

# Devops Helm Coding Exercise

We have a web service called mlflow, which is an open-source ML model registry, it allows us to track and compare the different AI experiments we perform using an API and a nice UI.

The screenshot shows the mlflow web interface for a 'Listing Price Prediction' experiment. At the top, there's a navigation bar with the mlflow logo and links to 'Github' and 'Docs'. Below the title, it shows 'Experiment ID: 0' and 'Artifact Location: /Users/matei/mlflow/demo/mlruns/0'. There are search and filter controls: 'Search Runs' with a text input 'metrics.R2 > 0.24' and a 'Search' button; 'Filter Params' with a text input 'alpha, lr' and a 'Filter Metrics' with a text input 'rmse, r2' and a 'Clear' button. Below these, it says '4 matching runs' and provides buttons for 'Compare Selected' and 'Download CSV'. A table lists the runs with columns for Time, User, Source, Version, Parameters (alpha, l1\_ratio), and Metrics (MAE, R2, RMSE).

Time	User	Source	Version	Parameters		Metrics		
				alpha	l1_ratio	MAE	R2	RMSE
<input type="checkbox"/> 17:37	matei	linear.py	3a1995	0.5	0.2	84.27	0.277	158.1
<input type="checkbox"/> 17:37	matei	linear.py	3a1995	0.2	0.5	84.08	0.264	159.6
<input type="checkbox"/> 17:37	matei	linear.py	3a1995	0.5	0.5	84.12	0.272	158.6
<input type="checkbox"/> 17:37	matei	linear.py	3a1995	0	0	84.49	0.249	161.2

We wish to deploy it on our Kubernetes cluster, So your task is to write an Helm Chart to allow you to deploy it on kubernetes.

1. Build an Helm deployment for mlFlow according to the following specification -
  - A web service deployment -
    - Use docker hub image - larribas/mlflow
    - You can run it by calling

```
mlflow server --host=0.0.0.0 --port=80 \
--backend-store-uri={postgres-connection-string} \
--serve-artifacts \
--artifacts-destination {s3 or some local path}
```
    - Plan your deployment for horizontal scaling and high availability.
  - An ingress controller
    - An https endpoint ingress controller for the service, so we can access it from outside the network.
  - A postgresql deployment
    - For simplicity, let's start with a K8S hosted PostgreSQL.
    - Allow me to replace it with a SAAS managed DB, if i wish.
  - A storage of your choice for the mlflow artifact files

2. How can you make sure this application does not go out of hand and consume all the cluster resources?
3. How would you collect logs from this app for later analysis? Where and how would you save it?

Few notes for the implementation -

- Feel free to use or reference any existing chart dependency (i.e i don't expect from you to implement your own postgresql chart).
- In a nutshell, if something already exists - use it. don't try to reinvent it.