

**MONGODB SHARDED SETUP GUIDE**

**The Below requirements are needed to establish a connection mongodb sharded .**

**Prerequisites**

Prerequisiteslink  
To connect your MongoDB database to Fivetran, you need:  
  
MongoDB version 2.6 to 5.0  
Your database host's IP (e.g., 1.2.3.4) or domain (your.server.com)  
Your database's port (usually 27017)  
TLS enabled on your database (if you want to connect to Fivetran directly)  
  
To perform incremental updates with change streams PRIVATE PREVIEW, you need:  
  
MongoDB version 3.6 or above  
Read Majority read concern activated  
A replication set using replica set protocol version 1.  
  
 cfg = rs.conf();   
 cfg.protocolVersion=1;   
 rs.reconfig(cfg);  
content\_copy  
  
Note: Fivetran does not support Mongo tiers M0, M2, or M5 because MongoDB's smaller managed tiers do not provide oplogs. We need oplogs to perform incremental updates. Mongo sharded cluster does not support SRV host. You must provide the MongoDB query router's IP instead. We do support SRV host in Mongo Replica Sets.

**Setup Guide**

* This step allows Fivetran to access your database.

1. Connect to MongoDB's query router

* Connect Fivetran to the mongos query router node in your MongoDB sharded cluster.

1. Allow port access

* Configure your firewall and/or other access control systems to allow incoming connections to all your mongo instances from Fivetran's IP or your SSH tunnel server's IP address.

1. SSL support

* Our MongoDB connector supports encrypted connection to MongoDB over SSL if your database is setup to allow clients to connect without a certificate.

1. Create user

* Create an identical Fivetran user in every primary shard node and the mongos query router. Replace <username> and <password> with a username and password of your choice.
* use admin

|  |
| --- |
| db.createUser({  user: "<username>",  pwd: "<password>",  roles: [ "readAnyDatabase", "clusterMonitor" ] }) |

* content\_copy
* You can pick any username and password, but it must be consistent across all primary shard nodes and the mongos query router.
* The Mongo shell command to create a user is db.AddUser() for versions 2.x.
* If you want to find out which shards are connected to the mongos query router, use the command db.adminCommand({ listShards: 1 }).

1. (Optional) Configure AWS PrivateLink BETA

* IMPORTANT: You must have a Business Critical plan to use AWS PrivateLink.
* AWS PrivateLink allows VPCs and AWS-hosted or on-premises services to communicate with one another without exposing traffic to the public internet. Learn more in AWS’ PrivateLink documentation.
* Follow our AWS PrivateLink setup guide to configure PrivateLink for your database.

1. Choose schema prefix

* Fivetran maps the schemas we discover in your source database to your destination and prepends the destination schema names with a prefix of your choice. For example, if your original database contains schemas foo and bar and if you choose the prefix pre, then your destination schemas will be pre\_foo and pre\_bar.

1. Setup tests

* Fivetran performs the following tests to ensure that we can connect to your MongoDB sharded cluster and that it is properly configured:
* The Connecting to SSH Tunnel Test validates the SSH tunnel details you provided in the setup form. It then checks that we can connect to your database using the SSH Tunnel. (We skip this test if you aren't connecting using SSH.)
* The Connecting to Host Test validates the database credentials you provided in the setup form. It then verifies that the database host is not private and checks that we can connect to the host.
* The Validating Certificate Test generates a pop-up window where you must choose which certificate you want Fivetran to use. It then validates that certificate and checks that we can connect to your database using TLS. (We skip this test if you aren't connecting directly.)
* The Connecting to Database Test connects to your database instance and verifies that your database is at least version 2.6. It then checks that we can access the schemas in your database. Lastly, it verifies that we can access the oplog and that it contains at least 24 hours' worth of changes.
* NOTE: The tests may take a few minutes to finish running.

1. Connecting with MongoDB Atlas

* In order to connect with MongoDB Atlas, follow the steps below:

1. Create an oplog user

* Log in to the MongoDB Atlas dashboard.
* In the Security section of the left menu, select Database Access.
* Click New Database User.
* Choose the password authentication method.
* Enter a username and password for the new user.
* In the Database User Privileges drop-down menu, select Grant Specific User Privileges.
* Under Specific Privileges, add the following roles/privileges:
* readAnyDatabase
* read on the local database
* clusterMonitor
* Click Add User.
* For general instructions, see MongoDB Atlas's documentation.

1. Note

* Set the oplog size so that it can retain at least 24 hours' worth of changes. We recommend increasing the size to accommodate seven days' worth of data. See this MongoDB tutorial to learn how to adjust the size of your oplog or this MongoDB Atlas Doc.
* If you are using MongoDB version 4.0 or later, you must change the read concern to read majority. Learn how in MongoDB's write and read concern documentation.

