SRE Image Resize Exercise

It is a simple image resize app document, in which I have created a GKE cluster with the help of terraform for deploying the app to the cluster and I have created the app with the help of helm chart and deploy on the cluster.

GKE Cluster:

I have created the GKE Cluster using github GKE Module repo and GCP best practices for creating the GKE. Best

1. Must create a regional cluster because it provides multiple control plans.
2. Cluster Autoscaler for adding and removing Nodes based on the scheduled workload
3. Node auto-provisioning, for dynamically creating new node pools with nodes that match the needs of users' Pods

References:

1. https://cloud.google.com/architecture/best-practices-for-running-cost-effective-kubernetes-applications-on-gke
2. https://cloud.google.com/kubernetes-engine/docs/best-practices/scalability
3. https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster
4. https://github.com/terraform-google-modules/terraform-google-kubernetes-engine

Monitoring & scaling:

We will monitor our cluster with prometheus and grafana. Prometheus is an open-source application used for metrics-based monitoring and alerting. It calls out to your application, pulls real-time metrics, compresses and stores them in a time-series database.

helm install prometheus stable/prometheus-operator

As prometheus pull the metrics and expose on /metric, we will pull those metrics with the adopter with every 15 second and HPA maintains the desired state by using the following formula.

```
desiredReplicas = ceil[currentReplicas * ( currentMetricValue / desiredMetricValue )]
```

`Let's suppose`

The current metric value is 200m.

Desired value is 100m.

1 X (200.0/100.0) == 2.0

the number of replicas will be **doubled**

Manual CPU Calculation:

1 CPU = 1000Mi
RPS=No. of CPU X (1/Task time)
let ust suppose our request take max 5ms per request
RPS = 1 X (1/5ms)
RPS = 1000/5 RPS = 200 of 100% CPU utilization
but we will configure our autoscaling on 50% utilization of the CPU so the next pod can be ready.
50% means 100 RPS

Cost Calculation:

let's suppose
cost of 1 CPU = 10$
1 CPU can handle approximate 180 request
100000/175 = 571.42 approximate
10$ X 572$ = 5720$

Load Testing:

Load testing will be performed with Jmeter.
We will start load testing with 10 RPS and increase the request up to 100000 and check where our request timeout of service didn't respond. Load testing performing steps are explain nicely on the gurue99 website with each and every steps.

References:

1. https://jmeter.apache.org/
2. https://www.guru99.com/jmeter-performance-testing.html