Instancias de bases de datos

Tiempo estimado: 20min

Para comenzar nuestra aventura por el mundo de las bases de datos, una de las primeras cosas que tenemos que tener clara es cómo trabajar mínimamente con el servidor de bases de datos. Vamos a verlo.

La lección comienza con la introducción del concepto de instancia de bases de datos. A continuación, se presenta algunos parámetros de configuración de la instancia. Después, se muestra cómo arrancar y detener una instancia. Y finalmente, se presenta el registro de mensajes.

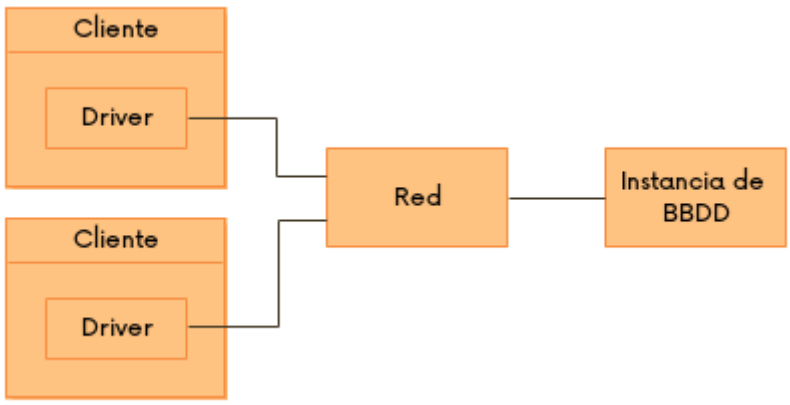
Al finalizar la lección, el estudiante sabrá:

- Qué es una instancia de bases de datos.
- Cómo configurar, arrancar y detener una instancia de bases de datos Redis.
- Qué es el registro de mensajes.
- Cómo rotar y purgar los archivos del registro de mensajes de una instancia de Redis.

Introducción

Un instancia de bases de datos (*database instance*) es una copia en ejecución del motor de bases de datos. Atendiendo al fabricante, una instancia de bases de datos puede servir una única base de datos, como Oracle y ValenciaDB, o múltiples bases de datos, como ArangoDB, MariaDB, MongoDB, PostgreSQL, Redis, RethinkDB o SQL Server. En el caso de Redis, la aplicación que implementa el servidor de bases de datos es redis-server.

Redis se ha implementado, como la mayoría de SGBDs, bajo una arquitectura cliente/servidor (*client/server architecture*). Ésta distingue entre, por un lado, aplicación cliente (*client*), aquella que accede y usa los datos; y por otro lado, aplicación servidora (*server*), aquella que se encarga de administrar y controlar el acceso a los datos. Cada una de ellas se ejecuta en un proceso aparte e incluso, lo más frecuente, en máquinas distintas. A veces incluso, el servidor puede ejecutarse en varios procesos.



En una misma máquina, puede haber tantas instancias en ejecución como sea necesario.

Básicamente, las instancias de bases de datos tienen como objeto administrar las bases de datos y sus archivos de datos así como atender y procesar las solicitudes de las aplicaciones clientes.

Instalación de Redis

Redis está disponible para varias plataformas, siendo Linux la más utilizada.

En distribuciones Debian como Ubuntu, podemos instalar Redis mediante APT. Concretamente, hay que

instalar el paquete redis-server.

Pero generalmente, los repositorios APT no van a la misma velocidad que Redis. Por lo que lo más extendido es descargar el código fuente de redis.io, compilarlo e instalar la versión compilada, todo ello manualmente.

Descarga del código fuente

Lo primero, hay que ir a redis.io y descargar el archivo .tar.gz de la versión a instalar.

También es posible descargarlo directamente usando, por ejemplo, wget o cURL. He aquí un ejemplo:

```
$ wget http://download.redis.io/releases/redis-3.2.8.tar.gz
```

Para versiones en desarrollo:

```
$ wget https://github.com/antirez/redis/archive/4.0-rc2.tar.gz
```

Descompresión del archivo .tar.gz

A continuación, descomprimir o extraer el contenido del archivo .tar.gz como sigue:

```
$ tar xzf 4.0-rc2.tar.gz
```

Compilación de Redis

Entrar en el directorio de código fuente y ejecutar make:

```
$ cd redis-4.0-rc2
$ make
```

Si todas las dependencias están instaladas, la compilación acabará bien y los comandos redis-server y redis-cli se encontrarán en el directorio src.

Tras la compilación, se recomienda ejecutar la batería de pruebas mediante el siguiente comando:

```
$ make test
```

Si todo va bien, recibiremos un mensaje final como:

```
All tests passed without errors!
```

Instalación

Lo siguiente es instalar la versión compilada. Para ello, lo más fácil es ejecutar el siguiente comando:

```
$ sudo make install
```

En sistemas Linux, los comandos se instalarán en el directorio /usr/local/bin/. Si necesitamos que se instalen en otro directorio, podemos instalar como sigue:

```
$ sudo make PREFIX=/el/directorio/donde/instalar install
```

He aquí un ejemplo ilustrativo:

```
$ sudo make PREFIX=/opt/redis install
cd src && make install
make[1]: Entering directory '/home/me/redis/redis-4.0-rc2/src'

Hint: It's a good idea to run 'make test' ;)

    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
make[1]: Leaving directory '/home/me/redis/redis-4.0-rc2/src'
$ ls -1 /opt/redis/bin
redis-benchmark
redis-check-aof
redis-check-rdb
redis-cli
redis-sentinel
redis-server
$
```

Si utilizamos un directorio específico, no hay que olvidar añadir al PATH el directorio bin de Redis, para que los comandos puedan ser accedidos directamente.

Prueba de la instalación

Finalmente, sólo queda probar que la instancia arranca y podemos acceder a ella. En primer lugar, arrancar la instancia servidora mediante el comando redis-server. La manera más fácil es como se muestra a continuación:

```
$ redis-server
```

Si arranca sin problemas, aparecerá un mensaje como el siguiente:

```
The server is now ready to accept connections on port 6379
```

Y a continuación, en otra terminal o consola, abrir una sesión al servidor mediante redis-cli:

```
$ redis-cli
127.0.0.1:6379>
```

Arranque de la instancia

El arranque de la instancia (*instance start*) es la operación mediante la cual se inicia la instancia. redis-server es el programa que implementa el software de una instancia de bases de datos Redis. Este programa presenta varias opciones, las cuales se pueden listar mediante la opción -h o --help.

He aquí un ejemplo ilustrativo usando un archivo de configuración específico:

```
$ redis-server /opt/redis/redis.conf
```

Por convenio y buenas prácticas, se recomienda ejecutar la instancia con una cuenta de usuario específica, aconsejándose la cuenta de usuario redis, la cual no necesita privilegios especiales.

Configuración de la instancia

La instancia contiene un conjunto de parámetros (*parameters*) u opciones (*options*) con los que fijar determinados aspectos del motor de bases de datos como, por ejemplo, el directorio de datos, el registro de eventos, etc. Estos parámetros se pueden fijar mediante un archivo de configuración o la línea de comandos de redis-server. Por convenio y buenas prácticas, se utiliza el archivo redis.conf.

Para obtener una copia del archivo predeterminado, se puede utilizar la que viene con el código fuente.

Si no se indica ningún archivo específico, se usará la configuración predterminada. Para especificar uno hay que indicarlo a continuación del nombre del comando:

```
$ redis-server /ruta/del/archivo/redis.conf
```

Ejemplo:

```
$ redis-server /opt/redis/redis.conf
```

Archivo de configuración

El archivo de configuración (*configuration file*) es un archivo de texto que contiene parámetros en formato:

```
parámetro valor
```

Como, por ejemplo:

```
bind 192.168.1.100 10.0.0.1
```

Extremos de conexión

Las aplicaciones clientes acceden a las instancias de bases de datos principalmente mediante conexiones de red. Por esta razón, los servidores deben escuchar en una o más interfaces de red. Cada uno de los puntos en el que escucha la instancia se conoce formalmente como extremo (*endpoint*). El extremo predeterminado es 127.0.0.1:6379.

Las interfaces de red se indican mediante el parámetro bind, mientras que el puerto con port:

```
bind 127.0.0.1
port 6379
```

Archivo PID

El archivo PID (*PID file*) es un archivo de texto en el que la instancia registra el PID, esto es, el

identificador del proceso redis-server. Este archivo se puede indicar mediante el parámetro pidfile. Por defecto, su valor es el siguiente:

```
pidfile /var/run/redis_6379.pid
```

Algunos DBAs prefieren:

```
pidfile /opt/redis/run/redis-server.pid
```

Comando CONFIG

Se puede utilizar el comando CONFIG para consultar o modificar los parámetros, que está usando la instancia, en tiempo de ejecución.

CONFIG GET

Mediante CONFIG GET se lista el valor actual de uno o más parámetros. Su sintaxis es:

```
CONFIG GET parámetro
CONFIG GET patrón
```

Ejemplos ilustrativos:

```
127.0.0.1:6379> CONFIG GET port
1) "port"
2) "6379"
127.0.0.1:6379> CONFIG GET *-max-*
 1) "hash-max-ziplist-entries"
 2) "512"
 3) "hash-max-ziplist-value"
 4) "64"
 5) "list-max-ziplist-size"
 6) "-2"
 7) "set-max-intset-entries"
 8) "512"
 9) "zset-max-ziplist-entries"
10) "128"
11) "zset-max-ziplist-value"
12) "64"
13) "hll-sparse-max-bytes"
14) "3000"
15) "slowlog-max-len"
16) "128"
17) "min-slaves-max-lag"
18) "10"
127.0.0.1:6379>
```

CONFIG SET

Para modificar el valor actual en tiempo de ejecución, se utiliza CONFIG SET:

```
CONFIG SET parámetro valor
```

Es importante recordar que este comando no actualiza el archivo de configuración, sólo el valor actual que está usando la instancia.

CONFIG REWRITE

El comando CONFIG REWRITE escribe los parámetros actuales en el archivo de configuración usado por redis-server para arrancar:

```
CONFIG REWRITE
```

Parada de la instancia

La parada de la instancia (*instance shutdown*) consiste en detener la instancia. Se puede hacer mediante el comando kill:

```
kill -SEÑAL pid-de-redis-server
```

Redis atiende las siguientes señales:

- SIGTERM. Parada formal. Espera que el comando en curso finalice y termina.
- SIGKILL. Parada informal. Como si la instancia hubiera caído de repente.

A continuación, un ejemplo de parada formal:

```
$ kill -SIGTERM `cat /opt/redis/run/redis-server.pid`
```

Registro de mensajes

El registro de mensajes (*log*), también conocido como registro de alertas (*alert log*) en algunas bases de datos, es un componente que registra mensajes de información relacionados con la actividad de la instancia como, por ejemplo, cuándo se arrancó, cuándo se paró, cuándo ha realizado una determinada operación, así como posibles errores que se han producido durante su funcionamiento. Lo utilizan sobre todo los DBAs como punto de partida cuando se detecta una incidencia en la instancia.

Este registro se encuentra formado por entradas, donde cada entrada de registro (*log entry*) representa un mensaje. Por lo general, estas entradas se mantienen en disco para su consulta posterior, en un archivo de registro (*log file*). Por convenio y buenas prácticas, se usa redis-server.log.

Es importante mantener la información del *log* un tiempo como, por ejemplo, dos semanas o un mes, por si posteriormente hubiera que usarla para abrir un caso de soporte detectado tardíamente.

He aquí un ejemplo de entrada:

```
11726:M 08 Apr 11:46:37.044 # WARNING: The TCP backlog setting of 511 cannot be enforced
because /proc/sys/net/core/somaxconn is set to the lower value of 128.
```

Cada entrada tiene un nivel (*level*) que indica la importancia o la categoría del mensaje. En Redis, se distingue los siguientes niveles:

- debug. Mensaje de información útil para desarrollo y pruebas.
- verbose. Similar a debug, pero no se genera tan a menudo.
- notice. Mensaje de información útil en entornos de producción.
- warning. Mensaje de información crítico.

El archivo se puede indicar mediante el parámetro logfile. Y el nivel mínimo de registro, mediante loglevel. Si no se indica archivo de registro, el caso predeterminado, los mensajes se mostrarán por la consola. Veamos un ejemplo de configuración:

```
logfile /opt/redis/log/redis-server.log
loglevel notice
```

Si deseamos que las entradas se escriban en el *syslog*, activar el parámetro syslog-enabled y configurar los parámetros syslog-ident y syslog-facility:

```
syslog-enabled yes
syslog-ident redis
syslog-facility local0
```

Rotación del registro

La rotación (*rotate*) es la operación mediante la cual se hace una copia del archivo de registro y se comienza con uno nuevo. En Redis, la rotación es un proceso manual de dos pasos:

1. Realizar una copia del archivo de registro, por ejemplo, mediante el comando cp.
2. Vaciar el registro.

Esta rotación se suele hacer todos los días, en las horas de menor carga que generalmente coincide con la noche.

Purgado del registro

La información de las copias de los archivos de registro, tras las rotaciones, no hay que mantenerla indefinidamente. Pero sí hay que hacerlo un tiempo, conocido formalmente como período de retención (*retention period*).

Transcurrido el período de retención, hay que llevar a cabo lo que se conoce como purgado del registro (*log purging*), operación mediante la cual eliminamos los archivos de *log* archivados, cuyo período de retención ha expirado. Por lo general, se recomienda utilizar un período de retención de como mínimo 15 días y como máximo uno o dos meses. Este purgado se suele hacer todos los días, en las horas de menor carga que generalmente coincide con la noche.

