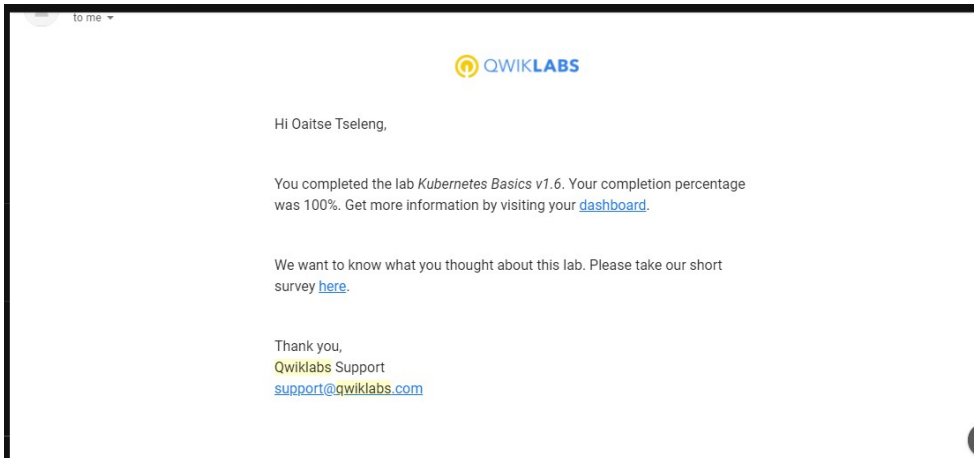
Part 1: Qwiklabs Completion(11 LAbs)

28 August:

1. Kubernetes Basics v1.6


to me

QWIKLABS

Hi Oaitse Tseleng,

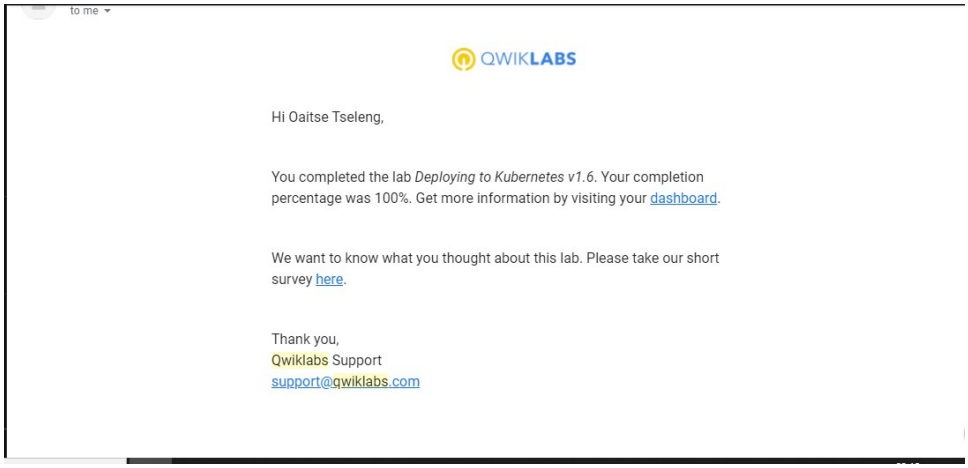
You completed the lab *Kubernetes Basics v1.6*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

29 August:

2. Deploying to Kubernetes v1.6


to me

QWIKLABS

Hi Oaitse Tseleng,

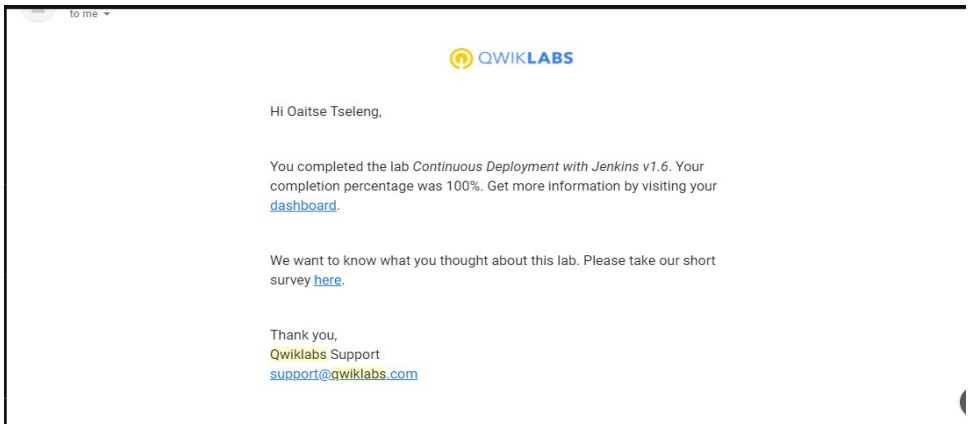
You completed the lab *Deploying to Kubernetes v1.6*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

6 September:

3. Continuous Deployment with Jenkins v1.6


to me

QWIKLABS

Hi Oaitse Tseleng,

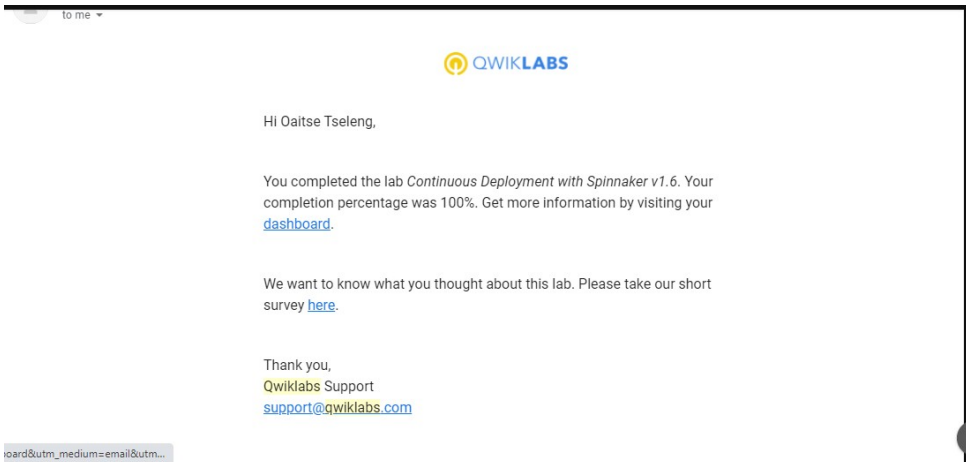
You completed the lab *Continuous Deployment with Jenkins v1.6*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

7 September:

4. Continuous Deployment with Spinnaker v1.6



to me

QWIKLABS

Hi Oaitse Tseleng,

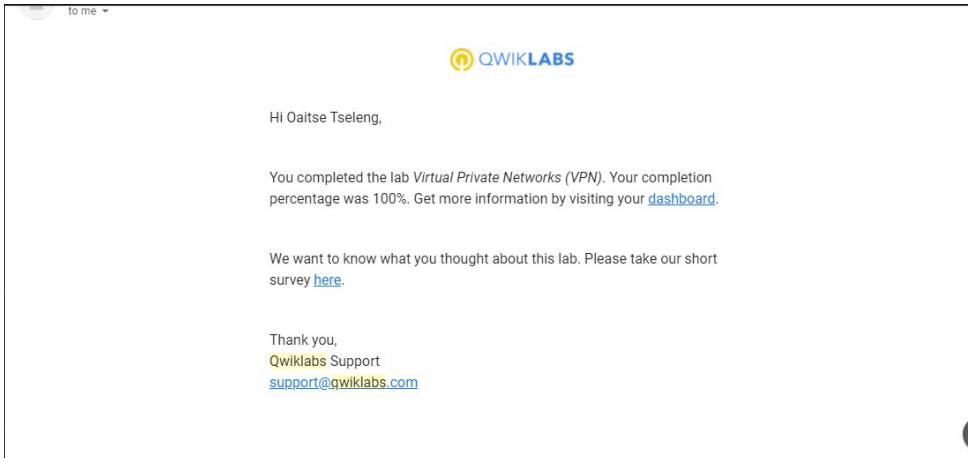
You completed the lab *Continuous Deployment with Spinnaker v1.6*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

oard&utm_medium=email&utm...

5. Virtual Private Networks (VPN)



to me

QWIKLABS

Hi Oaitse Tseleng,

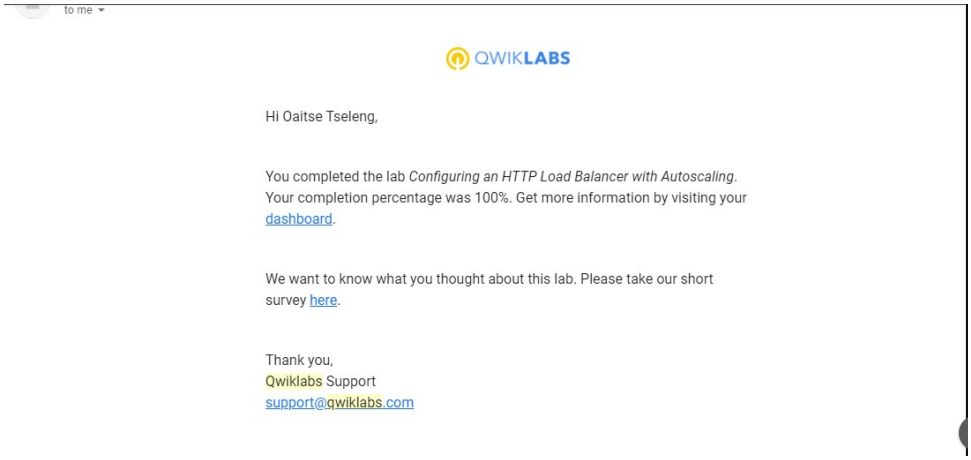
You completed the lab *Virtual Private Networks (VPN)*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

9 September:

6. HTTP Load Balancer with Autoscaling



to me

QWIKLABS

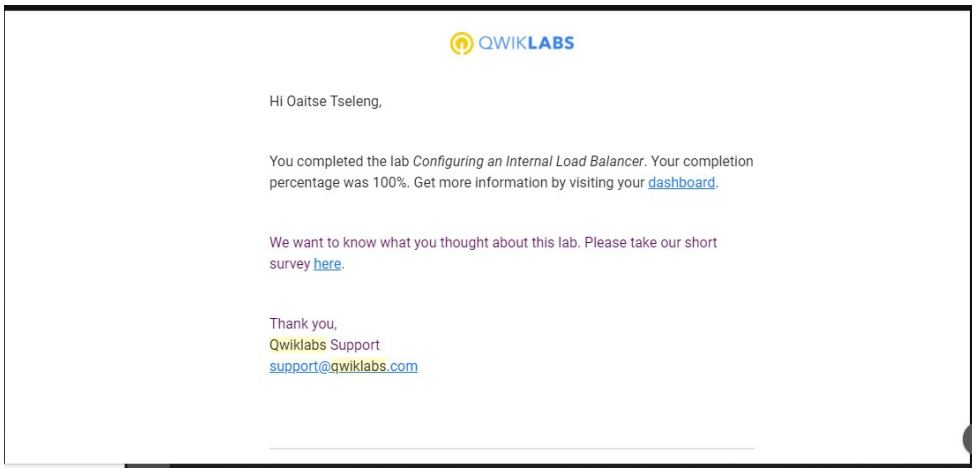
Hi Oaitse Tseleng,

You completed the lab *Configuring an HTTP Load Balancer with Autoscaling*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

7. Configuring an Internal Load Balancer



QWIKLABS

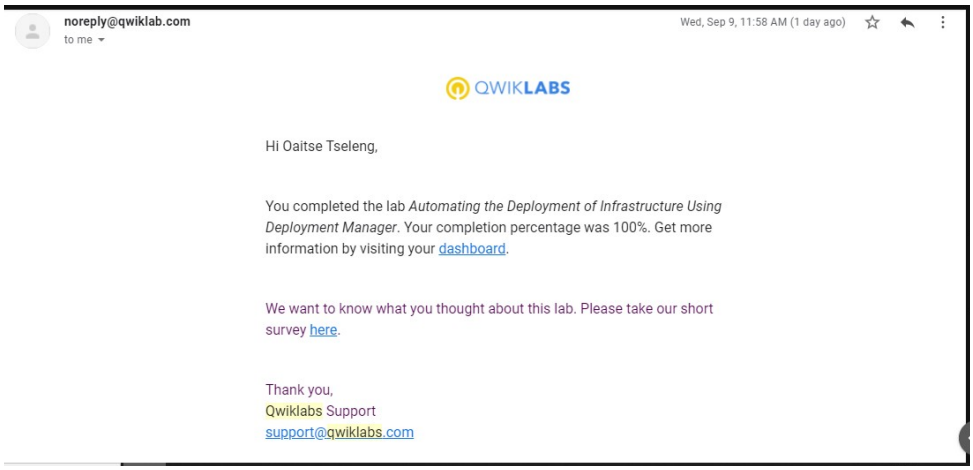
Hi Oaitse Tseleng,

You completed the lab *Configuring an Internal Load Balancer*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

8. Automating the Deployment of Infrastructure Using Deployment Manager



noreply@qwiklab.com
to me
Wed, Sep 9, 11:58 AM (1 day ago)

QWIKLABS

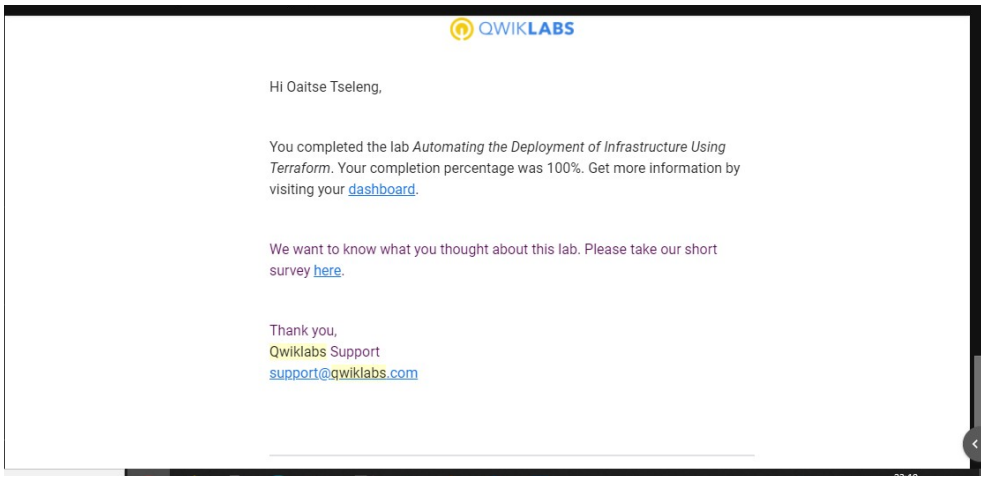
Hi Oaitse Tseleng,

You completed the lab *Automating the Deployment of Infrastructure Using Deployment Manager*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

9. Automating the Deployment of Infrastructure Using Terraform



QWIKLABS

Hi Oaitse Tseleng,

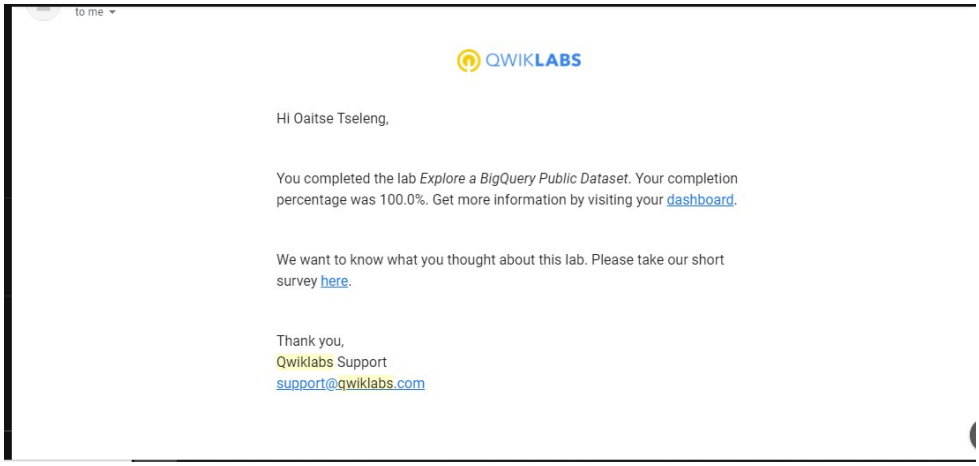
You completed the lab *Automating the Deployment of Infrastructure Using Terraform*. Your completion percentage was 100%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

10 September:

10. BigQuery Public Dataset



to me

QWIKLABS

Hi Oaitse Tseleng,

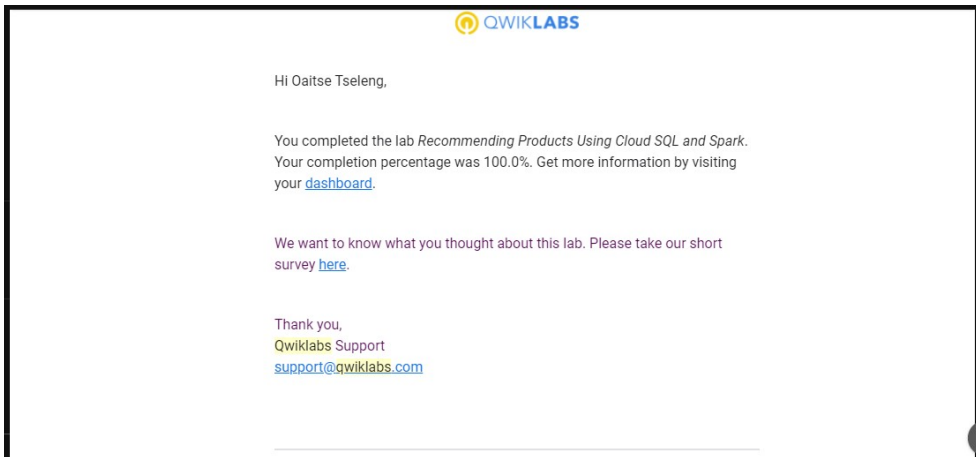
You completed the lab *Explore a BigQuery Public Dataset*. Your completion percentage was 100.0%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

11 September:

11. Recommending Products Using Cloud SQL and Spark



QWIKLABS

Hi Oaitse Tseleng,

You completed the lab *Recommending Products Using Cloud SQL and Spark*. Your completion percentage was 100.0%. Get more information by visiting your dashboard.

We want to know what you thought about this lab. Please take our short survey here.

Thank you,
Qwiklabs Support
support@qwiklabs.com

Phase 2: From Console to Command Line

Lab: Configuring an Internal Load Balancer

Task 1. Configure a health check firewall rule

- gcloud compute firewall-rules create fw-allow-health-checks \ --network default \ --target-tags allow-health -checks \ --source-ranges 130.211.0.0/22, 35.191.0.0/16 \ --rules tcp:80

Task 2: Create a NAT configuration using Cloud Router

- gcloud compute routers nats create nat-config \ --router nat-router-us-central1

Task 3: Create a custom image for a web server

- create vm instance:
  gcloud beta compute --project=qwiklabs-gcp-f72529adb527ca74 instances create webserver --zone=us-central1-a --machine-type=f1-micro --subnet=default --no-address --maintenance-policy=MIGRATE --service-account=984833531377-compute@developer.gserviceaccount.com --scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append --tags=allow-health-checks --image=debian-10-buster-v20200910 --image-project=debian-cloud --boot-disk-size=10GB --no-boot-disk-auto-delete --boot-disk-type=pd-standard --boot-disk-device-name=webserver --no-shielded-secure-boot --no-shielded-vtpm --no-shielded-integrity-monitoring --reservation-affinity=any
- gcloud compute ssh webserver
- Set the Apache service to start at boot:
  sudo apt-get update
  sudo apt-get install -y apache2
  sudo service apache2 start
  curl localhost
  sudo update-rc.d apache2 enable
  sudo service apache2 status
- Prepare the disk to create a custom image:
  gcloud compute instances describe webserver
  gcloud compute instances delete webserver --zone us-central1-a
  gcloud compute images create mywebserver \ --source-image webserver

Task 4. Configure an instance template and create instance groups

- Configure instance template:
  gcloud beta compute --project=qwiklabs-gcp-f72529adb527ca74 instance-templates create mywebserver-template --machine-type=f1-micro --network=projects/qwiklabs-gcp-f72529adb527ca74/global/networks/default --no-address --maintenance-policy=MIGRATE --service-account=984833531377-compute@developer.gserviceaccount.com --scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://

www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append --tags=allow-health-checks --image=mywebserver --image-project=qwiklabs-gcp-f72529adb527ca74 --boot-disk-size=10GB --boot-disk-type=pd-standard --boot-disk-device-name=mywebserver-template --no-shielded-secure-boot --shielded-vtpm --shielded-integrity-monitoring --reservation-affinity=any

- Create the managed instance groups:
  us-central1:
  gcloud compute --project "qwiklabs-gcp-f72529adb527ca74" health-checks create tcp "http-health-check" --timeout "5" --check-interval "10" --unhealthy-threshold "3" --healthy-threshold "2" --port "80"

  gcloud beta compute --project=qwiklabs-gcp-f72529adb527ca74 instance-groups managed create us-central1-mig --base-instance-name=us-central1-mig --template=mywebserver-template --size=1 --zones=us-central1-b,us-central1-c,us-central1-f --instance-redistribution-type=PROACTIVE --health-check=http-health-check --initial-delay=60

  gcloud beta compute --project "qwiklabs-gcp-f72529adb527ca74" instance-groups managed set-autoscaling "us-central1-mig" --region "us-central1" --cool-down-period "60" --max-num-replicas "2" --min-num-replicas "1" --target-load-balancing-utilization "0.8" --mode "on"

  europe-west1:
  gcloud beta compute --project=qwiklabs-gcp-f72529adb527ca74 instance-groups managed create europe-west1-mig --base-instance-name=europe-west1-mig --template=mywebserver-template --size=1 --zones=europe-west1-b,europe-west1-c,europe-west1-d --instance-redistribution-type=PROACTIVE --health-check=http-health-check --initial-delay=60

  gcloud beta compute --project "qwiklabs-gcp-f72529adb527ca74" instance-groups managed set-autoscaling "europe-west1-mig" --region "europe-west1" --cool-down-period "60" --max-num-replicas "2" --min-num-replicas "1" --target-load-balancing-utilization "0.8" --mode "on"

Task 5. Configure the HTTP load balancer

- Create static ip addresses:
  gcloud compute addresses create lb-ipv4-1 \--ip-version=IPV4 \ --global
  gcloud compute addresses create lb-ipv6-1 \--ip-version=IPV6 \ --global
- Health checks:
  gcloud compute health-checks create http http-health-check\ --port 80
- Create backend service:
  gcloud compute backend-services create http-lb \--protocol=HTTP \--port-name=http \--health-checks= http-health-check \--global
- Add backends:
  gcloud compute backend-services add-backend web-backend-service \--instance-group= europe-west1-mig
  gcloud compute backend-services add-backend web-backend-service \--instance-group= us-central1-mig
  gcloud compute forwarding-rules create http-content-rule \--address=lb-ipv4-1\--global \--target-http-proxy=http-lb-proxy \--ports=80

- Stress test the HTTP load balancer
  Creating new vm instance:
  gcloud beta compute --project=qwiklabs-gcp-f72529adb527ca74 instances create stress-test --zone=us-west1-c --machine-type=f1-micro --subnet=default --network-tier=PREMIUM --maintenance-policy=MIGRATE --service-account=984833531377-compute@developer.gserviceaccount.com --scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append --image=mywebserver --image-project=qwiklabs-gcp-f72529adb527ca74 --boot-disk-size=10GB --boot-disk-type=pd-standard --boot-disk-device-name=stress-test --no-shielded-secure-boot --shielded-vtpm --shielded-integrity-monitoring --reservation-affinity=any

  create an environment variable for your load balancer IP address:
  export LB_IP=<Enter your [LB_IP_v4] here>

  place a load on the load balancer:
  ab -n 500000 -c 1000 http://$LB_IP/

Lab: Exploring a Big Query Public Dataset

Task 1: Query a public dataset

- bq query --use_legacy_sql=false \ "SELECT name, gender FROM bigquery-public-data.usa_names.usa_1910_2013 GROUP BY name, gender ORDER BY total DESC LIMIT 10"

Task 3: Create Custom table

- bq mk babynames
- bq ls

Task 4: Load Data into new table

- bq load / --source_format=CSV / babynames.names2014 / C:\Users\oaits\Downloads\Compressed\yob2014.txt /name:string, gender:string, count:integer

Task 5: Query the Table

- bq query "SELECT name, count FROM `babynames.names_2014` WHERE gender = 'M' ORDER BY count DESC LIMIT 5"

Lab: Recommend Products using ML with Cloud SQL and Dataproc

Task 1: Create a cloud SQL instance

- gcloud sql instances create rentals --root-password=root

Task 2: Create Tables

- gcloud sql connect rentals --user=root
- provide password
- run command “SHOW DATABASES;”
- Run Script

```
CREATE DATABASE IF NOT EXISTS recommendation_spark;

USE recommendation_spark;

DROP TABLE IF EXISTS Recommendation;
DROP TABLE IF EXISTS Rating;
DROP TABLE IF EXISTS Accommodation;

CREATE TABLE IF NOT EXISTS Accommodation
(
 id varchar(255),
 title varchar(255),
 location varchar(255),
 price int,
 rooms int,
 rating float,
 type varchar(255),
 PRIMARY KEY (ID)
);

CREATE TABLE  IF NOT EXISTS Rating
(
 userId varchar(255),
 accoId varchar(255),
 rating int,
 PRIMARY KEY(accoId, userId),
 FOREIGN KEY (accoId)
  REFERENCES Accommodation(id)
);

CREATE TABLE  IF NOT EXISTS Recommendation
(
 userId varchar(255),
 accoId varchar(255),
 prediction float,
 PRIMARY KEY(userId, accoId),
 FOREIGN KEY (accoId)
  REFERENCES Accommodation(id)
);

SHOW DATABASES;
```

- USE recommendation_spark;
  SHOW TABLES;
- SELECT * FROM Accommodation;

Task 3: Stage data in cloud storage

- Creating bucket:
  gsutil mb gs://$DEVSHELL_PROJECT_ID
- Copying data from public dataset into our bucket:
  gsutil cp gs://cloud-training/bdml/v2.0/data/accommodation.csv
  gsutil cp gs://cloud-training/bdml/v2.0/data/rating.csv
- Show the files in our bucket:
  gsutil ls gs://$DEVSHELL_PROJECT_ID
- View some sample data:
  gsutil cat gs://$DEVSHELL_PROJECT_ID/accommodation.csv

Task 4: Load data from Cloud Storage into Cloud SQL tables

- gcloud sql import csv rentals gs:// qwiklabs-gcp-02-d4e65ce05de3/accomodation.csv \ --database= recommendation_spark --table= Accommodation
- gcloud sql import csv rentals gs:// qwiklabs-gcp-02-d4e65ce05de3/ratings.csv \ --database= recommendation_spark --table= Ratings

Task 5: Explore Cloud SQL data

- USE recommendation_spark;
- Query the ratings data:
  SELECT * FROM Rating LIMIT 15;
- Count Number of rows in table:
  SELECT COUNT(*) AS num_ratings FROM Rating;
- Average review rating of accommodations:
  ```
  SELECT
    COUNT(userId) AS num_ratings,
    COUNT(DISTINCT userId) AS distinct_user_ratings,
    MIN(rating) AS worst_rating,
    MAX(rating) AS best_rating,
    AVG(rating) AS avg_rating
  FROM Rating;
  ```
- Users who have provided the most ratings:
  ```
  SELECT
    userId,
    COUNT(rating) AS num_ratings
  FROM Rating
  GROUP BY userId
  ORDER BY num_ratings DESC;
  ```

Task 6. Launch Dataproc

- Create dataproc cluster:
  gcloud dataproc clusters create rentals –region=us-central1 --master-machine-type= n1-standard-2 --worker-machine-type= n1-standard-2
- Authorising cloud dataproc to connect to sql

  ```
  echo "Authorizing Cloud Dataproc to connect with Cloud SQL"
  CLUSTER=rentals
  ```

```
CLOUDSQL=rentals
ZONE=us-central1-a
NWORKERS=2

machines="$CLUSTER-m"
for w in `seq 0 $(($NWORKERS - 1))`; do
  machines="$machines $CLUSTER-w-$w"
done

echo "Machines to authorize: $machines in $ZONE ... finding their IP addresses"
ips=""
for machine in $machines; do
   IP_ADDRESS=$(gcloud compute instances describe $machine --zone=$ZONE --
format='value(networkInterfaces.accessConfigs[].natIP)' | sed "s/\['//g" | sed "s/'\]//g" )/32
   echo "IP address of $machine is $IP_ADDRESS"
   if [ -z  $ips ]; then
     ips=$IP_ADDRESS
   else
     ips="$ips,$IP_ADDRESS"
   fi
done

echo "Authorizing [$ips] to access cloudsql=$CLOUDSQL"
gcloud sql instances patch $CLOUDSQL --authorized-networks $ips
```

Task 7. Run the ML model

- gsutil cp gs://cloud-training/bdml/v2.0/model/train_and_apply.py train_and_apply.py
  cloudshell edit train_and_apply.py
- Edit apply and train file:
  # MAKE EDITS HERE
  CLOUDSQL_INSTANCE_IP = '<paste-your-cloud-sql-ip-here>'   # <---- CHANGE (database server IP)
  CLOUDSQL_PWD = '<type-your-cloud-sql-password-here>' # <---- CHANGE
- Copy file to cloud stoage:
  gsutil cp train_and_apply.py gs://$DEVSHELL_PROJECT_ID

Task 8. Run your ML job on Dataproc

- gcloud dataproc jobs submit pyspark gs:// qwiklabs-gcp-02-d4e65ce05de3 /train_and_apply.py --cluster=rentals

Task 9. Explore inserted rows with SQL

- USE recommendation_spark;
- Check to see if dataproc finished its job:
  SELECT COUNT(*) AS count FROM Recommendation;
- Find the recommendations for a user:
  SELECT r.userid, r.accoid, r.prediction, a.title, a.location, a.price, a.rooms, a.rating, a.type

```
FROM Recommendation as r JOIN Accommodation as a
ON r.accoid = a.id WHERE r.userid = 10;
```