



MongoDB – Up & Running



- Introducción
 - ¿Qué es?
 - Ventajas/Desventajas frente a BBDD Relacionales
 - Características Principales
 - ¿Por qué usar NoSQL?
 - Estructura
- Instalación & Configuración
- Comandos
 - Comandos más utilizados
 - Consultas en MongoDB



¿Qué es?

- MongoDB es un sistema de bases de datos NoSQL orientada a documentos, desarrollada bajo el concepto de código abierto.
- Las bases de datos NoSQL guardan los datos en formato JSON con un esquema dinámico , haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.
- Está disponible para PC, OS X, Windows y Solaris.

Ventajas/Desventajas frente a BDDR

Base de datos Tradicional	Base de datos NoSQL (MongoDB)
Los datos se parten en diferentes tablas. Estos datos se procesan en un servidor y se traducen a objetos de la vida real.	Simplemente guardas todos los datos, ya que NoSQL es libre de Schemas, luego no se diseñan las tablas por adelantado.
Requiere Operaciones Join. Escala verticalmente, a través de nuevas tablas y relaciones.	No requiere operaciones Join. Escala horizontalmente.
Define una arquitectura ACID rígida.	En NoSQL, sin embargo se presetan: <ul style="list-style-type: none">· Datos a escala web.· Alta frecuencia de lecturas y escrituras.· Cambios en el esquema de datos frecuentes.· Algunas aplicaciones funcionan con muy poco rendimiento siguiendo la arquitectura ACID. Además, utilizan estructuras de datos sencillas, tipo clave-valor.

Características Principales (I)

- Consultas Ad hoc: permite búsqueda de campos, consulta de rangos y expresiones regulares
- Indexación: cualquier campo en un documento de MongoDB puede ser indexado y es posible realizar índices secundarios, similar a las bases de datos relacionales.
- Replicación: soporta el tipo de replicación maestro-esclavo , muy útil para sistemas distribuídos.
- Balanceo de carga: balancea la carga en múltiples servidores.

Características Principales (II)

- Almacenamiento de archivos: utiliza un sistema llamado GridFS .
- Agregación: la función MapReduce puede ser utilizada para el procesamiento por lotes de datos y operaciones de agregación. Eso permite la operación de agrupación “Group By” .
- Ejecución de JavaScript del lado del servidor: tiene la capacidad de realizar consultas utilizando JavaScript, haciendo que sean ejecutadas para ser enviadas a bases de datos directamente.

¿Por qué usar NoSQL?

- 1. Tamaño: en los últimos dos años, se han generado más datos que en los últimos 10, alrededor de 1 Exabyte (10^{18}) por año.
- 2. Conectividad: los datos están cada vez más conectados y entrelazados.
- 3. Datos semi-estructurados: datos cada vez más desestructurados.
- 4. Arquitectura: actualmente cada servicio tiene su backend.
- 5. Buena conectividad con Kettle.
- 6. Tiene un gran soporte por parte de la comunidad.
- 7. Es muy rápido en la Entrada y Salida de datos.

Estructura

- Bases de datos: conjunto de colecciones. Son dinámicas, solo existen si hay almacenadas en ellas, colecciones o documentos.
- Colecciones: lo que sería una tabla para las bases de datos relacionales. Están compuestas de documentos.
- Documentos: son un conjunto de líneas en formato JSON. Cada línea JSON está compuesta atributos y cada atributo está compuesto de un par clave-valor.

Bajar en Ubuntu 32 Bits (usuario root)

- Ejecutar primero este comando, para obtener la clave pública:

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv  
7F0CEB10
```

- Crear el siguiente fichero:

```
$ echo 'deb http://downloads-  
distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | sudo  
tee /etc/apt/sources.list.d/10gen.list
```

- Hacer el update:

```
$ sudo apt-get update
```

- Instalar paquetes:

```
$ sudo apt-get install mongodb-10gen
```

Instalación & Puesta en marcha

- Instalar paquetes:

```
$ sudo apt-get install mongodb-10gen
```

- Si nos da error, ejecutar:

```
$ sudo apt-get uninstall mongodb
```

- Dar permisos de lectura:

```
$ chmod +x /etc/init.d/mongodb
```

- Para usar mongo, poner en la consola de comandos:

```
$ mongo
```

Comandos más utilizados (I)

- La creación de una base de datos se realiza de manera dinámica, es decir, esta se crea cuando almacenamos un documento.
- 1. Para utilizar una base de datos dinámica (en memoria pero no creada hasta guardar el documento):
>use Pruebas;
(Ojo, mongoDB es case sensitive)
- 2. Para almacenar un documento:
> j={name:"mongo"};
> t={x:3};

Comandos más utilizados (II)

- 3. Para almacenar estos documentos en la base de datos:
 - > db.Pruebas.save(j);
 - > db.Pruebas.save(t);
- 4. Para guardar elementos dinámicamente:
 - > for (var i=1; i<20; i++) db.things.save({x:4, j:i});
- 5. Para mostrar los elementos por pantalla:
 - > db.Pruebas.find()
- 6. Para movernos entre iteraciones (si es necesario):
 - > it

Comandos más utilizados (III)

- 7. Para guardar un “cursor” que apunte a una búsqueda o comando:
> var c=db.Pruebas.find();
- 8. Para recorrer ese cursor:
> while(c.hasNext()) printjson(c.next());
- 9. Usando características de JavaScript, podemos recorrerlo también:
> db.Pruebas.find().forEach(printjson);
- 10. Un cursor puede tener aspecto de array:
> printjson(cursor[4]);

Consultas en MongoDB

– Usando consultas:

- 1. `SELECT * FROM Pruebas WHERE name="mongo";`

En mongodb sería:

```
db.Pruebas.find({name:"mongo"}).forEach(printjson);
```

- 2. Para recuperar el primer elemento que cumpla alguna restricción:

```
printjson(db.Pruebas.findOne({name:"mongo"});
```

- 3. Limitar el número de resultados:

```
db.Pruebas.find().limit(3);
```

- 4. Para borrar algún documento:

```
db.Pruebas.remove({name:"mongo"});
```



Preguntas

