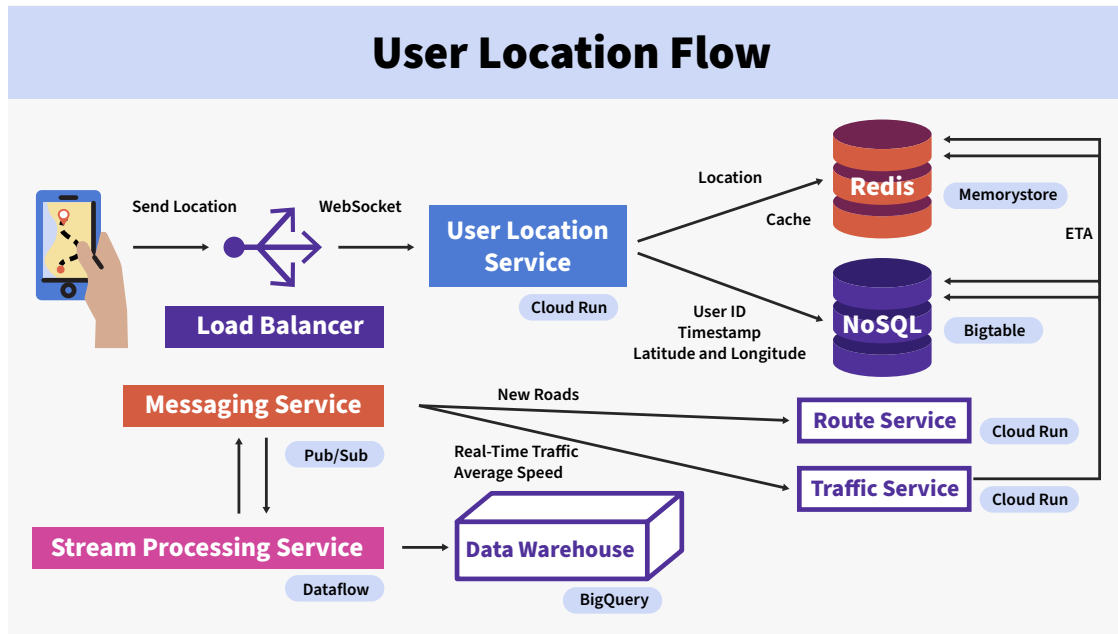


Map Service Architecture

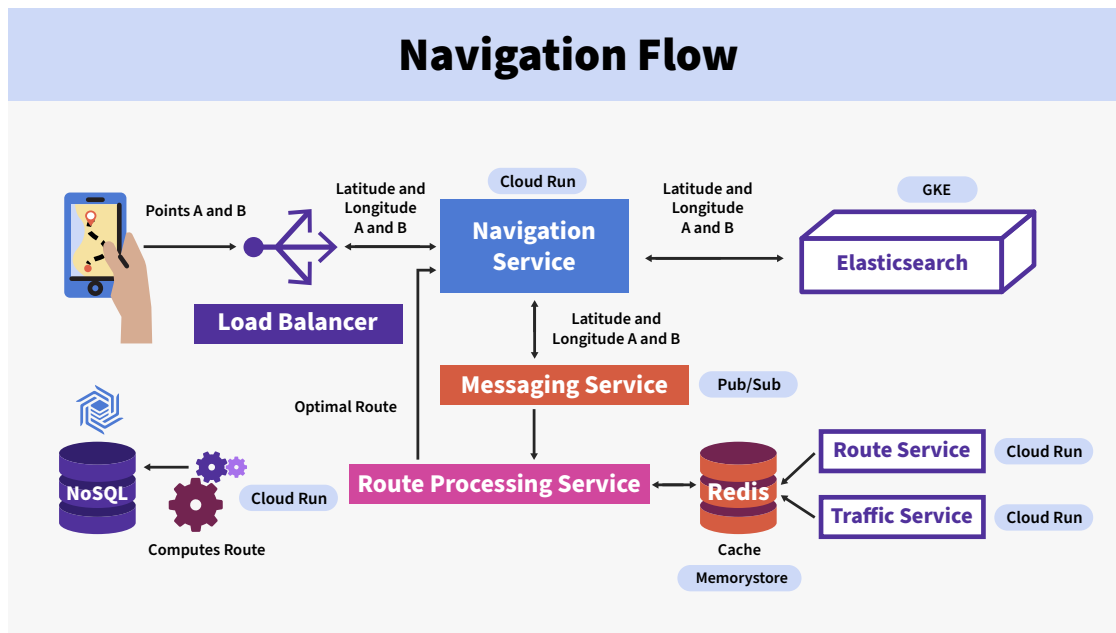
User Flow



1. The user location service is deployed on Cloud Run, and it receives regular pings every few seconds through the cloud load balancer. Cloud Run supports a persistent WebSocket connection.
2. To keep the connection information, I would use Memorystore for Redis for caching location information.
3. Our location service tracks user ID, timestamp, and latitude and longitude coordinates. It stores all that information in a nonrelational key-value permanent data store. Here, we use BigTable in Google Cloud.
4. Now, when it's getting location pings, it's also putting all those pings into a messaging service. Here, it's Pub/Sub in Google Cloud: a serverless messaging service that scales to millions of queries per second.
5. All the location pings that are going into this messaging service are read by a stream processing service. We use Dataflow (which is based on Apache Beam, is serverless, and scales seamlessly) for this stream processing.
6. It eventually puts that data into a data warehouse for further analysis. Here, we use BigQuery as a serverless data warehouse, where we can query the data and can even create machine learning models. Or, we can create our models in Vertex AI and the data can be ingested from BigQuery.

- The stream processing can update the “route service” with new roads, which we can deploy as a container in Cloud Run.
- It can also calculate average speed within an area, suggest real-time traffic information, and update the “traffic service” which we can deploy as a container in Cloud Run.
- Then, these route and traffic services will update our Bigtable and Memorystore so that the ETA can be predicted accurately based on this information.

Navigation Flow



- As a user, I type in my point A and point B. The request ends up on the cloud load balancer, and then finally on the “navigation service” running in a container on Cloud Run. This service is responsible for returning the latitude and longitude of points A and B.
- It does that using Elasticsearch; we will use managed Elastic service on GKE .
- The navigation service will also write those details to the messaging service for further processing. Now, we know that our messaging service in the cloud would be Pub/Sub.
- For the route processing service, we will use Cloud Run. This will return the optimal route to the user.
- It first looks for the route information within the cache in Memorystore, because route service and traffic service have been adding that in the database. If it finds the route in cache, it returns to the user.
- If it doesn't find it, then it computes the route, stores it in the Bigtable, and returns to the user.
- Now you know how you create a map service in Google Cloud!