# Running Custom Jobs & Hyperparameter Tuning

## Scope

This document specifies the procedure to run custom jobs and hyperparameter tuning jobs directly from your local machine.

## Prerequisites

**Step 1: Install Google Cloud SDK and Python Client**

1. Install [Google Cloud SDK](https://cloud.google.com/sdk/docs/install).
2. Install Python client for Vertex AI: **pip install google-cloud-aiplatform**.

**Step 2: Authenticate and Set Project**

1. Authenticate: **gcloud auth login**
2. Set your project: **gcloud config set project [YOUR\_PROJECT\_ID]**

## Overview

Note, you can either [launch custom jobs](https://cloud.google.com/vertex-ai/docs/training/create-custom-job) from python, command-line or from the GCP console GUI. Similarly, with [hyperparameter tuning jobs](https://cloud.google.com/vertex-ai/docs/training/hyperparameter-tuning-overview), you can launch them too from python, command-line or GCP console GUI, however, it is slightly more involved as it doesn’t automatically package your code, and as you will see, you must package the hyperparameter tuning jobs into a docker container first.

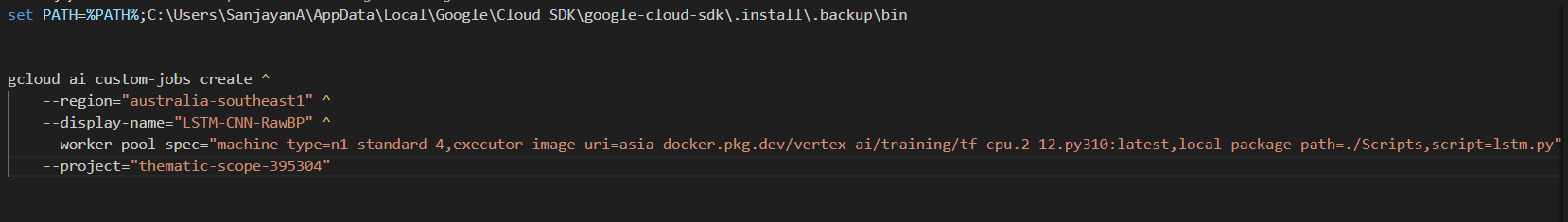
Cloud tools are constantly changing, and I also didn’t spend heaps of time exhausting the multitude of ways that jobs can be launched in GCP, so there may be easier ways that fit into your workflow a lot better. It’s always a good idea to reference the [official docs](https://cloud.google.com/vertex-ai/docs/training/create-custom-job) if you encounter errors or it doesn’t fit into your workflow.

## Running Custom Jobs

For running a custom job script (i.e. run a python script in cloud), you edit the “run\_gcloud.bat” file to specify the package path and the script to run inside of that package. Additionally, you can also specify the GCP image to use to build the docker container.

1. **run\_gcloud.bat**

* Set gcloud path if no admin access to set as environment var
* Specify machine specs to use
* Specify GCP image to use, and script location



**2. Run run\_gcloud.bat**

## Running Hyperparameter Tuning

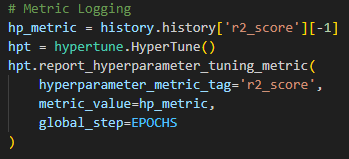
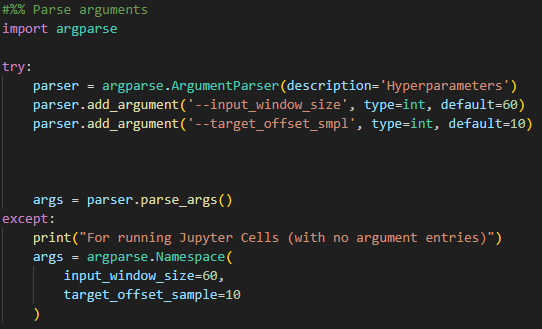
For running a hyperparameter tuning script, you need a Dockerfile to specify the container settings, and a “run\_hp\_tune.bat” file that specifies the commands to build the docker image, push it to the container registry, and then a command to spin up the job. Additionally, you require a hptuning\_config.yaml file that specifies the hyperparameters, their variable type and bounds to search, along with the machine settings to use.

These files should already be preconfigured to tune the LSTM.py for 2 variables, however there may be some PATHs that need to be modified.

The below shows each of the files used to run a hyperparameter tuning job, and explains what occurs in each.

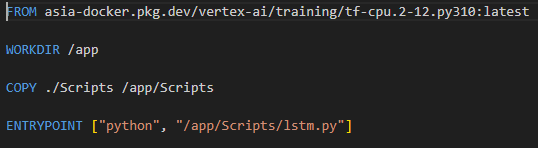
1. **Configure python script:**

* Configure hyperparameters as inputs from the command line arguments
* Configure the export of the metric



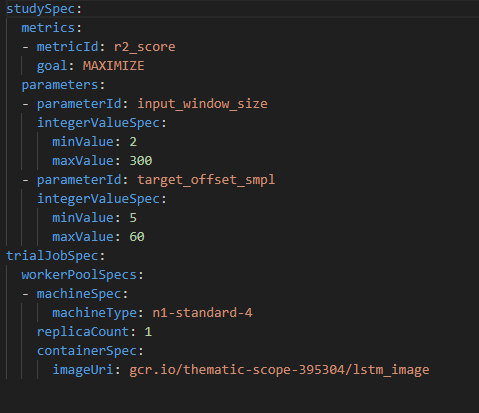
1. **Configure Dockerfile:**

* Set directory/script to run
* Choose GCP image to use



**4. Configure hptuning\_config.yaml**

* Specify metric, and goal for optimizing metric
* Specify hyperparameters, variable types and min/max values
* Specify machine specs, along with docker image in GCP artifact registry



**5. Run run\_hp\_tune.bat**

**3. Configure run\_hp\_tune.bat**

* Set gcloud path if no admin access to set as environment var
* Build docker image and upload to GCP artifact registry
* Create hp-tune job, specify yaml config file

