**MongoDB and Big Query**

**Running a MongoDB Database in Kubernetes with Stateful Sets:**

In this lab we create a Kubernetes cluster, a headless service and a Stateful Set. Initially, we connect to cloud shell, create a new cluster, set up a Storage Class, headless service and Stateful Set. After that we deploy the headless service and stateful set in which change the configurations within the yaml file and save it. Later, we connect to the MongoDB Replica Set by initiating, viewing, and scaling it. At the end, we clean up the cluster and the deployed resources.

**Text

Description automatically generated**

*Figure 1: Activating the Cloud Shell*

**Text

Description automatically generated**

*Figure 2: Integrating with MongoDB and Setting up*

**Text

Description automatically generated**

*Figure 3: Instantiating the Storage Class and Deploying the Headless Service and Stateful Set*

**Text

Description automatically generated**

*Figure 4: Connecting to the Mongo DB Replica Set and Scaling it*

**Text

Description automatically generated**

*Figure 5: Cleaning Up the Cluster and Services*

**Ingesting New Datasets into Big Query:**

In this to begin with we make a modern dataset by including a csv record which has subtle elements around ecommerce items. We at that point run a query to investigate the recently stacked data with SQL. After that we ingest the information from Cloud Network, which is additionally a csv record and attempt to overwrite the past table from work history. Afterward, we ingest another dataset from google sheets. Run a query and upload it within the form of google sheets. In conclusion, we include a column to that spreadsheet i.e., the comments column, import it as a table within the big query and query it from the google spreadsheet.

**Graphical user interface, text, application, email, website

Description automatically generated**

*Figure 1: Ingesting a new Dataset from a CSV File*

**Graphical user interface, text, application

Description automatically generated**

*Figure 2: Schema of the Product Table*

**Graphical user interface, text, application, email

Description automatically generated**

*Figure 3: Exploring the newly loaded data with SQL*

**Graphical user interface, text, application, email

Description automatically generated**

*Figure 4: Ingesting data from Cloud Storage and overwriting it with job history*

**Graphical user interface, application, table, Excel

Description automatically generated**

*Figure 5: Google Sheet Data used for query*

**Graphical user interface, text, application, email

Description automatically generated**

*Figure 6: Ingesting a new dataset from Google Spreadsheet*

**Graphical user interface, text, application, email

Description automatically generated**

*Figure 7: Creating a new table product comments with the added column*

**Graphical user interface, text, application, email

Description automatically generated**

*Figure 8: Composing a new query on products comments table*

**1) For MongoDB, include your understanding of a headless service, Stateful Set, and why running a DB on a container is not a contradictory task?**

Headless Service:

When load balancing or single-service IP addresses are not required. A headless service is created and utilized to establish a service grouping. This does not provide an IP address or allow traffic to be forwarded. You can accomplish this by setting Cluster IP to "None" in the manifest file, which implies no cluster IP will be assigned. For instance, suppose MongoDB is hosted on a single pod. On top of that, we need a service definition to handle the pod restart. As well as for obtaining a new IP address. However, we do not require load balancing or routing. The service just needs to patch the request to the back-end pod. So, you utilize Headless Service since it doesn't have an IP address.

Stateful Set:

MongoDB/Kubernetes maintains a set of Stateful Sets with unique, permanent identities and consistent hostnames regardless of where they are scheduled. The state information and other resilient data for every Stateful Set Pod is kept in the Stateful Set's persistent disk storage. Stateful Sets are used to deploy stateful and clustered applications on persistent storage, such as Compute Engine persistent disks.

DB as a container:

Containers are executable software units in which application code, as well as its libraries and dependencies, are packed in standard ways so that they may be run anywhere, whether on a desktop, in traditional IT, or in the cloud. MongoDB instances may be operated using Docker. The user may construct a portable and extensible NoSQL database by using MongoDB as a container. Without having to worry about the underlying settings, a containerized MongoDB instance operates just like a non-containerized MongoDB instance.

**2) Pros and Cons of Big Query:**

**Pros:**

1) If you know how to execute SQL scripts, that's a huge advantage.

2) The capacity to store and administer numerous data warehouses is a significant advantage that benefits expanding enterprises significantly.

3) Easy connection with data warehouse and data management technologies like Data Studio and Google Analytics.

**Cons:**

1) You can't utilize it outside of Google's cloud infrastructure, which is a disadvantage if you wish to put it up locally.

2) Managing it can be a little pricey.

3) It's a bit tough to manage someone with less technical knowledge because SQL understanding of joins, CTEs, and other things is required.