

# Instalación de Servidor MongoDB en Debian 12

---



Autor 

- Andrés Morales González
- 🏠 I.E.S. Gonzalo Nazareno 📍 (Dos Hermanas, Sevilla).

# Índice

---

- [Instalación de Servidor MongoDB en Debian 12](#)
  - [Autor](#) 
- [Índice](#)
- [Instalacion MONGODB eb Debian12](#)
  - [Actualizar el sistema](#)
  - [Importar la clave GPC de MongoDB](#)
  - [Agregar el repositorio de MongoDB](#)
  - [Instalar MongoDB](#)
  - [Configuración acceso remoto](#)
  - [Reinicio del servicio de MongoDB](#)
- [Creacion de un usuario](#)
- [Instalacion de cliente para mongodb](#)

# Instalacion MONGODB eb Debian12

---

## Actualizar el sistema

## Importar la clave GPG de MongoDB

Antes de agregar lo que sera el repositorio de MongoD, lo que vamos a necesitar es la clave pública:

```
wget -q0 - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
```

Verificación por pantalla:

```
andy@servidores:~$ wget -q0 - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

## Agregar el repositorio de MongoDB

Ahora agregaremos lo que sera el repo de MongoDB a nuestra lista de fuentes APT:

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/debian $(lsb_release -cs)/mongodb-org/6.0 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
```

Verificación:

```
andy@servidores:~$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/debian $(lsb_release -cs)/mongodb-org/6.0 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/debian bookworm/mongodb-org/6.0 main
```

## Instalar MongoDB

Para instalar mongo lo que tendremos que hacer l siguiiente:

- Paso 1

Al hacer lo que es la actualización de lo que seria nuestros repositorios, este nos dara un fallo, el cual será este:

```
andy@servidores:~$ sudo apt install -y mongodb-org
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
No se pudieron instalar algunos paquetes. Esto puede significar que
usted pidió una situación imposible o, si está usando la distribución
inestable, que algunos paquetes necesarios aún no se han creado o se
han sacado de «Incoming».
La siguiente información puede ayudar a resolver la situación:

Los siguientes paquetes tienen dependencias incumplidas:
mongodb-org-mongos : Depende: libssl1.1 (>= 1.1.0) pero no es instalable
mongodb-org-server : Depende: libssl1.1 (>= 1.1.0) pero no es instalable
mongodb-org-shell : Depende: libssl1.1 (>= 1.1.0) pero no es instalable
E: No se pudieron corregir los problemas, usted ha retenido paquetes rotos.
```

Esto sucede porque en nuestro entorno Debian 12, tenemos instalado *libssl1.3* y nos pide el siguiente *libssl1.1*.

Por lo que tendremos que proceder a meter su paquete:

```
andy@servidores:~$ sudo apt install libssl1.1
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  libssl1.1
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 1.566 kB de archivos.
Se utilizarán 4.227 kB de espacio de disco adicional después de esta
operación.
Des:1 http://deb.debian.org/debian bullseye/main amd64 libssl1.1 amd64
1.1.1w-0+deb11u1 [1.566 kB]
Descargados 1.566 kB en 16s (99,6 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete libssl1.1:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 39315 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar .../libssl1.1_1.1.1w-0+deb11u1_amd64.deb ...
Desempaquetando libssl1.1:amd64 (1.1.1w-0+deb11u1) ...
Configurando libssl1.1:amd64 (1.1.1w-0+deb11u1) ...
```

Con esto seguiremos lo que sera para la instalación.

- Paso 2

Ya que le hemos agregado el repositorio en *sudo nano /etc/apt/sources.list* procederemos a hacer la actualización del sistema y a la instalación:

```
sudo apt update
sudo apt install -y mongodb-org
```

- Paso 3 Después de la instalación, lo que haremos será iniciar el servicio de MongoDB, y habilitarlo para que se inicie automáticamente al momento que arranquemos el sistema:

```
sudo systemctl start mongod
sudo systemctl enable mongod
```

Comprobación:

 Enable mongo

## Configuración acceso remoto

Para habilitar lo que será el acceso remoto tendremos unos pasos que seguir:

1. Abrir el archivo de configuración de MongoDB:

```
sudo nano /etc/mongod.conf
```

Buscamos la línea que contiene `bindIp` y la cambiamos para que MongoDB escuche en todas las interfaces, es decir le ponemos esto:

```
net:
  bindIp: 0.0.0.0
  port: 27017
```

Dado que antes tenía esta forma, y no vamos a trabajar de forma local, a menos que sea estrictamente necesario:

```
"" net: bindIp: 127.0.0.1 port: 27017 ""
```

 Acceso remoto

## Reinicio del servicio de MongoDB

```
sudo systemctl restart mongod
```

## Creación de un usuario

Lo primero que haremos será meternos como admin en lo que será mongo, para ello no iremos a donde tengamos el servidor de mongo, e introduciremos lo siguiente:

**mongo**

esto nos dará lo siguiente por pantalla:

```
ndy@servidores:~$ mongo
MongoDB shell version v4.4.29
connecting to: mongodb://127.0.0.1:27017/?
compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("bebc5bca-9d30-4a8b-b7cd-
d3ec56f23673") }
MongoDB server version: 4.4.29
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
  https://community.mongodb.com
---
The server generated these startup warnings when booting:
    2024-10-11T13:22:08.983+02:00: Using the XFS filesystem is strongly
recommended with the WiredTiger storage engine. See
http://dochub.mongodb.org/core/prodnotes-filesystem
    2024-10-11T13:22:09.337+02:00: Access control is not enabled for
the database. Read and write access to data and configuration is
unrestricted
    2024-10-11T13:22:09.337+02:00:
/sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting
it to 'never'
---
>
```

Ahora vamos a crear un usuario en la base de datos de admin, el cual se llamara andy, y contraseña andy, por pantalla se verá así:

```
> use admin
switched to db admin
> db.createUser({
...   user: "andy",
...   pwd: "andy",
...   roles: [{ role: "readWrite", db: "admin" }]
... })
Successfully added user: {
  "user" : "andy",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "admin"
```

```
    }  
  ]  
}  
>
```

Con esto hemos creado con éxito el usuario junto a su contraseña y los permisos necesarios en la base de datos *admin*.

Como podemos observar ya podemos entrar:

```
andy@servidores:~$ mongo --authenticationDatabase "admin" -u "andy" -p  
"andy"  
MongoDB shell version v4.4.29  
connecting to: mongodb://127.0.0.1:27017/?  
authSource=admin&compressors=disabled&gssapiServiceName=mongodb  
Implicit session: session { "id" : UUID("1c9a5d0a-8944-4404-a147-  
0f5e2b9f5860") }  
MongoDB server version: 4.4.29  
---
```

He creado una base de datos llamada *concesionario*, la cual se ve reflejada aquí:

```
> show dbs  
admin          0.000GB  
concesionario  0.000GB  
config         0.000GB  
local          0.000GB  
>
```

Para poder usarla tenemos que volver a hacer uso de **use**:

```
> use concesionario  
switched to db concesionario
```

## Instalacion de cliente para mongodb

---

Para ello nos iremos a la maquina virtual donde tenemos todos los clientes a excepción del de oracle, y procederemos a instalar lo siguiente:

### 1. Importar la clave pública

```
andy@cliente-mariadb-mongo-postgres:~$ gpg --no-default-keyring --keyring
./mongo_key_temp.gpg --import ./mongoserver.asc
gpg: creado el directorio '/home/andy/.gnupg'
gpg: /home/andy/.gnupg/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave 160D26BB1785BA38: clave pública "MongoDB 7.0 Release Signing Key
<packaging@mongodb.com>" importada
gpg: Cantidad total procesada: 1
gpg:
importadas: 1
andy@cliente-mariadb-mongo-postgres:~$ gpg --no-default-keyring --keyring
./mongo_key_temp.gpg --export > ./mongoserver_key.gpg
andy@cliente-mariadb-mongo-postgres:~$ wget -qO -
https://www.mongodb.org/static/pgp/server-7.0.asc | sudo tee
/etc/apt/trusted.gpg.d/mongodb-server-7.0.asc
andy@cliente-mariadb-mongo-postgres:~$ echo "deb [ arch=amd64, arm64 ]
https://repo.mongodb.org/apt/debian buster/mongod b-org/7.0 main" | sudo
tee /etc/apt/sources.list.d/mongodb-org-7.0.list
deb [ arch=amd64, arm64 ]_https://repo.mongodb.org/apt/debian
buster/mongodb-org/7.0 main
deb [ arch=amd64, arm64 ] https://repo.mongodb.org/apt/debian buster/mongod
b-org/7.0 main
```

- ¿Qué hace? Importa una clave pública de MongoDB que asegura la autenticidad del software que se descargará.
- Resultado: Se crea un archivo temporal de claves en tu computadora y se guarda la clave de MongoDB.

### 2. Exportar la clave a otro archivo

```
andy@cliente-mariadb-mongo-postgres:~$ gpg --no-default-keyring --keyring
./mongo_key_temp.gpg --import ./mongoserver.asc
gpg: creado el directorio '/home/andy/.gnupg'
gpg: /home/andy/.gnupg/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave 160D26BB1785BA38: clave pública "MongoDB 7.0 Release Signing Key
<packaging@mongodb.com>" importada
gpg: Cantidad total procesada: 1
gpg:
importadas: 1
```

### 3. Agregar el repositorio MongoDB a nuestro sistema

```
andy@cliente-mariadb-mongo-postgres:~$ wget -qO -
https://www.mongodb.org/static/pgp/server-7.0.asc | sudo tee
/etc/apt/trusted.gpg.d/mongodb-server-7.0.asc
```



#### 4. Agregar la fuente de software de MongoDB

```
andy@cliente-mariadb-mongo-postgres:~$ echo "deb [ arch=amd64,arm64 ]
https://repo.mongodb.org/apt/debian buster/mongodb-org/7.0 main" | sudo tee
/etc/apt/sources.list.d/mongodb-org-7.0.list
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/debian
buster/mongodb-org/7.0 main
```

#### 5. Instalar MongoDB Shell y MongoDB Shell interactivo Para ello nos iremos a la maquina virtual donde tenemos todos los clientes a excepción del de oracle, y procederemos a instalar lo siguiente:

```
andy@cliente-mariadb-mongo-postgres:~$ sudo apt install mongodb-org-shell
mongodb-mongosh
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mongodb-org-shell ya está en su versión más reciente (6.0.18).
Se instalarán los siguientes paquetes NUEVOS:
  mongodb-mongosh
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 54,1 MB de archivos.
Se utilizarán 268 MB de espacio de disco adicional después de esta
operación.
¿Desea continuar? [S/n] s
Des:1 https://repo.mongodb.org/apt/debian bullseye/mongodb-org/4.4/main
amd64 mongodb-mongosh amd64 2.3.2 [54,1 MB]
Descargados 54,1 MB en 9s (6.335 kB/s)
Seleccionando el paquete mongodb-mongosh previamente no seleccionado.
(Leyendo la base de datos ... 39766 ficheros o directorios instalados
actualment
e.)
Preparando para desempaquetar .../mongodb-mongosh_2.3.2_amd64.deb ...
Desempaquetando mongodb-mongosh (2.3.2) ...
Configurando mongodb-mongosh (2.3.2) ...
Procesando disparadores para man-db (2.11.2-2) ...
```

Ahora procederemos a meternos remotamente, ya que como vimos anets tenemos configurado el servidor de *MongoDB*, lo haremos con el siguiente comando:

```
mongosh --host 192.168.1.159 --port 27017 -u andy -p andy --
authenticationDatabase admin
```

Y si lo probamos por terminal veremos lo siguiente:

```
ndy@cliente-mariadb-mongo-postgres:~$ mongosh --host 192.168.1.159 --port
27017 -u andy -p andy --authenticationDatabase admin
Current Mongosh Log ID: 670932be4dbc6f3d16fe6910
Connecting to:      mongodb://<credentials>@192.168.1.159:27017/?
directConnection=true&authSource=admin&appName=mongosh+2.3.2
Using MongoDB:      4.4.29
Using Mongosh:      2.3.2
```

For mongosh info see: <https://www.mongodb.com/docs/mongodb-shell/>

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (<https://www.mongodb.com/legal/privacy-policy>). You can opt-out by running the `disableTelemetry()` command.

-----

The server generated these startup warnings when booting  
2024-10-11T15:38:28.271+02:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See <http://dochub.mongodb.org/core/prodnotes-filesystem>

2024-10-11T15:38:28.661+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2024-10-11T15:38:28.661+02:00:  
/sys/kernel/mm/transparent\_hugepage/enabled is 'always'. We suggest setting it to 'never'

-----

test>

Y vemos podemos ver las siguientes base de datos:

```
test> show dbs
admin          132.00 KiB
concesionario  40.00 KiB
config         72.00 KiB
local          72.00 KiB
test>
```

y podremos cambiar a las distintas bases de datos:

```
test> use admin
switched to db admin
admin>
```

