passed status badge

## Project Overview

In this project, we have tested all the skills acquired in this course to operationalize a Machine Learning Microservice API.

We have been given a pre-trained, `sklearn` model that has been trained to predict housing prices in Boston according to several features, such as average rooms in a home and data about highway access, teacher-to-pupil ratios, and so on. You can read more about the data, which was initially taken from Kaggle, on [the data source site](https://www.kaggle.com/c/boston-housing). This project tests your ability to operationalize a Python flask app—in a provided file, `app.py`—that serves out predictions (inference) about housing prices through API calls. This project could be extended to any pre-trained machine learning model, such as those for image recognition and data labelling.

### Project Tasks

Our project goal is to operationalize this working, machine learning microservice using [kubernetes](https://kubernetes.io/), which is an open-source system for automating the management of containerized applications. In this project we have done below tasks:

\* Test the project code using linting

\* Complete a Dockerfile to containerize this application

\* Deploy our containerized application using Docker and make a prediction

\* Improve the log statements in the source code for this application

\* Configure Kubernetes and create a Kubernetes cluster

\* Deploy a container using Kubernetes and make a prediction

\* Upload a complete Github repo with CircleCI to indicate that your code has been tested

---

## Setup the Environment

\* Create a virtualenv and activate it

\* Run `make install` to install the necessary dependencies

### Running `app.py`

1. Standalone: `python app.py`

2. Run in Docker: `./run\_docker.sh`

3. Uploaded the Docker container to Docker Hub `./upload\_docker.sh`

3. Run in Kubernetes: `./run\_kubernetes.sh`

### Kubernetes Steps

\* Setup and Configure Docker locally

\* Setup and Configure Kubernetes locally

\* Create Flask app in Container

\* Run via kubectl