

## Instancias de bases de datos

Tiempo estimado: 10min

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.

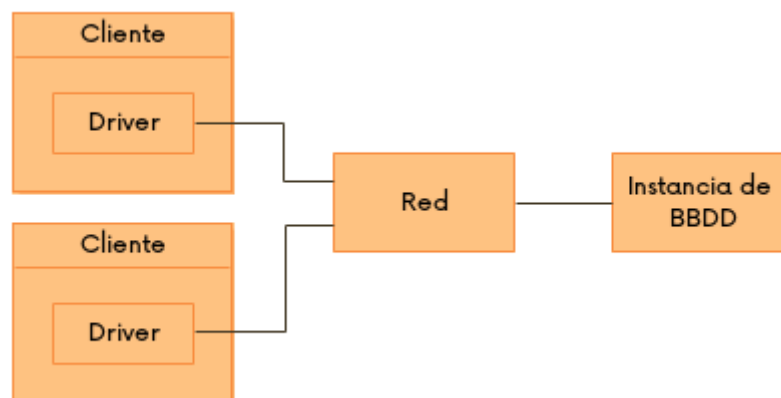
Al finalizar la lección, el estudiante sabrá:

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

### 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** o **Cassandra**, o múltiples bases de datos, como **ArangoDB**, **MariaDB**, **MongoDB**, **PostgreSQL**, **RethinkDB** o **SQL Server**. En el caso de **ArangoDB**, la aplicación que implementa el servidor de bases de datos es **arangod**.

**ArangoDB** se ha implementado, como la mayoría de SGBD, 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.

### 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 **arangod**. Por convenio y buenas prácticas, se utiliza el archivo **arangod.conf**, el cual se encuentra:

- En **Ubuntu**, en **/etc/arangodb3/arangod.conf**.

Se puede especificar el archivo que debe usar **arangod** mediante su opción **--configuration**:

```
--configuration archivo
```

Si se dispone de un archivo de configuración, se puede utilizar la opción **--check-configuration** para validar su sintaxis:

```
arangod --check-configuration archivo
```

## Archivo de configuración

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

```
sección.parámetro = valor
```

Como, por ejemplo:

```
log.level = trace
```

Los parámetros se pueden agrupar en secciones como sigue:

```
[sección]
parámetro = valor
parámetro = valor
...
```

Ejemplo:

```
[log]
level = info
file = /var/log/arangodb3/arangod.log
```

## 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 **tcp://127.0.0.1:8529**.

Cada extremo se configura mediante un parámetro **server.endpoint**, cuyo formato puede ser uno de los siguientes:

```
tcp://host:puerto
ssl://host:puerto
unix://socket
```

He aquí un ejemplo ilustrativo:

```
[server]
endpoint = tcp://localhost:8529
endpoint = tcp://127.0.0.1:8529
```

Para indicar que escuche en todas las interfaces de red del servidor, utilizar:

```
[server]
endpoint = tcp://0.0.0.0:8529
```

## Directorio de datos

El **directorio de datos** (*data directory*) es aquel en el que la instancia mantiene los archivos con los datos de las bases de datos. Se fija mediante el parámetro de configuración **database.directory**. Ejemplo:

```
[database]
directory = /var/lib/arangodb3
```

## 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 **arangod**. Este archivo se puede indicar mediante la opción **--pid-file**. Por convenio y buenas prácticas, se utiliza el archivo **arangod.pid**, el cual se encuentra:

- En **Ubuntu**, en `/var/run/arangodb/arangod.pid`.

## Arranque de la instancia

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

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

```
arangod --configuration /etc/arangodb3/arangod.conf
```

Y ahora usando opciones de línea de comandos:

```
arangod --database.directory data/ --server.endpoint tcp://localhost:9999 --log.file log/arangod.log
```

Si la instancia está registrada como servicio en un sistema **Ubuntu**, se puede usar:

```
service arangodb3 start
```

## Parada de la instancia

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

```
kill -9 pid-de-arangod
```

O bien, mediante el comando `service` si se encuentra registrado como servicio en **Ubuntu**:

```
service arangodb3 stop
```

## 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 relacionadas 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 **arangod.log**, el cual se encuentra:

- En **Ubuntu**, en `/var/log/arangodb3/arangod.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:

```
2016-12-30T18:17:09Z [8979] INFO ArangoDB 3.1.6 [linux] 64bit, using VPack 0.1.30, ICU 54.1, V8 5.0.71.39, OpenSSL 1.0.2g 1 Mar 2016
```

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

- **fatal**. Mensaje de error fatal.
- **error**. Mensaje de error.
- **warning**. Mensaje de aviso.
- **info**. Mensaje de información.
- **debug**. Mensaje de depuración.
- **trace**. Mensaje de traza.

El archivo se puede indicar mediante el parámetro `log.file`. Y el nivel mínimo de registro, mediante `log.level`.

Para que se use la hora local en cada entrada, se puede usar el parámetro `log.use-local-time`.

## 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 **ArangoDB**, la rotación es un proceso manual de dos pasos:

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

```
cp /var/log/arangodb3/arangod.log /var/log/arangodb3/arangod.log.$(date +%Y%m%d)
```
2. Vaciar el registro, enviando la señal **SIGHUP** a la instancia:  

```
kill -SIGHUP $(pidof arangod)
```

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.