# GSP319 : Build a Website on Google Cloud: Challenge Lab

Task 1: Download the monolith code and build your container

1git clone https://github.com/googlecodelabs/monolith-to-microservices.git

2

3cd ~/monolith-to-microservices

4./setup.sh

5

6cd ~/monolith-to-microservices/monolith

7npm start

8

9gcloud services enable cloudbuild.googleapis.com

10gcloud builds submit --tag gcr.io/${GOOGLE\_CLOUD\_PROJECT}/fancytest:1.0.0 .

### Task 2: Create a kubernetes cluster and deploy the application

1gcloud config set compute/zone us-central1-a

2gcloud services enable container.googleapis.com

3gcloud container clusters create fancy-cluster --num-nodes 3

4

5kubectl create deployment fancytest --image=gcr.io/${GOOGLE\_CLOUD\_PROJECT}/fancytest:1.0.0

6kubectl expose deployment fancytest --type=LoadBalancer --port 80 --target-port 8080

### Task 3: Create a containerized version of your Microservices

1cd ~/monolith-to-microservices/microservices/src/orders

2gcloud builds submit --tag gcr.io/${GOOGLE\_CLOUD\_PROJECT}/orders:1.0.0 .

3

4cd ~/monolith-to-microservices/microservices/src/products

5gcloud builds submit --tag gcr.io/${GOOGLE\_CLOUD\_PROJECT}/products:1.0.0 .

### Task 4: Deploy the new microservices

1kubectl create deployment orders --image=gcr.io/${GOOGLE\_CLOUD\_PROJECT}/orders:1.0.0

2kubectl expose deployment orders --type=LoadBalancer --port 80 --target-port 8081

3

4kubectl create deployment products --image=gcr.io/${GOOGLE\_CLOUD\_PROJECT}/products:1.0.0

5kubectl expose deployment products --type=LoadBalancer --port 80 --target-port 8082

### Task 5: Configure the Frontend microservice

1cd ~/monolith-to-microservices/react-app

2nano .env

### Task 6: Create a containerized version of the Frontend microservice

1cd ~/monolith-to-microservices/microservices/src/frontend

2gcloud builds submit --tag gcr.io/${GOOGLE\_CLOUD\_PROJECT}/frontend:1.0.0 .

### Task 7: Deploy the Frontend microservice

1kubectl create deployment frontend --image=gcr.io/${GOOGLE\_CLOUD\_PROJECT}/frontend:1.0.0

2

3kubectl expose deployment frontend --type=LoadBalancer --port 80 --target-port 8080