Build a Website on Google Cloud: Challenge Lab

Author: Vedant Kakde | **GitHub Profile:** github.com/vedant-kakde | **LinkedIn Profile:** linkedin.com/in/vedant-kakde/

Task 1: Download the monolith code and build your container

Open the cloud shell and start typing to set up the environment :

```
git clone https://github.com/googlecodelabs/monolith-to-microservices.gitcd ~/monolith-to-microservices./setup.shcd ~/monolith-to-microservices/monolith
```

```
gcloud services enable cloudbuild.googleapis.comgcloud builds submit — tag gcr.io/${GOOGLE_CLOUD_PROJECT}/fancytest:1.0.0 .
```

Task 2: Create a kubernetes cluster and deploy the application

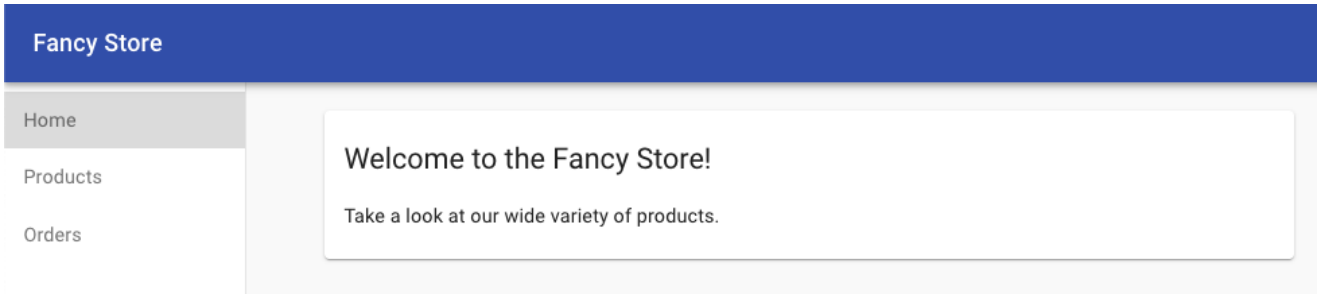
Create your cluster as follows:

```
gcloud services enable container.googleapis.com
```

```
gcloud container clusters create fancy-cluster --num-nodes 3 --zone us-central1-a --machine-type n1-standard-1
```

```
kubectl create deployment fancytest --image=gcr.io/${GOOGLE_CLOUD_PROJECT}/fancytest:1.0.0
```

```
kubectl expose deployment fancytest --name=fancytest --type=LoadBalancer --port=80 --target-port=8080
```



Task 3: Create a containerized version of your Microservices

```
cd ~/monolith-to-microservices/microservices/src/orders
gcloud builds submit --tag gcr.io/${GOOGLE_CLOUD_PROJECT}/orders:1.0.0 .
```

```
cd ~/monolith-to-microservices/microservices/src/products
gcloud builds submit --tag gcr.io/${GOOGLE_CLOUD_PROJECT}/products:1.0.0 .
```

Task 4: Deploy the new microservices

```
cd ~/monolith-to-microservices/microservices/src/orders

kubectl create deployment orders --
image=gcr.io/${GOOGLE_CLOUD_PROJECT}/orders:1.0.0

kubectl expose deployment orders --type=LoadBalancer --port 80 --target-
port 8081

For products :: use below commands ::

cd ~/monolith-to-microservices/microservices/src/products

kubectl create deployment products --
image=gcr.io/${GOOGLE_CLOUD_PROJECT}/products:1.0.0

kubectl expose deployment products --type=LoadBalancer --port 80 --target-
port 8082

kubectl get services
```

Task 5: Configure the Frontend microservice

Use the nano editor to replace the local URL with the IP address of the new Products microservices:

```
cd ~/monolith-to-microservices/react-app
nano .env

When the editor opens, your file should look like this:

REACT_APP_ORDERS_URL=http://localhost:8081/api/orders
REACT_APP_PRODUCTS_URL=http://localhost:8082/api/products

Replace the REACT_APP_PRODUCTS_URL to the new format while replacing with
your Orders and Product microservice IP addresses so it matches below:

REACT_APP_ORDERS_URL=http://<ORDERS_IP_ADDRESS>/api/orders
REACT_APP_PRODUCTS_URL=http://<PRODUCTS_IP_ADDRESS>/api/products

Press CTRL+O, press ENTER, then CTRL+X to save the file in the nano editor.

npm run build
```

Task 6: Create a containerized version of the Frontend microservice

```
cd ~/monolith-to-microservices/microservices/src/frontend
gcloud builds submit --tag gcr.io/${GOOGLE_CLOUD_PROJECT}/frontend:1.0.0 .
```

Task 7: Deploy the Frontend microservice

```
kubectl create deployment frontend --
image=gcr.io/${GOOGLE_CLOUD_PROJECT}/frontend:1.0.0

kubectl expose deployment frontend --type=LoadBalancer --port 80 --target-
port 8080
```

Congratulations! You completed this challenge lab.