

Configuración PostgreSQL + WHM/cPanel para Aplicación Multi-usuario

1. Configuración de PostgreSQL

Archivo `postgresql.conf` (generalmente en `/var/lib/pgsql/data/`):

```
ini

# Conexiones (optimizado para servidor pequeño)
max_connections = 100          # Reducido para 4GB RAM
superuser_reserved_connections = 3

# Memoria (optimizado para 4GB RAM)
shared_buffers = 1GB          # 25% de tu RAM
effective_cache_size = 3GB     # 75% de tu RAM
work_mem = 8MB                # Para consultas grandes (reducido)
maintenance_work_mem = 128MB  # Reducido para 4GB RAM
max_worker_processes = 2      # Igual a núcleos CPU
max_parallel_workers = 2      # Igual a núcleos CPU
max_parallel_workers_per_gather = 1  # Reducido para 2 núcleos

# Escritura y WAL
wal_buffers = 16MB
checkpoint_completion_target = 0.9
random_page_cost = 1.1        # Para SSD
effective_io_concurrency = 200 # Para SSD

# Timeouts para consultas largas
statement_timeout = 300000     # 5 minutos
lock_timeout = 30000           # 30 segundos
idle_in_transaction_session_timeout = 600000 # 10 minutos

# Logging para debug
log_min_duration_statement = 1000 # Log queries > 1 segundo
log_line_prefix = '%t [%p]: [%l-1] user=%u,db=%d,app=%a,client=%h '
```

Archivo `pg_hba.conf` para conexiones:

```
ini
```

```
# Conexiones locales
```

```
local all all peer
host all all 127.0.0.1/32 md5
host all all ::1/128 md5
```

```
# Conexiones desde aplicación
```

```
host all tu_usuario localhost md5
host all tu_usuario 127.0.0.1/32 md5
```

2. Optimización de conexiones con Connection Pooling

Instalar y configurar PgBouncer:

```
bash
```

```
# Instalar PgBouncer
```

```
yum install pgbouncer -y
```

```
# Configurar /etc/pgbouncer/pgbouncer.ini
```

```
ini
```

```
[databases]
```

```
tu_database = host=localhost port=5432 dbname=tu_database
```

```
[pgbouncer]
```

```
listen_port = 6432
```

```
listen_addr = 127.0.0.1
```

```
auth_type = md5
```

```
auth_file = /etc/pgbouncer/userlist.txt
```

```
pool_mode = transaction
```

```
server_reset_query = DISCARD ALL
```

```
max_client_conn = 100 # Reducido para 4GB RAM
```

```
default_pool_size = 15 # Reducido para 2 núcleos
```

```
server_lifetime = 3600
```

```
server_idle_timeout = 600
```

3. Configuración de Apache/Nginx para PostgreSQL

En WHM → **Apache Configuration** → **Include Editor**:

apache

Timeouts más largos para consultas PostgreSQL

Timeout 300

ProxyTimeout 300

ProxyBadHeader Ignore

Para aplicaciones PHP con PostgreSQL

<IfModule mod_php.c>

php_value pgsql.allow_persistent On

php_value pgsql.max_persistent 20

php_value pgsql.max_links 30

</IfModule>

4. Configuración de PHP para PostgreSQL

En WHM → **MultiPHP INI Editor**:

ini

Configuración PostgreSQL (optimizado para servidor pequeño)

pgsql.allow_persistent = On

pgsql.auto_reset_persistent = Off

pgsql.max_persistent = 10 # Reducido para 4GB RAM

pgsql.max_links = 15 # Reducido para 4GB RAM

pgsql.ignore_notice = Off

pgsql.log_notice = Off

; Para PDO PostgreSQL

pdo_pgsql.ignore_notice = Off

; Memoria y tiempo de ejecución (optimizado para 4GB RAM)

memory_limit = 256M # Reducido para 4GB RAM

max_execution_time = 300

max_input_time = 300

; Uploads de archivos

upload_max_filesize = 8M

post_max_size = 10M

max_file_uploads = 20

; Sesiones optimizadas

session.gc_maxlifetime = 3600

session.gc_probability = 1

session.gc_divisor = 1000

; OPcache para mejor rendimiento (optimizado para 4GB RAM)

opcache.enable = 1

opcache.memory_consumption = 128 # Reducido para 4GB RAM

opcache.max_accelerated_files = 10000 # Reducido

opcache.validate_timestamps = 0

opcache.revalidate_freq = 0

5. Configuración de índices y mantenimiento

Crea un script de mantenimiento `/root/pg_maintenance.sh`:

```
bash
```

```
#!/bin/bash
```

```
# Mantenimiento diario de PostgreSQL
```

```
# Variables
```

```
DB_NAME="tu_database"
```

```
DB_USER="tu_usuario"
```

```
# Vacuum y analyze automático
```

```
sudo -u postgres psql -d $DB_NAME -c "VACUUM ANALYZE;"
```

```
# Reindexar si es necesario
```

```
sudo -u postgres psql -d $DB_NAME -c "REINDEX DATABASE $DB_NAME;"
```

```
# Estadísticas
```

```
sudo -u postgres psql -d $DB_NAME -c "SELECT schemaname,tablename,n_tup_ins,n_tup_upd,n_tup_del FROM pg_stat_statements;"
```

Agregar a cron:

```
bash
```

```
# Ejecutar a las 2 AM diariamente
```

```
0 2 * * * /root/pg_maintenance.sh
```

6. Optimización de consultas grandes

Configurar límites en la aplicación:

```
sql
```

```
-- Para consultas que toman mucho tiempo
```

```
SET statement_timeout = '5min';
```

```
SET lock_timeout = '30s';
```

```
-- Para consultas con muchos datos
```

```
SET work_mem = '32MB';
```

```
SET temp_buffers = '32MB';
```

Índices recomendados para archivos:

```
sql
```

```
-- Para tablas que almacenan metadatos de archivos
```

```
CREATE INDEX idx_files_user_id ON files(user_id);
```

```
CREATE INDEX idx_files_upload_date ON files(upload_date);
```

```
CREATE INDEX idx_files_size ON files(file_size);
```

```
CREATE INDEX idx_files_type ON files(file_type);
```

```
-- Índice compuesto para consultas frecuentes
```

```
CREATE INDEX idx_files_user_date ON files(user_id, upload_date DESC);
```

7. Configuración para archivos grandes

Estructura de tabla recomendada:

```
sql
```

```
CREATE TABLE files (
```

```
  id BIGSERIAL PRIMARY KEY,
```

```
  user_id INTEGER NOT NULL,
```

```
  filename VARCHAR(255) NOT NULL,
```

```
  file_path TEXT NOT NULL,
```

```
  file_size INTEGER NOT NULL,
```

```
  file_type VARCHAR(50) NOT NULL,
```

```
  mime_type VARCHAR(100),
```

```
  upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
  metadata JSONB,
```

```
  checksum VARCHAR(64)
```

```
);
```

```
-- Índices para búsquedas rápidas
```

```
CREATE INDEX idx_files_user_id ON files(user_id);
```

```
CREATE INDEX idx_files_metadata ON files USING GIN(metadata);
```

8. Monitoreo de PostgreSQL

Script de monitoreo `/root/pg_monitor.sh`:

```
bash
```

```
#!/bin/bash
```

```
# Monitoreo de PostgreSQL
```

```
echo "=== Conexiones activas ==="
```

```
sudo -u postgres psql -c "SELECT count(*) FROM pg_stat_activity;"
```

```
echo "=== Consultas lentas ==="
```

```
sudo -u postgres psql -c "SELECT query, state, query_start FROM pg_stat_activity WHERE state = 'active' AND query_start < NOW() - INTERVAL '10 seconds';"
```

```
echo "=== Tamaño de bases de datos ==="
```

```
sudo -u postgres psql -c "SELECT datname, pg_size_pretty(pg_database_size(datname)) FROM pg_database;"
```

```
echo "=== Locks ==="
```

```
sudo -u postgres psql -c "SELECT mode, count(*) FROM pg_locks GROUP BY mode;"
```

9. Configuración de backup para PostgreSQL

```
bash
```

```
#!/bin/bash
```

```
# Backup automatizado
```

```
DB_NAME="tu_database"
```

```
BACKUP_DIR="/backup/postgresql"
```

```
DATE=$(date +%Y%m%d_%H%M%S)
```

```
# Crear directorio si no existe
```

```
mkdir -p $BACKUP_DIR
```

```
# Backup completo
```

```
sudo -u postgres pg_dump -Fc $DB_NAME > $BACKUP_DIR/backup_${DB_NAME}_${DATE}.dump
```

```
# Limpiar backups antiguos (mantener 7 días)
```

```
find $BACKUP_DIR -name "backup_${DB_NAME}_*.dump" -mtime +7 -delete
```

10. Configuración del sistema operativo

```
bash
```

```
# Optimizar para PostgreSQL
```

```
echo "kernel.shmmax = 17179869184" >> /etc/sysctl.conf
```

```
echo "kernel.shmall = 4194304" >> /etc/sysctl.conf
```

```
echo "vm.overcommit_memory = 2" >> /etc/sysctl.conf
```

```
echo "vm.overcommit_ratio = 80" >> /etc/sysctl.conf
```

```
# Aplicar cambios
```

```
sysctl -p
```

```
# Aumentar límites del sistema
```

```
echo "* soft nofile 65536" >> /etc/security/limits.conf
```

```
echo "* hard nofile 65536" >> /etc/security/limits.conf
```

```
# Optimizar kernel
```

```
echo "net.core.somaxconn = 4096" >> /etc/sysctl.conf
```

```
echo "net.ipv4.tcp_max_syn_backlog = 4096" >> /etc/sysctl.conf
```

11. Configuración de WHM/cPanel

En WHM → Server Configuration → Tweak Settings:

Max cPanel processes: 4

Max email per hour: 100

Max defer fail percentage: 25

Max hourly email per domain: 250

Apache MaxClients: 400

Apache ServerLimit: 16

Para cada cuenta en WHM → Account Functions → Modify Account:

Disk Space: Unlimited o según necesidades

Bandwidth: Unlimited

Max Databases: 10-20

Max Email Accounts: 50

Max Subdomains: 10

12. Configuración de .htaccess para la aplicación

apache

```
# Timeouts para uploads (optimizado para servidor pequeño)
php_value max_execution_time 300
php_value max_input_time 300
php_value memory_limit 256M      # Reducido para 4GB RAM
php_value upload_max_filesize 8M
php_value post_max_size 10M
```

```
# Cache para archivos estáticos
<IfModule mod_expires.c>
    ExpiresActive On
    ExpiresByType image/jpeg "access plus 1 month"
    ExpiresByType image/png "access plus 1 month"
    ExpiresByType image/gif "access plus 1 month"
</IfModule>
```

```
# Configuración PostgreSQL específica (optimizada)
php_value pgsql.allow_persistent On
php_value pgsql.max_persistent 10    # Reducido para 4GB RAM
php_value pgsql.max_links 15        # Reducido para 4GB RAM
```

13. Configuración de correo electrónico (SOLUCIÓN DMARC)

Configuración DNS para correo desde subdominio:

IP del VPS: 184.168.20.238

1. Modificar registro DMARC existente:

Tipo: TXT
Nombre: _dmarc
Valor: v=DMARC1; p=reject; sp=none; adkim=r; aspf=r; rua=mailto:dmarc-rua@onsecureserver.net;

2. Agregar registro SPF para el subdominio:

Tipo: TXT
Nombre: hcniep
Valor: v=spf1 ip4:184.168.20.238 ~all

3. Configurar DKIM en WHM:

- Ve a **Email Authentication**
- Habilita DKIM para `hcniep.redsummacloud.com`
- Copia la clave pública generada

4. Agregar registro DKIM al DNS:

Tipo: TXT

Nombre: default._domainkey.hcniep

Valor: [La clave pública que generó WHM]

14. Comandos útiles de mantenimiento

bash

Verificar estado de PostgreSQL

`systemctl status postgresql`

Reiniciar PostgreSQL

`systemctl restart postgresql`

Ver logs de PostgreSQL

`tail -f /var/lib/pgsql/data/log/postgresql-*.log`

Conectar a PostgreSQL

`sudo -u postgres psql`

Ver conexiones activas

`sudo -u postgres psql -c "SELECT * FROM pg_stat_activity;"`

Verificar configuración DNS

`nslookup -type=TXT hcniep.redsummacloud.com`

`nslookup -type=TXT _dmarc.redsummacloud.com`

`nslookup -type=TXT default._domainkey.hcniep.redsummacloud.com`

15. Checklist de implementación

- ☐ Configurar postgresql.conf
- ☐ Configurar pg_hba.conf
- ☐ Instalar y configurar PgBouncer
- ☐ Configurar PHP para PostgreSQL
- ☐ Crear scripts de mantenimiento
- ☐ Configurar cron jobs
- ☐ Optimizar sistema operativo
- ☐ Configurar DNS para correo
- ☐ Habilitar DKIM en WHM
- ☐ Configurar backups automatizados
- ☐ Configurar monitoreo
- ☐ Probar envío de correos
- ☐ Verificar rendimiento de la aplicación

16. Configuración específica para tu servidor (2 núcleos, 4GB RAM, 100GB)

Configuración adicional de PostgreSQL para hardware limitado:

```
ini

# Configuración específica para servidor pequeño
wal_level = replica
max_wal_size = 1GB          # Reducido para 100GB espacio
min_wal_size = 80MB
checkpoint_timeout = 15min  # Más frecuente para hardware limitado
archive_mode = off          # Desactivar si no necesitas replicación

# Configuración de memoria más conservadora
huge_pages = off            # Desactivar para 4GB RAM
temp_file_limit = 1GB       # Limitar archivos temporales
```

Configuración de Apache optimizada para 2 núcleos:

```
apache

# Configuración específica para 2 núcleos
ServerLimit 4                # Reducido para 2 núcleos
MaxRequestWorkers 100        # Reducido para 4GB RAM
ThreadsPerChild 25
MinSpareThreads 25
MaxSpareThreads 75
ThreadLimit 64
StartServers 2
```

Configuración de sistema operativo optimizada:

```
bash
```

```
# Optimizar swap para 4GB RAM
```

```
echo "vm.swappiness = 10" >> /etc/sysctl.conf
```

```
echo "vm.vfs_cache_pressure = 50" >> /etc/sysctl.conf
```

```
# Configuración de memoria compartida para 4GB RAM
```

```
echo "kernel.shmmax = 1073741824" >> /etc/sysctl.conf # 1GB
```

```
echo "kernel.shmall = 262144" >> /etc/sysctl.conf # 1GB/4KB
```

Límites recomendados para usuarios concurrentes:

- **Usuarios concurrentes máximos recomendados:** 20-30
- **Consultas simultáneas pesadas:** 5-8
- **Archivos simultáneos uploading:** 10-15
- **Espacio para logs:** 5GB máximo
- **Backups:** Rotar cada 3 días (en lugar de 7)

Script de monitoreo de recursos específico:

```
bash
```

```
#!/bin/bash
```

```
# Monitoreo específico para servidor pequeño
```

```
echo "=== Uso de memoria ==="
```

```
free -h
```

```
echo "=== Uso de CPU ==="
```

```
top -bn1 | grep "Cpu(s)"
```

```
echo "=== Uso de disco ==="
```

```
df -h
```

```
echo "=== Procesos PostgreSQL ==="
```

```
ps aux | grep postgres | wc -l
```

```
echo "=== Conexiones PostgreSQL ==="
```

```
sudo -u postgres psql -c "SELECT count(*) as conexiones_activas FROM pg_stat_activity WHERE state = 'active';"
```

```
# Alerta si el uso de memoria es alto
```

```
MEM_USAGE=$(free | grep Mem | awk '{printf "%.0f", $3/$2 * 100.0}')
```

```
if [ $MEM_USAGE -gt 85 ]; then
```

```
    echo "ALERTA: Uso de memoria alto: ${MEM_USAGE}%"
```

```
fi
```