Faculty of Engineering and Applied Science

SOFE 4790U Distributed Systems

Lab 2 Individual: Deploying a request splitting ambassador and a load balancer

Group 19

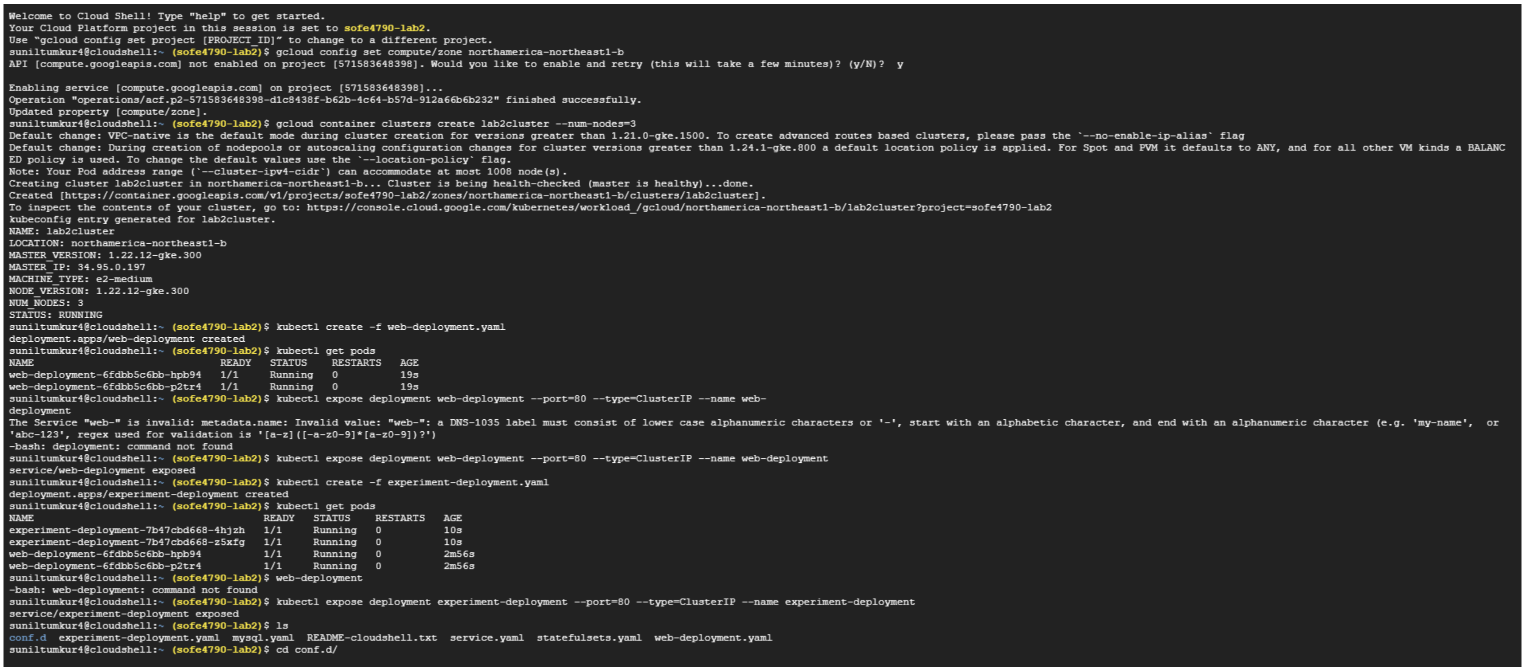
Sunil Tumkur 100620430

https://github.com/sunilt4/Distributed-Systems

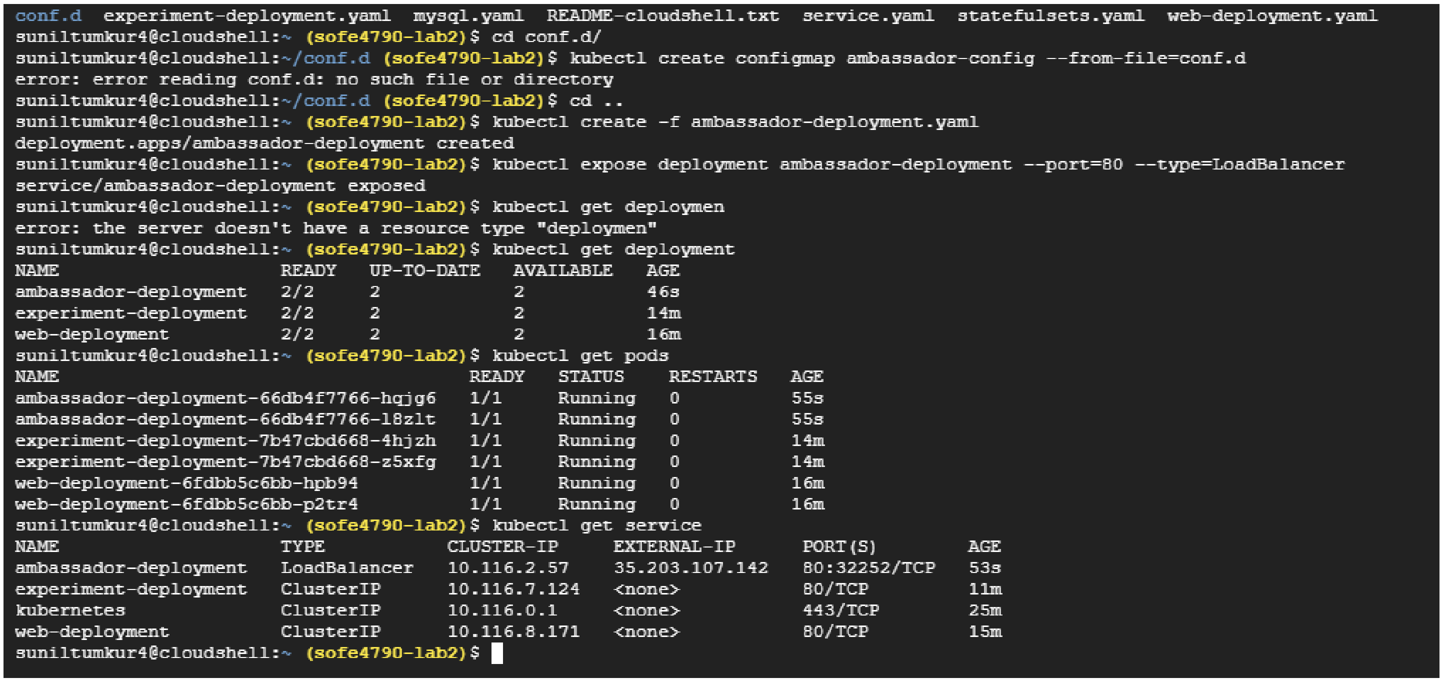
Part 1:

Scroll below to the Discussion section

Part 2:



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to sofe4790-lab2.
Use “gcloud config set project [PROJECT_ID]” to change to a different project.
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ gcloud config set compute/zone northamerica-northeast1-b
API [compute.googleapis.com] not enabled on project [571583648398]. Would you like to enable and retry (this will take a few minutes)? (y/N)?  y

Enabling service [compute.googleapis.com] on project [571583648398]...
Operation "operations/acf.p2-571583648398-d1c8438f-b62b-4c64-b57d-912a66b6b232" finished successfully.
Updated property [compute/zone].
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ gcloud container clusters create lab2cluster --num-nodes=3
Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1500. To create advanced routes based clusters, please pass the `--no-enable-ip-alias` flag
Default change: During creation of nodepools or autoscaling configuration changes for cluster versions greater than 1.24.1-gke.800 a default location policy is applied. For Spot and PVM it defaults to ANY, and for all other VM kinds a BALANC
ED policy is used. To change the default values use the `--location-policy` flag.
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
Creating cluster lab2cluster in northamerica-northeast1-b... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/sofe4790-lab2/zones/northamerica-northeast1-b/clusters/lab2cluster].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/northamerica-northeast1-b/lab2cluster?project=sofe4790-lab2
kubeconfig entry generated for lab2cluster.
NAME: lab2cluster
LOCATION: northamerica-northeast1-b
MASTER_VERSION: 1.22.12-gke.300
MASTER_IP: 34.95.0.197
MACHINE_TYPE: e2-medium
NODE_VERSION: 1.22.12-gke.300
NUM_NODES: 3
STATUS: RUNNING
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl create -f web-deployment.yaml
deployment.apps/web-deployment created
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get pods
NAME                             READY   STATUS    RESTARTS   AGE
web-deployment-6fdbb5c6bb-hpb94   1/1     Running   0          19s
web-deployment-6fdbb5c6bb-p2tr4   1/1     Running   0          19s
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl expose deployment web-deployment --port=80 --type=ClusterIP --name web-
deployment
The Service "web-" is invalid: metadata.name: Invalid value: "web-": a DNS-1035 label must consist of lower case alphanumeric characters or '-', start with an alphabetic character, and end with an alphanumeric character (e.g. 'my-name',  or
'abc-123', regex used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?')
-bash: deployment: command not found
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl expose deployment web-deployment --port=80 --type=ClusterIP --name web-deployment
service/web-deployment exposed
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl create -f experiment-deployment.yaml
deployment.apps/experiment-deployment created
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
experiment-deployment-7b47cbd668-4hjzh   1/1     Running   0          10s
experiment-deployment-7b47cbd668-z5xfg   1/1     Running   0          10s
web-deployment-6fdbb5c6bb-hpb94          1/1     Running   0          2m56s
web-deployment-6fdbb5c6bb-p2tr4          1/1     Running   0          2m56s
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ web-deployment
-bash: web-deployment: command not found
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl expose deployment experiment-deployment --port=80 --type=ClusterIP --name experiment-deployment
service/experiment-deployment exposed
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ ls
conf.d  experiment-deployment.yaml  mysql.yaml  README-cloudshell.txt  service.yaml  statefulsets.yaml  web-deployment.yaml
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ cd conf.d/
```

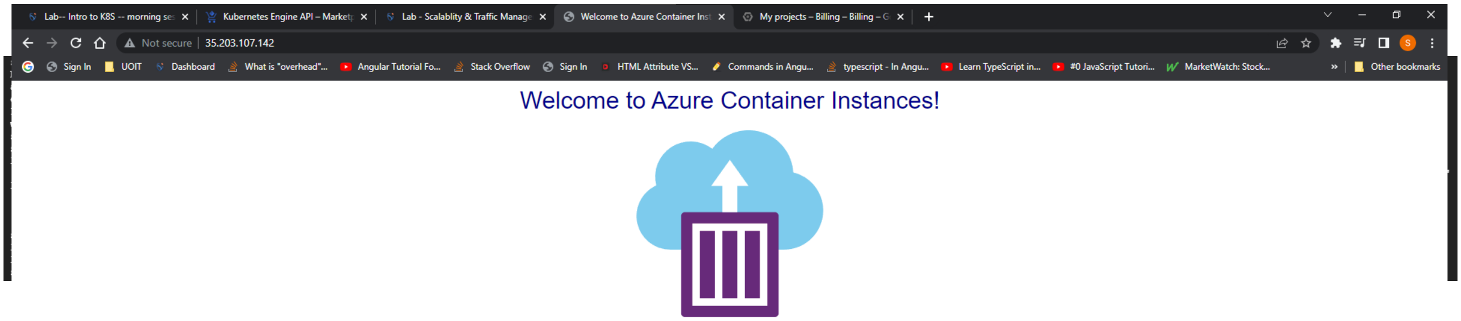


```
conf.d  experiment-deployment.yaml  mysql.yaml  README-cloudshell.txt  service.yaml  statefulsets.yaml  web-deployment.yaml
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ cd conf.d/
suniltumkur4@cloudshell:~/conf.d (sofe4790-lab2)$ kubectl create configmap ambassador-config --from-file=conf.d
error: error reading conf.d: no such file or directory
suniltumkur4@cloudshell:~/conf.d (sofe4790-lab2)$ cd ..
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl create -f ambassador-deployment.yaml
deployment.apps/ambassador-deployment created
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl expose deployment ambassador-deployment --port=80 --type=LoadBalancer
service/ambassador-deployment exposed
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get deploymen
error: the server doesn't have a resource type "deploymen"
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get deployment
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
ambassador-deployment   2/2     2            2           46s
experiment-deployment   2/2     2            2           14m
web-deployment          2/2     2            2           16m
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get pods
NAME                                     READY   STATUS    RESTARTS   AGE
ambassador-deployment-66db4f7766-hqjg6   1/1     Running   0          55s
ambassador-deployment-66db4f7766-l8zlt   1/1     Running   0          55s
experiment-deployment-7b47cbd668-4hjzh   1/1     Running   0          14m
experiment-deployment-7b47cbd668-z5xfg   1/1     Running   0          14m
web-deployment-6fdbb5c6bb-hpb94          1/1     Running   0          16m
web-deployment-6fdbb5c6bb-p2tr4          1/1     Running   0          16m
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get service
NAME                    TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)        AGE
ambassador-deployment   LoadBalancer   10.116.2.57    35.203.107.142   80:32252/TCP   53s
experiment-deployment   ClusterIP      10.116.7.124   <none>           80/TCP         11m
kubernetes              ClusterIP      10.116.0.1     <none>           443/TCP        25m
web-deployment          ClusterIP      10.116.8.171   <none>           80/TCP         15m
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ 
```

The commands you should use are:

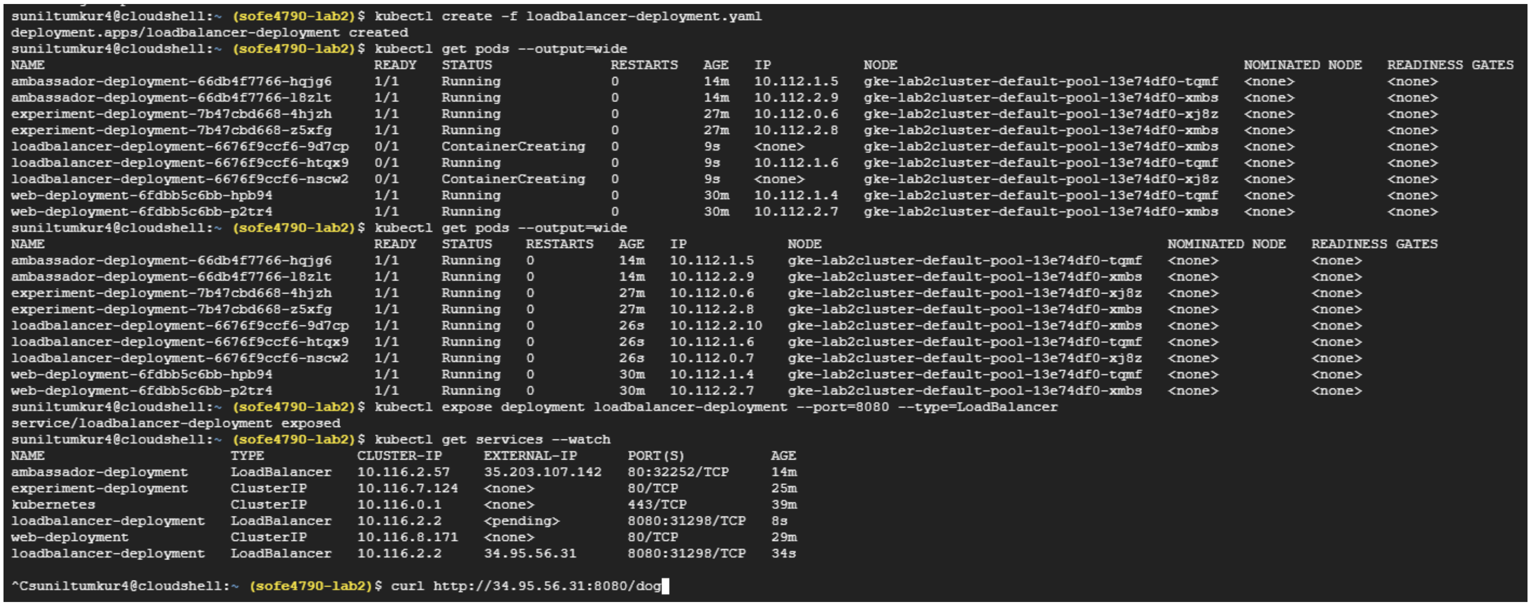
- kubectl get deployment
- kubectl get pods
- kubectl get service

The external IP associated with ambassador-deployment is 35.203.107.142.

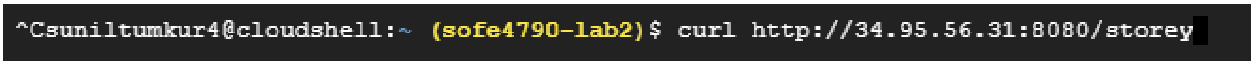


Welcome to Azure Container Instances!

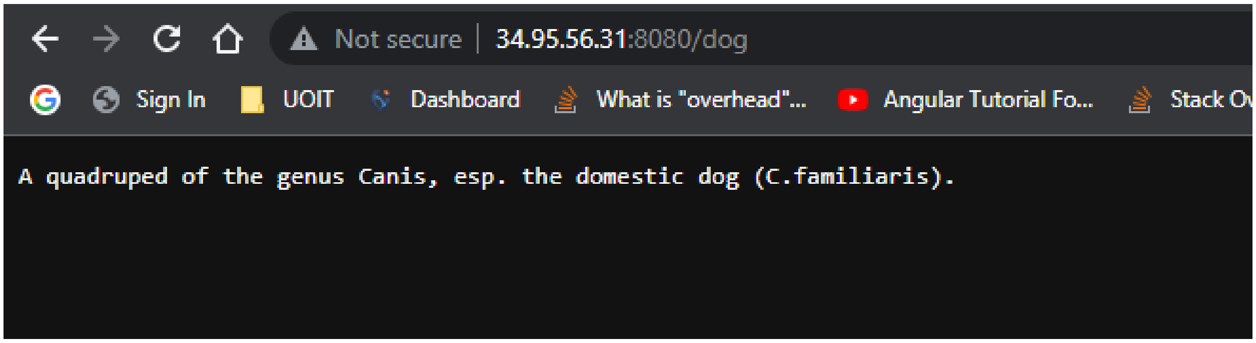
Part 3:



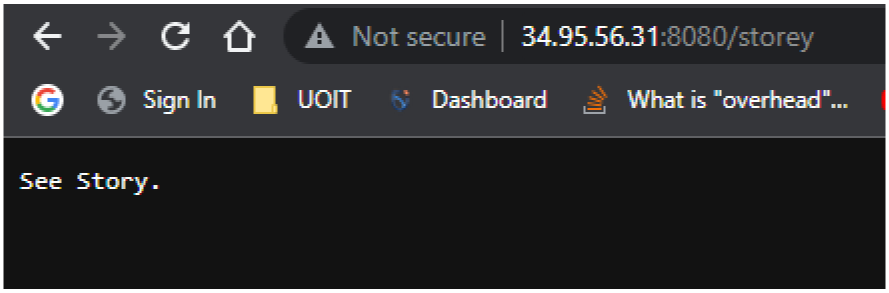
```
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl create -f loadbalancer-deployment.yaml
deployment.apps/loadbalancer-deployment created
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get pods --output=wide
NAME                                      READY   STATUS              RESTARTS   AGE   IP            NODE                                          NOMINATED NODE   READINESS GATES
ambassador-deployment-66db4f7766-hqjg6    1/1     Running             0          14m   10.112.1.5    gke-lab2cluster-default-pool-13e74df0-tqmf   <none>           <none>
ambassador-deployment-66db4f7766-18zlt    1/1     Running             0          14m   10.112.2.9    gke-lab2cluster-default-pool-13e74df0-xmbs   <none>           <none>
experiment-deployment-7b47cbd668-4hjzh    1/1     Running             0          27m   10.112.0.6    gke-lab2cluster-default-pool-13e74df0-xj8z   <none>           <none>
experiment-deployment-7b47cbd668-z5xfg    1/1     Running             0          27m   10.112.2.8    gke-lab2cluster-default-pool-13e74df0-xmbs   <none>           <none>
loadbalancer-deployment-6676f9ccf6-9d7cp  0/1     ContainerCreating   0          9s    <none>        gke-lab2cluster-default-pool-13e74df0-xmbs   <none>           <none>
loadbalancer-deployment-6676f9ccf6-htqx9  0/1     Running             0          9s    10.112.1.6    gke-lab2cluster-default-pool-13e74df0-tqmf   <none>           <none>
loadbalancer-deployment-6676f9ccf6-nscw2  0/1     ContainerCreating   0          9s    <none>        gke-lab2cluster-default-pool-13e74df0-xj8z   <none>           <none>
web-deployment-6fdbb5c6bb-hpb94           1/1     Running             0          30m   10.112.1.4    gke-lab2cluster-default-pool-13e74df0-tqmf   <none>           <none>
web-deployment-6fdbb5c6bb-p2tr4           1/1     Running             0          30m   10.112.2.7    gke-lab2cluster-default-pool-13e74df0-xmbs   <none>           <none>
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get pods --output=wide
NAME                                      READY   STATUS    RESTARTS   AGE   IP             NODE                                          NOMINATED NODE   READINESS GATES
ambassador-deployment-66db4f7766-hqjg6    1/1     Running   0          14m   10.112.1.5     gke-lab2cluster-default-pool-13e74df0-tqmf   <none>           <none>
ambassador-deployment-66db4f7766-18zlt    1/1     Running   0          14m   10.112.2.9     gke-lab2cluster-default-pool-13e74df0-xmbs   <none>           <none>
experiment-deployment-7b47cbd668-4hjzh    1/1     Running   0          27m   10.112.0.6     gke-lab2cluster-default-pool-13e74df0-xj8z   <none>           <none>
experiment-deployment-7b47cbd668-z5xfg    1/1     Running   0          27m   10.112.2.8     gke-lab2cluster-default-pool-13e74df0-xmbs   <none>           <none>
loadbalancer-deployment-6676f9ccf6-9d7cp  1/1     Running   0          26s   10.112.2.10    gke-lab2cluster-default-pool-13e74df0-xmbs   <none>           <none>
loadbalancer-deployment-6676f9ccf6-htqx9  1/1     Running   0          26s   10.112.1.6     gke-lab2cluster-default-pool-13e74df0-tqmf   <none>           <none>
loadbalancer-deployment-6676f9ccf6-nscw2  1/1     Running   0          26s   10.112.0.7     gke-lab2cluster-default-pool-13e74df0-xj8z   <none>           <none>
web-deployment-6fdbb5c6bb-hpb94           1/1     Running   0          30m   10.112.1.4     gke-lab2cluster-default-pool-13e74df0-tqmf   <none>           <none>
web-deployment-6fdbb5c6bb-p2tr4           1/1     Running   0          30m   10.112.2.7     gke-lab2cluster-default-pool-13e74df0-xmbs   <none>           <none>
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl expose deployment loadbalancer-deployment --port=8080 --type=LoadBalancer
service/loadbalancer-deployment exposed
suniltumkur4@cloudshell:~ (sofe4790-lab2)$ kubectl get services --watch
NAME                       TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)          AGE
ambassador-deployment      LoadBalancer   10.116.2.57    35.203.107.142   80:32252/TCP     14m
experiment-deployment      ClusterIP      10.116.7.124   <none>           80/TCP           25m
kubernetes                 ClusterIP      10.116.0.1     <none>           443/TCP          39m
loadbalancer-deployment    LoadBalancer   10.116.2.2     <pending>        8080:31298/TCP   8s
web-deployment             ClusterIP      10.116.8.171   <none>           80/TCP           29m
loadbalancer-deployment    LoadBalancer   10.116.2.2     34.95.56.31      8080:31298/TCP   34s

^Csuniltumkur4@cloudshell:~ (sofe4790-lab2)$ curl http://34.95.56.31:8080/dog
```



```
^Csuniltumkur4@cloudshell:~ (sofe4790-lab2)$ curl http://34.95.56.31:8080/storey
```



```
A quadruped of the genus Canis, esp. the domestic dog (C.familiaris).
```



```
See Story.
```

Discussion

Summary of Problem in Part 1:

- Setting up and managing multiple endpoints for multiple backend services. When the API changes, the client must change

Solution in Part 1:

- A gateway is placed in front of a set of applications and application layer 7 routing is used to route the request to the appropriate instances.
- The client only needs to know about a single endpoint in which it will only communicate with that single endpoint
- If a service is replaced, the client doesn't need to update and it can continue to make requests to the gateway and routing changes
- The gateway will also acquire backend services from clients which will help keep client calls simple when changes are made in backend services behind the gateway. Client calls will then be routed to any service/services that need to handle the client behavior. This will allow for adding and splitting of services behind the gateway without modifying the client.

Requirements needed Pattern in Part 1?

- Services
- Single endpoint
- API

Which of these requirements can be achieved by the procedures shown in parts 2 and 3?

- All of them (Services, Endpoints, and an API)

Design

Why is autoscaling usually used?

- Autoscaling is used as it allows the optimal use of resource utilization and cloud spending.
- Without autoscaling you have to manually provision resources depending on conditions and scale down/up.

How is autoscaling implemented?

- Automatically scaling a cluster up and down spending on the demand/resources being used.
- Horizontal Pod Autoscaler. Pods will get deployed in response to a growing load. The scaler will automatically order the workload resource to scale down if the load drops.
- Vertical Pod Autoscaler. Modify the resources such as CPU and RAM of each node in the cluster. It will resize pods for optimal CPU and memory resources. Autoscaling is used to provide the recommended values for CPU and memory requests and limits. Can automatically update the values.

How is autoscaling different from load balancing and request splitter?

- Autoscaling is used for automatic scaling up and down instances
- Load balancing is used to distribute incoming traffic across to multiple targets/services
- Request splitter will split the incoming request/message into multiple parts and send it to be processed individually.

Video Link 1:
https://drive.google.com/file/d/1oGwlZe_duQeWy_G7fCbmkhnnmDXnuLnU/view?usp=sharing

Video Link 2:

https://drive.google.com/file/d/1Gxq5PH-bs996QGOjAXRVX2_ConsL3gB-/view?usp=sharing

References

[1] Erjosito, “Gateway Routing Pattern - Azure Architecture Center,” *Azure Architecture Center | Microsoft Learn*. [Online]. Available: https://learn.microsoft.com/en-us/azure/architecture/patterns/gateway-routing. [Accessed: 26-Sep-2022].

[2] “Horizontal pod autoscaling,” *Kubernetes*, 10-Jun-2022. [Online]. Available: https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/. [Accessed: 28-Sep-2022].

[3] R. Shivalkar, “Kubernetes Autoscaling: How to use the kubernetes autoscaler,” *ClickIT*, 29-Jul-2022. [Online]. Available: https://www.clickittech.com/devops/kubernetes-autoscaling/. [Accessed: 28-Sep-2022].