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.

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 temenos 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.

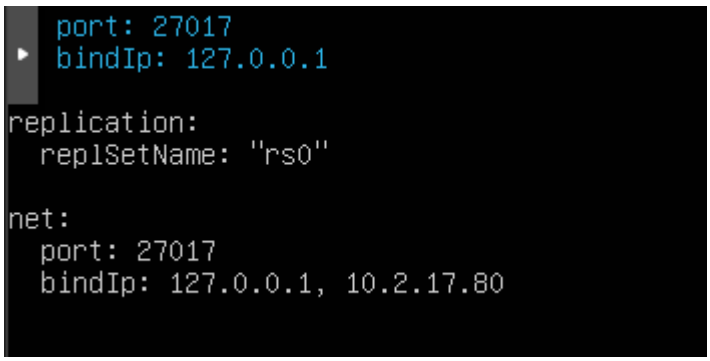
"replication:

replSetName: "rs0"

net:

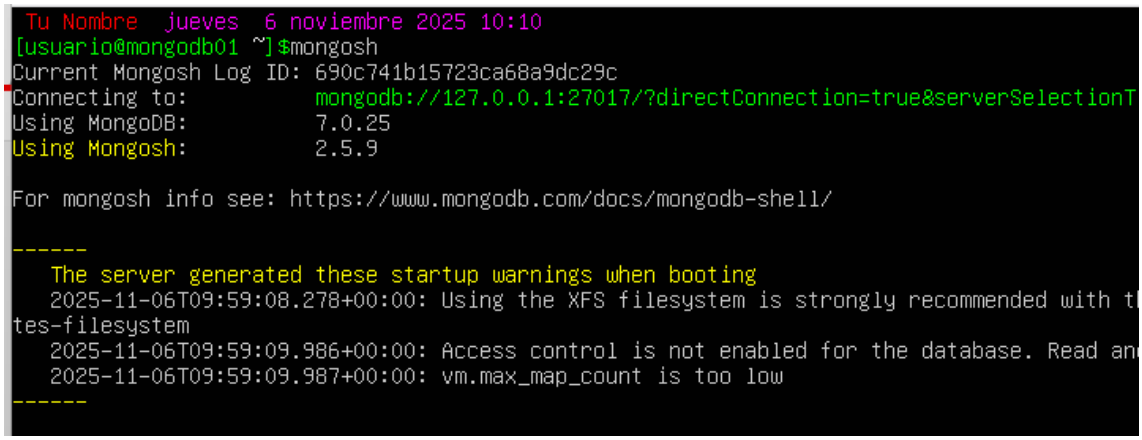
port: 27017

bindIp: 127.0.0.1, 10.2.17.80"



Cada servidor debe poner su IP.

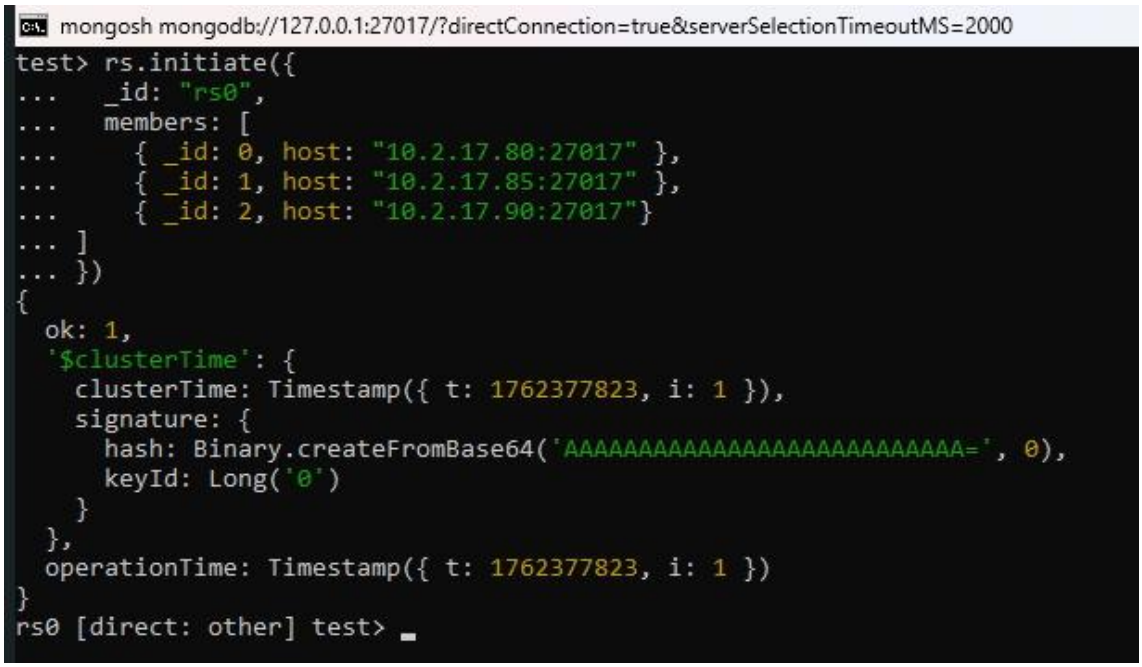
Ahora nos metemos en la "Shell" de MongoDB usando el comando "mongosh".



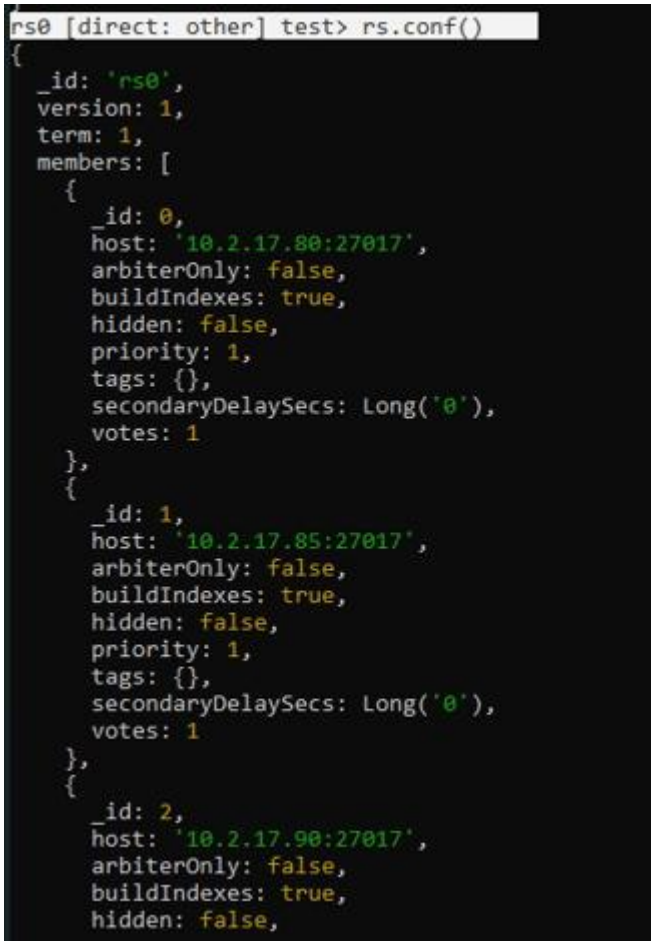
Ahí debemos introducir el siguiente código:

```
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.



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



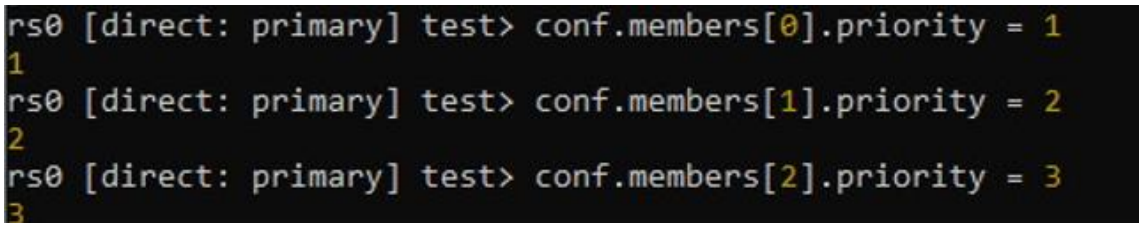
Cambiar la prioridad:

Usamos los siguientes comandos:

conf.members[0].priority = 1

conf.members[1].priority = 2

conf.members[2].priority = 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('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 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: [
    {
```

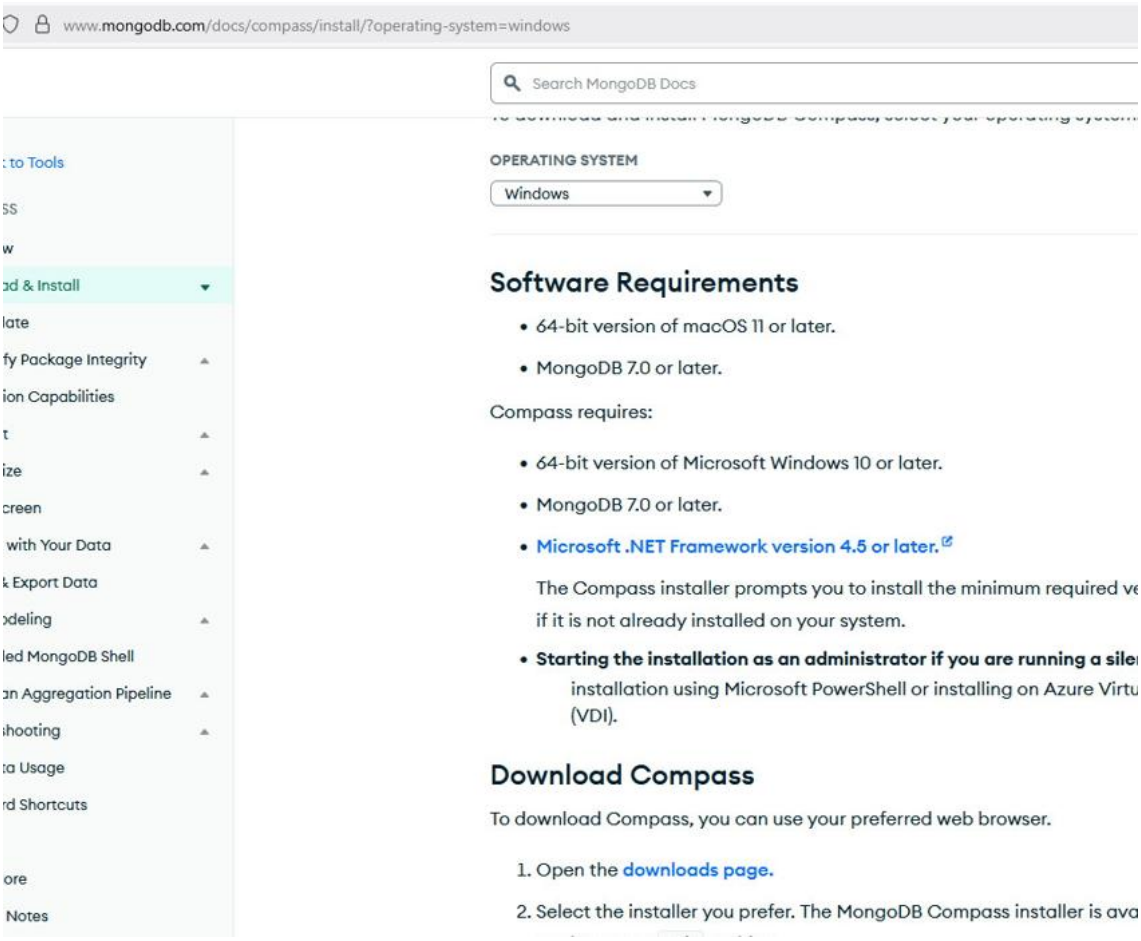
Ahí nos deben salir información sobre cada uno de los servidores. Nos importa en este caso el valor de la prioridad, que puede ser PRIMARY o SECONDARY.

```
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
}
,
```

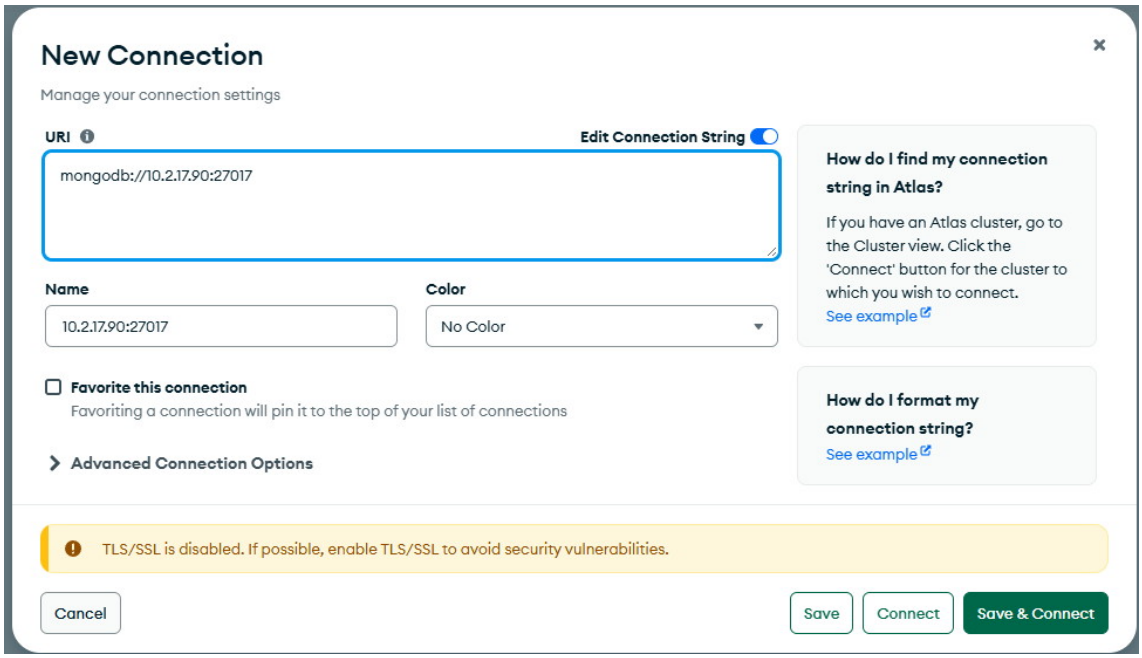
Instala un cliente en tu equipo local y conéctate a la base de datos de mongo.

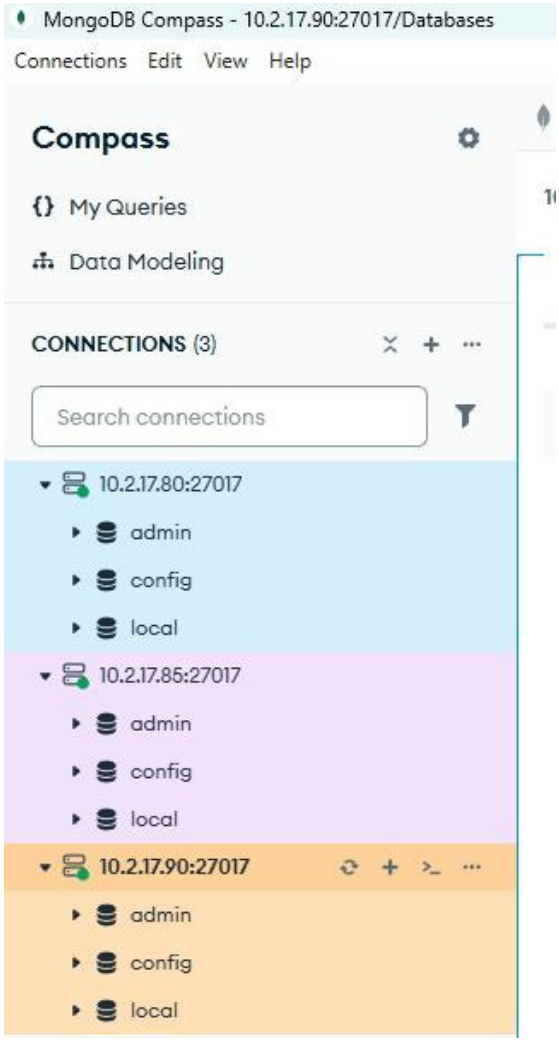
Nos dirigimos a la página oficial de mongo y elegimos el sistema operativo donde lo vamos a instalar: https://www.mongodb.com/docs/compass/install/?operating-system=windows En nuestro caso, Windows.



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

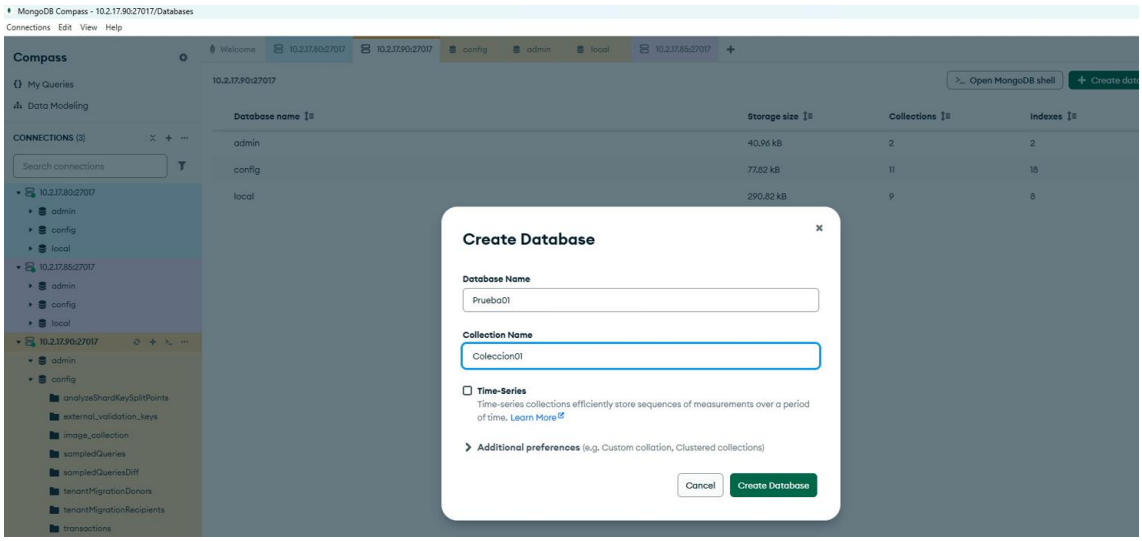
Introducimos el nombre o IP de nuestros servidores:



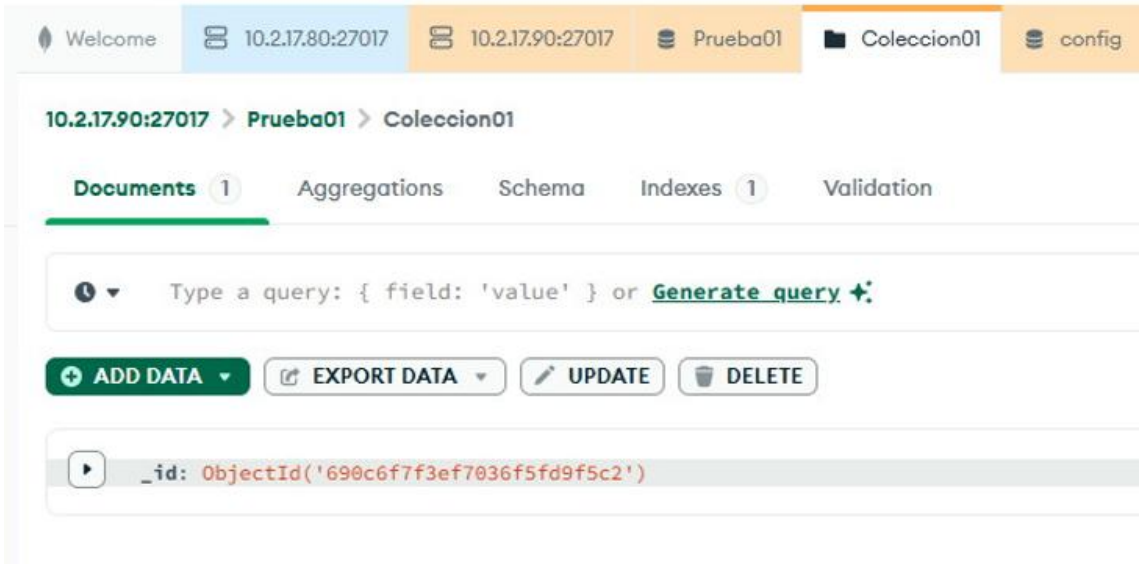


Crea una nueva colección e inserta datos de prueba.

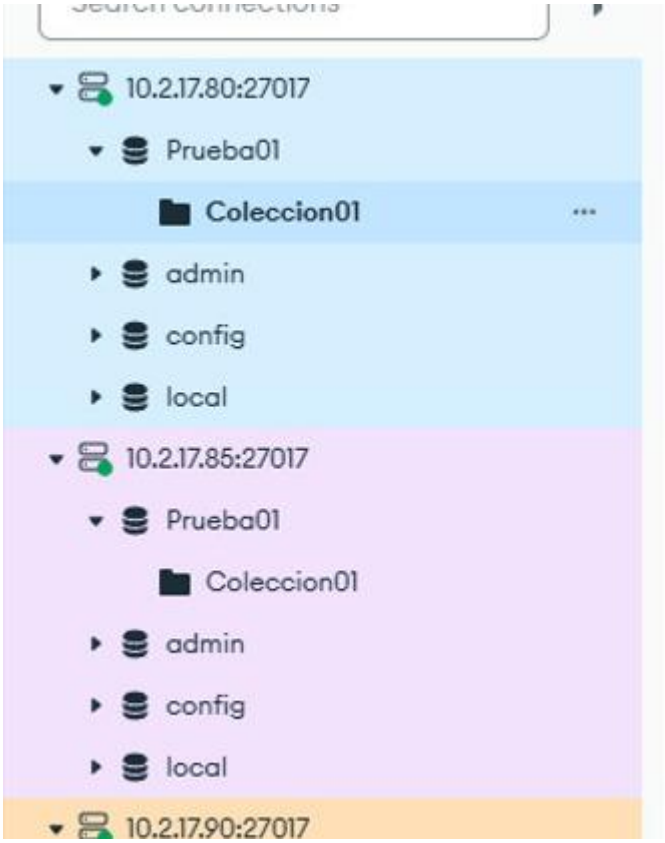
Seleccionamos un servidor y en este caso le vamos a crear una nueva base de datos y dentro de ella una nueva colección.



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.

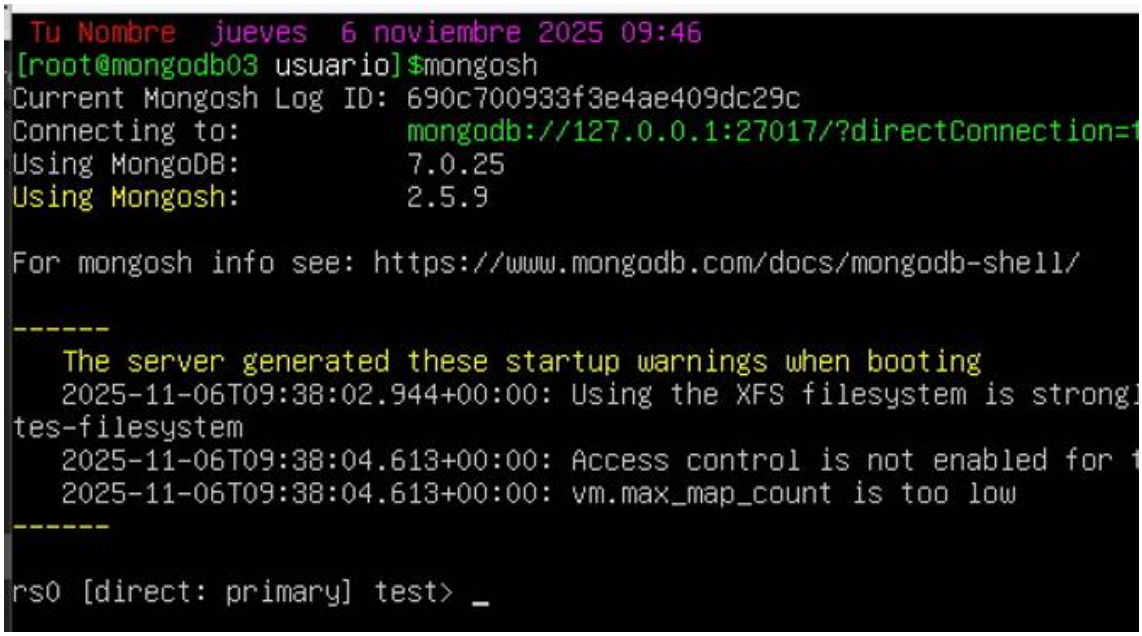


Si actualizamos el estado de los otros dos servidores, podemos ver como se ha replicado la base de datos y la colección (y también la información insertada).



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.



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