

Build a Website on Google Cloud: Challenge Lab

This lab demonstrates how to deploy a monolithic application and convert it into microservices on Google Kubernetes Engine (GKE). Below are the steps to clone the application, build the necessary images, and deploy them to your Kubernetes cluster.

Setup

1. Export Environment Variables

Make sure to set the following environment variables:

```
export ZONE=
```

```
export MONOLITH_IDENTIFIER=
```

```
export CLUSTER_NAME=
```

```
export ORDERS_IDENTIFIER=
```

```
export PRODUCTS_IDENTIFIER=
```

```
export FRONTEND_IDENTIFIER=
```

2. Clone the Repository

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

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

```
./setup.sh
```

3. Run the Monolith Locally

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

```
npm start
```

You can stop the application by pressing Ctrl+C.

Deploy the Monolithic Application

4. Enable Cloud Build API and Submit the Build

```
gcloud services enable cloudbuild.googleapis.com gcloud builds submit --tag  
gcr.io/${GOOGLE_CLOUD_PROJECT}/${MONOLITH_IDENTIFIER}:1.0.0 .
```

5. Create a GKE Cluster and Deploy the Monolith

```
gcloud config set compute/zone $ZONE gcloud services enable container.googleapis.com
gcloud container clusters create $CLUSTER_NAME --num-nodes 3
```

```
kubectl create deployment $MONOLITH_IDENTIFIER --
image=gcr.io/${GOOGLE_CLOUD_PROJECT}/${MONOLITH_IDENTIFIER}:1.0.0 kubectl
expose deployment $MONOLITH_IDENTIFIER --type=LoadBalancer --port 80 --target-port
8080
```

6. Deploy the Microservices

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

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

```
kubectl create deployment $ORDERS_IDENTIFIER --
image=gcr.io/${GOOGLE_CLOUD_PROJECT}/${ORDERS_IDENTIFIER}:1.0.0 kubectl expose
deployment $ORDERS_IDENTIFIER --type=LoadBalancer --port 80 --target-port 8081
```

```
kubectl create deployment $PRODUCTS_IDENTIFIER --
image=gcr.io/${GOOGLE_CLOUD_PROJECT}/${PRODUCTS_IDENTIFIER}:1.0.0 kubectl
expose deployment $PRODUCTS_IDENTIFIER --type=LoadBalancer --port 80 --target-port
8082
```

7. Deploy the Frontend

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

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

```
kubectl create deployment $FRONTEND_IDENTIFIER --
image=gcr.io/${GOOGLE_CLOUD_PROJECT}/${FRONTEND_IDENTIFIER}:1.0.0 kubectl
expose deployment $FRONTEND_IDENTIFIER --type=LoadBalancer --port 80 --target-port
8080
```