1. Deploy Kubernetes App

Objective:

MNC Corp. needs to add a simple counter to their high traffic(100 req/s) website. Of course this is a microservice and has to be deployed to the Kubernetes cluster. The app is now ready and you have to deploy it in a scalable way.

Specification:

Platform:

This app should be deployed on Kubernetes. Docker image for this app can be found here. (https://hub.docker.com/r/tarunbhardwaj/flask-counter-app).

- GCP provides managed Kubernetes and free credits.
- This micro service depends on Redis to store count. The URL to access redis is provided to the app as REDIS_URL.

Test Case:

You can test your deployment via apache benchmark (ab). We will be using the same to see how scalable the app is.

```
$> ab -l -c 100 -t 10 https://app_url

# Result should look something like below
...
Concurrency Level:      100
Time taken for tests:   10.000 seconds
Requests per second:    1200 [#/sec] (mean)
Failed requests:        0
...
```

Requests per second in above result should be >= 1200/sec.

Level Up:

- Requests should not break when app is being deployed
- Redis should be HA and on K8s only, for performance

Deliverables:

- A working service - send us the URL
- Code on Github for the deployment itself. Feel free to use Ansible, Chef or Google's deployment manager - the choice is yours.

Points and Rating scheme:

- Approach and Code Quality
- Commit History
- Deployment

2.Automate the process of stop

Automate the process of stop to a group of instances (based on tags). Ensure that there is no user logged into the servers, and CPU usage is idle for the particular period of time before the stop. The idle period and tag will be passed in as the arguments.

usage: autostop <Tag name> < idle period> For example:

autostop <development> 30

If the current time is 7 PM, the script needs to check the idle development instances in the last 30 minutes(means 6.30 PM to 7 PM) and no users logged into the instances before stop. Don't set up a permanent cloudwatch alarm to stop the instances. The script needs to run on-demand for stopping the instances.