

[◀ Return to Classroom](#)

Operationalize a Machine Learning Microservice API

REVIEW

CODE REVIEW 3

HISTORY

Meets Specifications

You have taken ml project and written scripts to test and launch it as a container. Additionally, you have written a script to launch it as a cluster. Finally you have integrated circleci, so your deployments are tested as they are pushed to your repository. Well done!

You have listened to feedback and applied the advice in quick succession. This is a very valuable skill which will serve you well as a developer in your career.

Congratulations and good luck with the rest of your degree!

Files Submitted

The submitted repository includes a `.circleci` folder, a `README.md` file, a `Dockerfile` and `Makefile`, as well as an `app.py` file, a prediction script, and the necessary scripts to run and upload a microservice on Docker and Kubernetes.

There should also be two output text files: `docker_out.txt` and `kubernetes_out.txt` that include the log output after a prediction is made, given some sample input data.

NOTE: Before submitting a link to your complete, project repository, make sure you have included all required and complete files (including `run_kubernetes.sh`, `run_docker.sh`, `docker_out.txt`, `kubernetes_out.txt`, and a `.circleci` build directory).

All necessary files are included and the output files contain successfully ran cluster with a prediction.

A `.circleci` folder is included in the Github repository. The directory holds a `config.yml` that checks the project code for errors. Your project should pass, as indicated by a CircleCI status badge in the repository README.

Excellent, you have added the rightly configured circleci to your project and you have added circleci badge to your github.



Code Quality & Enhancement

Add an additional logging statement to `app.py` that prints as "info" the output prediction for some given input data.

The prediction output is in the `docker_out.txt` file.

The README file includes a summary of the project, how to run the Python scripts and web app, and an explanation of the files in the repository.

You have provided a readme file explaining how to run the project. You can remove the instructional part provided by Udacity.

Both the Dockerfile and the python file pass linting using pylint and hadolint. This may involve selectively customizing lint overrides in both tools. The lint should be run for both tools via the command `make lint`. Circleci build server validates step.

You have setup linting for the docker file and your app.py. Well done !

Docker Configuration

The Dockerfile should create a working directory, install the necessary dependencies, expose port 80, and specify that `app.py` run at container launch.

Excellent, your docker file creates working directory, installs dependencies, exposes port 80 and runs our app.py at container launch.

The Dockerfile should pass `make lint` without errors. Circleci build server validates step.

While running the docker container, call the prediction script, `make_predictions.sh`; the log output, which includes the output prediction value, should be included in your submission as a text file, `docker_out.txt`.

The prediction with the server response is outputted in the respective txt files.

Build, list, and run steps are completed in `run_docker.sh`.

The docker file successfully builds in my environment created through your make file and executes the command `app.py` to run the service. Your script builds the image and runs it, fantastic !

The built docker image is uploaded to your own personal Docker ID, as indicated by a complete `upload_docker.sh`.

Using the `upload_docker.sh` you are first logging in with your credentials and then uploading the image to your docker registry. Well done !

Kubernetes Configuration

This script runs a docker image with kubernetes, lists the kubernetes pod(s), and forwards the container port to a host, using `kubect1` appropriately.

Running the command `./run_kubernetes.sh` starts the cluster. And when the prediction is made through the script it responds appropriately.

While running on kubernetes, call `make_predictions.sh`; the terminal output should be included in your submission as a text file, `kubernetes_out.txt`.

Fantastic you have updated your log files and your app works as intended running on a cluster.

 [DOWNLOAD PROJECT](#)