# Deployment Methodology

For deployment, I first created the python script with flask for testing it on colab with ngrok.

The api request should be in the form:-

{

'model': 'MODEL\_TYPE',

'modelfile': 'MODEL\_FILE\_NAME',

'text': 'TEXT',

'aspect': 'ASPECT'

}

# Example:

For example, inferencing for an logistic regression model with modelfile value should be

{

'model': 'lr',

'modelfile': '',

'text': 'text for inferencing sentence',

'aspect': 'text for aspect terms'

}

The resulting output received should be of the form:-

{

"sentiment":0

}

# Deploy on AWS:

After testing the api on postman with the help of ngrok on colab notebook, the AWS instance was created and the files were uploaded using WinSCP.

Then putty was used to connect to the instance with the private key. The necessary libraries were installed and the api script was run.

The instance id for deployment is given by:-

ec2-3-109-208-2.ap-south-1.compute.amazonaws.com

with port 8080 for accepting requests.

The server is made using waitress.

# API URL Example

An example of api url is given by:

http://ec2-3-109-208-2.ap-south-1.compute.amazonaws.com:8080/predict?model=svc&modelname=&text=hey%20marilyn%20thanks%20for%20your%20answer%20the%20soc2%20type%201%20certification%20is%20a%20document%20that%20can%20be%20shared%20with%20external%20parties%20for%20security%20assessment%20do%20you%20have%20a%20copy%20of%20the%20report%20we%20could%20explore?&aspect=Security

which gives result

{

"sentiment":1

}

# Cost:

Using apache benchmark, I made 1000 requests to the server for testing.

The results were

**[cloudshell-user@ip-10-0-98-236 run]$ ab -c 10 -n 1000 -r 'http://ec2-3-109-208-2.ap-south-1.compute.amazonaws.com:8080/predict?model=svc&modelname=&text=hey marilyn thanks for your answer the soc2 type 1 certification is a document that can be shared with external parties for security assessment do you have a copy of the report we could explore?&aspect=Security'**

**This is ApacheBench, Version 2.3 <$Revision: 1879490 $>**

**Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/**

**Licensed to The Apache Software Foundation, http://www.apache.org/**

**Benchmarking ec2-3-109-208-2.ap-south-1.compute.amazonaws.com (be patient)**

**Completed 100 requests**

**Completed 200 requests**

**Completed 300 requests**

**Completed 400 requests**

**Completed 500 requests**

**Completed 600 requests**

**Completed 700 requests**

**Completed 800 requests**

**Completed 900 requests**

**Completed 1000 requests**

**Finished 1000 requests**

**Server Software: waitress**

**Server Hostname: ec2-3-109-208-2.ap-south-1.compute.amazonaws.com**

**Server Port: 8080**

**Document Path: /predict?model=svc&modelname=&text=hey marilyn thanks for your answer the soc2 type 1 certification is a document that can be shared with external parties for security assessment do you have a copy of the report we could explore?&aspect=Security**

**Document Length: 178 bytes**

**Concurrency Level: 10**

**Time taken for tests: 0.789 seconds**

**Complete requests: 1000**

**Failed requests: 0**

**Non-2xx responses: 1000**

**Total transferred: 375000 bytes**

**HTML transferred: 178000 bytes**

**Requests per second: 1267.48 [#/sec] (mean)**

**Time per request: 7.890 [ms] (mean)**

**Time per request: 0.789 [ms] (mean, across all concurrent requests)**

**Transfer rate: 464.16 [Kbytes/sec] received**

**Connection Times (ms)**

**min mean[+/-sd] median max**

**Connect: 0 0 0.7 0 9**

**Processing: 2 7 0.5 7 12**

**Waiting: 1 7 0.5 7 12**

**Total: 2 8 0.6 8 14**

**Percentage of the requests served within a certain time (ms)**

**50% 8**

**66% 8**

**75% 8**

**80% 8**

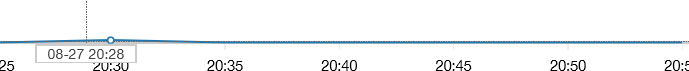
**90% 8**

**95% 8**

**98% 9**

**99% 11**

**100% 14 (longest request)**

****

After running the benchmark, I observed no noticeable change in CPUCreditUsage Metric as it fell under the load necessary to cause CPU credit usage. Hence, no cost other than basic instance costs necessary for api. While, that is the case, when I used the same request as 1000 iterations in postman, the CPUCreditUsage peaked at 2 while the iterations were running.