Assessment: Events log microservice

Jordi Neil Sánchez A

February 10, 2022

1 Summary

Microservice for storing and querying event logs. Developed using Python, FastAPI, and AWS.

2 Introduction

This PoC is aimed to be a short overview of my software development skills. This demo lets the user create a read events, those events have the following information:

1. Event Id
2. Description: A description of the event, it could by for example "Transaction of X amount." or "Customer closed his account".
3. Date: Event date.
4. Customer Id: Id of the customer whom is related with the event, e.g. the transaction or the account closing.
5. Event type: This field looks to be a easier way to classify the events, for example, all the transaction events should share the same type, e.g "Transaction".

In addition, exists the table Customer what is looking to normalize the database, what would helps to avoid of data duplicity or redundancy. You can see the database relations diagram on figure 1

3 Development

I tried to make it autoscalable, that's why I used an Fargate Instance and not a EC2 on the ECS. That means it cans scale if needed, but, it's not necesary for this PoC.

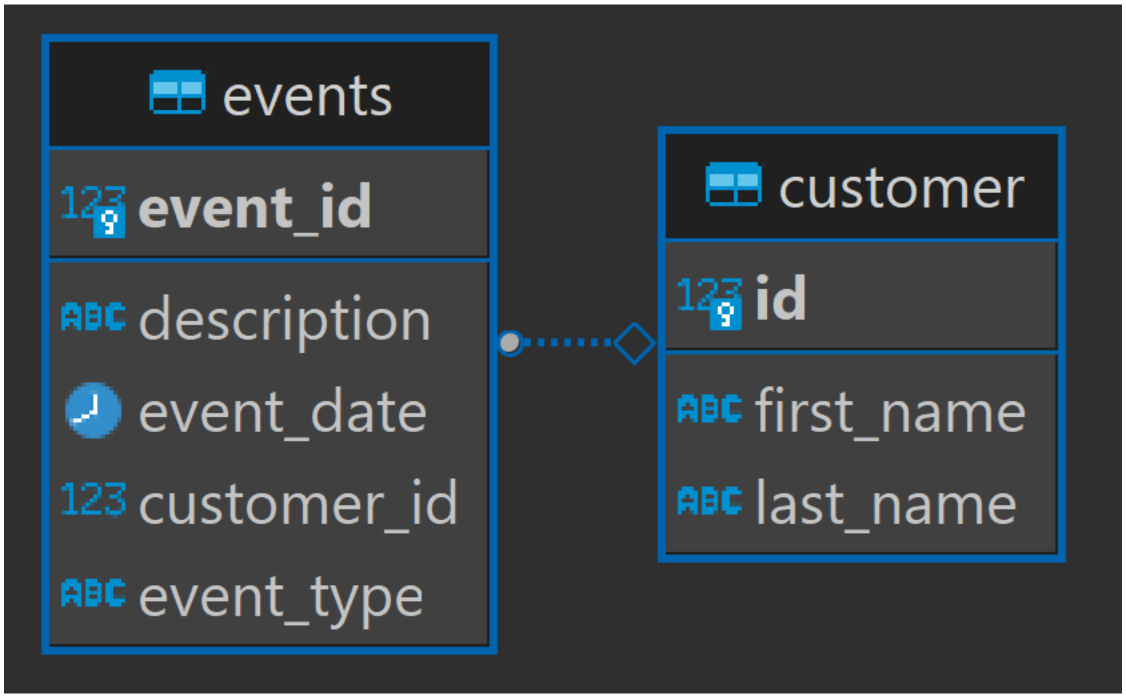

Figure 1: Database relations diagram

I also used Postgres because I have more familiarity working on it. I chose an SQL database because for this case, because we don't need really fast queries, it's going to be used by a person who wants to know the events logs. But, if this solution would scale to much more data, it would be good to implement a datawarehouse architecture, in that way we could store much more data without increasing so dramatically the costs. For example, RedShift could be a good solution, or Athena.

I also used a loadbalancer because it helps to balance the load between multiples instances and do not affect the response speed (in case this solution would be used by more people), and also give us a endpoint, it doesn't feel good to connect directly to an IP address.

ECR is used to storing the docker image. By the way, I used docker to make it easier to deploy and maintain.

For the API implementation I used FastAPI what is the easiest (and fastest) tool I know for creating APIs. It lets me create models for the requests and the response, that way is easier to declare how I want the response and requests to be, which fields are required and which are not. Also it lets me use libraries as SQLalchemy to connect with databases and use an ORM (Object-relational mapping) what make easier the queries executions (and avoid SQL injections).

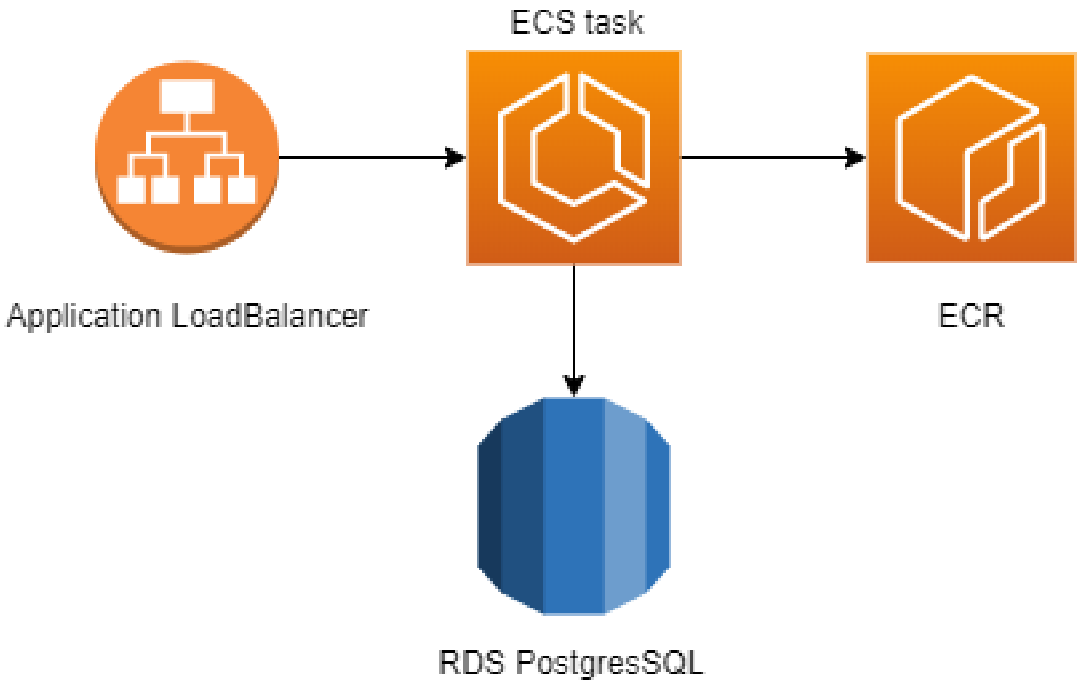

Figure 2: Solution architecture

4 Future work

For this application I would develop a way to could get events filtered by event_type. That way we could do further analysis about customers, e.g. what are the most commons events, how often they do different types of events. How often they create an account, the median time to close their accounts and more.

5 Conclusion

This PoC allows user to monitor customers events. If anyone wants to know if his account was closed, then getting the events by his customer id, we could see all the events and answer his question.