**Journal**

Day 1

It took my a while to find a ‘computer’, but finally I made it.

I logged in to the ‘Internet’ and found the so called ‘GitHub’ code repository, and forked the repository to my local ‘computer’.

I learned that ‘Forking’ is different than ‘Cloning’ – when I fork I create my own seperate repository, while cloning keeps a connection and continues to synchronize with the original repository.

So far everything is going well.

I quickly began to inspect the files in the repository and soon realized I couldn’t understand the glyphs inside them. After some research on ‘Google’, I learned that the glyphs were some sort of ‘Code’ written in the high-level ‘Java programming language’. I also learned that there was a tool called ‘Maven’, that helps manage and build the code to an executable ‘Java’ file.

The tool was reading instructions from a ‘pom.xml’ file on how to build the executable file, and what dependencies are required for the build process. It was fascinating.

Day 2

I decided I need a plan for this project. I have to leave as soon as possible.

These Terran like to watch others of their kind chasing a circular object on a green field. they call it ‘Soccer’. Very primitive.

I started to design the project and set times for each task. The design can be seen [**here**](https://drive.google.com/file/d/1psyEifH-8UYLO1IJXtMP4JJP6wEkFC3Z/view?usp=sharing).

The first thing I did was edit the project I forked to suit the task I was given. This step was pretty easy, as all was needed was to edit the ‘<version>’ field of the pom file, and the ‘println’ field of the App.java file.

I then proceeded to build a Dockerfile. after some researched, it was clear that using a multi-stage Dockerfile was going to be far more beneficial, because of features like:

a) Smaller image size

b) Increased security

c) Faster builds by taking advantage of caching

d) Easier debugging at different stages

I tagged the image I built with 1.0.0 and pushed it to a repository I created on Dockerhub.

Automating the build incrementation proved to be a rather difficult task.

I started by running a script in Bash that automates the incrementation with the following code:

VERSION=$(mvn help:evaluate -Dexpression=project.version -q -DforceStdout)

NEXTVERSION=$(echo ${VERSION} | awk -F. -v OFS=. '{$NF += 1 ; print}')

mvn versions:set -DnewVersion=${NEXTVERSION}

But I soon realized I had to come up with a different plan, that will have to include some sort of ‘state’, so the version won’t reset on each build.

I decided to get the latest version from Dockerhub, and increment it each time:

VERSION=$(curl -s https://registry.hub.docker.com/v2/repositories/danieliko/maven-hw-testings/tags/ | jq '.results[0].name')

This way the version will increment every time a build succeeds.

I made some Taco and called it a day.

Day 3

I saw a ‘TV Show’ last night about people who were participating in an act of ‘Singing’. I never thought that manipulating voice frequency can achieve such positive experience. I will definitly try to immitate this culture back at home.

I decided to run the pipeilne in GitHub Actions, as it is simpler than Azure Pipeline.

I found some libraries that can help with the pipeline steps:

*actions/checkout@v3:* For fetching the last commit from github.

*docker/setup-buildx-action@v2: Enables* building dockers images on the runner.

*docker/login-action@v2:* For logging into Dockerhub

*docker/build-push-action@v2:* For building and pushing to Dockerhub.

These ‘actions’ made it far easier to implement the pipeline - for example, using the ‘ *docker/setup-buildx-action@v2’* yielded a runner that had Docker Engine pre installed. Very useful.

I decided to work on the pipeline on a different branch, named ‘workflow\_ci’, so that the ‘master’ branch will remain clean.

Some of GitHub Actions’ features like secrets and environment variables came in really useful in different parts of the pipeline.

I edited the Dockerfile so it will run as non root. This highly improves security.

Day 4

It took me over 70 commits to finally to get to the complete pipeline, but it is finally ready.

I will merge back to ‘Master’ tomorrow morning and deliver the final output.

Day 5

I delivered the project.

I hope this is my final entry of the journal, I can’t wait to get back to planet Kepler-186f.

I will definitely remember this expidition for the rest of my life!