

Una de las funciones de un sistema de gestión de bases de datos es controlar el acceso a los datos. Para ello, proporciona principalmente dos mecanismos de regulación de acceso: la autenticación y la autorización. La autenticación se describe en esta lección. La autorización en una posterior.

Comenzamos introduciendo los conceptos de autenticación y usuario. A continuación, se describe cómo configurar una instancia para que requiera que los usuarios se identifiquen y proporcionen credenciales válidas para así poder acceder. Finalmente, se examina cómo crear, consultar, actualizar y suprimir usuarios.

Al finalizar la lección, el estudiante sabrá:

- Qué es la autenticación.
- Qué es un usuario.
- Cómo configurar la instancia para que autentique a los usuarios.
- Cómo crear, actualizar y suprimir usuarios mediante el *shell* y la interfaz web de ArangoDB.
- Cómo consultar la información de uno o más usuarios mediante el *shell* y la interfaz web de ArangoDB.

Introducción

Un aspecto muy importante es el control de acceso a los usuarios. Por un lado, tenemos la autenticación y, por otro lado, la autorización.

La **autenticación** (*authentication*) es el proceso mediante el cual un usuario se identifica ante la instancia y ésta le permite su acceso o se lo deniega. Es necesaria si hay que controlar el acceso de los usuarios a la instancia. Por su parte, la **autorización** (*authorization*) regula el acceso a los datos, una vez permitido el acceso a la instancia. En esta lección, nos centramos en la autenticación. Dejamos para más adelante la autorización.

Para proporcionar autenticación, ArangoDB proporciona el concepto de **usuario** (*user*) o **cuenta de usuario** (*user account*), un objeto que representa a una persona o aplicación que puede acceder a la instancia. Cada usuario tiene un **nombre de usuario** (*username*) único que lo identifica del resto. Y una **contraseña** (*password*), un texto secreto que sirve para que el sistema confirme que es quien dice ser. De tal manera que si la autenticación está activada, quien esté detrás de una conexión o sesión deberá proporcionar las credenciales de un usuario dado de alta en el sistema.

En ArangoDB, los usuarios además tienen otros campos:

- **active** (boolean). Indica si la cuenta está activa o no.
Un **usuario activo** (*active user*) es aquel que se puede usar para acceder a la instancia. Si no está activo, la cuenta existe pero no se puede usar.
- **extra** (object). Consiste en un objeto que proporciona información adicional del usuario como, por ejemplo, su nombre completo, su centro de trabajo, su teléfono, etc.
Es libre y cada organización puede usarlo como considere oportuno.

Configuración de autenticación

Para que ArangoDB aplique autenticación, es necesario fijar a **true** el parámetro de configuración **server.authentication**. Si se encuentra a **false**, no la aplicará. Por otra parte, tenemos el parámetro **server.authentication-system-only**, mediante el cual se le indica a la instancia si la autenticación sólo debe aplicarla a las consultas dirigidas a la API interna de ArangoDB: **true**, sí; **false**, no.

Por lo general, la autenticación se configura como sigue:

```
[server]
authentication = true
authentication-system-only = false
```

Creación de usuarios

Una vez configurada la instancia para que trabaje con autenticación, lo siguiente, que parece bastante razonable, es conocer cómo trabajar con usuarios. Cómo darles de alta, modificarlos o suprimirlos. Vamos a comenzar con la operación de **creación de usuario** (*user creation*), aquella que da de alta un nuevo usuario en la instancia. Hay que hacerlo usando una sesión a la base de datos **_system**. Se puede realizar mediante el *shell* y la interfaz web de **ArangoDB**.

Permisos

Para crear un usuario, es necesario usar una cuenta con permiso de lectura/escritura en la base de datos **_system**.

Creación de usuarios mediante arangosh

Cuando se trabaja con usuarios en el *shell* de **ArangoDB**, se hace mediante el módulo **@arangodb/users**, el cual proporciona funciones para crear, actualizar, suprimir y acceder a documentos de usuario. Por convenio y buenas prácticas, este módulo se importa como **users**. En este punto, la función que nos interesa es **save()**, con la que se añade un nuevo usuario a la instancia:

```
function save(user, passwd)
function save(user, passwd, active)
function save(user, passwd, active, extra)
```

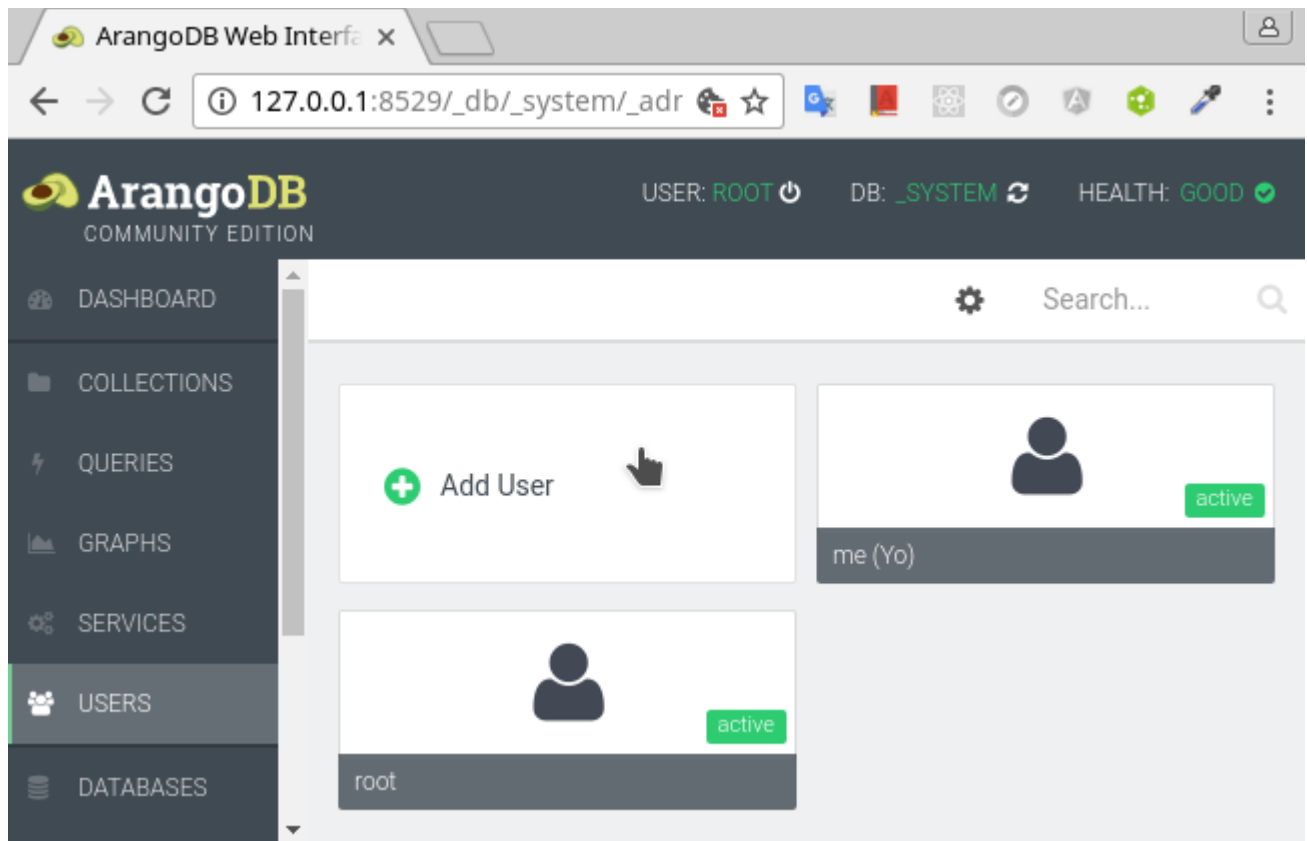
Parámetro	Tipo de datos	Descripción
user	string	Nombre del usuario a crear.
passwd	string	Contraseña del nuevo usuario.
active	boolean	¿Cuenta activa? true , sí; false , no. Valor predeterminado: true .
extra	object	Propiedades adicionales del usuario.

He aquí un ejemplo ilustrativo:

```
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.save("yo", "contraseña", true, {name: "Raúl G."})
{
  "user" : "yo",
  "active" : true,
  "extra" : {
    "name" : "Raúl G."
  },
  "changePassword" : false,
  "code" : 201
}
127.0.0.1:8529@_system>
```

Creación de usuarios mediante interfaz web

Para crear un nuevo usuario mediante la interfaz web de **ArangoDB**, hay que ir a **USERS** y hacer clic en **+ Add User**:



A continuación, hay que rellenar la información del usuario y hacer clic en **Create**:

ArangoDB Web Interface

127.0.0.1:8529/_db/_system/_adr

ArangoDB USER: ROOT DB: _SYSTEM HEALTH: GOOD

Create New User

Username*:

Name:

Password:

Active: ☒

Consulta de usuarios

A veces, es necesario consultar la información de uno o más usuarios dados de alta en la instancia. Esto se puede hacer mediante el *shell* o la interfaz web.

Permisos

Para consultar la información de los usuarios, es necesario que el usuario que realiza la consulta tenga el permiso de lectura/escritura en la base de datos `_system`.

Consulta de usuarios mediante arangosh

Por un lado, podemos acceder al documento de un usuario particular mediante la función `document()` del módulo `@arangodb/users`:

```
function document(user) : object
```

Parámetro	Tipo de datos	Descripción
<code>user</code>	string	Nombre del usuario a consultar.

Ejemplo:

```
127.0.0.1:8529@_system> users.document("yo")
{
  "user" : "yo",
  "active" : true,
  "extra" : {
    "name" : "Raúl G."
  },
  "changePassword" : false,
  "code" : 200
}
```

```
127.0.0.1:8529@_system>
```

También es posible obtener los documentos de todos los usuarios dados de alta, mediante la función

`all()` del módulo `@arangodb/users`:

```
function all() : object[]
```

Consulta de usuarios mediante interfaz web

También podemos consultar los usuarios mediante la interfaz web, para ello, ir a **USERS**. Para ver la de un determinado usuario, basta con hacer clic en el usuario en cuestión.

Actualización de usuarios

La **actualización de usuario** (*user alter*) es la operación mediante la cual se cambia alguno de los campos del usuario como, por ejemplo, su contraseña o los campos **active** y **extra**. Veamos cómo hacerlo mediante el *shell* y la interfaz web de **ArangoDB**.

Permisos

Para actualizar un documento de usuario, el usuario que realiza la modificación debe tener el permiso de lectura/escritura sobre la base de datos `_system`.

Actualización de usuarios mediante arangosh

Si la actualización se realiza mediante el *shell*, se puede utilizar la función `update()` del módulo `@arangodb/users`:

```
function update(user, passwd)
function update(user, passwd, active)
function update(user, passwd, active, extra)
```

Parámetro	Tipo de datos	Descripción
<code>user</code>	string	Nombre del usuario a actualizar.
<code>passwd</code>	string	Nueva contraseña.
<code>active</code>	boolean	¿Usuario activo? true , sí; false , no.
<code>extra</code>	object	Propiedades adicionales del documento.

He aquí un ejemplo para cambiar la contraseña de un usuario:

```
127.0.0.1:8529@_system> users.update("yo", "nuevacontraseña")
{
  "user" : "yo",
  "active" : true,
  "extra" : {
    "name" : "Raúl G."
  },
  "changePassword" : false,
  "code" : 200
}
```

```
127.0.0.1:8529@_system>
```

Generalmente, una vez añadido un nuevo usuario a la instancia o cambiada la contraseña de un usuario existente, lo habitual es comprobar que se tiene acceso a la instancia, o sea, que cuando el usuario tire a conectar no le fallará, a menos que intente conectar a una base de datos en la que no puede. Para este fin, se puede utilizar la función `isValid()` del módulo `@arangodb/users`:

```
function isValid(user, password) : boolean
```

Parámetro	Tipo de datos	Descripción
<code>user</code>	string	Nombre del usuario a comprobar.
<code>password</code>	string	Contraseña del usuario a comprobar.

La función devuelve **true** si las credenciales proporcionadas son válidas; **false**, en otro caso.

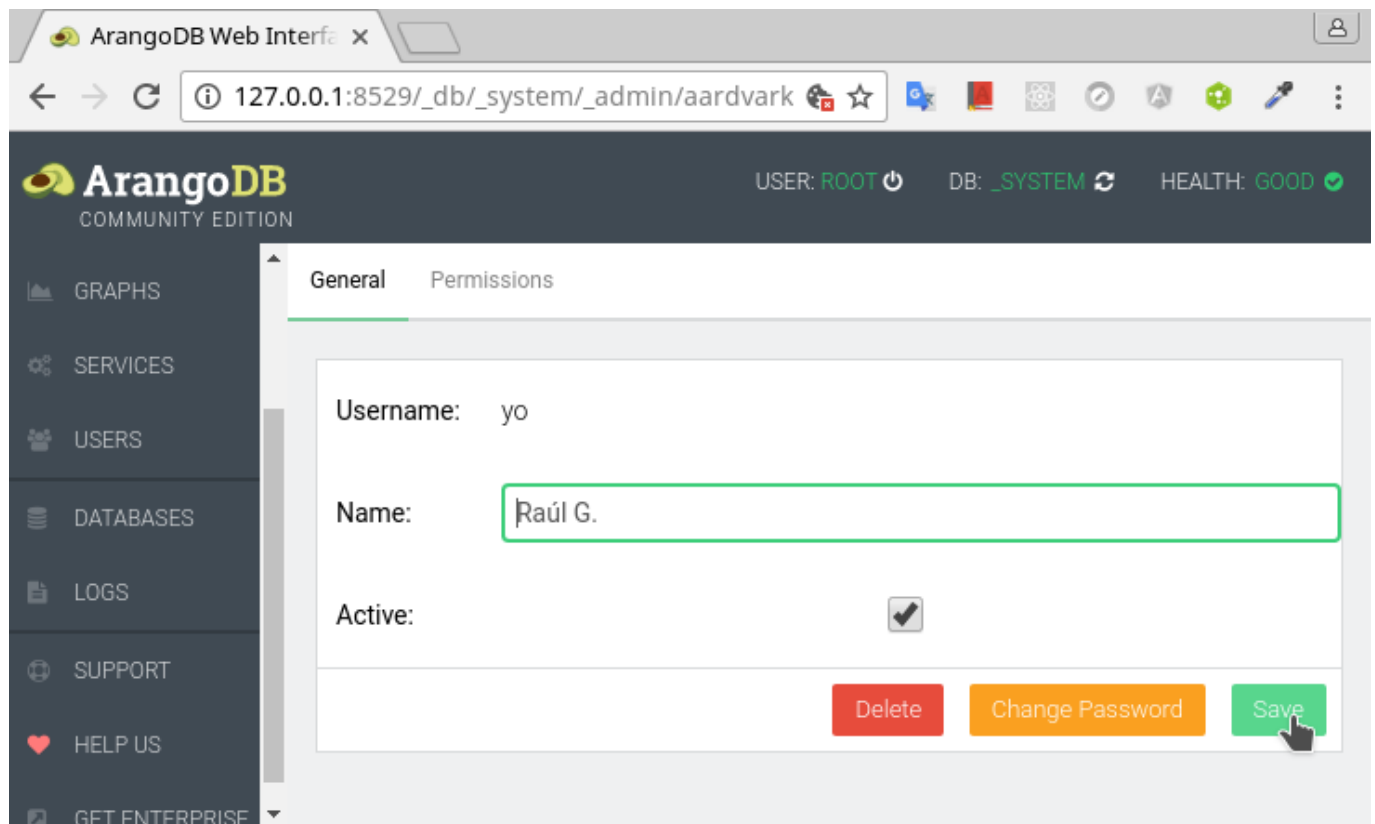
Veamos unos ejemplos ilustrativos:

```
127.0.0.1:8529@_system> users.isValid("yo", "mal")
false
```

```
127.0.0.1:8529@_system> users.isValid("yo", "contraseña")
true
127.0.0.1:8529@_system>
```

Actualización de usuarios mediante interfaz web

Para modificar el documento de un usuario mediante la interfaz web de **ArangoDB**, hay que ir a **USERS**, hacer clic en el usuario en cuestión, modificar el campo correspondiente y hacer clic en **Save**:



Para cambiar la contraseña de un usuario, hacer clic en **Change Password**. La interfaz despliega un formulario modal que solicita la nueva contraseña del usuario.

Supresión de usuarios

La **supresión de un usuario** (*user drop*) es la operación mediante la cual se elimina un usuario de la instancia. Se puede realizar mediante el *shell* o la interfaz web de **ArangoDB**.

Permisos

Para suprimir un usuario, hay que tener el permiso de lectura/escritura sobre la base de datos **_system**.

Supresión de usuarios mediante arangosh

Para suprimir un usuario mediante el *shell*, hay que utilizar la función **remove()** del módulo **@arangodb/users**:

```
function remove(user)
```

Parámetro	Tipo de datos	Descripción
user	string	Nombre del usuario a suprimir.

Ejemplo:

```
127.0.0.1:8529@_system> users.remove("yo")
127.0.0.1:8529@_system>
```

Supresión de usuarios mediante interfaz web

La supresión de un usuario mediante la interfaz web es también muy sencilla. Hay que ir a **USERS**, hacer clic el usuario y, a continuación, en **Delete**.