Discussion about the CI/CD pipeline implemented.

The free account of DockerHub doesn't allow tracking a GitHub repository in order to automatically build new images when changes are made to the code. So there is a manual process of, when making a change in the model or in the Flask application, building the image by hand and pushing it to the repository in DockerHub.

Considering this scenario, tests were conducted mainly changing the deployment YAML file, more precisely: changing the docker image version to be used in the containers. So, after a change is pushed to the repository which ArgoCD listens to (arthursl12/Cloud_TP2_k8s), we must wait for ArgoCD to refresh its status on the git repository, which happens every 3 minutes by default.

During the tests, after around 2 minutes, ArgoCD detected the change in the repository and started the process of synchronizing the container images. Downloading and updating the pods take less than 10 seconds. However, in that meantime, the service becomes unavailable. The client scripts, if run during that roughly 10 second interval, will fail, reporting a "Connection Refused" error. Changes unrelated to versioning, such as increasing or decreasing the number of pods (parameter "replicas" in the YAML file) didn't have an impact on service availability.

In an attempt to estimate an duration for the whole process of deploying a change in the code, we can say that, for the first step (building and updating the image), if you have the commands at hand, it will take around a minute. The change in the YAML file is quick, because, most of the time, we'll just change a number or some small string, so in less than 30 seconds these changes are pushed to the repository. Finally, we must account for the 3 minute interval for ArgoCD to detect those changes and update the pods, which takes around 10 seconds. Adding up everything, the whole process takes almost five minutes.