Build a Website on Google Cloud:

export MONOLITH=

export CLUSTER=

export ORDERS=

export PRODUCTS=

export FRONTEND=

gcloud config set compute/zone [ZONE]

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

Task 1: Download the monolith code and build your container

```
git clone https://github.com/googlecodelabs/monolith-to-microservices.git
cd ~/monolith-to-microservices
```

./setup.sh

nvm install --lts

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

Task 2. Create a kubernetes cluster and deploy the application

gcloud container clusters create $CLUSTER --num-nodes 3

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

kubectl expose deployment $MONOLITH --type=LoadBalancer --port 80 --target-port 8080

Task 3. Create new 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

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

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 service

Task 5. Configure and deploy the Frontend microservice

export ORDERS_IP=$(kubectl get services -o
jsonpath="{.items[1].status.loadBalancer.ingress[0].ip}")

export PRODUCTS_IP=$(kubectl get services -o
jsonpath="{.items[2].status.loadBalancer.ingress[0].ip}")

cd ~/monolith-to-microservices/react-app

sed -i "s/localhost:8081/$ORDERS_IP/g" .env

sed -i "s/localhost:8082/$PRODUCTS_IP/g" .env

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

kubectl get svc
```