**Vertex AI Automation**

In order to deploy model to vertex ai from command line or using python we need to get authenticated first

We can use either

1. gcloud auth login --key-file=/path/to/service/account/keyfile.json

Description- Obtains access credentials for your user account via a web-based authorization flow. When this command comp

letes successfully, it sets the active account in the current configuration to the account specified. If no configuration exists, it creates a configuration named default.If valid credentials for an account are already available from a prior authorization, the account is set to active without rerunning the flow.

OR

1. gcloud auth activate-service-account =‘service-account-email’ --key-file= /path/to/service/account/keyfile.json

Description- To allow [gcloud](https://cloud.google.com/sdk/gcloud/reference) (and other tools in Google Cloud CLI) to use service account credentials to make requests, use this command to import these credentials from a file that contains a private authorization key, and activate them for use in [gcloud](https://cloud.google.com/sdk/gcloud/reference). gcloud auth activate-service-account serves the same function as [gcloud auth login](https://cloud.google.com/sdk/gcloud/reference/auth/login) but uses a service account rather than Google user credentials.

We can also pass the project-id with auth for ease:

**gcloud auth activate-service-account SERVICE\_ACCOUNT@DOMAIN.COM --key-file=/path/key.json --project=PROJECT\_ID**

Once authentication is done we declare the desired model names and and locations for models and endpoints and export them

Eg.:

export SERVICE\_ACCOUNT - 'qwertytestvai@pritish-vertex-ai.iam.gserviceaccount.com'

export REGION= 'us-central1'

export ENDPOINT\_NAME = 'vertex-ai-ep'

export MODEL\_NAME1 = 'Model\_1'

export MODEL1\_ID = 'MY\_MODEL\_01'

export MODEL\_NAME2= 'Model\_2'

export MODEL2\_ID = 'MY\_Model\_02'

export CONTAINER\_IMAGE = 'gcr.io/pritish-vertex-ai/mpg:v1'

export PROJECT= pritish-vertex-ai

**Deploying the first model**

gcloud ai models upload --region=$REGION \

--display-name=$MODEL\_NAME1 \

--container-image-uri=$CONTAINER\_IMAGE \

--models-id$MODEL1\_ID

It uploads the model from the provided container-image in the region chosen

If model id is specified than the corresponding model id is used else a unique number is assigned by default

**Deploy the second model similarly**

gcloud ai models upload --region=$REGION \

--display-name=$MODEL\_NAME2 \

--container-image-uri=$CONTAINER\_IMAGE \

--models-id$MODEL2\_ID

**Create an endpoint in vertex ai**

gcloud ai endpoints create --region=$REGION

--display-name=$ENDPOINT\_NAME

It creates a vertex ai endpoint on the region specified with the display name specified

These two flags are compulsory

**Deploy Model to an endpoint in vertex ai**

To deploy a model 456 to an endpoint 123 under project example in region us-central1, run:

gcloud ai endpoints deploy-model 123 --project=example --region=us-central1 --model=456 --display-name=my\_deployed\_model

**Deploy the first Model**

gcloud ai endpoints deploy-model --region=$REGION \

--model=$MODEL1\_ID

--display-name=$MODEL\_NAME1\

--machine-type=n1-standard-4 \

--min-replica-count=1 \

--max-replica-count=2 \

--traffic-split=0=100

**Deploy the Second Model with traffic split**

gcloud ai endpoints deploy-model --region=$REGION \

--model=$MODEL2\_ID

--display-name=$MODEL\_NAME2\

--machine-type=n1-standard-4 \

--min-replica-count=1 \

--max-replica-count=2 \

--traffic-split=0=40 , $MODEL1\_ID=60

--traffic-split=0=40 , $MODEL1\_ID=60 🡸 This line splits the traffic between the two models uses 0 to refer the current model being deployed and the old model id to reference the older model

**Gattling Script**

Gatling is a powerful load testing tool that uses Akka actors for simulating a huge load. The tests are written in Scala. Gatling uses DSL, which makes it easy to understand and write tests with very minimal knowledge of Scala. Gatling comes in both open source and enterprise versions.

Inorder to test the performance of the model we need to get first authentication token from gcloud of our account

Inorder to get the auth token : type – Bearer use :

gcloud auth application-default print-access-token

The json format structure for the post request can be found in the link below and the used format for our model is present in INPUT.json

https://cloud.google.com/vertex-ai/docs/predictions/online-predictions-custom-models#:~:text=Vertex%20AI%20online%20prediction%20is,your%20predictions%20in%20the%20response.