Yogeshwaran

Case study 10

Module 10: GCP DevOps Services

Step 1

She have to enable Compute Engine API

```
(qwiklabs-gcp-03-86dbc49ca6c3)
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud services enable compute.googleapis.com
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 2

And she have to create a bucket

```
Cloud Shell Editor
(qwiklabs-gcp-03-86dbc49ca6c3)
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gsutil mb gs://fancy-store-$DEVSHELL_PROJECT_ID
Creating gs://fancy-store-qwiklabs-gcp-03-86dbc49ca6c3/...
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 3

Clone the application from github

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gsutil mb gs://fancy-store-$DEVSHELL_PROJECT_ID
Creating gs://fancy-store-qwiklabs-gcp-03-86dbc49ca6c3/...
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ git clone https://github.com/googlecodelabs/monolith-to-microservices.git
Cloning into 'monolith-to-microservices'...
remote: Enumerating objects: 952, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 952 (delta 9), reused 0 (delta 0), pack-reused 936
Receiving objects: 100% (952/952), 2.81 MiB | 2.77 MiB/s, done.
Resolving deltas: 100% (410/410), done.
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 4

Execute the startup script to run the application locally

```
student_03_76b8a7cfbeb8@cloudshell:~/monolith-to-microservices (qwiklabs-gcp-03-86dbc49ca6c3)$ ls
CONTRIBUTING.md  deploy-monolith.sh  LICENSE  logs  microservices  monolith  package-lock.json  react-app  README.md  setup.sh
student_03_76b8a7cfbeb8@cloudshell:~/monolith-to-microservices (qwiklabs-gcp-03-86dbc49ca6c3)$ ./ setup.sh
-bash: ./: Is a directory
student_03_76b8a7cfbeb8@cloudshell:~/monolith-to-microservices (qwiklabs-gcp-03-86dbc49ca6c3)$ ./setup.sh
Checking for required npm version...Completed.
Installing monolith dependencies...Completed.
Installing microservies dependencies...Completed.
Installing React app dependencies...Completed.
Building React app and placing into sub projects...
```

Step 5

Now open to microservices and start the web server

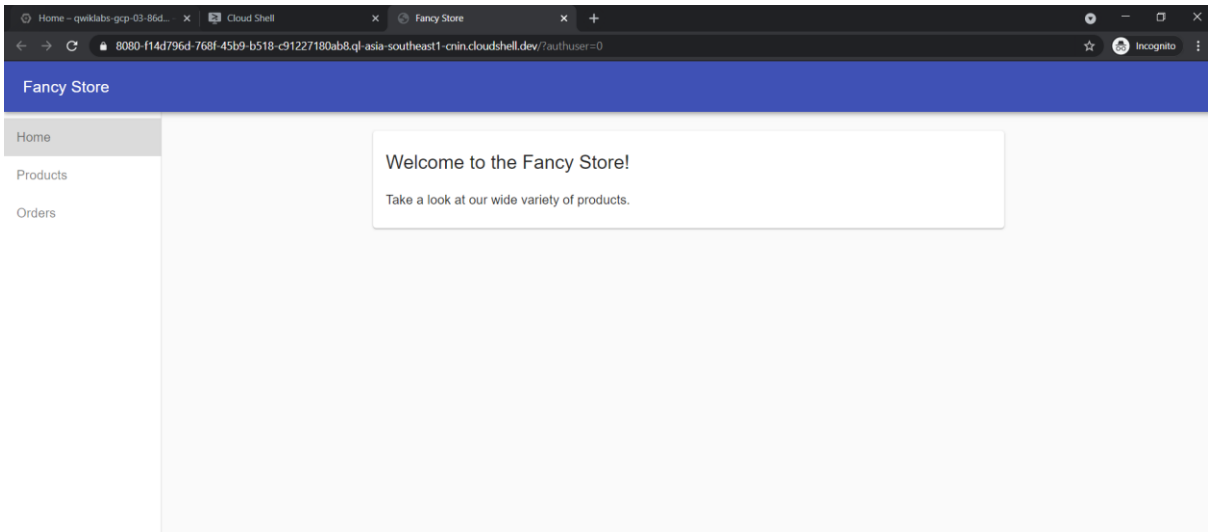
```
Script completed successfully!
student_03_76b8a7cfbeb8@cloudshell:~/monolith-to-microservices (qwiklabs-gcp-03-86dbc49ca6c3)$ cd microservices
student_03_76b8a7cfbeb8@cloudshell:~/monolith-to-microservices/microservices (qwiklabs-gcp-03-86dbc49ca6c3)$ npm start

> microservices@1.0.0 start
> concurrently "npm run frontend" "npm run products" "npm run orders"

[2]
[2] > microservices@1.0.0 orders
[2] > node ./src/orders/server.js
[2]
[2] Orders microservice listening on port 8081!
[0]
[0] > microservices@1.0.0 frontend
[0] > node ./src/frontend/server.js
[0]
[1]
[1] > microservices@1.0.0 products
[1] > node ./src/products/server.js
[1]
[0] Frontend microservice listening on port 8080!
[1] Products microservice listening on port 8082!
```

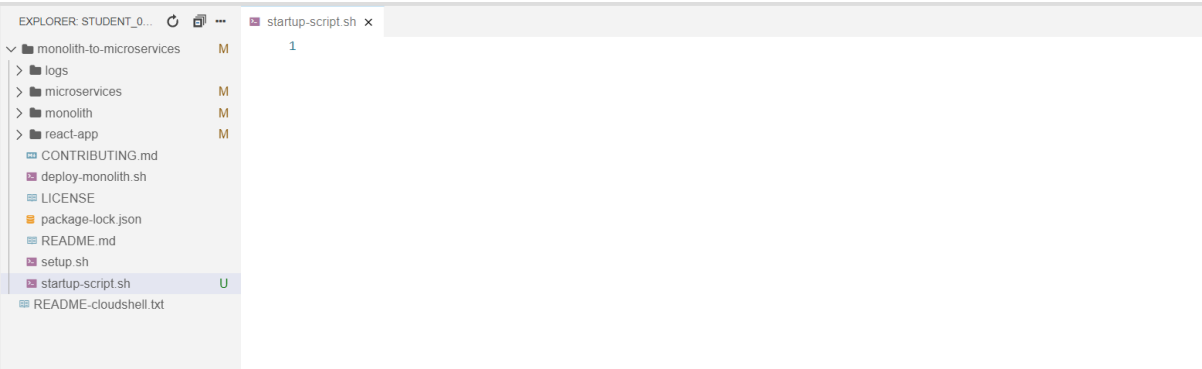
Step 6

Check the port of the website, and check it is running or not



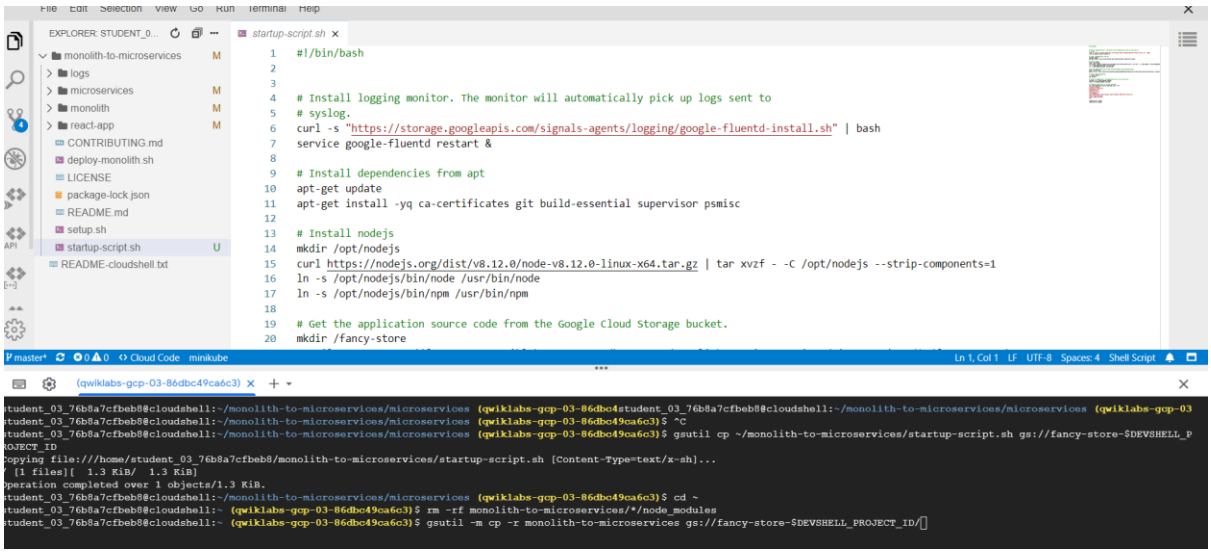
Step 7

She have to create new start-up script so A startup script will be used to instruct the instance what to do each time it is started. This way the instances are automatically configured.



Step 8

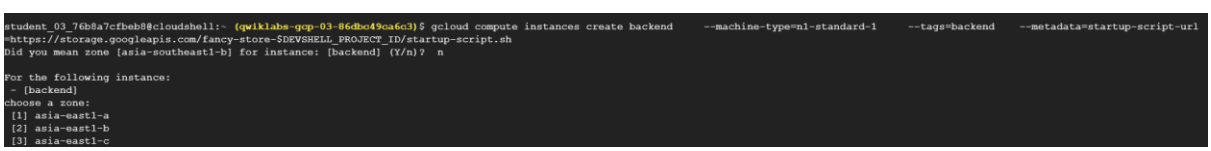
After copy the startup script to the bucket, Copy the cloned code into same bucket



Step 9

Deploy backend instance

The first instance to be deployed will be the backend instance which will house the Orders and Products microservices, with firewall rules allow http



Step 10

Change the env file local host into external ip of backend instances, so that we can make it live on internet

Step 11

After the change we must rebuild the application

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instances list
NAME     ZONE            MACHINE_TYPE   PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP     STATUS
backend  us-central1-f   n1-standard-1               10.128.0.3   35.224.121.151  RUNNING
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ ^C
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ cd ~/monolith-to-microservices/react-app
student_03_76b8a7cfbeb8@cloudshell:~/monolith-to-microservices/react-app (qwiklabs-gcp-03-86dbc49ca6c3)$ npm install && npm run-script build
[        ........] \ reify:fsevents: timing reifyNode:node_modules/jest-haste-map/node_modules/fsevents Completed in 321ms
```

Step 12

Now we have to Deploy frontend instance

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instances create frontend \
>     --machine-type=n1-standard-1 \
>     --tags=frontend \
>     --metadata=startup-script-url=https://storage.googleapis.com/fancy-store-$DEVSHELL_PROJECT_ID/startup-script.sh
Did you mean zone [asia-southeast1-b] for instance: [frontend] (Y/n)?
  n
```

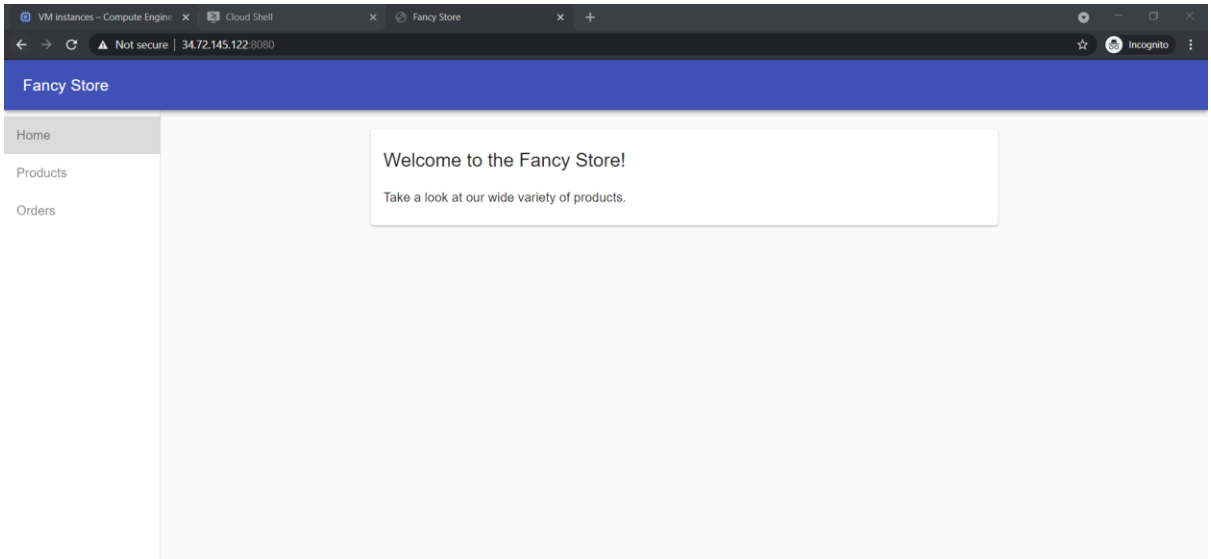
Step 13

Create firewall rules to allow access to port 8080 for the frontend, and ports 8081-8082 for the backend. These firewall commands use the tags assigned during instance creation for application

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute firewall-rules create fw-fe \
>     --allow tcp:8080 \
>     --target-tags=frontend
Creating firewall...⠶Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/firewalls/fw-fe].
Creating firewall...done.
NAME   NETWORK  DIRECTION  PRIORITY  ALLOW     DENY  DISABLED
fw-fe  default  INGRESS    1000      tcp:8080        False
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute firewall-rules create fw-be \
>     --allow tcp:8081-8082 \
>     --target-tags=backend
Creating firewall...⠶Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/firewalls/fw-be].
Creating firewall...done.
NAME   NETWORK  DIRECTION  PRIORITY  ALLOW          DENY  DISABLED
fw-be  default  INGRESS    1000      tcp:8081-8082        False
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 14

After redirect the external ip for front end instances to out application we can see the application running on our front-end instances



Step 15

Create Managed Instance Groups

To allow the application to scale, managed instance groups will be created and will use the frontend and backend instances as Instance Templates.

So we must Create Instance Template from Source Instance, so here we have to create template for both front end and back end

```
(qwiklabs-gcp-03-86dbc49ca6c3)
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-templates create fancy-fe \
>     --source-instance=frontend
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/instanceTemplates/fancy-fe].
NAME      MACHINE_TYPE   PREEMPTIBLE  CREATION_TIMESTAMP
fancy-fe  n1-standard-1               2021-04-24T10:51:17.196-07:00
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-templates create fancy-be \
>     --source-instance=backend
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/instanceTemplates/fancy-be].
NAME      MACHINE_TYPE   PREEMPTIBLE  CREATION_TIMESTAMP
fancy-be  n1-standard-1               2021-04-24T10:51:27.950-07:00
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 16

Create managed instance group

Next, create two managed instance groups, one for the frontend and one for the backend:

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed create fancy-fe-mig \
>     --base-instance-name fancy-fe \
>     --size 2 \
>     --template fancy-fe
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroupManagers/fancy-fe-mig].
NAME          LOCATION       SCOPE  BASE_INSTANCE_NAME  SIZE  TARGET_SIZE  INSTANCE_TEMPLATE  AUTOSCALED
fancy-fe-mig  us-central1-f  zone   fancy-fe            0     2            fancy-fe           no
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed create fancy-be-mig \
>     --base-instance-name fancy-be \
>     --size 2 \
>     --template fancy-be
```

Step 17

For application, the frontend microservice runs on port 8080, and the backend microservice runs on port 8081 for orders and port 8082 for products

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups set-named-ports fancy-fe-mig \
>     --named-ports frontend:8080
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroups/fancy-fe-mig].
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups set-named-ports fancy-be-mig \
>     --named-ports orders:8081,products:8082
```

Step 18

To improve the availability of the application itself and to verify it is responding, configure an autohealing policy for the managed instance groups.

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute health-checks create http fancy-fe-hc \
>     --port 8080 \
>     --check-interval 30s \
>     --healthy-threshold 1 \
>     --timeout 10s \
>     --unhealthy-threshold 3
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/healthChecks/fancy-fe-hc].
NAME         PROTOCOL
fancy-fe-hc  HTTP
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute health-checks create http fancy-be-hc \
>     --port 8081 \
>     --request-path=/api/orders \
>     --check-interval 30s \
>     --healthy-threshold 1 \
>     --timeout 10s \
>     --unhealthy-threshold 3
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/healthChecks/fancy-be-hc].
NAME         PROTOCOL
fancy-be-hc  HTTP
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 19

Create a firewall rule to allow the health check probes to connect to the microservices on ports 8080-8081

```
fancy-be-hc  HTTP
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute firewall-rules create allow-health-check \
>     --allow tcp:8080-8081 \
>     --source-ranges 130.211.0.0/22,35.191.0.0/16 \
>     --network default
Creating firewall...⠶Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/firewalls/allow-health-check].
Creating firewall...done.
NAME                 NETWORK  DIRECTION  PRIORITY  ALLOW          DENY  DISABLED
allow-health-check   default  INGRESS    1000      tcp:8080-8081        False
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed update fancy-fe-mig \
>     --health-check fancy-fe-hc \
>     --initial-delay 300
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroupManagers/fancy-fe-mig].
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed update fancy-be-mig \
>     --health-check fancy-be-hc \
>     --initial-delay 300
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroupManagers/fancy-be-mig].
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 20

Create http load balancer

Create health checks that will be used to determine which instances are capable of serving traffic for each service

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute http-health-checks create fancy-fe-frontend-hc \
>   --request-path / \
>   --port 8080
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/httpHealthChecks/fancy-fe-frontend-hc].
NAME                 HOST  PORT  REQUEST_PATH
fancy-fe-frontend-hc       8080  /
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute http-health-checks create fancy-be-orders-hc \
>   --request-path /api/orders \
>   --port 8081
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/httpHealthChecks/fancy-be-orders-hc].
NAME                 HOST  PORT  REQUEST_PATH
fancy-be-orders-hc         8081  /api/orders
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute http-health-checks create fancy-be-products-hc \
>   --request-path /api/products \
>   --port 8082
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/httpHealthChecks/fancy-be-products-hc].
NAME                 HOST  PORT  REQUEST_PATH
fancy-be-products-hc       8082  /api/products
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 21

Create backend services that are the target for load-balanced traffic. The backend services will use the health checks and named ports you created

```
>
Cr[illegible]/www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/httpHealthChecks/fancy-be-products-hc].
NAME                  HOST  PORT  REQUEST_PATH
fancy-be-products-hc        8082  /api/products
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute backend-services create fancy-fe-frontend \
>   --http-health-checks fancy-fe-frontend-hc \
>   --port-name frontend \
>   --global
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/backendServices/fancy-fe-frontend].
NAME                BACKENDS  PROTOCOL
fancy-fe-frontend             HTTP
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute backend-services create fancy-be-orders \
>   --http-health-checks fancy-be-orders-hc \
>   --port-name orders \
>   --global
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/backendServices/fancy-be-orders].
NAME              BACKENDS  PROTOCOL
fancy-be-orders             HTTP
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute backend-services create fancy-be-products \
>   --http-health-checks fancy-be-products-hc \
>   --port-name products \
>   --global
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/backendServices/fancy-be-products].
NAME                BACKENDS  PROTOCOL
fancy-be-products             HTTP
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute backend-services add-backend fancy-fe-frontend \
>   --instance-group fancy-fe-mig \
>   --instance-group-zone us-central1-f \
>   --global
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/backendServices/fancy-fe-frontend].
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute backend-services add-backend fancy-be-orders \
>   --instance-group fancy-be-mig \
>   --instance-group-zone us-central1-f \
>   --global
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/backendServices/fancy-be-orders].
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute backend-services add-backend fancy-be-products \
>   --instance-group fancy-be-mig \
>   --instance-group-zone us-central1-f \
>   --global
```

Step 22

Create a url map, The URL map defines which URLs are directed to which backend services

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute url-maps create fancy-map \
>   --default-service fancy-fe-frontend
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/urlMaps/fancy-map].
NAME       DEFAULT_SERVICE
fancy-map  backendServices/fancy-fe-frontend
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 23

Create a path matcher to allow the /api/orders and /api/products paths to route to their respective services:

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute url-maps add-path-matcher fancy-map \
>    --default-service fancy-fe-frontend \
>    --path-matcher-name orders \
>    --path-rules "/api/orders=fancy-be-orders,/api/products=fancy-be-products"
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/urlMaps/fancy-map].
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 24

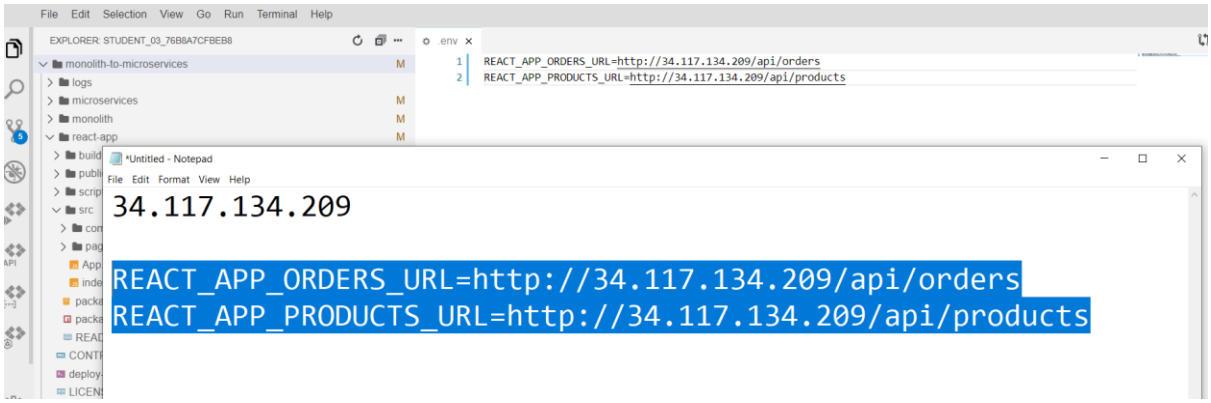
Create the proxy which ties to the URL map

```
tudent_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute target-http-proxies create fancy-proxy \
    --url-map fancy-map
reated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/targetHttpProxies/fancy-proxy].
AME          URL_MAP
ancy-proxy  fancy-map
tudent_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute forwarding-rules create fancy-http-rule \
    --global \
    --target-http-proxy fancy-proxy \
    --ports 80
```

Step 25

Update Configuration

new static IP address, update the code on the frontend to point to this new address instead of the ephemeral address used earlier that pointed to the backend instance.



Step 26

Rebuild the application with load balencer ip

```
student_03_76b8a7cfbeb8@cloudshell:~/monolith-to-microservices/react-app (qwiklabs-gcp-03-86dbc49ca6c3)$ cd ~/monolith-to-microservices/react-app
student_03_76b8a7cfbeb8@cloudshell:~/monolith-to-microservices/react-app (qwiklabs-gcp-03-86dbc49ca6c3)$ npm install && npm run-script build
[                    ........] \ reify:fsevents: timing reifyNode:node_modules/jest-haste-map/node_modules/fsevents Completed in 333ms
```

Step 27

Update the frontend instances

Now that there is new code and configuration, we want the frontend instances within the managed instance group to pull the new code. Since our instances pull the code at startup

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed rolling-action replace fancy-fe-mig \
>     --max-unavailable 100%
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroupManagers/fancy-fe-mig].
---
autoHealingPolicies:
- healthCheck: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/healthChecks/fancy-fe-hc
  initialDelaySec: 300
baseInstanceName: fancy-fe
creationTimestamp: '2021-04-24T10:53:40.966-07:00'
currentActions:
  abandoning: 0
  creating: 0
  creatingWithoutRetries: 0
  deleting: 0
  none: 2
```

Step 28

We have to test the application

```
Every 2.0s: gcloud compute backend-services get-health fancy-fe-frontend --global          cs-856802657058-default-default-ht4j8: Sat Apr 24 18:16:04 2021

---
backend: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroups/fancy-fe-mig
status:
  healthStatus:
  - healthState: HEALTHY
    instance: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instances/fancy-fe-c8x0
    ipAddress: 10.128.0.9
    port: 8080
  - healthState: HEALTHY
    instance: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instances/fancy-fe-v53t
    ipAddress: 10.128.0.10
    port: 8080
  kind: compute#backendServiceGroupHealth
```

Our instances are healthy

Step 29

Scaling Compute Engine

Now will create an autoscaling policy based on utilization to automatically scale each managed instance group.

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed set-autoscaling \
>   fancy-fe-mig \
>   --max-num-replicas 2 \
>   --target-load-balancing-utilization 0.60
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/autoscalers/fancy-fe-mig-2216].
---
autoscalingPolicy:
  coolDownPeriodSec: 60
  loadBalancingUtilization:
    utilizationTarget: 0.6
  maxNumReplicas: 2
  minNumReplicas: 2
  mode: ON
creationTimestamp: '2021-04-24T11:17:42.063-07:00'
id: '7538955436434251305'
kind: compute#autoscaler
name: fancy-fe-mig-2216
selfLink: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/autoscalers/fancy-fe-mig-2216
status: ACTIVE
target: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroupManagers/fancy-fe-mig
zone: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed set-autoscaling \
>   fancy-be-mig \
>   --max-num-replicas 2 \
>   --target-load-balancing-utilization 0.60
```

These commands create an autoscaler on the managed instance groups that automatically adds instances when utilization is above 60% utilization, and removes instances when the load balancer is below 60% utilization.

Step 30

Enable Content Delivery Network and Updating Instance Template

```
namedPorts:
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed set-autoscaling \
>   fancy-fe-mig \
>   --max-num-replicas 2 \
>   --target-load-balancing-utilization 0.60
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/autoscalers/fancy-fe-mig-2216].
---
autoscalingPolicy:
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instances set-machine-type frontend --machine-type custom-4-3840
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instances/frontend].
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-templates create fancy-fe-new \
>     --source-instance=frontend \
>     --source-instance-zone=us-central1-f
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/instanceTemplates/fancy-fe-new].
NAME          MACHINE_TYPE   PREEMPTIBLE  CREATION_TIMESTAMP
fancy-fe-new  custom-4-3840               2021-04-24T11:19:43.757-07:00
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instance-groups managed rolling-action start-update fancy-fe-mig \
>     --version template=fancy-fe-new
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroupManagers/fancy-fe-mig].
---
autoHealingPolicies:
- healthCheck: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/global/healthChecks/fancy-fe-hc
  initialDelaySec: 300
baseInstanceName: fancy-fe
creationTimestamp: '2021-04-24T10:53:40.966-07:00'
```

Step 31

Find the description of a one healthy vm

```
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$ gcloud compute instances describe fancy-fe-q5sg | grep machineType
machineType: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/machineTypes/custom-4-3840
student_03_76b8a7cfbeb8@cloudshell:~ (qwiklabs-gcp-03-86dbc49ca6c3)$
```

Step 32

Everything running successfully

```
Every 2.0s: gcloud compute backend-services get-health fancy-fe-frontend --global                                    cs-856802657058-defa

---
backend: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instanceGroups/fancy-fe-mig
status:
  healthStatus:
  - healthState: UNHEALTHY
    instance: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-86dbc49ca6c3/zones/us-central1-f/instances/fancy-fe-5fvh
    ipAddress: 10.128.0.13
    port: 8080
  kind: compute#backendServiceGroupHealth
```

I complete this assignment from this lab

End Lab 00:05:29

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Console

Username

student-03-76b8a7cfbeb8@

Password

YvBgKP97Nt

GCP Project ID

qwiklabs-gcp-03-86dbc49c

Congratulations!

You successfully deployed, scaled, and updated your website on C
You are now experienced with Compute Engine, Managed Instance
Balancers, and Health Checks!

Finish Your Quest

This self-paced lab is part of the Qwiklabs Website on Google Clou
is a series of related labs that form a learning path. Enroll in this Q
immediate completion credit if you've taken this lab. See other ava
Quests.

Looking for a hands-on challenge lab to demonstrate your skills ar
knowledge? On completing this quest, finish this additional challer
an exclusive Google Cloud digital badge.

Checkpoints

Create GCS bucket	Check my progress	10 / 10
Copy startup script and code to Cloud Storage bucket	Check my progress	10 / 10
Deploy instances and configure network	Check my progress	20 / 20
Create managed instance groups	Check my progress	20 / 20
Create HTTP(S) load balancers	Check my progress	10 / 10
Update the frontend instances	Check my progress	10 / 10
Scaling GCE	Check my progress	10 / 10
Update the website	Check my progress	10 / 10