Using Cloud sql with Kubernetes Engine

To list active account name:

***gcloud auth list***

To list project id :

***gcloud config list project***

To create environment variables , cluster name and configure tab completion of kubectl command:

***export my\_zone=us-central1-a***

***export my\_cluster=standard-cluster-1***

***source <(kubectl completion bash)***

To Create a VPC-native Kubernetes cluster and configure access to it for kubectl:

***gcloud container clusters create $my\_cluster \***

***--num-nodes 3 --enable-ip-alias --zone $my\_zone***

***gcloud container clusters get-credentials $my\_cluster --zone $my\_zone***

To clone a repository and change to the repo directory:

***git clone*** [***https://github.com/GoogleCloudPlatformTraining/training-data-analyst***](https://github.com/GoogleCloudPlatformTraining/training-data-analyst)

***cd ~/training-data-analyst/courses/ak8s/18\_Cloud\_SQL/***

To enable cloud sql api in cloud shell:

[***gcloud services list***](https://cloud.google.com/sdk/gcloud/reference/services/list) ***--available***

***gcloud services enable cloudsqlapi***

To create a cloud sql instance and also create a account and set its root password:

***gcloud sql instances create sql-instance --tier=db-n1-standard-2 --region=us-central1***

***Copy your instance connection name and export it to cloudshell using this command:export SQL\_NAME=qwiklabs-gcp-01-f126b144dd0b:us-central1:sql-instance***

To connect to that Sql Instance and use a word press database :

***gcloud sql connect sql-instance***

***use wordpress;***

***Inorder to show the tables in the wordpress i used:***

***show tables;***

To create a service account in cloud IAM and assign it a cloud Sql client role:

Use GCP cloud console

To create secrets use the kubectl command as shown below:

***kubectl create secret generic sql-credentials \***

***--from-literal=username=sqluser\***

***--from-literal=password=sqlpassword***

***Upload the previously downloaded credentials file and move it using this command:***

***mv ~/credentials.json .***

To create a secret for the gcloud service accounts credentials:

***kubectl create secret generic google-credentials\***

***--from-file=key.json=credentials.json***

To deploy the a wordpress database and cloud sql proxy agent as a side container a **sql-proxy.yaml** is provided

The sed command is used to update placeholder variable for the SQL connection name to the instance:

***sed -i 's/<INSTANCE\_CONNECTION\_NAME>/'"${SQL\_NAME}"'/g'\***

***Sql-proxy.yaml***

To deploy the application and later check its status use these commands:

***kubectl apply -f sql-proxy.yaml***

***kubectl get deployment wordpress***

***kubectl get services is used to list the services in the GKE clusters.***

Connect to the wordpress and install it then run the flowing command in your cloud shell to connect to the SQL instance:

***gcloud sql connect sql-instance***

***use wordpress;***

***show tables;***

***To exit Sql use ‘exit’ command.***

Explore Big query dataset

Open big query and load a public dataset say USA names. Click on that names dataset and Query bigquery-public-data.usa\_names.usa\_1910\_2013 for the name and gender of the babies in this dataset, and then list the top 10 names in descending order, paste the command below in the cloud shell:

*bq query --use\_legacy\_sql=false \*

*‘SELECT*

*name, gender,*

*SUM(number) AS total*

*FROM*

*`bigquery-public-data.usa\_names.usa\_1910\_2013`*

*GROUP BY*

*name, gender*

*ORDER BY*

*total DESC*

*LIMIT*

*10’*

Create an dataset in gcp console and load the downloaded tables into the dataset:

First use *bq ls* to show if your project has existing data sets

*bq mk babynames*

Use the *bq ls* to show if the dataset has been created

To create and upload the table use *bq load*:

*bq load babynames.names2014 yob2014.txt name:string,gender:string,count:integer*

* *datasetID: babynames*
* *tableID: names2014*
* *source: yob20104.txt: if necessary, include the full path to the file*
* *schema: name:string,gender:string,count:integer*

To query the table;

*Bq query ‘*

*SELECT*

*name, count*

*FROM*

*`babynames.names\_2014`*

*WHERE*

*gender = 'M'*

*ORDER BY count DESC LIMIT 5*

*‘*