## TD 2: Deploy a MLFlow tracking server on Kubernetes

## Goal 🎯

Now, since you are part of the Data Engineering Team you've been asked to deploy a tool that is able to track and serve Machine Learning models at scale.

After a thorough study on the market, you decided that MLFlow would be the best candidate for your usecase. Now the final component is to make this tool scalable accross the whole Comcast company and its customers. Therefore you need to deploy it on Kubernetes. This way:

* Data Scientists will be able to build models and track performance
* Comcast will be able to serve models at scale with no downtime

## **Exercise - PART I**

To run properly, Mlflow needs 3 components:

* An MLFlow tracking server
* A PostgreSQL DB to store models' metadata, performance and metrics
* A S3 bucket to dump model's artifact into

Feel free to read this documentation to get a better understanding 👉 [MLflow with remote Tracking Server, backend and artifact stores](https://mlflow.org/docs/latest/tracking.html#scenario-4-mlflow-with-remote-tracking-server-backend-and-artifact-stores).

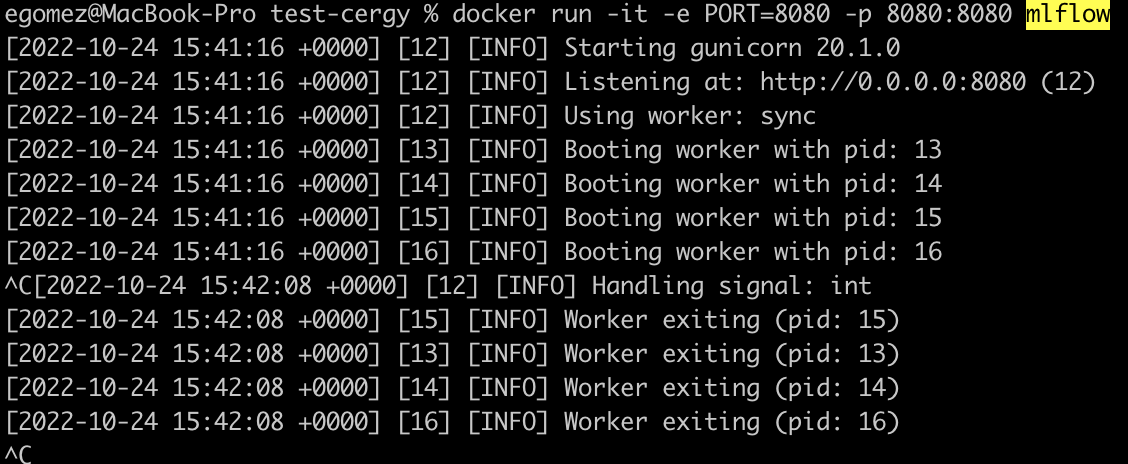
Therefore, before doing anything on Kubernetes, you will need first to:

* Have an AWS Account (if not done already)
* Have credentials (AWS\_ACCESS\_KEY\_ID & AWS\_SECRET\_ACCESS\_KEY) 👉 [Here is how to get them](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html)
* Create a PostgreSQL DB. You can do it by creating a free one on Heroku. Just:
  + Create an [Heroku account](http://heroku.com/) (if you did not already)
  + Create an [Heroku app](https://devcenter.heroku.com/articles/creating-apps)
  + Create an PostgreSQL DB [How to do it here](https://data.heroku.com/) then get your credentials under Settings > View Credentials
* Create a folder within an S3 bucket
  + How to create an S3 bucket - [here](https://docs.aws.amazon.com/AmazonS3/latest/userguide/create-bucket-overview.html)
  + How to create a folder within your S3 bucket - [here](https://cloudaffaire.com/faq/how-to-create-a-folder-in-aws-s3-bucket/)

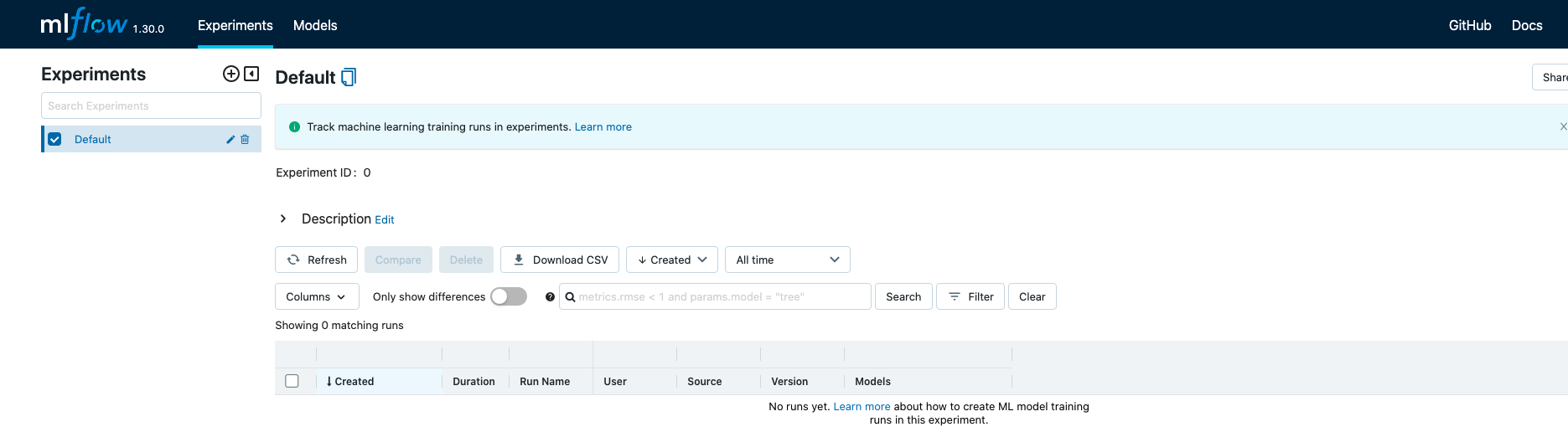
## **Exercise - PART II**

We have provided you with a Dockerfile with and a requirements.txt. You first need to create an image called mlflow that you will deploy to your kubernetes cluster.

Build the image and when the image is build you can test it using the following command :  
  
docker run -it -e PORT=8080 -p 8080:8080 mlflow

you should see an output like this:

You can access your browser to be able to access your mlflow server :

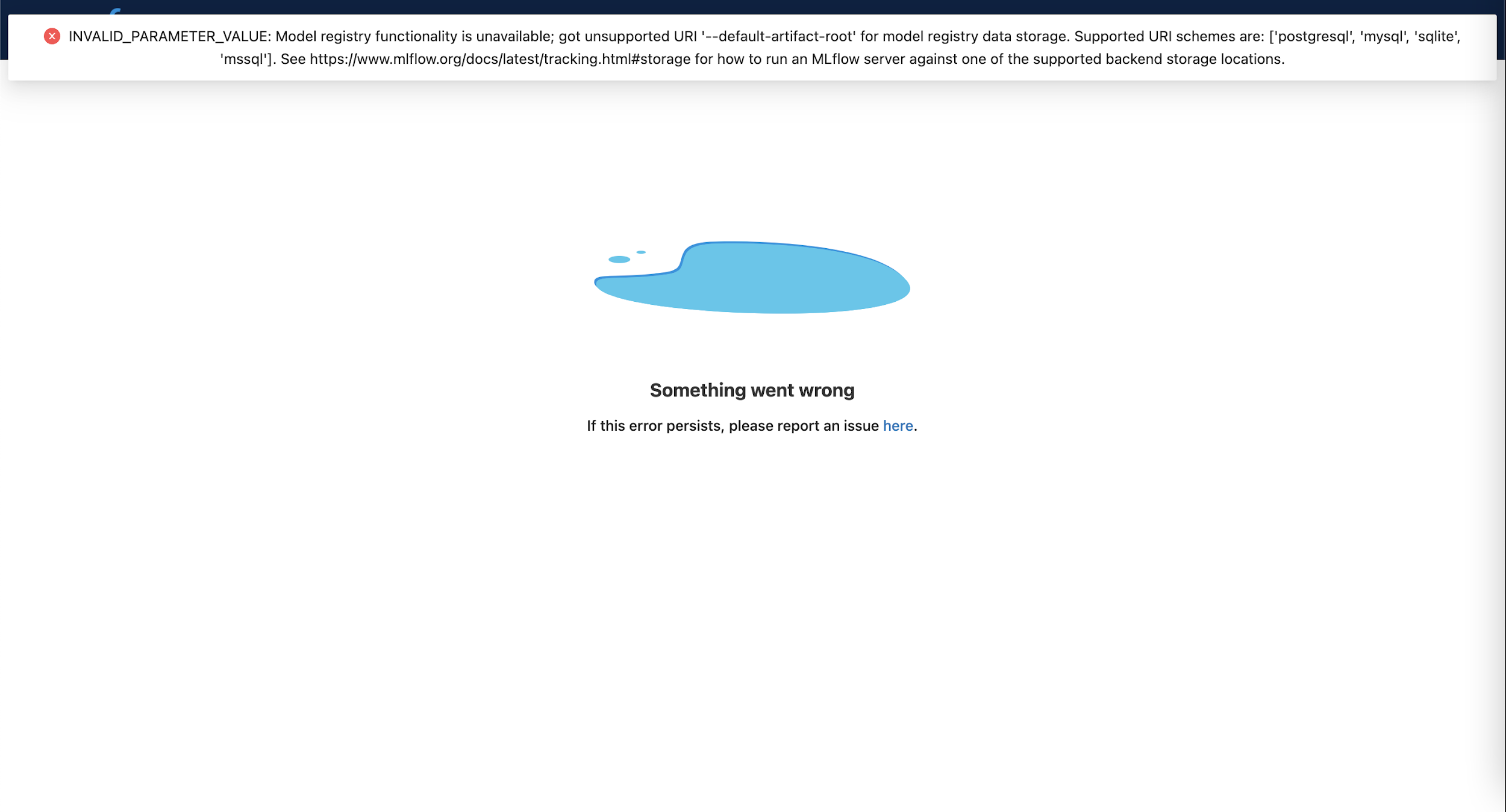
http://0.0.0.0:8080   
  


## **Exercise - PART III**

Now we can move on to Kubernetes. First let's start with the basics:

* Create Deployment, Service & Ingress Resources
* Use mlflow image for your Deployment resource
* You will need to add one environment variable called PORT into your Deployment for the app to work. Read this to know how to do it 👉 [Define Environment Variables for a Container](https://kubernetes.io/docs/tasks/inject-data-application/define-environment-variable-container/)

You can consider this step a success once you open up your web browser, go to mlflow.MINIKUBE\_IP.nip.io and see the mlflow dashboard. If you navigate, you might see an error like this one:



It is completely okay, we'll tackle that in the last part of this exercise.

## **Exercise - PART IV** Now for the last part of this exercise, we actually need to setup environment variables into our cluster. As we mentionned at the beginning of this exercise, you need to have:

## an MLFLOW server up and running

## an S3 bucket associated to it

## a PostgreSQL DB also associated to it

## So far we only did the first point. In this part, you need to add a few more environment variables:

## BACKEND\_STORE\_URI 👉 Get your credentials under Settings > View Credentials in your Heroku App Resources

## ARTIFACT\_ROOT 👉 How to get it here: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-bucket-intro.html> (under Accessing a bucket using S3:// section)

## AWS\_ACCESS\_KEY\_ID

## AWS\_SECRET\_ACCESS\_KEY

## 👉 You need to create a user with programmatic access on your AWS Account

## 👉 How to do it <https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html>

## Now add them all in your Deployment resource. Read this to know how to do it 👉 [Define Environment Variables for a Container](https://kubernetes.io/docs/tasks/inject-data-application/define-environment-variable-container/)

## **Exercise - PART V 🔒**

## Now that you have your mlflow app onto Kubernetes, there is one flaw that is important to fix: all environment variables are hard coded in Deployment. This is really not a good practice and we should use Secrets instead. Therefore:

## Create a Secret resource on your cluster

## Apply this Secret values to your Deployment

## 

## 