)   
6. Screenshots of testing and monitoring   
   
**Rubric:**

40% - functionality – the application works to the specification; good use of integration   
25% - use of Docker and Docker Compose or Kubernetes (docker files, use of Docker Hub, creation of   
Compose YML file or Kubernetes manifest files)   
15% - architecture diagram and explanation of integration and storage choices   
20% - Testing and monitoring implementations and discussion (half to each)

**To be submitted:**

1. A zip file containing all your code.
2. Dockerfiles.
3. Compose file / Kubernetes manifests.
4. Any other relevant files.