

# ALTA DISPONIBILIDAD WEB

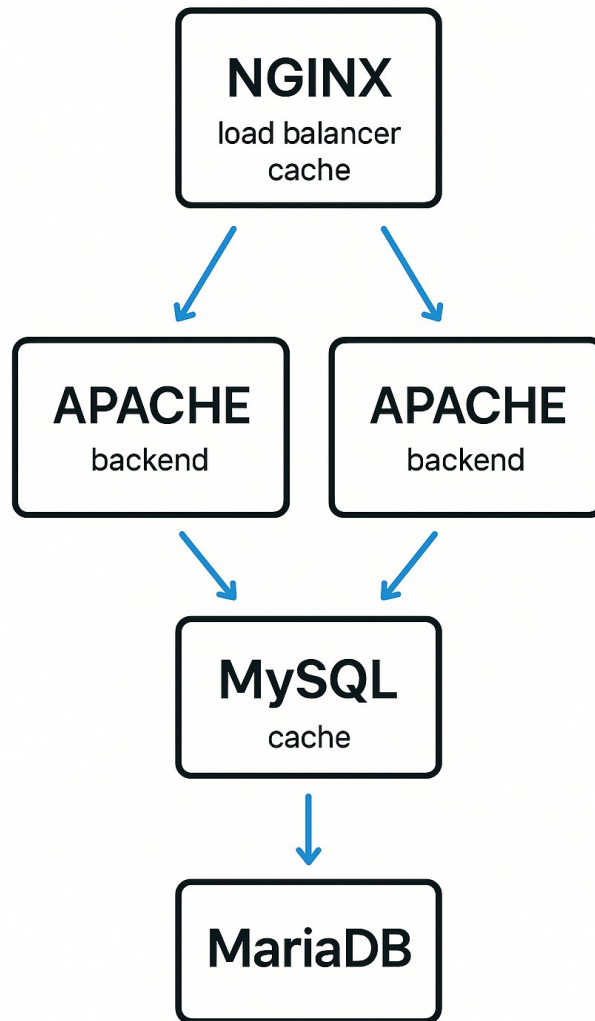
BALANCEO DE CARGA Y CACHE DE ACCESO A BASE DE DATOS



## Índice de contenidos

|  |    |
|--|----|
| 1. Arquitectura del sistema.....         | 3  |
| 1.1. Interfaces de red.....              | 3  |
| 1.2. Flujo del tráfico.....              | 3  |
| 2. Balanceador de carga y web caché..... | 4  |
| 2.1. Estructura de archivos.....         | 4  |
| 2.2. Archivos de configuración.....      | 4  |
| 2.3. Análisis de la configuración.....   | 6  |
| 2.4. Ajuste del servicio.....            | 6  |
| 3. Servidores web.....                   | 7  |
| 4. Proxy-Cache de base de datos.....     | 8  |
| 4.1. Instalación.....                    | 8  |
| 4.2. Configuración.....                  | 8  |
| 4.3. Verificación y monitorización.....  | 9  |
| 5. Servidor de bases de datos.....       | 10 |

# 1. Arquitectura del sistema



## 1.1. Interfaces de red

| <b>nginx</b>     | <b>apache 2</b> | <b>apache2</b> | <b>proxymysql</b> | <b>mariadb</b> |
|------------------|-----------------|----------------|-------------------|----------------|
| 192.168.59.40/24 | 192.168.59.10   | 192.168.59.20  | 192.168.59.30     | 192.168.59.100 |
| 172.20.0.40/24   | 172.20.0.10     | 172.20.0.20    | 172.21.0.30       | 172.22.0.100   |
|                  | 172.21.0.10     | 172.21.0.20    | 172.22.0.30       |                |

## 1.2. Flujo del tráfico

```
Cliente --> NGINX (192.168.59.40)
           |
           |> Apache2-1 (172.20.0.10)
           |> Apache2-2 (172.20.0.20)
           |
           |> ProxySQL (172.21.0.30)
           |
           |> MariaDB (172.22.0.100)
```

## 2. Balanceador de carga y web caché

### 2.1. Estructura de archivos

```
/etc/nginx/  
├── nginx.conf          (Configuración principal)  
├── conf.d/            (Configuraciones adicionales)  
│   ├── proxy.conf    (Configuración específica del proxy)  
├── sites-available/  (Configuraciones de sitios disponibles)  
│   └── default        (Tu configuración de proxy)  
├── sites-enabled/    (Sitios activos - enlaces simbólicos)  
│   └── default -> ../sites-available/default  
└── snippets/         (Fragmentos reutilizables)
```

### 2.2. Archivos de configuración

```
apt update && sudo apt install -y nginx
```

```
nano /etc/nginx/nginx.conf
```

```
user www-data;  
worker_processes auto;  
pid /run/nginx.pid;  
  
events {  
    worker_connections 1024;  
    multi_accept on; # Mejora para aceptar múltiples conexiones a la vez  
    use epoll;      # Mejor método de polling para Linux  
}  
  
http {  
    include      /etc/nginx/mime.types;  
    default_type application/octet-stream;  
  
    # Configuración básica de logs, compresión, etc.  
    log_format main '$remote_addr - $remote_user [$time_local] "$request" ' '  
                    '$status $body_bytes_sent "$http_referer" ' '  
                    '"$http_user_agent" "$http_x_forwarded_for"';  
  
    access_log /var/log/nginx/access.log main buffer=32k flush=1m;  
    error_log  /var/log/nginx/error.log warn;  
  
    # Optimizaciones de rendimiento  
    sendfile on;  
    tcp_nopush on;  
    tcp_nodelay on;  
    keepalive_timeout 65;  
    keepalive_requests 1000;  
    server_tokens off;  
  
    # Compresión  
    gzip on;  
    gzip_comp_level 5;  
    gzip_min_length 256;  
    gzip_types text/plain text/css application/json application/javascript text/xml  
    application/xml application/xml+rss text/javascript;  
  
    include /etc/nginx/conf.d/*.conf;  
    include /etc/nginx/sites-enabled/*;  
}
```

## nano /etc/nginx/sites-available/default

```
# Configuración de caché
proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=cache_backend:100m inactive=60m
use_temp_path=off;

# Configuración de upstream con health checks
upstream backend_servers {
    least_conn; # Balanceo por menor conexión en lugar de round-robin
    server 172.20.0.10:80 max_fails=3 fail_timeout=30s;
    server 172.20.0.20:80 max_fails=3 fail_timeout=30s;

    # Mejora: mantener conexiones persistentes con los backends
    keepalive 32;
}

# Configuración del servidor virtual
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    server_name example.com www.example.com; # Cambiar por tu dominio real

    # Mejora: manejo de favicon y robots.txt
    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt { access_log off; log_not_found off; }

    location / {
        proxy_pass http://backend_servers;
        proxy_http_version 1.1; # Necesario para keepalive

        # Headers
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Connection "";

        # Timeouts
        proxy_connect_timeout 5s;
        proxy_read_timeout 30s;
        proxy_send_timeout 30s;

        # Configuración de caché
        proxy_cache cache_backend;
        proxy_cache_valid 200 302 10m;
        proxy_cache_valid 404 1m;
        proxy_cache_use_stale error timeout updating http_500 http_502 http_503 http_504;
        proxy_cache_background_update on;
        proxy_cache_lock on;
        proxy_cache_lock_timeout 5s;

        # Headers para el cliente
        add_header X-Cache-Status $upstream_cache_status;
        add_header X-Backend $upstream_addr;

        # Mejora: reintentos en caso de error
        proxy_next_upstream error timeout invalid_header http_500 http_502 http_503
http_504;
        proxy_next_upstream_tries 3;
        proxy_next_upstream_timeout 10s;
    }

    # Mejora: endpoint para health checks
    location /nginx-health {
        access_log off;
    }
}
```

```
        return 200 "OK";  
        add_header Content-Type text/plain;  
    }  
}
```

## 2.3. Análisis de la configuración

### Configuración Global

`worker_processes auto`: Automáticamente configura tantos workers como CPUs tenga el servidor.  
`worker_connections 1024`: Número máximo de conexiones por worker (podría aumentarse si el servidor tiene mucha RAM).  
`multi_accept on`: Permite aceptar múltiples conexiones a la vez, mejorando rendimiento.  
`use epoll`: Método eficiente para manejar conexiones en Linux.

### Optimizaciones HTTP

`sendfile on`: Usa el sistema operativo para enviar archivos directamente, mejorando rendimiento.  
`tcp_nopush on + tcp_nodelay on`: Optimizaciones para el envío de paquetes TCP.  
`keepalive_timeout 65`: Tiempo que se mantiene una conexión abierta para peticiones posteriores.  
`server_tokens off`: Oculta la versión de Nginx por seguridad.

### Upstream (Balanceo de Carga)

`least_conn`: Estrategia de balanceo que envía tráfico al servidor con menos conexiones activas.  
`max_fails=3 fail_timeout=30s`: Después de 3 fallos en 30 segundos, marca el servidor como no disponible temporalmente.  
`keepalive 32`: Mantiene conexiones persistentes con los backends, mejorando rendimiento.

### Configuración de Proxy

`proxy_http_version 1.1`: Necesario para conexiones persistentes.  
`proxy_set_header`: Configura headers importantes para los backends.  
`proxy_cache_*`: Configuración detallada del caché con bloqueo y actualizaciones en background.  
`proxy_next_upstream`: Reintenta con otro servidor si hay errores.

### Seguridad y Monitoreo

`server_name`: Debe especificarse con dominios reales en producción.  
Health check endpoint (`/nginx-health`) para monitoreo.

## 2.4. Ajuste del servicio

```
sudo mkdir -p /var/cache/nginx
```

```
sudo chown www-data:www-data /var/cache/nginx
```

```
sudo systemctl restart nginx.service
```

### 3. Servidores web

```
sudo apt install apache2 php libapache2-mod-php php-mysql mariadb-client
```

```
sudo rm /var/www/html/index.html
```

```
sudo nano /var/www/html/index.php
```

```
<?php
$host = '172.21.0.30';
$port = 6033;
$dbname = 'webdb'; // Puedes cambiarlo si creas otro
$user = 'webuser';
$pass = 'webpass';

try {
    $pdo = new PDO("mysql:host=$host;port=$port;dbname=$dbname", $user, $pass);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $stmt = $pdo->query("SELECT NOW() AS ahora");
    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    echo "<h1>Conexión exitosa a través de ProxySQL</h1>";
    echo "<p>Hora actual desde MariaDB: " . htmlspecialchars($row['ahora']) . "</p>";
} catch (PDOException $e) {
    echo "<h1>Error de conexión</h1>";
    echo "<p>" . htmlspecialchars($e->getMessage()) . "</p>";
}

echo '<p>Respuesta obtenida desde: '.$_SERVER['SERVER_ADDR'].'</p>';
?>
```

```
sudo chown www-data:www-data /var/www/html/index.php
```

## 4. Proxy-Cache de base de datos

### 4.1. Instalación

```
sudo apt-get install -y --no-install-recommends lsb-release wget apt-transport-https ca-certificates gnupg
```

```
sudo wget -O - 'https://repo.proxysql.com/ProxySQL/proxysql-3.0.x/repo_pub_key' | sudo gpg --dearmor -o /etc/apt/keyrings/proxysql.gpg
```

```
echo "deb [signed-by=/etc/apt/keyrings/proxysql.gpg]  
https://repo.proxysql.com/ProxySQL/proxysql-3.0.x/$(lsb_release -sc)/ ." | sudo tee  
/etc/apt/sources.list.d/proxysql.list
```

```
sudo apt update && sudo apt install -y proxysql mariadb-client
```

#### Verificar estado:

```
sudo systemctl status proxysql
```

#### Si no está activo, iniciar:

```
sudo systemctl start proxysql
```

### 4.2. Configuración

Entrar al shell de ProxySQL

#### mysql -u admin -padmin -h 127.0.0.1 -P6032

```
-- Añadir el servidor MariaDB como backend  
INSERT INTO mysql_servers (hostgroup_id, hostname, port, weight)  
VALUES (0, '172.22.0.100', 3306, 1);  
  
-- Crear usuario de acceso desde los Apache  
INSERT INTO mysql_users (username, password, default_hostgroup)  
VALUES ('webuser', 'webpass', 0);  
  
-- CONFIGURACIÓN GLOBAL DEL CACHE  
  
-- Habilitar caché y asignar 100MB de memoria  
UPDATE global_variables SET variable_value='true'  
WHERE variable_name='mysql-query_cache_enabled';  
  
UPDATE global_variables SET variable_value='104857600' -- 100MB  
WHERE variable_name='mysql-query_cache_size_MB';  
  
-- Tiempo mínimo de vida en caché (10 segundos)  
UPDATE global_variables SET variable_value='10000' -- 10,000ms = 10s  
WHERE variable_name='mysql-query_cache_ttl_ms';  
  
-- Optimización para alta frecuencia de consultas  
UPDATE global_variables SET variable_value='50000' -- 50 resultados/seg  
WHERE variable_name='mysql-query_cache_results_per_second';  
  
-- Aplicar cambios  
LOAD MYSQL VARIABLES TO RUNTIME;  
SAVE MYSQL VARIABLES TO DISK;
```

```
-- REGLAS DE CACHÉ INTELIGENTES

-- Consultas genéricas (TTL: 30 segundos)
INSERT INTO mysql_query_rules
(rule_id, active, match_pattern, cache_ttl, apply, comment)
VALUES
(10, 1, '^SELECT .* WHERE ', 30000, 1, 'SELECT con WHERE - Cache 30s');

-- Consultas de solo lectura (TTL: 5 minutos)
INSERT INTO mysql_query_rules
(rule_id, active, match_pattern, cache_ttl, apply, comment)
VALUES
(20, 1, '^SELECT .* FROM (report_|log_|view_)', 300000, 1, 'Consultas de reportes - Cache 5m');

-- Excluir consultas NO cacheables
INSERT INTO mysql_query_rules
(rule_id, active, match_pattern, cache_ttl, apply, comment)
VALUES
(30, 1, '^SELECT .*(NOW())|RAND()|UUID())', 0, 1, 'Excluir funciones variables'),
(31, 1, '^SELECT .* FOR UPDATE', 0, 1, 'Excluir SELECT bloqueantes');

-- Tu consulta de prueba (TTL: 60 segundos)
INSERT INTO mysql_query_rules
(rule_id, active, match_pattern, cache_ttl, apply, comment)
VALUES
(40, 1, '^SELECT SLEEP\ (2\), 1;?$', 60000, 1, 'Consulta específica - Cache 60s');

-- Aplicar configuración y reglas
LOAD MYSQL SERVERS TO RUNTIME;
LOAD MYSQL USERS TO RUNTIME;
SAVE MYSQL SERVERS TO DISK;
SAVE MYSQL USERS TO DISK;
```

### 4.3. Verificación y monitorización

Desde un servidor apache2:

```
# Primera ejecución (debe ser lenta)
time mysql -h 172.21.0.30 -P6033 -u webuser -pwebpass -e "SELECT SLEEP(2), 1;"

# Ejecuciones consecutivas (deben ser inmediatas)
for i in {1..3}; do
    time mysql -h 172.21.0.30 -P6033 -u webuser -pwebpass -e "SELECT SLEEP(2), 1;"
done

# Espera 30 segundos y prueba
sleep 30
time mysql -h 172.21.0.30 -P6033 -u webuser -pwebpass -e "SELECT SLEEP(2), 1;"

# Espera 60 segundos y prueba final
sleep 30
time mysql -h 172.21.0.30 -P6033 -u webuser -pwebpass -e "SELECT SLEEP(2), 1;"
```

Desde shell ProxySQL:

```
SELECT
    Variable_name,
    Variable_value
FROM stats.stats_mysql_global
WHERE Variable_name LIKE 'Query_Cache%';
```

```
SELECT
  digest_text,
  count_star AS total_queries,
  sum_time
  AS total_time_ms,
  sum_time/count_star AS avg_time_ms
FROM stats.stats_mysql_query_digest
ORDER BY count_star DESC
LIMIT 10;
```

```
SELECT * FROM stats.stats_mysql_global WHERE Variable_name IN ('Query_Cache_count_GET',
'Query_Cache_count_GET_OK', 'Query_Cache_count_SET');
```

## 5. Servidor de bases de datos

```
sudo apt install mariadb-server
```

```
sudo sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
sudo systemctl restart mariadb.service
```

```
CREATE USER 'webuser'@'172.22.0.30' IDENTIFIED BY 'webpass';
GRANT ALL PRIVILEGES ON *.* TO 'webuser'@'172.22.0.30';
FLUSH PRIVILEGES;
```

```
CREATE DATABASE webdb;
GRANT ALL PRIVILEGES ON webdb.* TO 'webuser'@'172.22.0.30';
```