

# Improving query performance in PostGIS using a distributed environment

**Lukas Schilcher,  
Florian Steinwider,  
Daniel Roncelli**

19. Jänner 2023

# Table of Contents

1. Introduction
2. Setting up PostGIS on Kubernetes
3. Performance Comparison
4. Conclusion

# What is Kubernetes?

- container orchestration platform that automates processes involved in managing containerized applications
- clusters together groups of hosts running linux containers
- enables cloud-based development and makes it easier to transform and optimize existing applications



**kubernetes**

# overcoming single instance database challenges

- scalability: huge influx of concurrent requests / huge amounts of data
- high availability: software failure & hardware failure
- backup and restore: no downtime

## deploying on GKE - the hard way

1. configure kubernetes cluster ('kubectl' CLI tool / GCP interface)
2. build a custom PostGIS docker image
3. upload image to Artifact Registry/Github/Bitbucket
4. deploy image to the Cluster
5. configure autoscaling and load balancer service
6. expose application to the internet

## deploying on GKE - the easy way

1. configure kubernetes cluster ('kubectl' CLI tool / GCP interface)
2. deploy PostgreSQL app from marketplace via GCP interface
3. ssh into kubernetes pod and install PostGIS binaries
4. create PostGIS extension via sql-query

# challenges when running PostGIS on kubernetes

- harder to troubleshoot and pinpoint problems due to distributed environment
- plethora of tools needed to develop and deploy distributed clusters
- additional costs relative to single instanced db

# benchmarking

- PostgreSQL: EXPLAIN ANALYZE command
- for presentation: manual execution
- for report: python script



# single instance vs. kubernetes cluster specs

## single instance, Cloud SQL

- DB-Version:  
PostgreSQL 14.4
- CPU: 2 vCPUs
- memory: 8 GB

## GKE cluster

- DB-Version:  
PostgreSQL 15.1
- number of nodes: 3  
(no autoscaling)
- per node:
  - CPU: 2 vCPUs
  - memory: 4 GB

# results of performance tests - Union

```
explain analyze create table merged_census_blocks as  
select ST_AsText(ST_Union(geom)) as geom from nyc_census_blocks;
```

	Planning Time	Execution Time
Single Instance	0.071 ms	11808.699 ms
Kubernetes Cluster	0.058 ms	4936.106 ms

# results of performance tests - Convex Hull

```
explain analyze select ST_ConvexHull(ST_Union(st.geom))  
  from nyc_streets as st  
  join nyc_neighborhoods as nb  
  on ST_Contains(nb.geom, st.geom)  
group by nb.boriname  
having nb.boriname = 'Brooklyn';
```

	Planning Time	Execution Time
Single Instance	0.322 ms	591.110 ms
Kubernetes Cluster	0.410 ms	84.950 ms

# discussion

Q&A