

Una vez aclarado qué es una instancia de bases de datos, cómo configurarla, arrancarla y detenerla, así como las dos principales herramientas con las que ejecutar consultas y administrar, ya tenemos los ingredientes principales para poder cocinar datos. Vamos a comenzar presentando el concepto de base de datos. Y poco a poco, iremos describiendo cada uno de los elementos o componentes que todo programador y DBA deben conocer.

Comenzamos la lección con una introducción al concepto de base de datos. A continuación, se describe varias operaciones de bases de datos: la creación, el listado, la supresión y el cambio de base de datos activa. Finalmente, presentamos el concepto de autorización para controlar el acceso a los recursos, en nuestro caso, a bases de datos.

Al finalizar la lección, el estudiante sabrá:

- Qué es una base de datos.
- Qué es una colección.
- Cuál es la diferencia entre una base de datos de sistema y una de usuario.
- Cómo crear y suprimir bases de datos mediante el *shell* y la interfaz web de ArangoDB.
- Cómo conocer las bases de datos de una instancia mediante el *shell* y la interfaz web de ArangoDB.
- Qué es la base de datos activa y cómo cambiarla.
- Cómo controlar el acceso de los usuarios a las bases de datos.

Introducción

Una **base de datos** (*database*) es una colección estructurada de datos y/u objetos, organizada atendiendo a un determinado **modelo de base de datos** (*database model*), el cual especifica la manera en que la base de datos se estructura y manipula. Recordemos que ArangoDB es multimodelo y proporciona soporte para:

- Almacenes clave-valor.
- Almacenes de documentos.
- Almacenes de grafos.

A lo largo del curso, iremos viendo los dos primeros. El tercer tipo de almacén lo dejamos para un curso aparte.

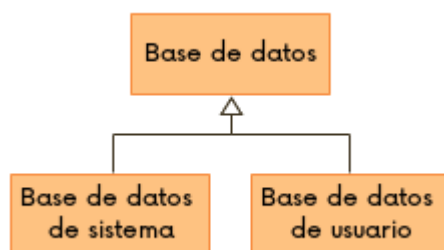
En ArangoDB, el contenedor principal de datos es la colección. Una **colección** (*collection*) no es más que un objeto o repositorio donde se almacenan datos. Una base de datos puede tener tantas colecciones como sea necesario.



Independientemente del tipo de modelo de bases de datos, una base de datos está formada por objetos, tanto de datos como de otro tipo, cada uno de los cuales se conoce formalmente como **objeto de base de datos** (*database object*). Se pueden clasificar atendiendo a su contenido. Por ejemplo, en ArangoDB, encontramos que las colecciones y los índices son los contenedores donde se almacenan datos, mientras que la lógica se puede almacenar en funciones AQL o en servicios ArangoDB Foxx, ambos escritos en JavaScript.

Base de datos de sistema

ArangoDB, al igual que muchos motores de bases de datos, distingue entre bases de datos de sistema y de usuario.



Una **base de datos de sistema** (*system database*) es aquella en la que la propia instancia almacena datos internos. En el caso de ArangoDB, es la de nombre `_system` y viene de fábrica con el producto. En cambio, una **base de datos de usuario** (*user database*) es aquella que creamos nosotros para almacenar nuestros propios datos.

Generalmente, los usuarios sólo acceden a bases de datos de usuario. Algunos usuarios especiales como, por ejemplo, los DBAs sí suelen trabajar con bases de datos de sistema.

Creación de bases de datos

La **creación de una base de datos** (*database creation*) es la operación mediante la cual se crea una nueva base de datos en la instancia de ArangoDB. Se puede hacer con la utilidad de línea de comandos `arangosh` o la interfaz web de ArangoDB. También es posible hacerlo con algunos *drivers* como, por ejemplo, el de `JavaScript`, pero los *drivers* quedan fuera del ámbito del curso.

Para crear una base de datos es necesario utilizar una sesión que tenga la base de datos `_system` como base de datos activa.

Permisos

Para crear una base de datos, es necesario que el usuario tenga permiso de lectura/escritura en la base de datos `_system`.

Creación de una base de datos mediante arangosh

Si se usa el *shell* de ArangoDB, hay que utilizar el método `_createDatabase()` del objeto `db`, recordemos que este objeto representa la base de datos activa de la conexión o sesión al *shell*:

```
_createDatabase(name)
_createDatabase(name, opts)
_createDatabase(name, opts, users)
```

Parámetro	Tipo de datos	Descripción
-----------	---------------	-------------

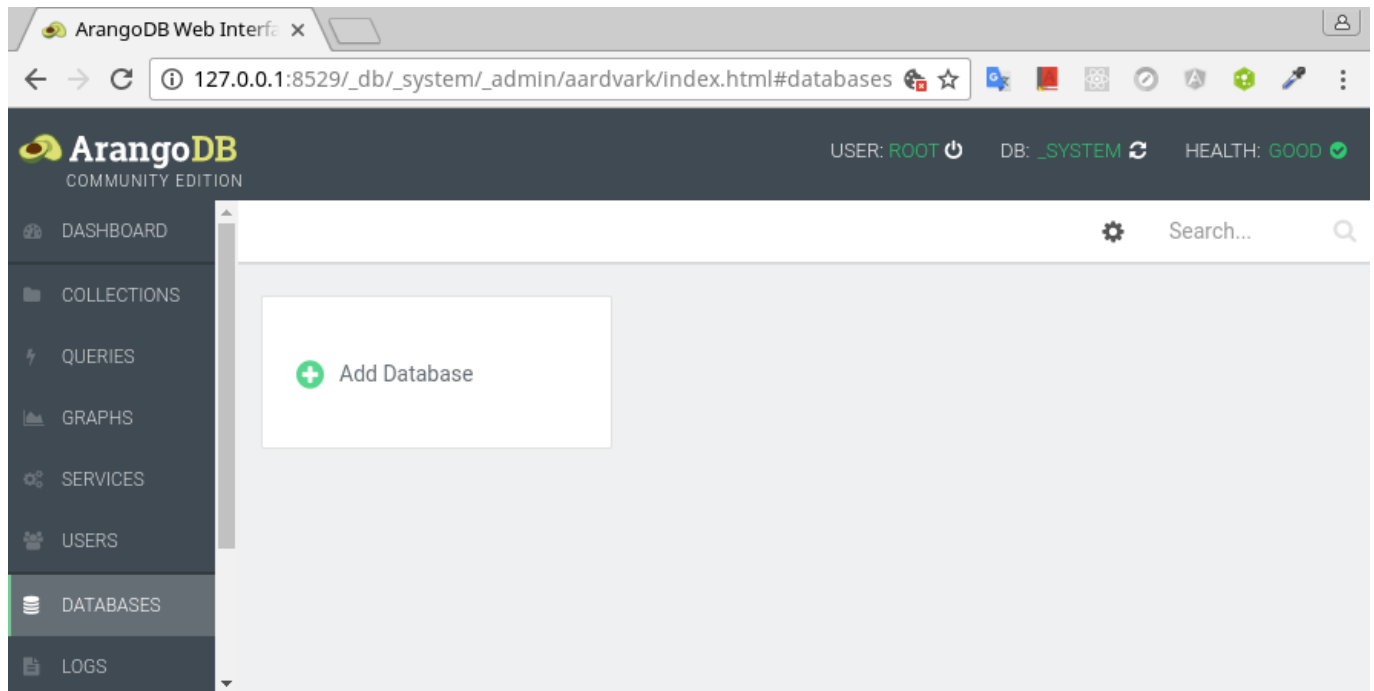
<code>name</code>	string	Nombre de la base de datos a crear.
<code>opts</code>	object	Opciones de la base de datos. Actualmente, reservado para futuro uso.
<code>users</code>	object[]	Usuarios de la base de datos: <ul style="list-style-type: none"><code>username</code> (string). Nombre de la cuenta de usuario.<code>passwd</code> (string). Contraseña del usuario. Valor predeterminado: la cadena vacía.<code>active</code> (boolean). ¿Cuenta de usuario activa? <code>true</code>, sí; <code>false</code>, no. Valor predeterminado: <code>true</code>.<code>extra</code> (object). Información adicional del usuario.

Ejemplo ilustrativo:

```
127.0.0.1:8529@_system> db._createDatabase("prueba")
true
127.0.0.1:8529@_system>
```

Creación de una base de datos mediante interfaz web

También es posible crear bases de datos mediante la interfaz web de **ArangoDB**. Con una sesión a la base de datos `_system`, ir a **DATABASES** y hacer clic en **+ Add Database**:



Listado de bases de datos

Para conocer las bases de datos de una instancia **ArangoDB**, se puede utilizar el *shell* o la interfaz web. En ambos casos, hay que utilizar una sesión a la base de datos `_system`.

Permisos

La cuenta de usuario debe tener permiso de lectura/escritura en la base de datos `_system`.

Listado de bases de datos mediante arangosh

Si hacemos la consulta mediante el *shell*, utilizar el método `_databases()` del objeto predefinido `db`:

```
_databases() : string[]
```

Listado de bases de datos mediante interfaz web

En el caso de la interfaz web, ir a **DATABASES**.

Supresión de bases de datos

La **supresión de una base de datos** (*database drop*) es la operación mediante la cual se suprime una base de datos y todos sus datos. Se puede ejecutar mediante el *shell* y la interfaz web.

Permisos

Para suprimir una base de datos, es necesario que la cuenta de usuario tenga permiso de lectura/escritura en la base de datos `_system`.

Supresión de una base de datos mediante arangosh

Para suprimir una base de datos mediante **arangosh**, hay que utilizar el método `_dropDatabase()` del objeto `db`:

```
_dropDatabase(name) : boolean
```

Parámetro	Tipo de datos	Descripción
-----------	---------------	-------------

<code>name</code>	string	Nombre de la base de datos a suprimir.
-------------------	--------	--

Ejemplo:

```
127.0.0.1:8529@_system> db._dropDatabase("prueba")
true
127.0.0.1:8529@_system>
```

Supresión de una base de datos mediante interfaz web

Si se usa la interfaz web, ir a **DATABASES** y hacer clic en el botón de opciones de la base de datos a suprimir:



A continuación, hacer clic en **Delete**:



Cambio de base de datos activa

Toda sesión a la instancia de **ArangoDB** debe estar asociada a una de sus bases de datos, la cual se conoce como **base de datos activa** (*active database*) o **base de datos actual** (*current database*). En cualquier momento podemos cambiarla, tanto en el *shell* como en la interfaz web.

Permisos

Para poder cambiar a una determinada base de datos, es necesario tener permiso de lectura/escritura en la base de datos a la que deseamos cambiar.

Cambio de base de datos activa en arangosh

Para cambiar la base de datos activa de una sesión de **arangosh**, hay que utilizar el método `_useDatabase()` del objeto `db`:

```
_useDatabase(name)
```

Parámetro	Tipo de datos	Descripción
-----------	---------------	-------------

<code>name</code>	string	Nombre de la base de datos a la que cambiar.
-------------------	--------	--

Ejemplo ilustrativo:

```
127.0.0.1:8529@_system> db._useDatabase("prueba")
true
127.0.0.1:8529@prueba>
```

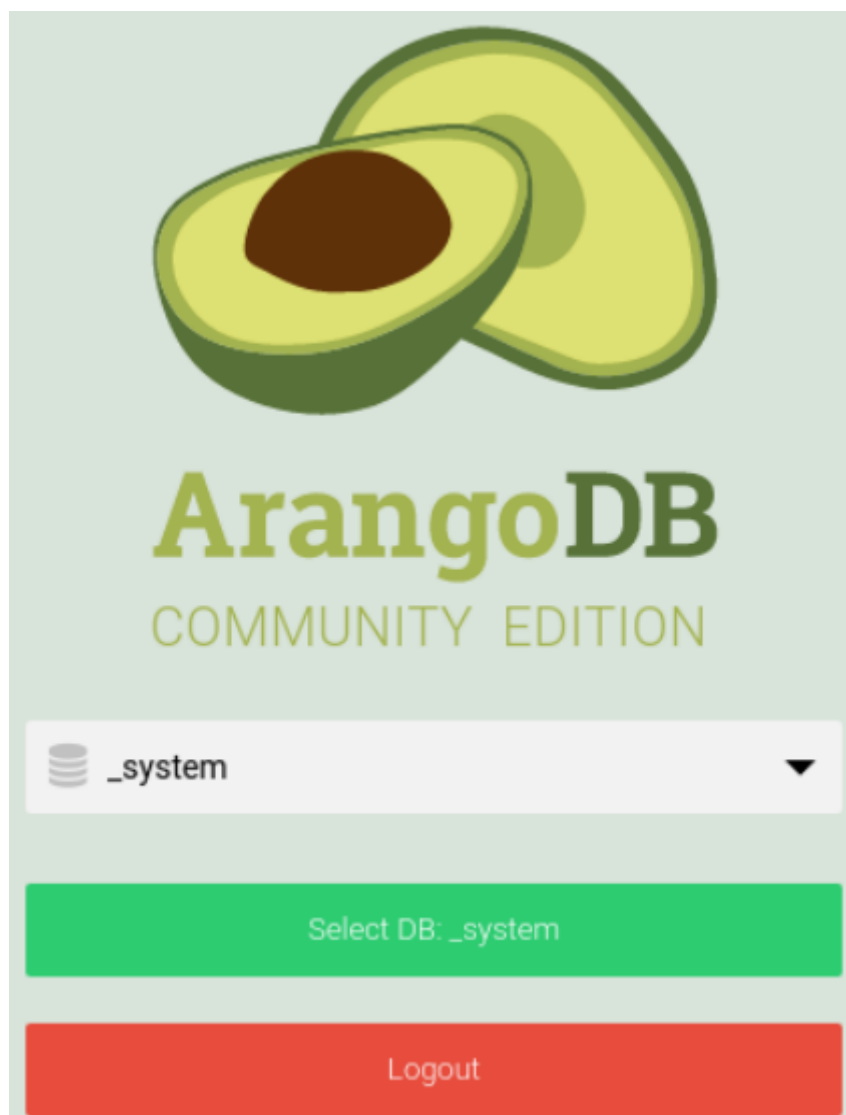
Para saber el nombre de la base de datos activa, podemos apoyarnos en el *prompt* del *shell* o bien en el método `_name()` del objeto `db`:

```
_name() : string
```

Cambio de base de datos activa en interfaz web

En la interfaz web, hay que hacer clic en el nombre de la base de datos ubicado en la barra de tareas de la consola:

A continuación, seremos remitidos a un formulario, similar al que aparece en la apertura de la sesión, donde elegir la base de datos a la que cambiar o a usar como activa:



Control de acceso a bases de datos

La **autorización** (*authorization*) es la operación mediante la cual se valida si un usuario puede realizar una determinada acción, es decir, regula el acceso a los recursos. En **ArangoDB**, los recursos son las bases de datos. Se puede regular qué usuarios pueden acceder a una base de datos y, por lo tanto, a su contenido mediante permisos.

Un **permiso** (*permission*) o **privilegio** (*privilege*) es un derecho para hacer algo, en nuestro caso, para hacerlo sobre una base de datos. En **ArangoDB**, sólo existe un permiso, el de lectura/escritura que permite hacer *cualquier cosa* en la base de datos.

La **concesión** (*grant*) es la operación mediante la cual se concede un permiso. En nuestro caso, se concede el permiso de acceso a una base de datos y su contenido. Mientras que la **revocación** (*revoke*) es la que lo quita.

La información de los usuarios y los permisos se almacena en la base de datos **_system**. Para poder conceder o revocar el permiso de lectura/escritura sobre una determinada base de datos a un usuario, debemos hacerlo desde la base de datos **_system** con una cuenta con el permiso de lectura/escritura sobre **_system**.

Concesión y revocación del permiso de lectura/escritura mediante arangosh

En el *shell* de ArangoDB, se usa las funciones `grantDatabase()` y `revokeDatabase()` del módulo `@arangodb/users`:

```
function grantDatabase(user, database)
function revokeDatabase(user, database)
```

Parámetro	Tipo de datos	Descripción
-----------	---------------	-------------

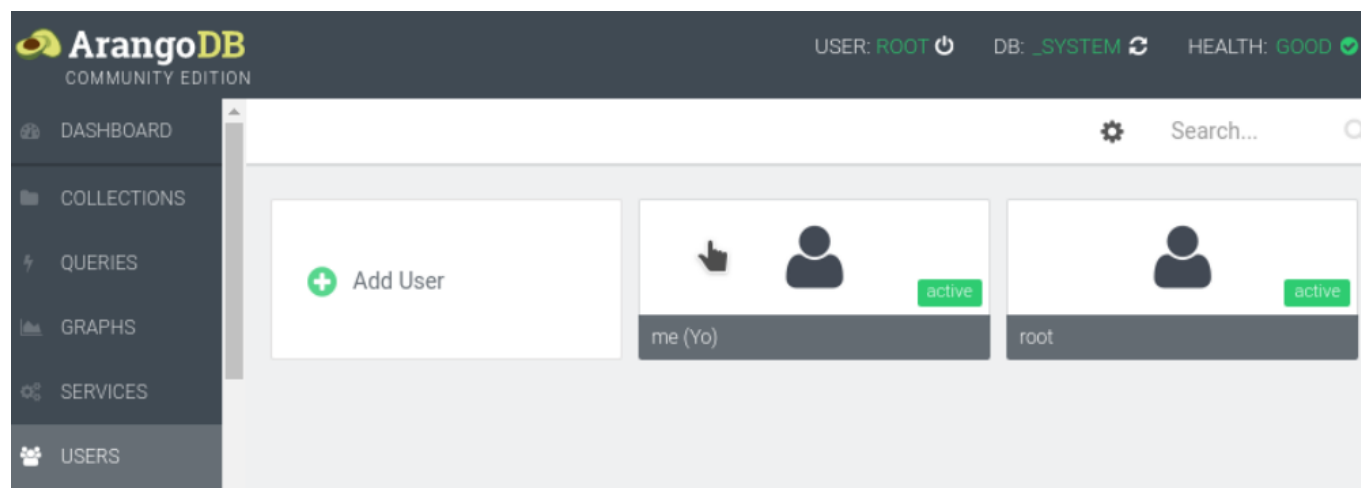
<code>user</code>	string	Nombre de la cuenta de usuario a la que conceder o revocar el permiso.
<code>database</code>	string	Nombre de la base de datos sobre la que conceder o revocar el permiso.

He aquí un ejemplo que muestra cómo conceder el permiso al usuario `me` sobre la base de datos `prueba`:

```
127.0.0.1:8529@_system> users = require("@arangodb/users")
{
  "save" : [Function "(user, passwd, active, extra, changePasswor" ...],
  "replace" : [Function "(user, passwd, active, extra, changePasswor" ...],
  "update" : [Function "(user, passwd, active, extra, changePasswor" ...],
  "remove" : [Function "(use" ...],
  "document" : [Function "(use" ...],
  "isValid" : [Function "(user, passwor" ...],
  "all" : [Function "" ...],
  "reload" : [Function "" ...],
  "grantDatabase" : [Function "(username, database, typ" ...],
  "revokeDatabase" : [Function "(username, databas" ...],
  "updateConfigData" : [Function "(username, key, valu" ...],
  "configData" : [Function "(username, ke" ...],
  "permission" : [Function "(username, ke" ...]
}
127.0.0.1:8529@_system> users.grantDatabase("me", "prueba")
127.0.0.1:8529@_system>
```

Concesión y revocación del permiso de lectura/escritura mediante interfaz web

Si deseamos conceder o revocar el permiso de lectura/escritura sobre una base de datos mediante la interfaz web, hay que ir a **USERS** y hacer clic en el usuario en cuestión:



A continuación, hacer clic en **Permissions** y seleccionar o deseleccionar el cuadro de comprobación de la base de datos:

General **Permissions**

Database	Read & Write (rw)
_system	<input type="checkbox"/>
prueba	<input checked="" type="checkbox"/>