Microservice Application on a VM

Goal

Create a microservice based application on a vm. Later we will compare this with running the same app containerized.

Instructions

Clone my repo instead of the one in the instructions below. Here it is.

Complete all of step 1 on this page.

Make sure you stop when you get to step 2.

To Pass off

In class:

Demonstrate that you can view sentiment when accessing the single ip that hosts this application.

Hints

Here is the quick and dirty to get part 1 done:

```
git clone https://github.com/nachofree/k8s-mastery.git
```

- **frontend**
  - cd sa-frontend
  - sudo apt install npm nginx
  - npm install
  - npm run build
  - sudo cp -r build/* /var/www/html
  - service nginx restart
  - Make sure it loads in the browser now

- **webapp**
  - cd sa-webapp
  - sudo apt install maven
  - mvn install
  - cd target
  - java -jar sentiment-analysis-web-0.0.1-SNAPSHOT.jar --sa.logic.api.url=http://localhost:5000 (leave it running in the terminal)

- **logic** (from a different terminal)
  - cd sa-logic
  - sudo apt install python3
  - sudo apt install python3-pip
  - cd sa
  - Edit requirements.txt so that it will also have the following line (if it’s not already there):
    - itsdangerous==2.0.1
  - pip3 install -r requirements.txt
  - python3 -m textblob.download_corpora
  - python3 sentiment_analysis.py (leave it running in the terminal)

Now when you load your web interface, the sentiment app should work. Note: If you are running this on a remote vm, you will have to construct a few tunnels to make everything work. My command was something like this: `ssh jfrancom@ssh.cs.dixie.edu -L 8888:144.38.193.248:80 -L 8080:144.38.193.248:8080`. Then you will visit [http://localhost:8888](http://localhost:8888) in the browser. So, nginx is hosting the frontend at port 80, it then sends some information to the python module on port 8080.