Cangrejo SA Digital Transformation

A brief story of crab

Mykhailo Lieibenson - 31. March 2021

Scenario

A prospective customer, Cangrejo SA, a large multinational seafood retailer, approached us and asked for advice regarding modernising their cloud infrastructure and seeing if Aiven services could be used to help with their new needs.

Their current cloud setup is a B2B -facing Spring Boot web application that handles orders by stores and other retailers reselling their wares.

The application is a bit dated, a Spring Boot web app running on GCP Elastic Compute instances managed with puppet and using a CloudSQL Postgres 11 database.

Much of the data in the Postgres is typical webapp data updated by the customer, but the database also has a set of large tables related to pricing and assorted analytics that are updated once a day.

The daily update of those tables is done via a legacy service running on-premise that gathers and aggregates data from a large set of suppliers and affiliates and then pushed to the Postgres database.

Currently database queries on the webapp related to the large pricing/analytics tables can take up to 20 seconds and is a critical pain point for them that would need to be addressed fast.

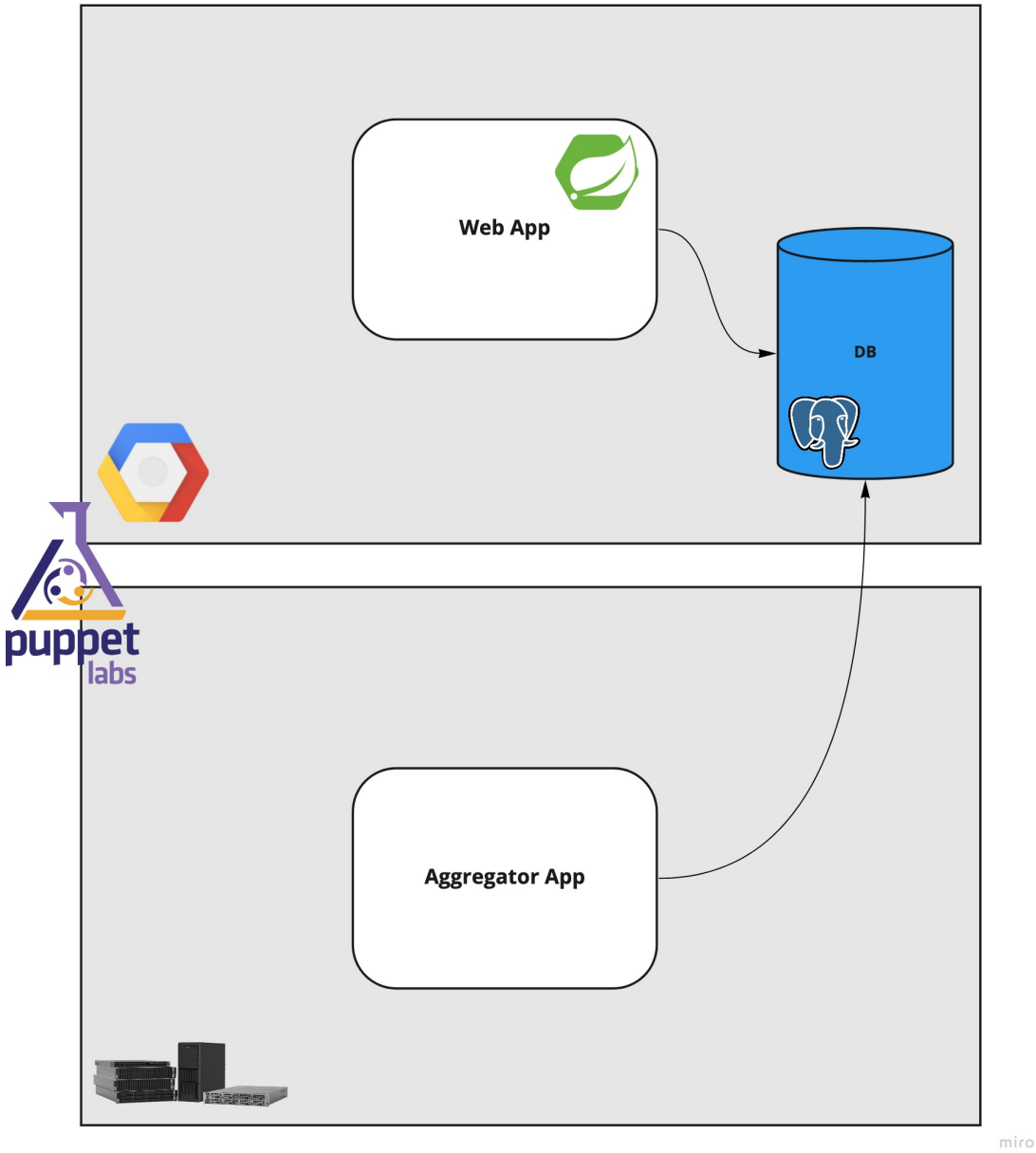
They have also recognised the need to update their infra to a more modern setup and are starting work on that, but will keep the current setup running until that is done.

Cangrejo SA intend to move all the rest of their services to a public cloud, but have not yet decided whether it will be GCP or something else. They are worried about vendor lock in, after a bad experience with a proprietary database provider in the '90s. Data security and GDPR compliance is another concern they want to make sure that is taken into account from the start.

As part of the modernization project, the company would like to further extend their business to logistics and expect to have to integrate more with third parties and be able to react faster to pricing and other business data updates. Their CIO is also hoping to improve their analytics handling once the initial webapp update has been done.

How would you go about helping Cangrejo SA get things running smoothly and planning a good setup for handling future business?

Status Quo



Basically as scenario describes we have puppet-orchestrated DC and GCP where two apps are running and writing to the same DB.

In general I consider it is anti-pattern to share same DB (not a server, but database) for multiple application due to potential data conflicts and interference between applications, but since this application is a bit dated and company is already taking actions to improve this situation this is not as bad as it sounds. No judging.

The Plan

So situation is the following:

1. We need to provide immediate pain relief for slow queries
2. We need to come up with the plan how client could improve situation

As far as slow queries concerned I would try to understand where time is spent the most. Is it on DB side or App side.
Some sort of APM could be useful, but if not available slow query log is nice as well.

Then if it turns out to be on DB side of things, I would try to understand the query itself. Maybe it could be optimised to avoid table locking.

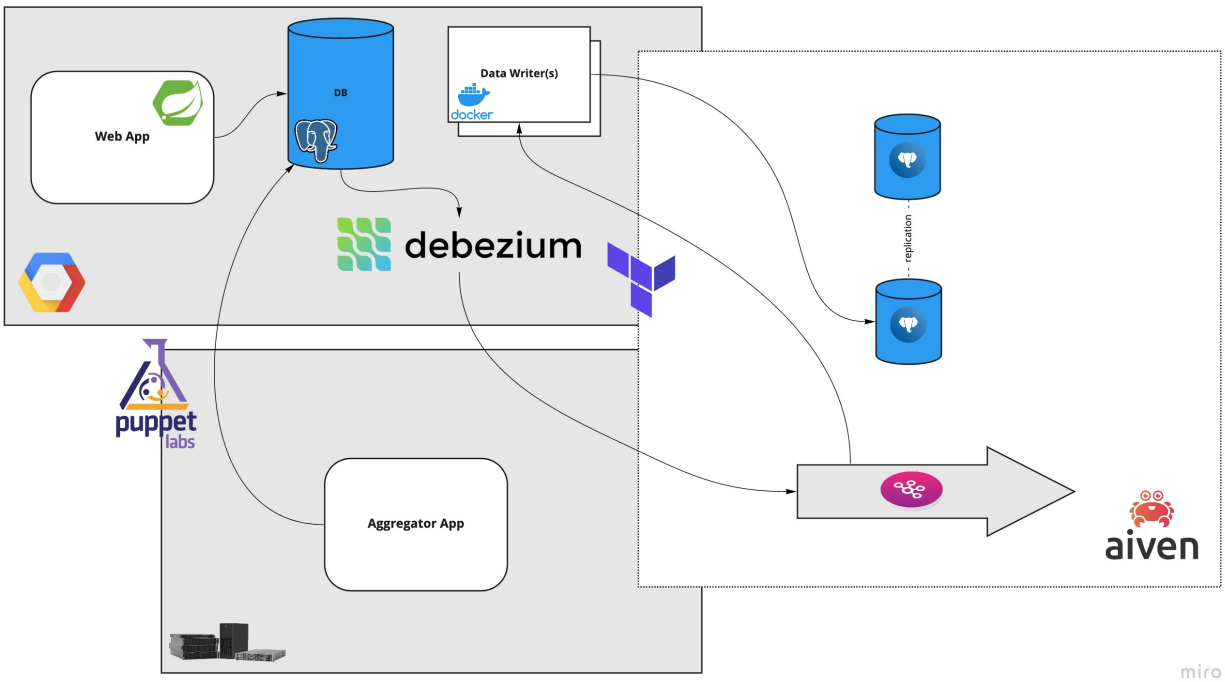
Another thing which could be useful is to look at amount of data in certain tables and maybe some soft of partitioning and/or sharding should be introduced.

Checking resources on DB server could be useful as well, i.e. disk I/O. Maybe faster SSD could be helpful. Or any other OS level optimisations.

Building some sort of caching layer be it on App side (i.e. in-mem cache or Redis) or DB side (materialised views) could definitely improve the situation.

Separation of reads and writes could be helpful as well and maybe we can do some sort of master-slave replication for DB where App will do reads from slave and Aggregator App can do writes to master.

Intermediate Step



Once slow DB queries situation has improved we could consider transition plan to the new setup while still maintaining old application.

For that I think we can do the following:

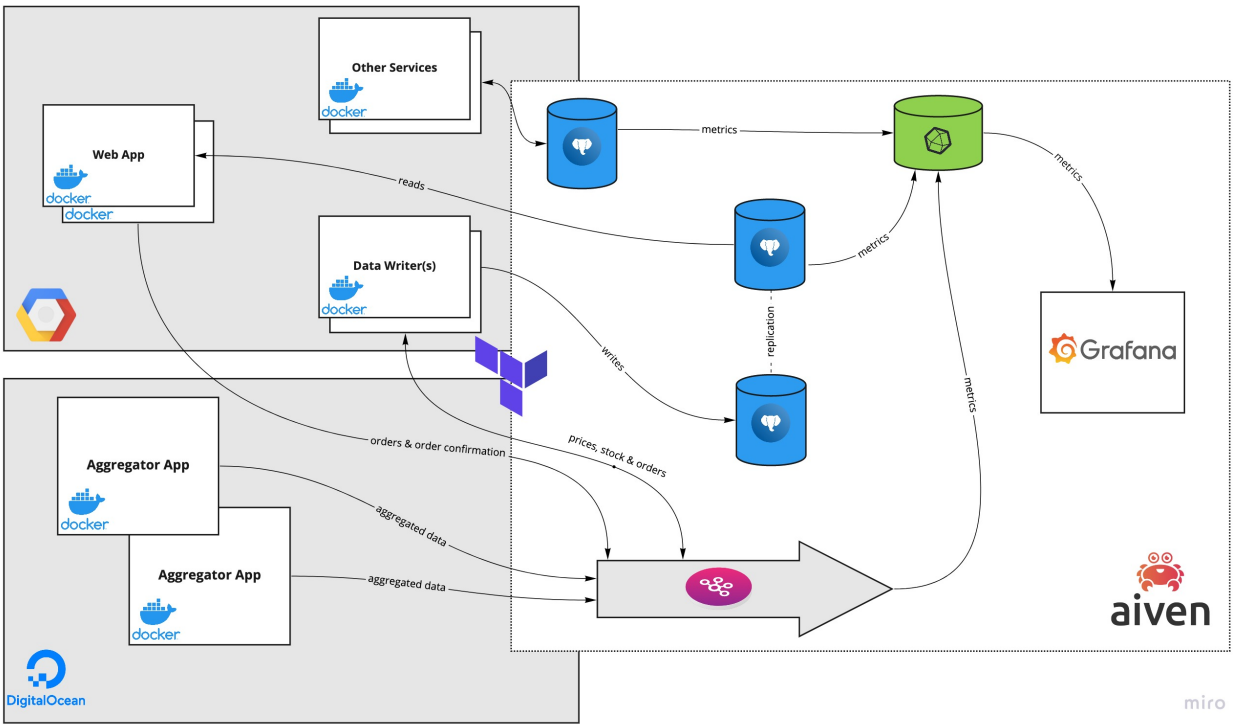
Provision new PG cluster with master-slave replication at Aiven.
Provision Apache Kafka cluster at Aiven.
Provisioning could be done using Terraform instead of Puppet.

Deploy Debezium to capture CDC events from the old DB and publish them to Kafka cluster.

Start writing “Data Writer” App which will consume those CDC events from Kafka, do some ETL to transform those events to completely new and improved schema and apply them to the newly created PG cluster.

This way we will keep old setup working while starting to replicate data for completely new setup which will set customer up for the success.

The Result



Once we are done with intermediate step and we have “Data Writer” application up and running we can be sure that data is being replicated and transformed in to the new data structure.

As next step we can replace old on-prem “Aggregator App” with set of new application which will crawl the supplies faster and in more concurrent way and publish stock and price updates directly to Kafka skipping old DB and Debezium setup.

“Data Writer” could consume these new events and write them to PG cluster.

Now we can deploy new version of “Web App” which can start reading data from the new PG cluster and place orders via Kafka so “Data Writer” could “write” them to PG cluster. This is sort of optional step because I think it could be beneficial to separate read and writes in CQRS style.

This approach will create microservice environment and will allow to have more async and decoupled components. Potentially it could increase concurrency and throughput of the system but it will not be real time. Also it will make system eventually consistent with higher availability. Strong consistency could be achieved at cost of other properties according to CAP theorem.

All infra components could push metrics to InfluxDB and Grafana could be used to visualise data and metrics.

New microservices could be added to this setup at any point and new databases could be created for them.

Since it was mentioned that customer is cautious about vendor lock for illustration purposes only I've added DO and GCP as public clouds to host their (micro-)services in form of docker containers. They could run them in k8s or by any other means.

Aiven can assure them that all components managed by Aiven could run in many more clouds and migration is possible upon request.

Thank you!