Después de una larga ausencia en el blog, vamos a explicar una configuración para la base de datos PostgreSQL que pueda soportar altas cargas y aumentar la disponibilidad del servicio, comentaremos sus opciones y justificaremos los valores que decidimos aplicar en función de los recursos de la máquina.

Nuestra máquina es una servidor cloud que tiene 4 núcleos y 8G de memoria RAM, la máquina actualmente está compartida con contenedores docker de otros servicios por lo que en este contenedor vamos a consumir como máximo 3G de RAM y un pool de 250 conexiones, ya que el software que la usa (*Elixir* en este caso) es muy rápido y necesita de una alta disponibilidad, así mismo precisa de un alto número de conexiones concurrentes para no saturarla.

Opciones de configuración

Estas opciones las hemos sacado de la página https://pgtune.leopard.in.ua que nos ayudará a afinar la configuración en función de los valores de versión, núcleos, memoria y conexiones que necesitemos, hemos decidido usar el perfil *online transaction processing system* para nuestro caso.

```
# DB Version: 12
# OS Type: linux
# DB Type: oltp
# Total Memory (RAM): 3 GB
# CPUs num: 4
# Connections num: 250

max_connections = 250 # máximo número de conexiones concurrentes
shared_buffers = 768MB # cantidad de memoria dedicada a datos en caché
effective_cache_size = 2304MB # cantidad de memoria disponible para memoria intermedia en el disco
maintenance_work_mem = 192MB # cantidad de memoria usada para operaciones de mantenimiento
checkpoint_completion_target = 0.9 # permite escribir lentamente en la instancia:
checkpoint_completion_target * checkpoint_timeout (5min)
wal_buffers = 16MB # pequeña memoria que sincroniza los datos, aumentándola se permiten inserciones más
grandes
```

```
default_statistics_target = 100 # recolecta estadísticas de cada una de las tablas para decidir como se
ejecutarán las consultas sobre ellas
random_page_cost = 1.1 # sugiere al optimizador cuanto tiempo le llevará al disco encontrar una página
aleatoria de disco
effective_io_concurrency = 200 # número de operaciones de disco I/O concurrentes
work_mem = 1572kB # para operaciones complejas realiza ordenamientos más distendidos en memoria
min_wal_size = 2GB # cantidad mínima de la memoria usada para integridad de los datos
max_wal_size = 4GB # cantidad máxima de la memoria usada para integridad de los datos
max_worker_processes = 4 # cantidad máxima de procesos
max_parallel_workers_per_gather = 2 # cantidad máxima de subprocesos en paralelo por nodo, no puede exceder
a max_parallel_workers
max_parallel_workers = 4 # cantidad máxima de subprocesos en paralelo
```

Con esta configuración el contenedor de PostgreSQL 12 está preparado para trabajar con 250 conexiones concurrentes, pudiendo gestionar correctamente la carga de las consultas que se realicen y mantener la estabilidad del sistema a lo largo del tiempo.

Esperamos que os haya gustado, nos vemos en la próxima entrada, saludos