



## **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.

```
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$ echo "select * from myDB.test;" | kubectl exec -i mysql-7dcb5fd764-z7ng4 -- mysql -uuser -pSOF4630U
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
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$
```

```

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to galvanized-sled-344592.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
pmihir2699@cloudshell: (galvanized-sled-344592)$ gcloud config set compute/zone us-central1-a
Updated property [compute/zone].
pmihir2699@cloudshell: (galvanized-sled-344592)$ gcloud container clusters create gk-cluster
Default change: gke-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1590. 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 nodes(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-344592/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-344592
kubectl entry generated for gk-cluster.
NAME: gk-cluster
LOCATION: us-central1-a
MASTER VERSION: 1.21.0-gke.1593
MASTER IP: 34.68.170.189
MACHINE TYPE: e2-medium
NODE VERSION: 1.21.6-gke.1593
NUM_NODES: 3
STATUS: RUNNING
pmihir2699@cloudshell: (galvanized-sled-344592)$ kubectl get pvc
No resources found in default namespace.
pmihir2699@cloudshell: (galvanized-sled-344592)$ kubectl apply -f mysql-pvc.yaml
error: the path "mysql-pvc.yaml" does not exist
pmihir2699@cloudshell: (galvanized-sled-344592)$ ls
index.html  README-cloudshell.txt  SOF4630U-tut3
pmihir2699@cloudshell: (galvanized-sled-344592)$ cd SOF4630U-tut3/
pmihir2699@cloudshell:~/SOF4630U-tut3 (galvanized-sled-344592)$ cd GKE/
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$ kubectl apply -f mysql-pvc.yaml
persistentvolumeclaim/mysql-volumeclaim created
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$ kubectl get pvc
NAME          STATUS  VOLUME          CAPACITY  ACCESS MODES  STORAGECLASS  AGE
mysql-volumeclaim  Bound  pvc-1b0884ea-8942-47cf-9b34-06e74d4eac72  10Gi      RWO           standard      24s
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$ kubectl apply -f mysql-app.yaml
service/mysql created
deployment.apps/mysql created
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$ kubectl get pods
NAME          READY  STATUS  RESTARTS  AGE
mysql-7dcb5fd764-z7ng4  1/1    Running  0          25s
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$ kubectl get deployment
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
mysql         1/1    1           1          69s
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$ kubectl get services
NAME          TYPE          CLUSTER-IP  EXTERNAL-IP  PORT(S)  AGE
kubernetes    ClusterIP    10.12.0.1    <none>        443/TCP   7m5s
mysql         LoadBalancer 10.12.15.34  34.132.201.110 3306:31407/TCP 69s
pmihir2699@cloudshell:~/SOF4630U-tut3/GKE (galvanized-sled-344592)$
```

```

pmihir2609@cloudshell:~/S0FE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl delete -f mysql-app.yaml
error: the path "mysql-app.yaml" does not exist
pmihir2609@cloudshell:~/S0FE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl delete -f mysql-app.yaml
service "mysql" deleted
deployment.apps "mysql" deleted
pmihir2609@cloudshell:~/S0FE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl apply -f mysql-app.yaml
service/mysql created
deployment.apps/mysql created
pmihir2609@cloudshell:~/S0FE4630U-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:~/S0FE4630U-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:~/S0FE4630U-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:~/S0FE4630U-tut3/GKE (galvanized-sled-344502)$ mysql -user -pS0FE4630U -h35.193.165.175 <<< select * from 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:~/S0FE4630U-tut3/GKE (galvanized-sled-344502)$ mysql -user -pS0FE4630U -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:~/S0FE4630U-tut3/GKE (galvanized-sled-344502)$ kubectl delete -f mysql-app.yaml
service "mysql" deleted
deployment.apps "mysql" deleted
pmihir2609@cloudshell:~/S0FE4630U-tut3/GKE (galvanized-sled-344502)$

```

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

```

mihirkumar@mihirkumar-Precision-3520:~/Documents/cloudComputing/Milestone3/S0FE4630U-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.

```
CLOUD SHELL
Terminal (galvanized-sled-344502) x + -
CPU (%)
All services normal
Open Editor

pmihir2609@cloudshell:~ (galvanized-sled-344502)$ mysql -uuser -pS0FE4630U -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 14:51:45.850
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	Message
<a href="#">MySQLSinkConnector_0</a>	Running	Sink	lcc-zmm93z	MySQLSink	1	0	--	0	0
<a href="#">MySQLSourceConnector_0</a>	Running	Source	lcc-pggoqm	MySQLSource	1	0	0B/s	0	0

## Connectors

[+ Add Connector](#)

Connector name	Status	Category	ID	Plugin name	Tasks	Messages/sec	Bytes/sec	Messages behind	Message
<a href="#">MySQLSinkConnector_0</a>	Running	Sink	lcc-zmm93z	MySQLSink	1	0	--	0	0
<a href="#">MySQLSourceConnector_0</a>	Running	Source	lcc-pggoqm	MySQLSource	1	0	0B/s	0	0
<a href="#">RedisSinkConnector_0</a>	Provisionin	Sink	lcc-6kk18q	RedisSink	0 → 1	0	--	0	0


Connect with popular connectors

[See all connectors](#)

**Snowflake Sink**  
Sink

**Google Cloud Storage Sink**  
Sink

**Elasticsearch Service Sink**  
Sink

**MongoDB Atlas Source**  
Source

**MongoDB Atlas Sink**  
Sink

**Microsoft SQL Server CDC Source**  
Source

**Postgres CDC Source**  
Source

**Amazon Kinesis Source**  
Source

```
Processing triggers for libc-bin (2.31-13+deb11u2) ...
pmihir2609@cloudshell:~ (galvanized-sled-344502)$ kubectl get services
error: the server doesn't have a resource type "services"
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 S0FE4630U
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>
```

