



INSTALAR MONGODB

Actividad 4



CRISTÓBAL SUÁREZ ABAD

ADMINISTRACIÓN DE SISTEMAS GESTORES DE BASES DE DATOS
2º ASIR

Introducción:

Sigue los siguientes pasos para instalar un cluster de MongoDB:

<https://serverspace.io/es/support/help/mongodb-cluster-configuration/>

- En el paso 1 de configuración del cluster sustituyelo por esto

Añade estas líneas en `/etc/mongod.conf` en todos los nodos.

replication:

replSetName: "rs0"

net:

bindIp: localhost, domainname

--> domainname sera el nombre de dominio del propio nodo

- Una vez realizado esto
 - Instala un cliente en tu equipo local y conéctate a la base de datos de mongo
 - Crea una nueva colección e inserta datos de prueba
 - Apaga el nodo primario y comprueba que algunos de los nodos secundario se ha transformado en primario.

Configuración de los servidores.

Detalles de los servidores utilizados:

1. Servidor primario: nombre — mongodb01, IP 10.2.17.80
2. Servidor secundario: nombre — mongodb02, IP 10.2.17.85
3. Servidor secundario: nombre — mongodb03, IP 10.2.17.90

Les cambiamos el nombre con el comando:

```
hostnamectl set-hostname mongodb01
```

En el archivo “/etc/hosts” ponemos las direcciones completas de cada uno, en cada uno de los servidores.

```
Tu Nombre jueves 6 noviembre 2025 10:00
[usuario@mongodb01 ~]$cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 mongodb01.suarez.abad mongodb01
10.2.17.80 mongodb01.suarez.abad mongodb01
10.2.17.85 mongodb02.suarez.abad mongodb02
10.2.17.90 mongodb03.suarez.abad mongodb03

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
Tu Nombre jueves 6 noviembre 2025 10:00
```

Instalamos mongo.

- 1) ***sudo apt update***
- 2) ***sudo apt install -y gnupg curl***
- 3) ***curl -fsSL https://pgp.mongodb.com/server-7.0.asc | ***
sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg --dearmor
echo "deb [signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg]
\ https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0
multiverse" |
\ sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list
- 4) ***sudo apt update***
- 5) ***sudo apt install -y mongodb-org***

Configuración del Cluster.

Una vez que tenemos instalado MongoDB en cada uno de los servidores, debemos realizar los siguientes pasos.

Podemos habilitarlo para que se inicie el servicio de MongoDB cada vez que se inicia el servidor.

systemctl enable mongod

systemctl restart mongod

systemctl status mongod

Si lo activas ahora tendrás que hacerle un “restart” cuando acabes de modificarlo.

Ahora, en el archivo “/etc/mongod.conf” de cada servidor debemos introducir la información que se mencionó en la introducción del trabajo. En nuestro caso lo hemos hecho así.

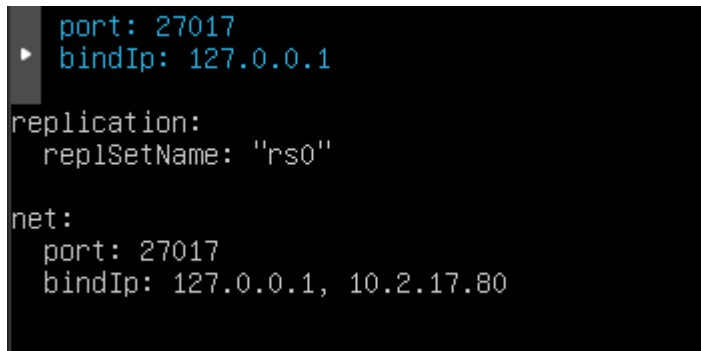
“replication:

replSetName: “rs0”

net:

port: 27017

bindIp: 127.0.0.1, 10.2.17.80”



```
port: 27017
bindIp: 127.0.0.1
replication:
  replSetName: "rs0"
net:
  port: 27017
  bindIp: 127.0.0.1, 10.2.17.80
```

Cada servidor debe poner su IP.

Ahora nos metemos en la “Shell” de MongoDB usando el comando “mongosh”.

```

Tu Nombre  jueves  6 noviembre 2025 10:10
[usuario@mongodb01 ~]$ mongosh
Current Mongosh Log ID: 690c741b15723ca68a9dc29c
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionT
Using MongoDB:      7.0.25
Using Mongosh:      2.5.9

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

-----
The server generated these startup warnings when booting
2025-11-06T09:59:08.278+00:00: Using the XFS filesystem is strongly recommended with the
tes-filesystem
2025-11-06T09:59:09.986+00:00: Access control is not enabled for the database. Read an
2025-11-06T09:59:09.987+00:00: vm.max_map_count is too low
-----

```

Aquí debemos introducir el siguiente código:

```

rs.initiate({
  _id: "rs0",
  members: [
    { _id: 0, host: "10.2.17.80:27017" },
    { _id: 1, host: "10.2.17.85:27017" },
    { _id: 2, host: "10.2.17.90:27017" }
  ]
})

```

“rs0” es el nombre del clúster y el resto de “id” identifican a los servidores. Nos debe salir algo como lo de la imagen de abajo.

```

C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
test> rs.initiate({
...   _id: "rs0",
...   members: [
...     { _id: 0, host: "10.2.17.80:27017" },
...     { _id: 1, host: "10.2.17.85:27017" },
...     { _id: 2, host: "10.2.17.90:27017" }
...   ]
... })
... {
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1762377823, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1762377823, i: 1 })
}
rs0 [direct: other] test>

```

Si ahora usamos el comando “test> rs.conf()”, podremos obtener más información sobre el resto de servidores.

```
rs0 [direct: other] test> rs.conf()
{
  _id: 'rs0',
  version: 1,
  term: 1,
  members: [
    {
      _id: 0,
      host: '10.2.17.80:27017',
      arbiterOnly: false,
      buildIndexes: true,
      hidden: false,
      priority: 1,
      tags: {},
      secondaryDelaySecs: Long('0'),
      votes: 1
    },
    {
      _id: 1,
      host: '10.2.17.85:27017',
      arbiterOnly: false,
      buildIndexes: true,
      hidden: false,
      priority: 1,
      tags: {},
      secondaryDelaySecs: Long('0'),
      votes: 1
    },
    {
      _id: 2,
      host: '10.2.17.90:27017',
      arbiterOnly: false,
      buildIndexes: true,
      hidden: false,
```

Cambiar la prioridad:

Usamos los siguientes comandos:

```
conf.members[0].priority = 1
```

```
conf.members[1].priority = 2
```

```
conf.members[2].priority = 3
```

```
rs0 [direct: primary] test> conf.members[0].priority = 1
1
rs0 [direct: primary] test> conf.members[1].priority = 2
2
rs0 [direct: primary] test> conf.members[2].priority = 3
3
```

Luego usamos “rs.reconfig(conf)” para que se aplique esta nueva configuración.

```
rs0 [direct: primary] test> rs.reconfig(conf)
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1762379007, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1762379007, i: 1 })
}
```

Ahora podemos verificar el estado de la configuración con el comando “rs.status()”.

```
rs0 [direct: secondary] test> rs.status()
{
  set: 'rs0',
  date: ISODate('2025-11-05T21:48:14.817Z'),
  myState: 2,
  term: Long('2'),
  syncSourceHost: '10.2.17.90:27017',
  syncSourceId: 2,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
    lastCommittedWallTime: ISODate('2025-11-05T21:48:08.157Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
    appliedOpTime: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
    durableOpTime: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
    lastAppliedWallTime: ISODate('2025-11-05T21:48:08.157Z'),
    lastDurableWallTime: ISODate('2025-11-05T21:48:08.157Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1762379258, i: 1 }),
  electionParticipantMetrics: {
    votedForCandidate: true,
    electionTerm: Long('2'),
    lastVoteDate: ISODate('2025-11-05T21:43:38.123Z'),
    electionCandidateMemberId: 2,
    voteReason: '',
    lastAppliedOpTimeAtElection: { ts: Timestamp({ t: 1762379007, i: 1 }), t: Long('1') },
    maxAppliedOpTimeInSet: { ts: Timestamp({ t: 1762379007, i: 1 }), t: Long('1') }
  },
  priorityAtElection: 1,
  newTermStartDate: ISODate('2025-11-05T21:43:38.141Z'),
  newTermAppliedDate: ISODate('2025-11-05T21:43:38.148Z')
},
members: [
  {

```

Ahí nos deben salir información sobre cada uno de los servidores. Nos importa en este caso el valor de la prioridad, que puede ser PRIMARY o SECONDARY.


```

members: [
  {
    _id: 0,
    name: '10.2.17.80:27017',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 1732,
    optime: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
    optimeDate: ISODate('2025-11-05T21:48:08.000Z'),
    lastAppliedWallTime: ISODate('2025-11-05T21:48:08.157Z'),
    lastDurableWallTime: ISODate('2025-11-05T21:48:08.157Z'),
    syncSourceHost: '10.2.17.90:27017',
    syncSourceId: 2,
    infoMessage: '',
    configVersion: 2,
    configTerm: 2,
    self: true,
    lastHeartbeatMessage: ''
  },
  {
    _id: 1,
    name: '10.2.17.85:27017',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 1471,
    optime: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
    optimeDurable: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
    optimeDate: ISODate('2025-11-05T21:48:08.000Z'),
    optimeDurableDate: ISODate('2025-11-05T21:48:08.000Z'),
    lastAppliedWallTime: ISODate('2025-11-05T21:48:08.157Z'),
    lastDurableWallTime: ISODate('2025-11-05T21:48:08.157Z'),
    lastHeartbeat: ISODate('2025-11-05T21:48:14.805Z'),
    lastHeartbeatRecv: ISODate('2025-11-05T21:48:14.231Z'),
    pingMs: Long('0'),
    lastHeartbeatMessage: '',
    syncSourceHost: '10.2.17.80:27017',
    syncSourceId: 0,
    infoMessage: ''
  }
]

```

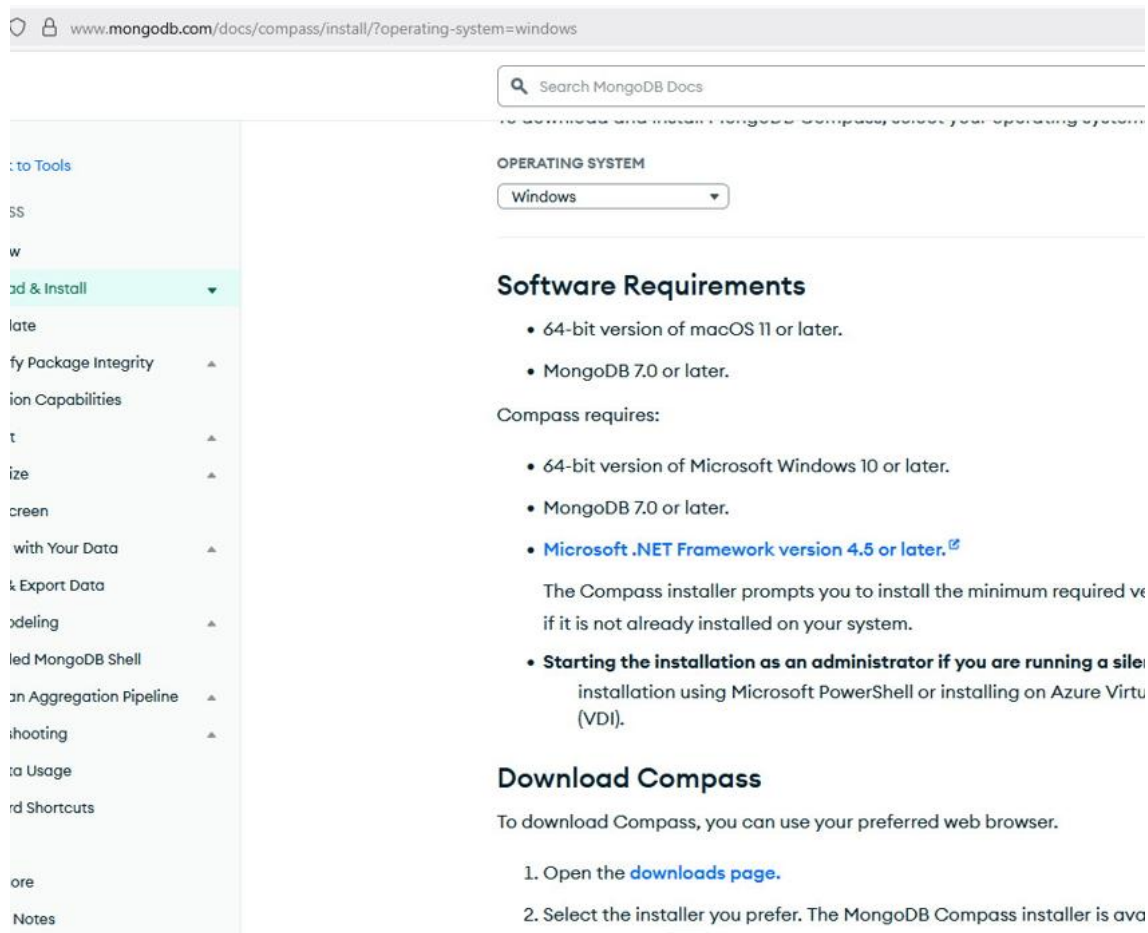
```

{
  _id: 2,
  name: '10.2.17.90:27017',
  health: 1,
  state: 1,
  stateStr: 'PRIMARY',
  uptime: 1471,
  optime: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
  optimeDurable: { ts: Timestamp({ t: 1762379288, i: 1 }), t: Long('2') },
  optimeDate: ISODate('2025-11-05T21:48:08.000Z'),
  optimeDurableDate: ISODate('2025-11-05T21:48:08.000Z'),
  lastAppliedWallTime: ISODate('2025-11-05T21:48:08.157Z'),
  lastDurableWallTime: ISODate('2025-11-05T21:48:08.157Z'),
  lastHeartbeat: ISODate('2025-11-05T21:48:14.806Z'),
  lastHeartbeatRecv: ISODate('2025-11-05T21:48:14.271Z'),
  pingMs: Long('0'),
  lastHeartbeatMessage: '',
  syncSourceHost: '',
  syncSourceId: -1,
  infoMessage: '',
  electionTime: Timestamp({ t: 1762379018, i: 1 }),
  electionDate: ISODate('2025-11-05T21:43:38.000Z'),
  configVersion: 2,
  configTerm: 2
}

```


Instala un cliente en tu equipo local y conéctate a la base de datos de mongo.

Nos dirigimos a la página oficial de mongo y elegimos el sistema operativo donde lo vamos a instalar: <https://www.mongodb.com/docs/compass/install/?operating-system=windows> En nuestro caso, Windows.



The screenshot shows the MongoDB Compass installation page for Windows. The browser address bar displays www.mongodb.com/docs/compass/install/?operating-system=windows. A search bar at the top right is labeled "Search MongoDB Docs". On the left, a sidebar lists navigation options, with "Add & Install" highlighted. The main content area features a dropdown menu for "OPERATING SYSTEM" set to "Windows". Below this, the "Software Requirements" section lists the following:

- 64-bit version of macOS 11 or later.
- MongoDB 7.0 or later.

Compass requires:

- 64-bit version of Microsoft Windows 10 or later.
- MongoDB 7.0 or later.
- **Microsoft .NET Framework version 4.5 or later.**

The text states: "The Compass installer prompts you to install the minimum required version if it is not already installed on your system."

Additional requirements include:

- **Starting the installation as an administrator if you are running a silent installation using Microsoft PowerShell or installing on Azure Virtual Desktop (VDI).**

The "Download Compass" section provides instructions:

To download Compass, you can use your preferred web browser.

1. Open the [downloads page](#).
2. Select the installer you prefer. The MongoDB Compass installer is available for download.

Le damos al archivo ejecutable y una vez abierto el programa nos vamos a añadir nueva conexión.



The screenshot shows the "Welcome to MongoDB Compass" screen. On the left, there is a logo featuring a magnifying glass over a yellow bar. The main text reads "Welcome to MongoDB Compass" and "To get started, connect to an existing server or". Below this is a green button labeled "+ Add new connection". A light blue box contains the text "New to Compass and don't have a cluster?" followed by "If you don't already have a cluster, you can create one for free using [MongoDB Atlas](#)". At the bottom of this box is a button labeled "CREATE FREE CLUSTER".

Introducimos el nombre o IP de nuestros servidores:

New Connection

Manage your connection settings

URI ⓘEdit Connection String

mongodb://10.2.17.90:27017

Name
10.2.17.90:27017

Color
No Color

☐ **Favorite this connection**
Favoriting a connection will pin it to the top of your list of connections

[Advanced Connection Options](#)

How do I find my connection string in Atlas?
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.
[See example](#)

How do I format my connection string?
[See example](#)

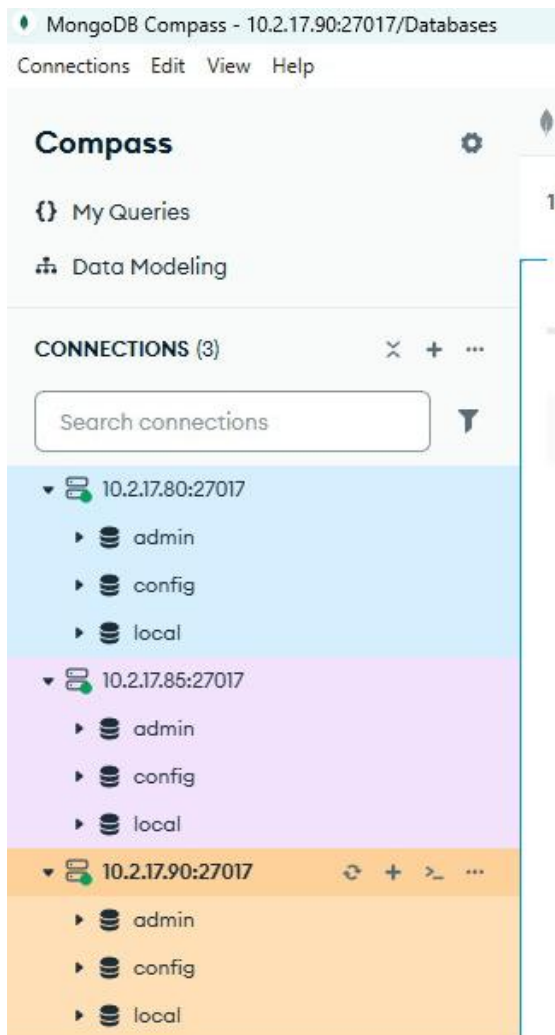
TLS/SSL is disabled. If possible, enable TLS/SSL to avoid security vulnerabilities.

Cancel

Save

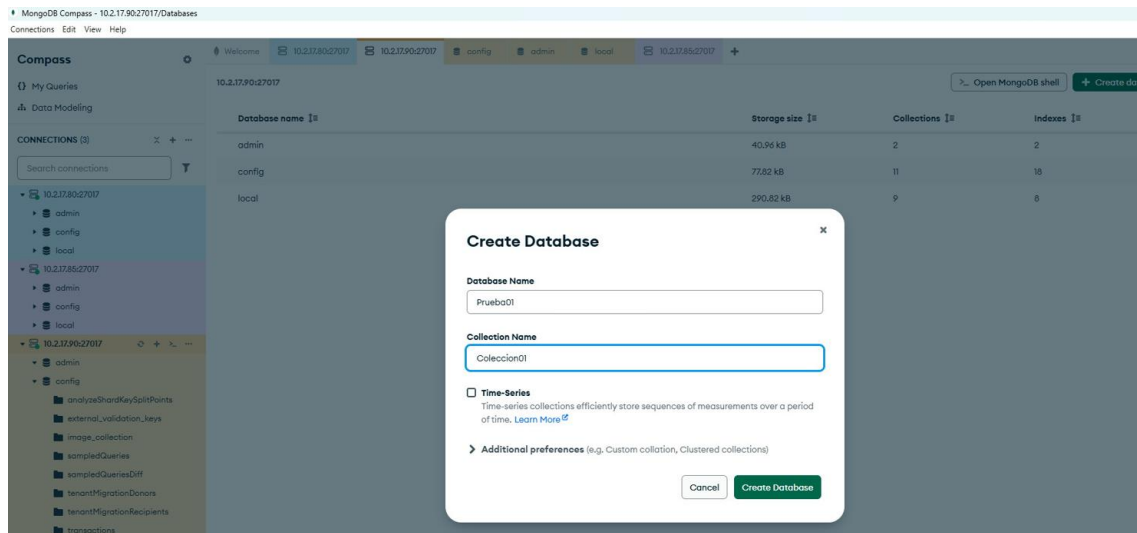
Connect

Save & Connect

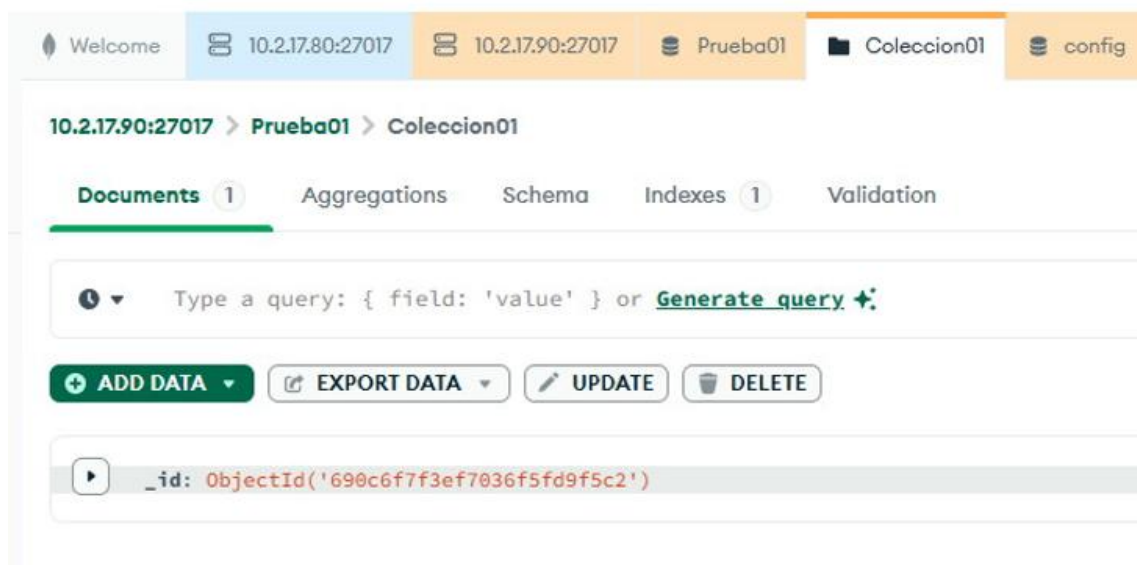


Crea una nueva colección e inserta datos de prueba.

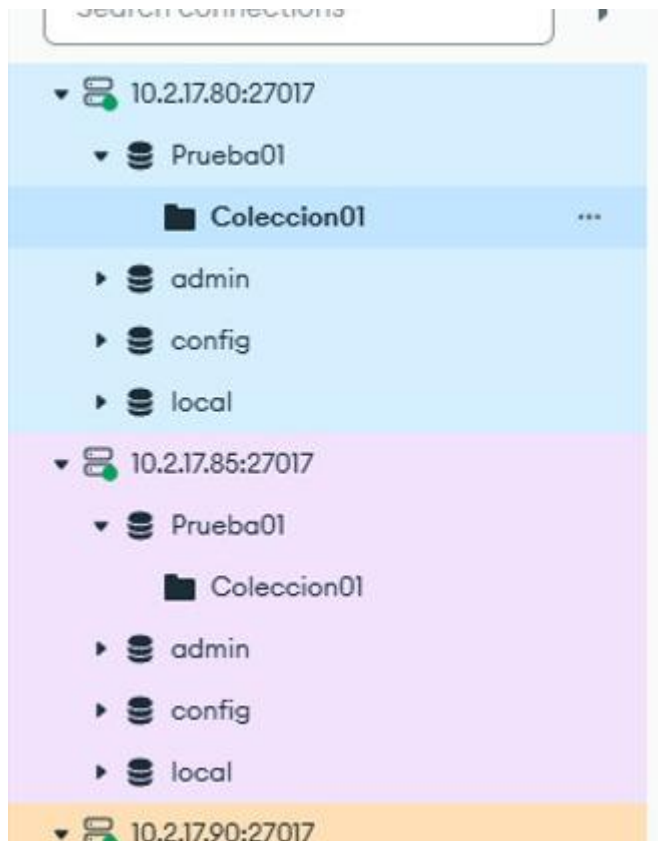
Seleccionamos un servidor y en este caso le vamos a crear una nueva base de datos y dentro de ella una nueva colección.



Seleccionamos la colección creada y le damos a “Add Data”. Ahí nos da la opción de modificar una plantilla genérica para meter un documento. Como en este caso no se ha especificado nada sobre el contenido de la información, se ha decidido simplemente meter la plantilla.



Si actualizamos el estado de los otros dos servidores, podemos ver como se ha replicado la base de datos y la colección (y también la información insertada).



Apaga el nodo primario y comprueba que algunos de los nodos secundario se han transformado en primario.

Vamos a apagar el servidor “mongodb03” que en nuestro caso es el primario.

```
Tu Nombre  jueves  6 noviembre 2025 09:46
[root@mongodb03 usuario]$mongosh
Current Mongosh Log ID: 690c700933f3e4ae409dc29c
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true
Using MongoDB:          7.0.25
Using Mongosh:          2.5.9

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

-----
  The server generated these startup warnings when booting
  2025-11-06T09:38:02.944+00:00: Using the XFS filesystem is strongly
  tes-filesystem
  2025-11-06T09:38:04.613+00:00: Access control is not enabled for t
  2025-11-06T09:38:04.613+00:00: vm.max_map_count is too low
  -----

rs0 [direct: primary] test> _
```

Una vez que lo apagamos, el servidor “mongodb02” se convierte en primario.

```
Tu Nombre  jueves  6 noviembre 2025 09:54
[root@mongodb02 usuario]$mongosh
Current Mongosh Log ID: 690c704113bf9dc9b139dc29c
Connecting to:          mongodb://127.0.0.1:27017/?directConnec
Using MongoDB:          7.0.25
Using Mongosh:          2.5.9

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

To help improve our products, anonymous usage data is collected
You can opt-out by running the disableTelemetry() command.

-----
  The server generated these startup warnings when booting
  2025-11-06T09:38:00.042+00:00: Using the XFS filesystem is s
  tes-filesystem
  2025-11-06T09:38:01.666+00:00: Access control is not enabled
  2025-11-06T09:38:01.667+00:00: vm.max_map_count is too low
  -----

rs0 [direct: primary] test> _
```