OntarioTech UNIVERSITY

Project Milestone

Data Storage Implementation: KV + relational

Name: Mihir Patel
Student Number: 100702168
Date: January 31, 2022

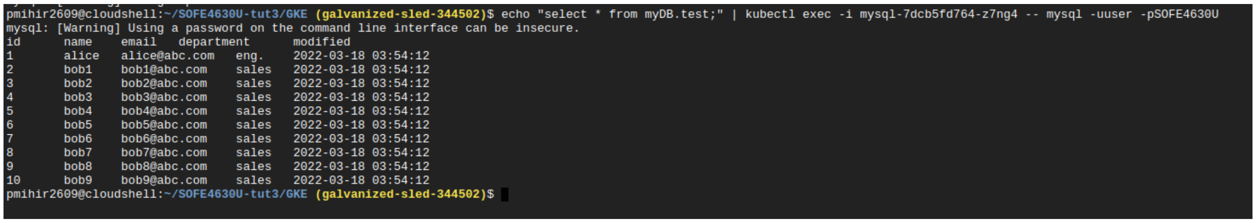
Procedure:

1. **Watch the first three videos for Kafka connectors (focus on the concepts, not the details)**
    - https://www.confluent.io/blog/kafka-connect-tutorial/ (Links to an external site.).

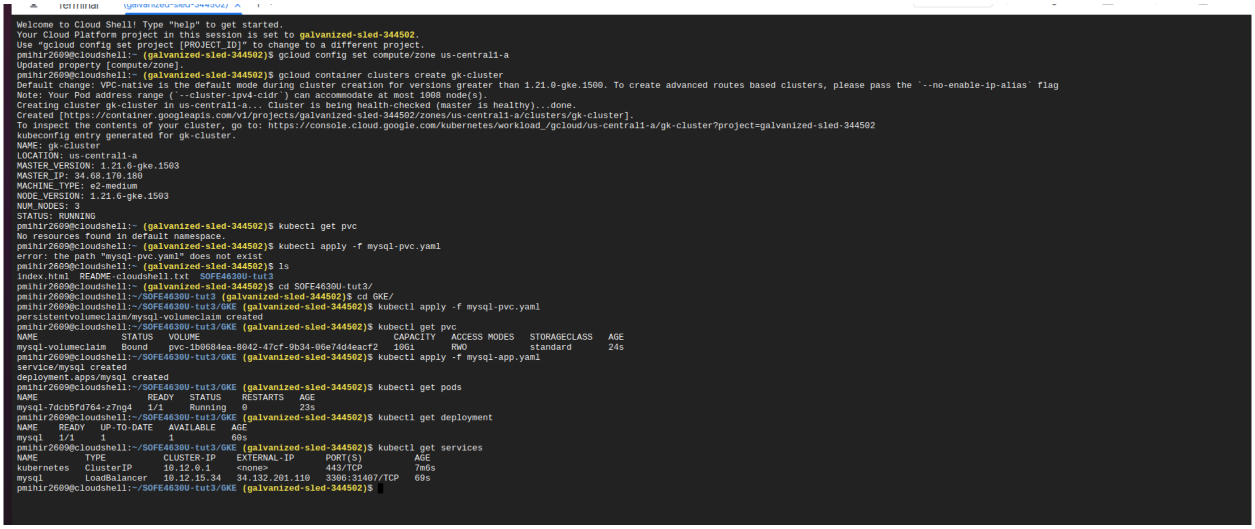
2. **Describe the following:**
    - Sink and Source connectors.
    - The applications/advantages of using Kafka Connectors with data storage.
    - How do Kafka connectors maintain availability?
    - List the popular Kafka converters for values and the properties/advantages of each.

3. **Search the internet to answer the following question:**
    - What's a Key-Value (KV) database?
    - What are KV databases' advantages and disadvantages?
    - List some popular KV databases.

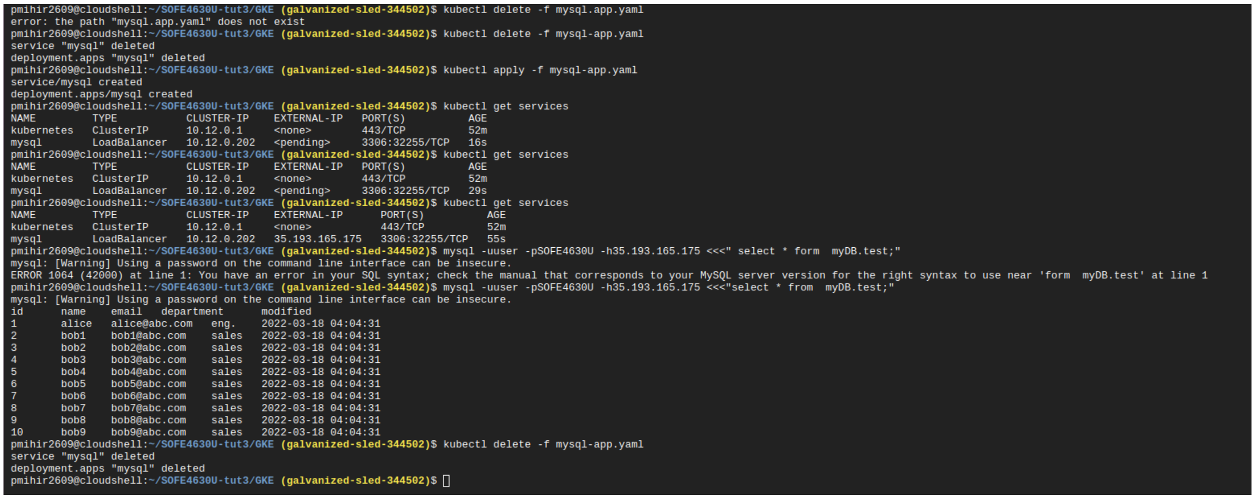
4. **Follow the following videos to deploy and use Redis and MySQL databases using GKE.**



```
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ echo "select * from myDB.test;" | kubectl exec -i mysql-7dcb5fd764-z7ng4 -- mysql -uuser -pSOFE4630U
mysql: [Warning] Using a password on the command line interface can be insecure.
id      name    email   department      modified
1       alice   alice@abc.com   eng.    2022-03-18 03:54:12
2       bob1    bob1@abc.com    sales   2022-03-18 03:54:12
3       bob2    bob2@abc.com    sales   2022-03-18 03:54:12
4       bob3    bob3@abc.com    sales   2022-03-18 03:54:12
5       bob4    bob4@abc.com    sales   2022-03-18 03:54:12
6       bob5    bob5@abc.com    sales   2022-03-18 03:54:12
7       bob6    bob6@abc.com    sales   2022-03-18 03:54:12
8       bob7    bob7@abc.com    sales   2022-03-18 03:54:12
9       bob8    bob8@abc.com    sales   2022-03-18 03:54:12
10      bob9    bob9@abc.com    sales   2022-03-18 03:54:12
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$
```

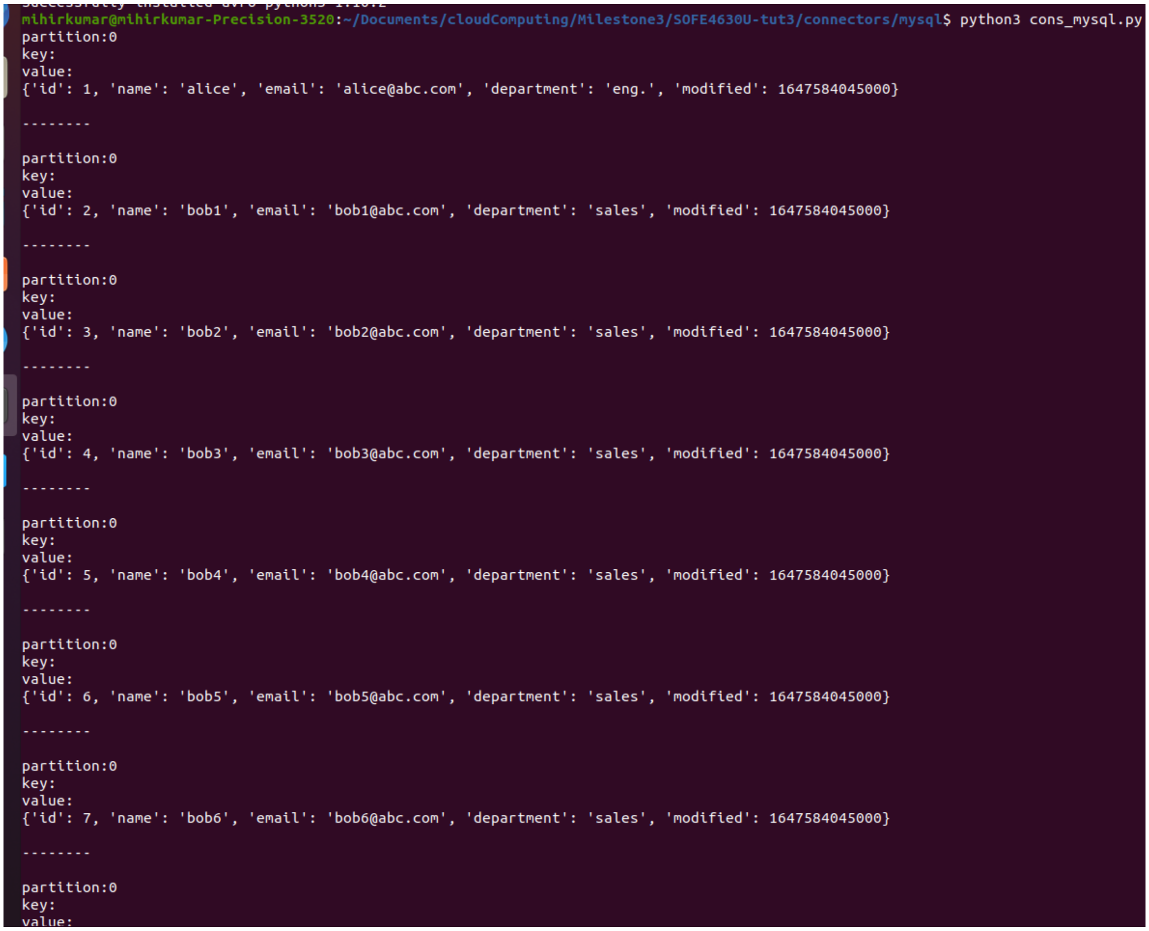


```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to galvanized-sled-344502.
Use “gcloud config set project [PROJECT_ID]” to change to a different project.
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ gcloud config set compute/zone us-central1-a
Updated property [compute/zone].
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ gcloud container clusters create gk-cluster
Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1500. To create advanced routes based clusters, please pass the `--no-enable-ip-alias` flag
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
Creating cluster gk-cluster in us-central1-a... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/galvanized-sled-344502/zones/us-central1-a/clusters/gk-cluster].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/us-central1-a/gk-cluster?project=galvanized-sled-344502
kubeconfig entry generated for gk-cluster.
NAME: gk-cluster
LOCATION: us-central1-a
MASTER_VERSION: 1.21.6-gke.1503
MASTER_IP: 34.68.170.180
MACHINE_TYPE: e2-medium
NODE_VERSION: 1.21.6-gke.1503
NUM_NODES: 3
STATUS: RUNNING
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ kubectl get pvc
No resources found in default namespace.
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ kubectl apply -f mysql-pvc.yaml
error: the path "mysql-pvc.yaml" does not exist
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ ls
index.html  README-cloudshell.txt  SOFE4630U-tut3
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ cd SOFE4630U-tut3/
pmihir2609@cloudshell:~/SOFE4630U-tut3 (galvanized-sled-344502)$ cd GKE/
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl apply -f mysql-pvc.yaml
persistentvolumeclaim/mysql-volumeclaim created
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl get pvc
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mysql-volumeclaim   Bound    pvc-1b0684ea-8042-47cf-9b34-06e74d4eacf2   10Gi       RWO            standard       24s
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl apply -f mysql-app.yaml
service/mysql created
deployment.apps/mysql created
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
mysql-7dcb5fd764-z7ng4   1/1     Running   0          23s
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl get deployment
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
mysql   1/1     1            1           60s
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl get services
NAME         TYPE           CLUSTER-IP    EXTERNAL-IP      PORT(S)          AGE
kubernetes   ClusterIP      10.12.0.1     <none>           443/TCP          7m6s
mysql        LoadBalancer   10.12.15.34   34.132.201.110   3306:31407/TCP   69s
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$
```

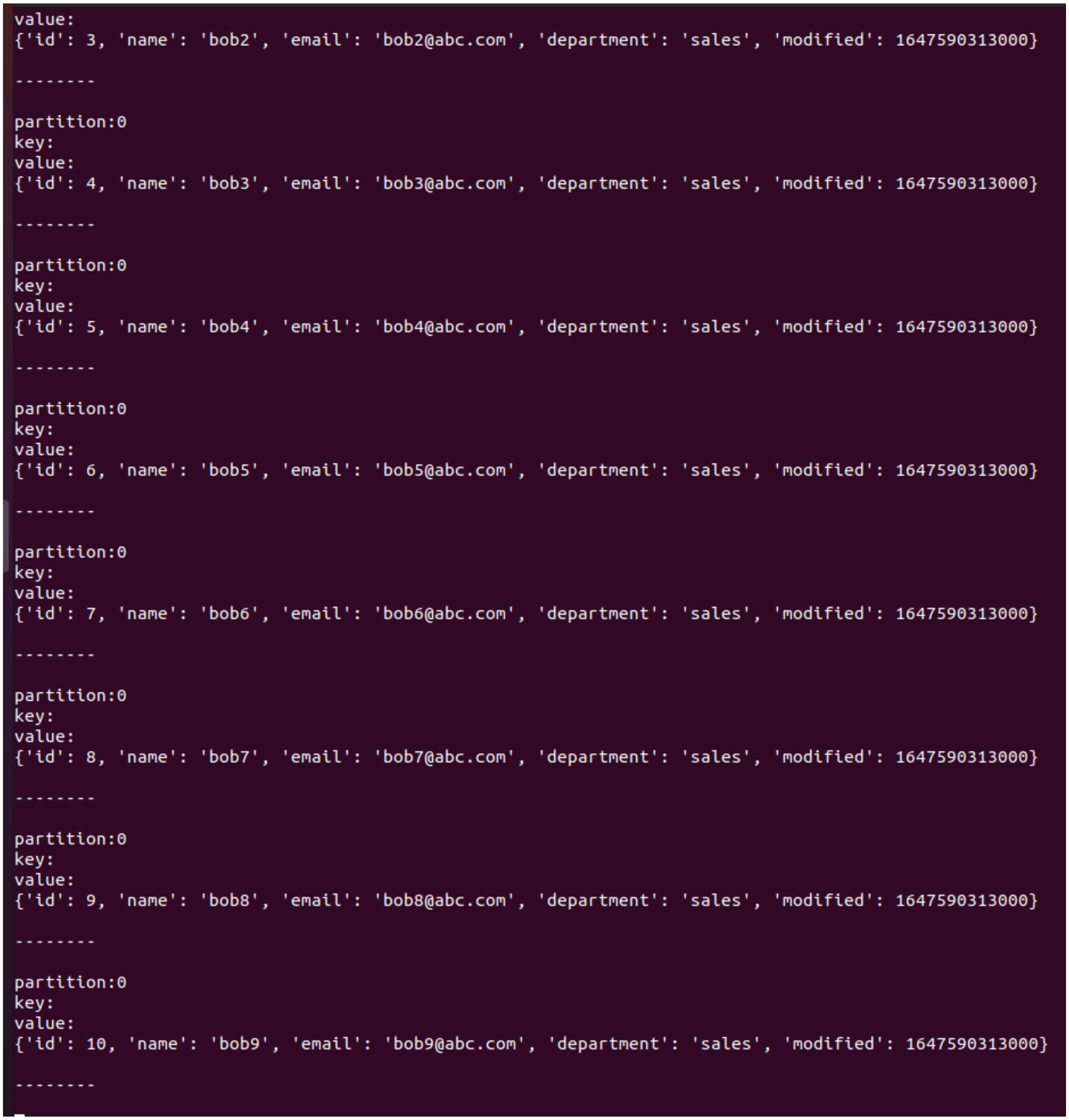


```
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl delete -f mysql.app.yaml
error: the path "mysql.app.yaml" does not exist
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl delete -f mysql-app.yaml
service "mysql" deleted
deployment.apps "mysql" deleted
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl apply -f mysql-app.yaml
service/mysql created
deployment.apps/mysql created
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl get services
NAME         TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP      10.12.0.1     <none>        443/TCP          52m
mysql        LoadBalancer   10.12.0.202   <pending>     3306:32255/TCP   16s
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl get services
NAME         TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP      10.12.0.1     <none>        443/TCP          52m
mysql        LoadBalancer   10.12.0.202   <pending>     3306:32255/TCP   29s
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl get services
NAME         TYPE           CLUSTER-IP    EXTERNAL-IP      PORT(S)          AGE
kubernetes   ClusterIP      10.12.0.1     <none>           443/TCP          52m
mysql        LoadBalancer   10.12.0.202   35.193.165.175   3306:32255/TCP   55s
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ mysql -uuser -pSOFE4630U -h35.193.165.175 <<<" select * form  myDB.test;"
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 1064 (42000) at line 1: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'form  myDB.test' at line 1
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ mysql -uuser -pSOFE4630U -h35.193.165.175 <<<"select * from  myDB.test;"
mysql: [Warning] Using a password on the command line interface can be insecure.
id      name    email   department      modified
1       alice   alice@abc.com   eng.    2022-03-18 04:04:31
2       bob1    bob1@abc.com    sales   2022-03-18 04:04:31
3       bob2    bob2@abc.com    sales   2022-03-18 04:04:31
4       bob3    bob3@abc.com    sales   2022-03-18 04:04:31
5       bob4    bob4@abc.com    sales   2022-03-18 04:04:31
6       bob5    bob5@abc.com    sales   2022-03-18 04:04:31
7       bob6    bob6@abc.com    sales   2022-03-18 04:04:31
8       bob7    bob7@abc.com    sales   2022-03-18 04:04:31
9       bob8    bob8@abc.com    sales   2022-03-18 04:04:31
10      bob9    bob9@abc.com    sales   2022-03-18 04:04:31
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl delete -f mysql-app.yaml
service "mysql" deleted
deployment.apps "mysql" deleted
pmihir2609@cloudshell:~/SOFE4630U-tut3/GKE (galvanized-sled-344502)$ 
```

5. Follow the following video to set up sink and source Kafka connectors to the deployed MySQL database

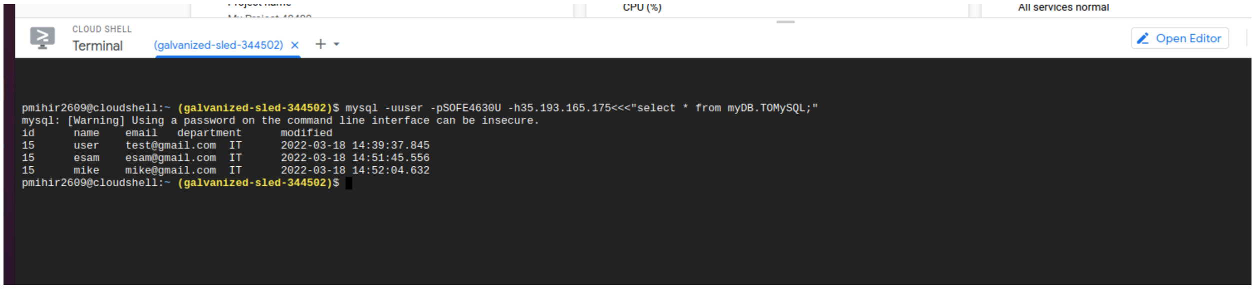


```
Successfully installed avro-python3 1.10.2
mihirkumar@mihirkumar-Precision-3520:~/Documents/cloudComputing/Milestone3/SOFE4630U-tut3/connectors/mysql$ python3 cons_mysql.py
partition:0
key:
value:
{'id': 1, 'name': 'alice', 'email': 'alice@abc.com', 'department': 'eng.', 'modified': 1647584045000}

--------

partition:0
key:
value:
{'id': 2, 'name': 'bob1', 'email': 'bob1@abc.com', 'department': 'sales', 'modified': 1647584045000}

--------

partition:0
key:
value:
{'id': 3, 'name': 'bob2', 'email': 'bob2@abc.com', 'department': 'sales', 'modified': 1647584045000}

--------

partition:0
key:
value:
{'id': 4, 'name': 'bob3', 'email': 'bob3@abc.com', 'department': 'sales', 'modified': 1647584045000}

--------

partition:0
key:
value:
{'id': 5, 'name': 'bob4', 'email': 'bob4@abc.com', 'department': 'sales', 'modified': 1647584045000}

--------

partition:0
key:
value:
{'id': 6, 'name': 'bob5', 'email': 'bob5@abc.com', 'department': 'sales', 'modified': 1647584045000}

--------

partition:0
key:
value:
{'id': 7, 'name': 'bob6', 'email': 'bob6@abc.com', 'department': 'sales', 'modified': 1647584045000}

--------

partition:0
key:
value:
```

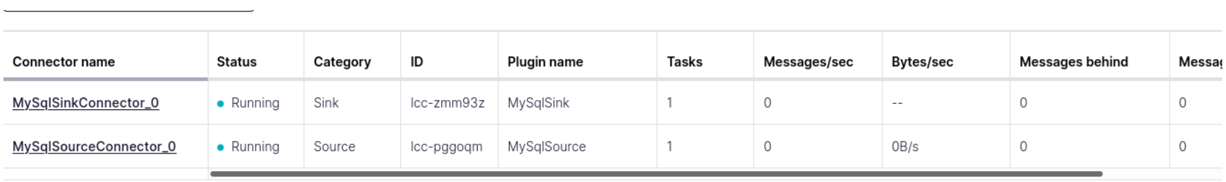


```
value:
{'id': 3, 'name': 'bob2', 'email': 'bob2@abc.com', 'department': 'sales', 'modified': 1647590313000}

--------

partition:0
key:
value:
{'id': 4, 'name': 'bob3', 'email': 'bob3@abc.com', 'department': 'sales', 'modified': 1647590313000}

--------

partition:0
key:
value:
{'id': 5, 'name': 'bob4', 'email': 'bob4@abc.com', 'department': 'sales', 'modified': 1647590313000}

--------

partition:0
key:
value:
{'id': 6, 'name': 'bob5', 'email': 'bob5@abc.com', 'department': 'sales', 'modified': 1647590313000}

--------

partition:0
key:
value:
{'id': 7, 'name': 'bob6', 'email': 'bob6@abc.com', 'department': 'sales', 'modified': 1647590313000}

--------

partition:0
key:
value:
{'id': 8, 'name': 'bob7', 'email': 'bob7@abc.com', 'department': 'sales', 'modified': 1647590313000}

--------

partition:0
key:
value:
{'id': 9, 'name': 'bob8', 'email': 'bob8@abc.com', 'department': 'sales', 'modified': 1647590313000}

--------

partition:0
key:
value:
{'id': 10, 'name': 'bob9', 'email': 'bob9@abc.com', 'department': 'sales', 'modified': 1647590313000}

--------
```

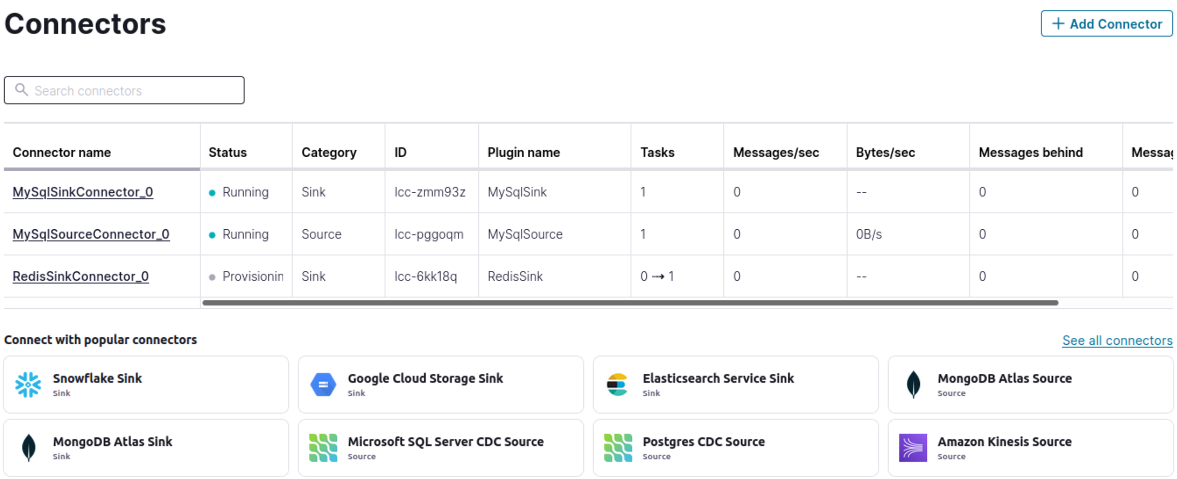
6. **Follow the following video to set up a Kafka connector to the deployed Redis database.**



```
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ mysql -uuser -pSOFE4630U -h35.193.165.175<<<"select * from myDB.TOMySQL;"
mysql: [Warning] Using a password on the command line interface can be insecure.
id      name    email   department      modified
15      user    test@gmail.com  IT      2022-03-18 14:39:37.845
15      esam    esam@gmail.com  IT      2022-03-18 14:51:45.556
15      mike    mike@gmail.com  IT      2022-03-18 14:52:04.632
pmihir2609@cloudshell:~ (galvanized-sled-344502)$
```

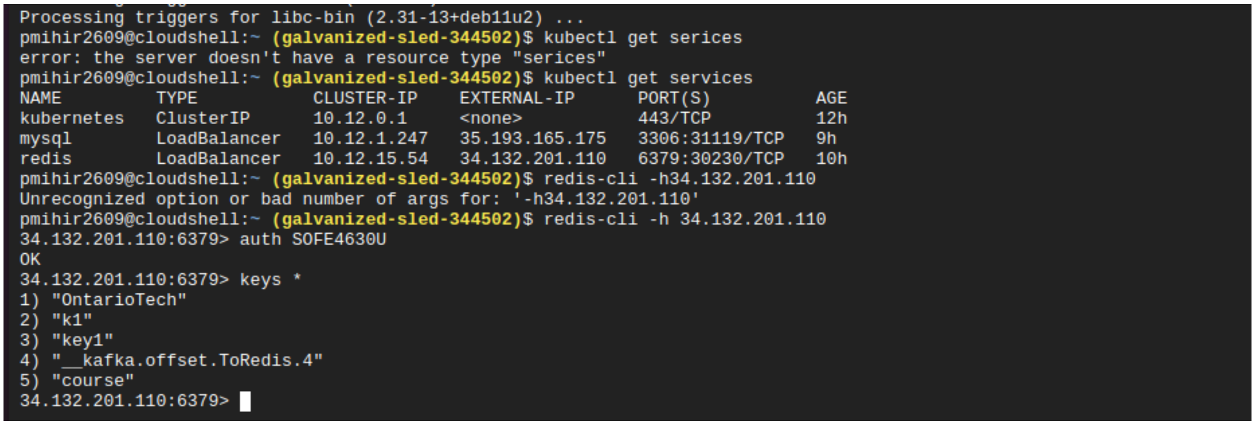


Connector name	Status	Category	ID	Plugin name	Tasks	Messages/sec	Bytes/sec	Messages behind	Messag
MySqlSinkConnector_0	● Running	Sink	lcc-zmm93z	MySqlSink	1	0	--	0	0
MySqlSourceConnector_0	● Running	Source	lcc-pggoqm	MySqlSource	1	0	0B/s	0	0

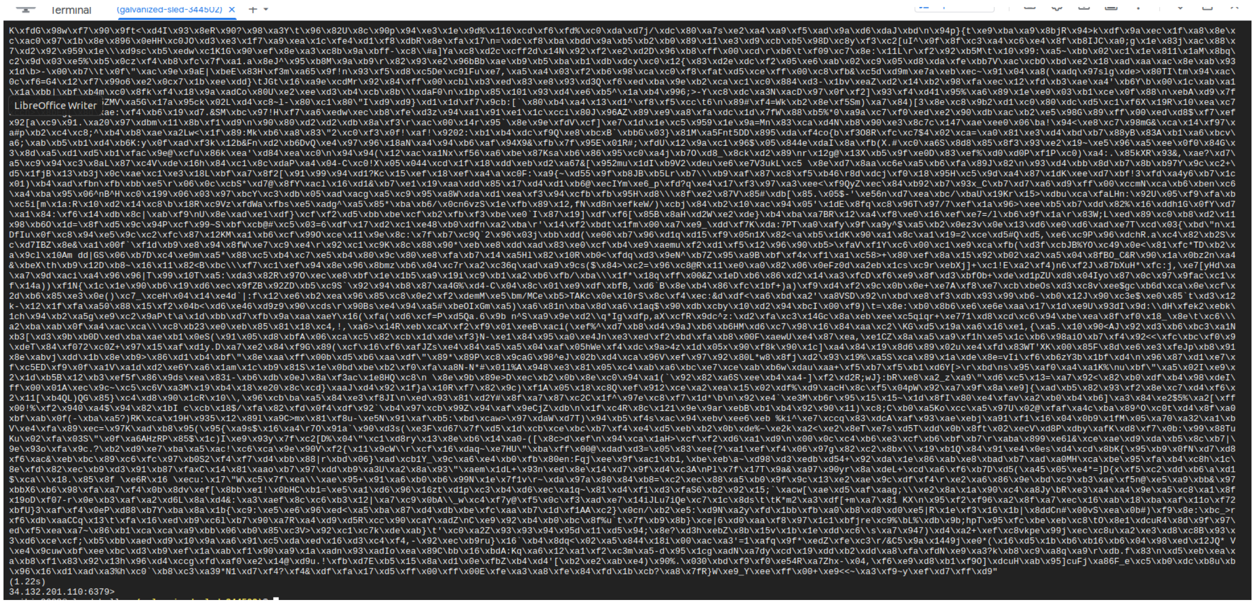


Connectors

Connector name	Status	Category	ID	Plugin name	Tasks	Messages/sec	Bytes/sec	Messages behind	Messag
MySqlSinkConnector_0	● Running	Sink	lcc-zmm93z	MySqlSink	1	0	--	0	0
MySqlSourceConnector_0	● Running	Source	lcc-pggoqm	MySqlSource	1	0	0B/s	0	0
RedisSinkConnector_0	● Provisionin	Sink	lcc-6kk18q	RedisSink	0 → 1	0	--	0	0



```
Processing triggers for libc-bin (2.31-13+deb11u2) ...
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ kubectl get serices
error: the server doesn't have a resource type "serices"
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ kubectl get services
NAME         TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)          AGE
kubernetes   ClusterIP      10.12.0.1      <none>           443/TCP          12h
mysql        LoadBalancer   10.12.1.247    35.193.165.175   3306:31119/TCP   9h
redis        LoadBalancer   10.12.15.54    34.132.201.110   6379:30230/TCP   10h
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ redis-cli -h34.132.201.110
Unrecognized option or bad number of args for: '-h34.132.201.110'
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ redis-cli -h 34.132.201.110
34.132.201.110:6379> auth SOFE4630U
OK
34.132.201.110:6379> keys *
1) "OntarioTech"
2) "k1"
3) "key1"
4) "__kafka.offset.ToRedis.4"
5) "course"
34.132.201.110:6379>
```



```
(1.22s)
34.132.201.110:6379>
```

7. Now, you will store a dataset into cloud storage. The dataset has to be sent into Kafka topics and connectors have to be configured to automatically store the dataset into the data storage. The producer that will send the dataset to Kafka topics should run on your local machine as it will simulate real sensors while Kafka, connectors, and data storage should be on the cloud. Use MySQL for the CSV files and Redis for images. Feel free to update the Yaml files from the given repository to fit your dataset

8. Use the sensors, images, ground truth Pose in the lastest session in http://robots.engin.umich.edu/nclt/ (Links to an external site.) as your dataset.

9. List some possible applications that can be implemented by using the uploaded dataset.