Faculty of Engineering and Applied Science

SOFE 4790U Distributed Systems

Lab 2: Deploying a request splitting ambassador and a load balancer with Kubernetes

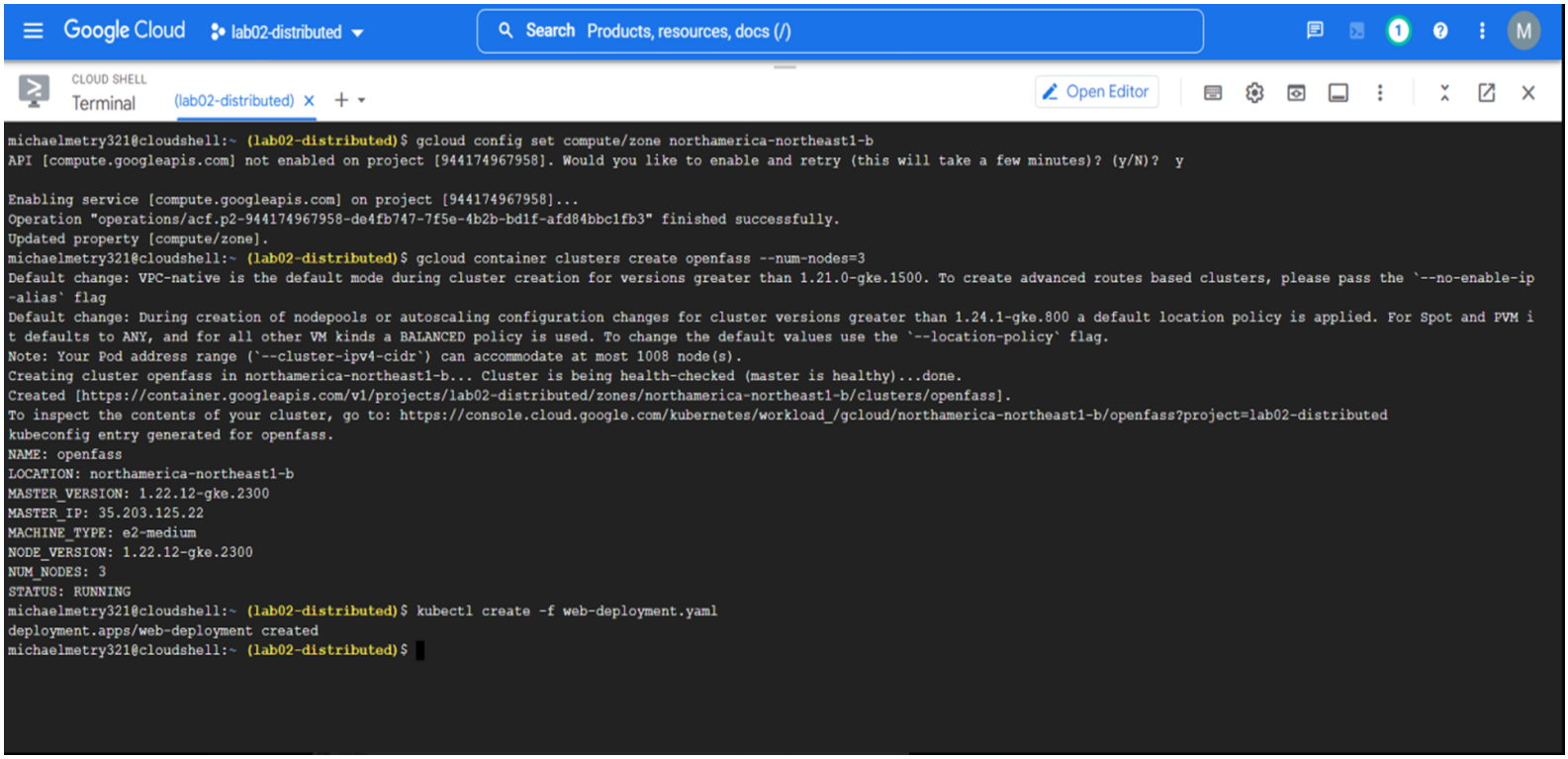
Group 19

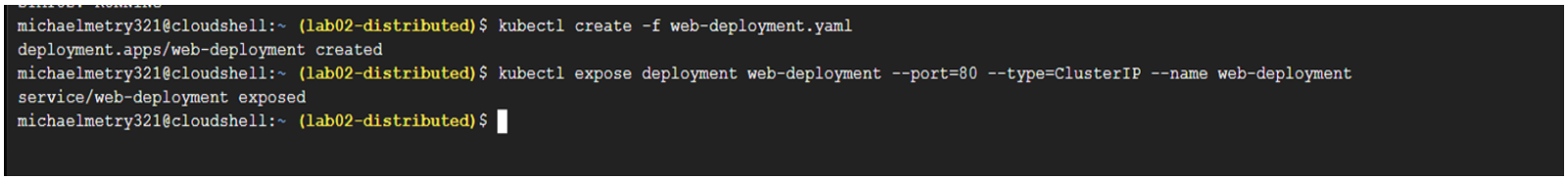
Michael Metry 100747141

GitHub Link:

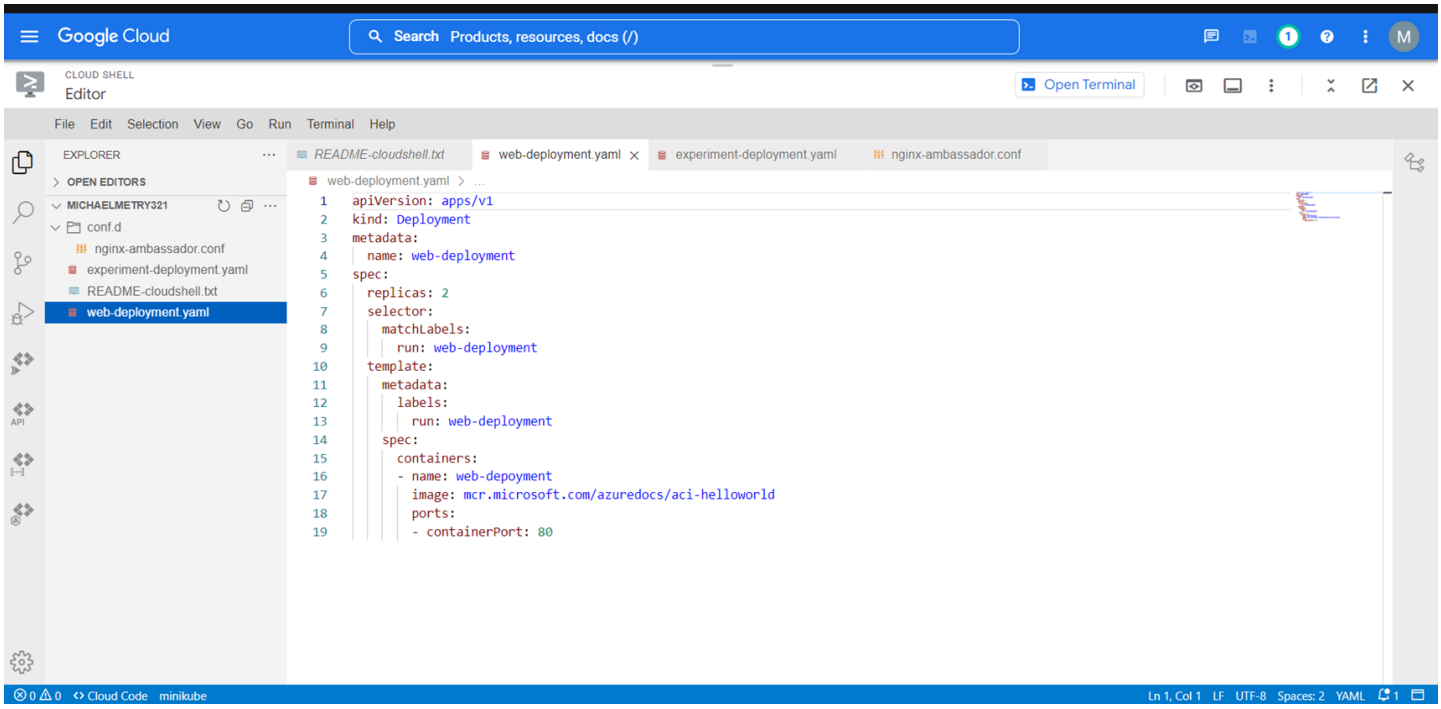
https://github.com/sunilt4/Distributed-Systems/tree/main/Lab%202

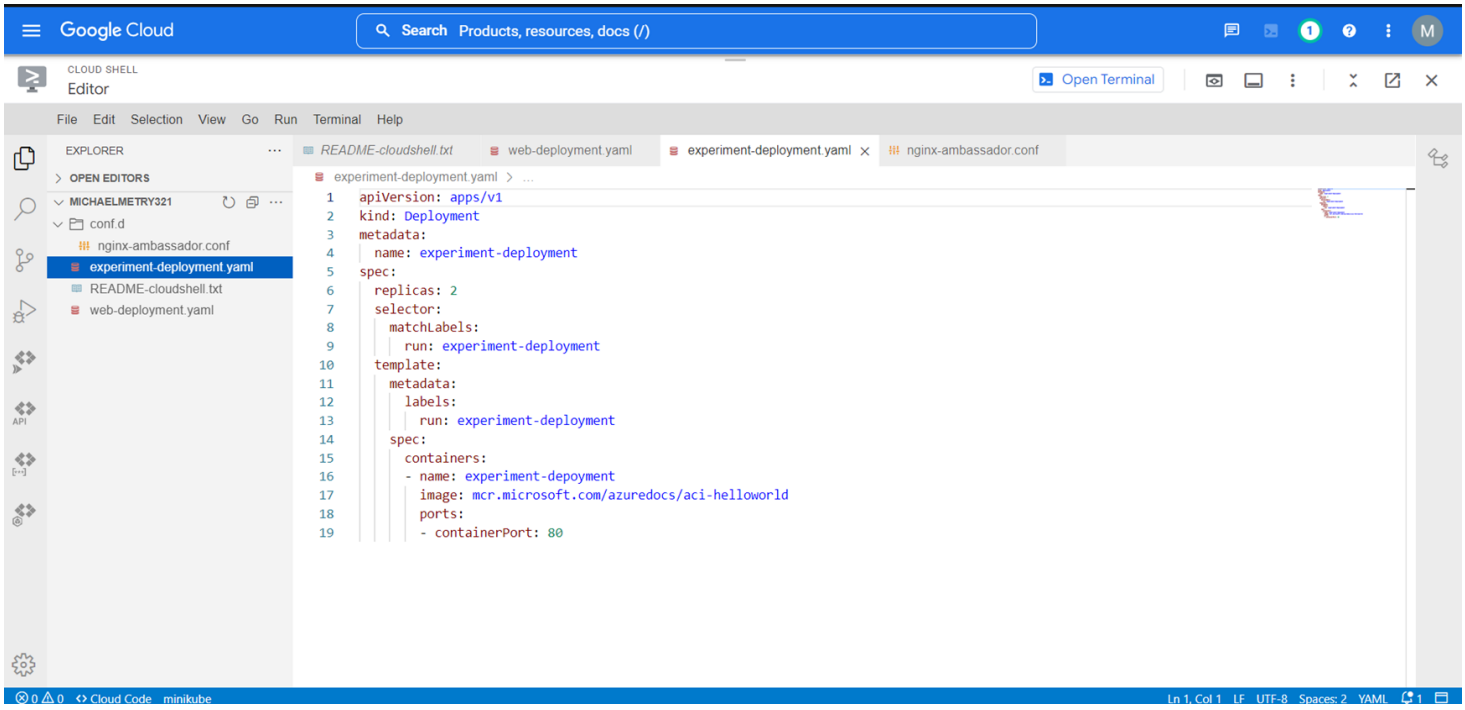
Part 2:

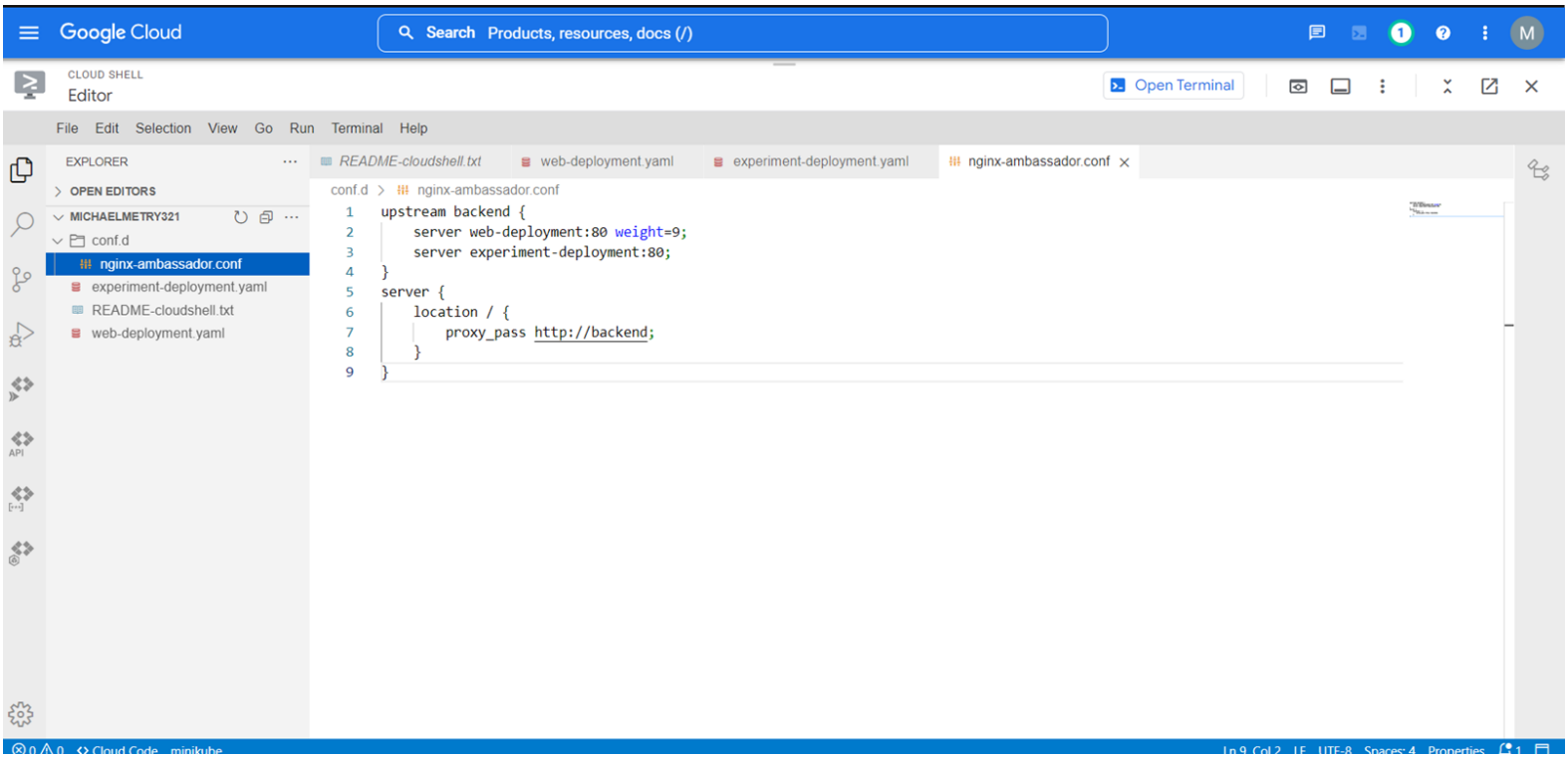

Google Cloud
lab02-distributed
Search Products, resources, docs (/)
CLOUD SHELL
Terminal
(lab02-distributed)
Open Editor
michaelmetry321@cloudshell:~ (lab02-distributed)$ gcloud config set compute/zone northamerica-northeast1-b
API [compute.googleapis.com] not enabled on project [944174967958]. Would you like to enable and retry (this will take a few minutes)? (y/N)? y
Enabling service [compute.googleapis.com] on project [944174967958]...
Operation "operations/acf.p2-944174967958-de4fb747-7f5e-4b2b-bd1f-afd84bbc1fb3" finished successfully.
Updated property [compute/zone].
michaelmetry321@cloudshell:~ (lab02-distributed)$ gcloud container clusters create openfass --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 BALANCED 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 openfass in northamerica-northeast1-b... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/lab02-distributed/zones/northamerica-northeast1-b/clusters/openfass].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/northamerica-northeast1-b/openfass?project=lab02-distributed
kubeconfig entry generated for openfass.
NAME: openfass
LOCATION: northamerica-northeast1-b
MASTER_VERSION: 1.22.12-gke.2300
MASTER_IP: 35.203.125.22
MACHINE_TYPE: e2-medium
NODE_VERSION: 1.22.12-gke.2300
NUM_NODES: 3
STATUS: RUNNING
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl create -f web-deployment.yaml
deployment.apps/web-deployment created
michaelmetry321@cloudshell:~ (lab02-distributed)$


michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl create -f web-deployment.yaml
deployment.apps/web-deployment created
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl expose deployment web-deployment --port=80 --type=ClusterIP --name web-deployment
service/web-deployment exposed
michaelmetry321@cloudshell:~ (lab02-distributed)$


michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl create -f web-deployment.yaml
deployment.apps/web-deployment created
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl expose deployment web-deployment --port=80 --type=ClusterIP --name web-deployment
service/web-deployment exposed
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl create -f experiment-deployment.yaml
deployment.apps/experiment-deployment created
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl expose deployment experiment-deployment --port=80 --type=ClusterIP --name experiment-deployment
service/experiment-deployment exposed
michaelmetry321@cloudshell:~ (lab02-distributed)$

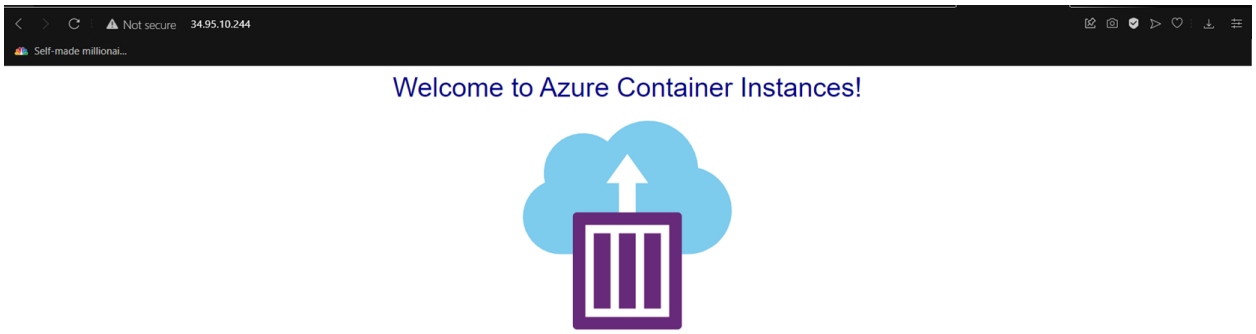

Google Cloud
Search Products, resources, docs (/)
CLOUD SHELL
Editor
Open Terminal
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
MICHAELMETRY321
conf.d
nginx-ambassador.conf
experiment-deployment.yaml
README-cloudshell.txt
web-deployment.yaml
README-cloudshell.txt
web-deployment.yaml
experiment-deployment.yaml
nginx-ambassador.conf
web-deployment.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 name: web-deployment
5 spec:
6 replicas: 2
7 selector:
8 matchLabels:
9 run: web-deployment
10 template:
11 metadata:
12 labels:
13 run: web-deployment
14 spec:
15 containers:
16 - name: web-depoyment
17 image: mcr.microsoft.com/azuredocs/aci-helloworld
18 ports:
19 - containerPort: 80
0 0 Cloud Code minikube
Ln 1, Col 1 LF UTF-8 Spaces: 2 YAML 1


Google Cloud
Search Products, resources, docs (/)
CLOUD SHELL
Editor
Open Terminal
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
MICHAELMETRY321
conf.d
nginx-ambassador.conf
experiment-deployment.yaml
README-cloudshell.txt
web-deployment.yaml
README-cloudshell.txt
web-deployment.yaml
experiment-deployment.yaml
nginx-ambassador.conf
experiment-deployment.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 name: experiment-deployment
5 spec:
6 replicas: 2
7 selector:
8 matchLabels:
9 run: experiment-deployment
10 template:
11 metadata:
12 labels:
13 run: experiment-deployment
14 spec:
15 containers:
16 - name: experiment-depoyment
17 image: mcr.microsoft.com/azuredocs/aci-helloworld
18 ports:
19 - containerPort: 80
0 0 Cloud Code minikube
Ln 1, Col 1 LF UTF-8 Spaces: 2 YAML 1




```
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl create configmap ambassador-config --from-file=conf.d
configmap/ambassador-config created
michaelmetry321@cloudshell:~ (lab02-distributed)$
```

```
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl create configmap ambassador-config --from-file=conf.d
configmap/ambassador-config created
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl create -f ambassador-deployment.yaml
deployment.apps/ambassador-deployment created
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl expose deployment ambassador-deployment --port=80 --type=LoadBalancer
service/ambassador-deployment exposed
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
ambassador-deployment-66db4f7766-6472g  1/1     Running   0          112s
ambassador-deployment-66db4f7766-qgxn7  1/1     Running   0          112s
experiment-deployment-7b47cbd668-8j6xt  1/1     Running   0          30m
experiment-deployment-7b47cbd668-kkqzr  1/1     Running   0          30m
web-deployment-6fdbb5c6bb-qck78         1/1     Running   0          37m
web-deployment-6fdbb5c6bb-vmx7s         1/1     Running   0          37m
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get deployments
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
ambassador-deployment   2/2     2            2           2m2s
experiment-deployment   2/2     2            2           30m
web-deployment          2/2     2            2           37m
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get services
NAME                    TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
ambassador-deployment   LoadBalancer   10.124.7.197    <pending>     80:32681/TCP   36s
experiment-deployment   ClusterIP      10.124.6.36     <none>        80/TCP         30m
kubernetes              ClusterIP      10.124.0.1      <none>        443/TCP        41m
web-deployment          ClusterIP      10.124.5.223    <none>        80/TCP         33m
michaelmetry321@cloudshell:~ (lab02-distributed)$
```




```
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
ambassador-deployment-66db4f7766-6472g  1/1     Running   0          15m
ambassador-deployment-66db4f7766-qgxn7  1/1     Running   0          15m
experiment-deployment-7b47cbd668-8j6xt  1/1     Running   0          44m
experiment-deployment-7b47cbd668-kkqzr  1/1     Running   0          44m
web-deployment-6fdbb5c6bb-qck78         1/1     Running   0          51m
web-deployment-6fdbb5c6bb-vmx7s         1/1     Running   0          51m
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get deployments
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
ambassador-deployment  2/2     2            2           15m
experiment-deployment  2/2     2            2           44m
web-deployment         2/2     2            2           51m
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get services
NAME                   TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
ambassador-deployment  LoadBalancer   10.124.7.197   34.95.10.244   80:32681/TCP   14m
experiment-deployment  ClusterIP      10.124.6.36    <none>         80/TCP         44m
kubernetes             ClusterIP      10.124.0.1     <none>         443/TCP        55m
web-deployment         ClusterIP      10.124.5.223   <none>         80/TCP         46m
```

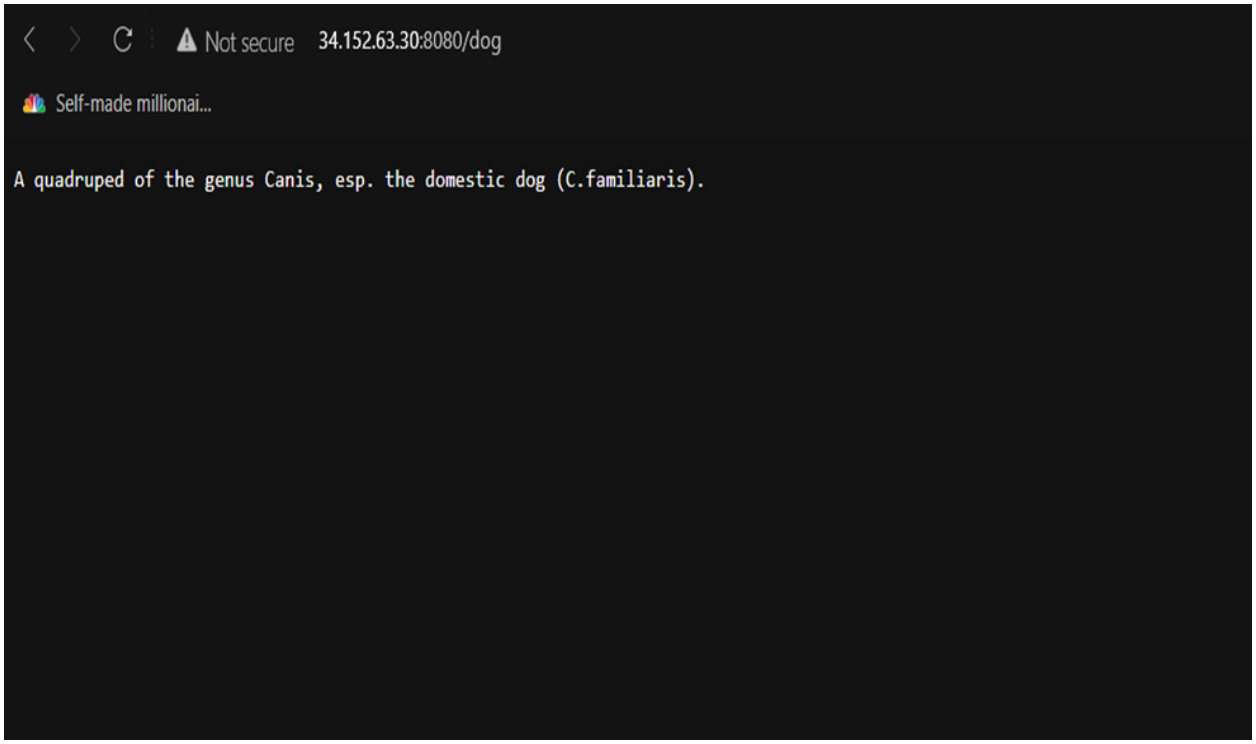
```
michaelmetry321@cloudshell:~ (lab02-distributed)$ curl http://34.95.10.244
<html>
<head>
  <title>Welcome to Azure Container Instances!</title>
</head>
<style>
h1 {
    color: darkblue;
    font-family:arial, sans-serif;
    font-weight: lighter;
}
</style>

<body>

<div align="center">
<h1>Welcome to Azure Container Instances!</h1>

<svg id="Layer_1" data-name="Layer 1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 49.8 49.9" width="250px" height="250px">
  <title>ContainerInstances_rgb_UI</title>
  <path d="M41.9,11.368A11.929,11.929,0,0,0,20.3,5.061a9.444,9.444,0,0,0-14.932,9.8A8.969,8.969,0,0,0,9.064,32H39.442A10.463,10.463,0,0,0,41.9,11.368Z" transform="translate(-0.1 -0.1)" fi
ll="#fff"/>
        <path d="M41.9,11.368A11.929,11.929,0,0,0,20.3,5.061a9.444,9.444,0,0,0-14.932,9.8A8.969,8.969,0,0,0,9.064,32H39.442A10.463,10.463,0,0,0,41.9,11.368Z" transform="translate(-0.1 -0.
1)" fill="#27a9e1" opacity="0.6" style="isolation:isolate"/>
        <path d="M13,22a1,1,0,0,0-1,1V49a1,1,0,0,0,1,1H37a1,1,0,0,0,1-1V23a1,1,0,0,0-1-1Z" transform="translate(-0.1 -0.1)" fill="#672a7a"/>
        <path d="M26.95,16" transform="translate(-0.1 -0.1)" fill="none"/>
        <path d="M34.95,20" transform="translate(-0.1 -0.1)" fill="none"/>
        <polygon points="22.9 21.9 22.9 14.9 19.9 14.9 24.9 7.9 29.9 14.9 26.9 14.9 26.9 21.9 22.9 21.9" fill="#fff"/>
        <path d="M26.95,16" transform="translate(-0.1 -0.1)" fill="#814a98"/>
        <path d="M33,25H15V47H35V25ZM21,45H17V27h4Zm6,0H23V27h4Zm6,0H29V27h4Z" transform="translate(-0.1 -0.1)" fill="#b92025" opacity="0.3" style="isolation:isolate"/>
        <path d="M33,25H15V47H35V25ZM21,45H17V27h4Zm6,0H23V27h4Zm6,0H29V27h4Z" transform="translate(-0.1 -0.1)" fill="#fff" style="isolation:isolate"/>
</svg>
</div>

</body>
</html>
michaelmetry321@cloudshell:~ (lab02-distributed)$
```

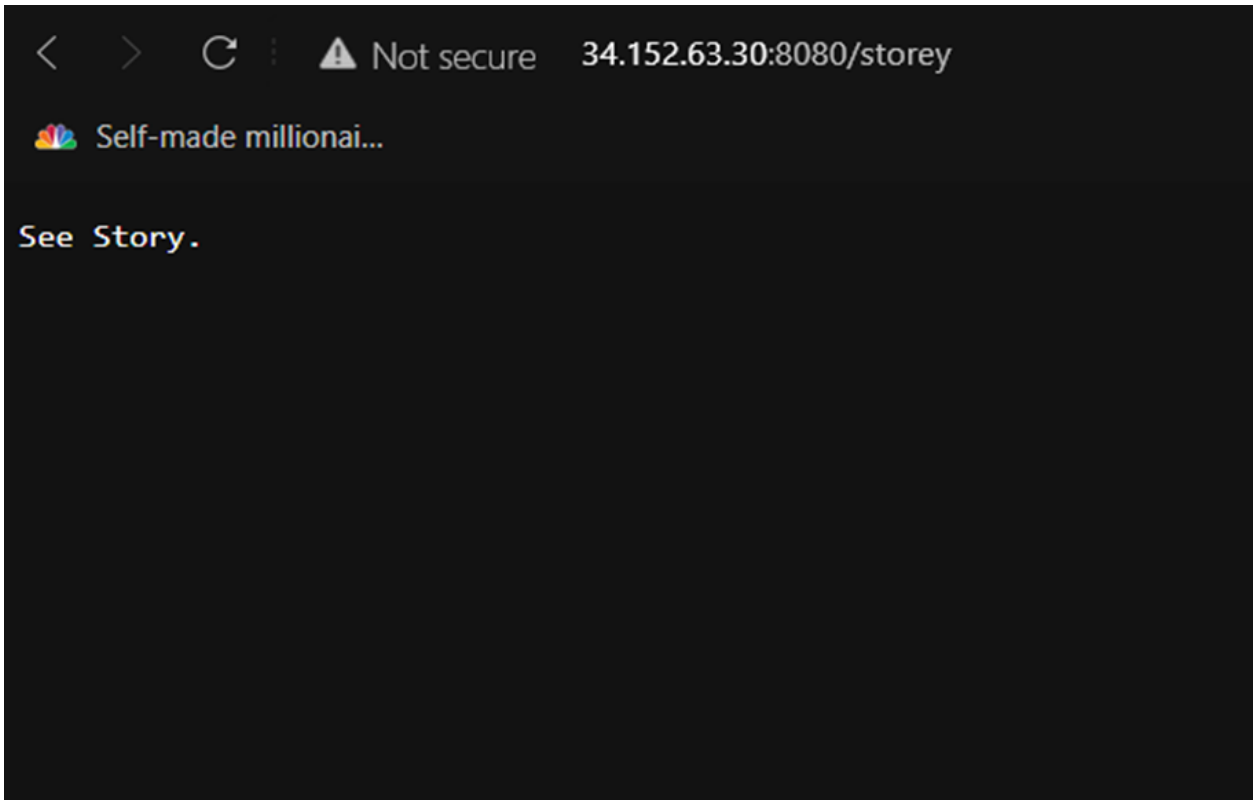
```
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl logs -l run=web-deployment
listening on port 80
::ffff:10.120.0.7 - - [04/Oct/2022:19:03:25 +0000] "GET / HTTP/1.0" 200 1663 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 Sa
fari/537.36 OPR/90.0.4480.117"
::ffff:10.120.0.7 - - [04/Oct/2022:19:03:26 +0000] "GET /favicon.ico HTTP/1.0" 404 150 "http://34.95.10.244/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
cko) Chrome/104.0.5112.102 Safari/537.36 OPR/90.0.4480.117"
::ffff:10.120.0.7 - - [04/Oct/2022:19:03:37 +0000] "GET / HTTP/1.0" 200 1663 "-" "curl/7.74.0"
listening on port 80
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl logs -l run=experiment-deployment
listening on port 80
listening on port 80
michaelmetry321@cloudshell:~ (lab02-distributed)$
```

Part 3:

```
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl create -f loadbalancer-deployment.yaml
deployment.apps/loadbalancer-deployment created
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get pods --output=wide
NAME                                     READY   STATUS    RESTARTS   AGE   IP           NODE                                          NOMINATED NODE   READINESS GATES
ambassador-deployment-66db4f7766-6472g   1/1     Running   0          25m   10.120.2.7   gke-openfass-default-pool-ada9681e-wb8s   <none>           <none>
ambassador-deployment-66db4f7766-qgxn7   1/1     Running   0          25m   10.120.0.7   gke-openfass-default-pool-ada9681e-ccpp   <none>           <none>
experiment-deployment-7b47cbd668-8j6xt   1/1     Running   0          54m   10.120.0.6   gke-openfass-default-pool-ada9681e-ccpp   <none>           <none>
experiment-deployment-7b47cbd668-kkqzr   1/1     Running   0          54m   10.120.1.6   gke-openfass-default-pool-ada9681e-g5cr   <none>           <none>
loadbalancer-deployment-6676f9ccf6-bs2f9 1/1     Running   0          25s   10.120.2.8   gke-openfass-default-pool-ada9681e-wb8s   <none>           <none>
loadbalancer-deployment-6676f9ccf6-sxv8h 1/1     Running   0          25s   10.120.1.7   gke-openfass-default-pool-ada9681e-g5cr   <none>           <none>
loadbalancer-deployment-6676f9ccf6-wrv4x 1/1     Running   0          25s   10.120.0.8   gke-openfass-default-pool-ada9681e-ccpp   <none>           <none>
web-deployment-6fdbb5c6bb-qck78          1/1     Running   0          60m   10.120.0.5   gke-openfass-default-pool-ada9681e-ccpp   <none>           <none>
web-deployment-6fdbb5c6bb-vmx7s          1/1     Running   0          60m   10.120.2.6   gke-openfass-default-pool-ada9681e-wb8s   <none>           <none>
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl expose deployment loadbalancer-deployment --port=8080 --type=LoadBalancer
service/loadbalancer-deployment exposed
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get services --watch
NAME                      TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE
ambassador-deployment     LoadBalancer   10.124.7.197    34.95.10.244   80:32681/TCP     24m
experiment-deployment     ClusterIP      10.124.6.36     <none>         80/TCP           54m
kubernetes                ClusterIP      10.124.0.1      <none>         443/TCP          66m
loadbalancer-deployment   LoadBalancer   10.124.11.158   <pending>      8080:30721/TCP   12s
web-deployment            ClusterIP      10.124.5.223    <none>         80/TCP           57m
loadbalancer-deployment   LoadBalancer   10.124.11.158   34.152.63.30   8080:30721/TCP   38s
```

```
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get services --watch
NAME                      TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE
ambassador-deployment     LoadBalancer   10.124.7.197    34.95.10.244   80:32681/TCP     27m
experiment-deployment     ClusterIP      10.124.6.36     <none>         80/TCP           57m
kubernetes                ClusterIP      10.124.0.1      <none>         443/TCP          68m
loadbalancer-deployment   LoadBalancer   10.124.11.158   34.152.63.30   8080:30721/TCP   3m
web-deployment            ClusterIP      10.124.5.223    <none>         80/TCP           60m
^Z
[4]+  Stopped                 kubectl get services --watch
michaelmetry321@cloudshell:~ (lab02-distributed)$ kubectl get services
NAME                      TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE
ambassador-deployment     LoadBalancer   10.124.7.197    34.95.10.244   80:32681/TCP     27m
experiment-deployment     ClusterIP      10.124.6.36     <none>         80/TCP           57m
kubernetes                ClusterIP      10.124.0.1      <none>         443/TCP          69m
loadbalancer-deployment   LoadBalancer   10.124.11.158   34.152.63.30   8080:30721/TCP   3m19s
web-deployment            ClusterIP      10.124.5.223    <none>         80/TCP           60m
michaelmetry321@cloudshell:~ (lab02-distributed)$ curl http://34.152.63.30:8080/dog
A quadruped of the genus Canis, esp. the domestic dog (C.familiaris).michaelmetry321@cloudshell:~ (lab02-distributed)$ curl http://34.152.63.30:8080/storey
See Story.michaelmetry321@cloudshell:~ (lab02-distributed)$
```


Not secure
34.152.63.30:8080/dog
Self-made millionai...
A quadruped of the genus Canis, esp. the domestic dog (C.familiaris).


Not secure
34.152.63.30:8080/storey
Self-made millionai...
See Story.

Discussion:

https://learn.microsoft.com/en-us/azure/architecture/patterns/gateway-routing

Summary of Problem in Part 1:

- Configuration of multiple endpoints for multiple backend services. When the API changes, the client must be modified as well. When refactoring a service and making separate services, the code must be changed for the service and the client.

Solution in Part 1:

- The solution to the design pattern is to place a gateway in front of a set of applications, services, or deployments. Application Layer 7 is then used to route the request to the appropriate instances.
- If a service is consolidated, the client isn't required to update and it can continue to make requests to the gateway and routing changes.
- Incorporating the gateway design pattern will allow multiple services on a gateway that can only be exposed to the client on a single endpoint and routing external addresses to internal endpoints.
- Ensure that multiple services are available so that these services that split and manage all communications in that gateway

Requirements needed Pattern in Part 1?

- Services
- API
- Single Endpoint

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

- In part 2, two deployments were used under the same web page in one cluster.
- Part 3, load balancer is deploying 3 pods of an image and extra pod for transition into the next instruction.
- Requires services, API, and endpoints

Design

Why is auto-scaling used?

- Autoscaling is used for optimal utilization of resources in an upward/downward direction based on traffic or demand. It is to ensure that the cloud services costs are reduced (use required resources) in a way businesses utilize those resources for their tasks. Autoscaling allows the resources to automatically meet the demands which assist in reduced costs and helps to retain service availability

How is autoscaling implemented?

- Automatically scale up and down clusters on demands and resources
- Horizontal Pod Autoscaler: Pods will increase based on the increase of load. The scalar will then automatically arrange the workload resources to scale down if the load drops
- Vertical Pod Autoscaler: Modify resources of components such as CPU, and RAM of each node in the cluster

How is auto-scaling different from load balancing and request splitter?

- Used for scaling up and down instances
- Load balancing is used to distribute traffic calls across various services.
- Request splitter is used to divide incoming requests into multiple segments and each component is processed separately.

Video Link 1:
https://drive.google.com/file/d/1J1t6go7kqm1k9Mprr27jlJh0J8mc9TWd/view?usp=sharing

Video Link 2:
https://drive.google.com/file/d/1MRQTxbapxAZ4jC7njOImmRJBNRvPVBlF/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: 06-Oct-2022].

[2] “Horizontal pod autoscaling,” *Kubernetes*, 10-Jun-2022. [Online]. Available: https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/. [Accessed: 07-Oct-2022].