

**MYSQL AURORA SETUP GUIDE**

**The Below requirements are needed to establish a connection mysql aurora .**

**Prerequisites**

Prerequisiteslink  
  
IMPORTANT: We do not support serverless Aurora.  
  
To connect your Amazon Aurora MySQL database to Fivetran, you need:  
  
MySQL version 5.6.13 or above  
  
NOTE: If you upgrade your database version to 8.0.23 or above, you must set the binlog\_row\_metadata value to MINIMAL before you do the upgrade. If you set binlog\_row\_metadata to MINIMAL after the upgrade, your existing Aurora MySQL connectors will fail and you may need re-sync them.  
  
  
Your database host's IP (e.g., 1.2.3.4) or domain (your.server.com)  
A unique replica ID for Fivetran. We need this ID because we connect to your database as a replica. We provide a random replica ID in your setup form, but you can provide your own if you'd prefer or if the form's replica ID conflicts with one of your existing replica IDs.  
  
NOTE: The replica ID is a unique ID within the MySQL replica set. By default, the replica ID is a random integer greater than 1000.

**Setup Guide**

1. Choose connection method

* First, decide whether to connect Fivetran to your Amazon Aurora MySQL database directly, using an SSH tunnel, or using AWS PrivateLink. How you configure your security groups in later steps will differ depending on this decision.

1. Connect directly (TLS required)

* IMPORTANT: You must have TLS enabled on your database to connect directly to Fivetran. Follow Amazon's Aurora MySQL TLS setup instructions to enable TLS on your database.
* Fivetran connects directly to your Amazon Aurora MySQL database. This is the simplest method.
* If you connect directly, you will create a rule in a security group that allows Fivetran access to your database instance.

1. Connect using SSH (TLS optional)

* Fivetran connects to a separate server in your network that provides an SSH tunnel to your database. You must connect through SSH if your database is in an inaccessible subnet.
* If you connect using SSH, you will configure your tunnel server's security group to allow Fivetran access and configure your database's security to allow access from the tunnel.
* Before you proceed to the next step, you must follow our SSH connection instructions. If you want Fivetran to tunnel SSH over TLS, follow Amazon's Aurora MySQL TLS setup instructions to enable TLS on your database.

1. Connect using 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. Find endpoint and port

* Find the endpoint and port for the database that you want to connect to Fivetran. You must connect Fivetran to your master/writer database because AWS does not allow you to activate binary logging on an Aurora MySQL reader. We require binary logs to perform incremental updates.
* In your Amazon RDS dashboard, click on the Amazon Aurora MySQL database that you want to connect to Fivetran.
* NOTE: The database's role must be "writer."
* (If you're connecting directly) In the Connectivity & security section, ensure that the Publicly Accessibility setting is Yes.
* IMPORTANT: If you're connecting using SSH, skip this step. You do not have to make your database publicly accessible.
* Find the endpoint and port and make a note of them. You will need them to configure Fivetran.

1. Enable database access

* Grant Fivetran's data processing servers access to your master/writer database.

1. Configure security groups

* Configure your Amazon Aurora MySQL cluster's VPC security group to allow Fivetran to access the cluster.

1. In the Security column, click the to your database's security group.

* On the Security Groups page, click on the security group ID.
* On the Inbound tab, click Edit inbound rules.
* Click Add Rule. This creates a new Custom TCP Rule at the bottom of the list.
* Fill in the new Custom TCP Rule.
* In the Port Range field, enter your database's port number that you copied in the previous section.
* What you enter in the Source Custom IP field depends on whether you're connecting directly or using SSH.
* If you're connecting directly, enter Fivetran's IPs for your database's region.
* If you're connecting using SSH, enter {your-ssh-tunnel-server-ip-address}/32.
* (Optional) Enter a brief description in the Description field.
* Click Save rules.

1. Configure Network ACLs (VPC only)

* If your database is not in a VPC, skip ahead to Step 4.
* Return to the instance details page.

1. In the Connectivity and security section, click the to the VPC.

* Click the VPC ID.

1. In the Details section, click the Network ACL .

* Click the Network ACL ID.
* You will see tabs for Inbound Rules and Outbound Rules. You must edit both.
* Edit inbound rules
* Select Inbound Rules.
* If you have a default VPC that was automatically created by AWS, the settings already allow all incoming traffic. To verify that the settings allow incoming traffic, confirm that the Source value is 0.0.0.0/0 and that the ALLOW entry is listed above the DENY entry.
* If your inbound rules don't include an ALL - 0.0.0.0/0 - ALLOW entry, edit the rules to allow the Source to access the port number of your database instance. (The port will be 3306 for direct connections, unless you changed the default.) For additional help, see Amazon's Network ACL documentation.
* If you're connecting directly, enter Fivetran's IPs for your database's region.
* If you're connecting using an SSH tunnel, enter {your-ssh-tunnel-server-ip-address}/32.
* Edit outbound rules
* Select Outbound Rules.
* If you have a default VPC that was automatically created by AWS, the settings already allow all outbound traffic. To verify that the settings allow outbound traffic, confirm that the Destination value is 0.0.0.0/0 and that the ALLOW entry is listed above the DENY entry.
* If your outbound rules don't include an ALL - 0.0.0.0/0 - ALLOW entry, edit the rules to allow outbound traffic to all ports 1024-65535 for the following Destination(s):
* If you're connecting directly, enter Fivetran's IPs for your database's region.
* If you're connecting using an SSH tunnel, enter {your-ssh-tunnel-server-ip-address}/32.

1. Create user

* In your Amazon Aurora MySQL master database, create a database user for Fivetran's exclusive use.
* WARNING: This user must be reserved for Fivetran use only and must be unique to your connector. For more information, see our MySQL documentation.
* Open a connection to your master database in your favorite SQL tool (for example, MySQL Workbench or the "mysql" command in your operating system's terminal window).
* Create a user for Fivetran and grant replication permissions by executing the following SQL command. Replace <username> and password with a username and password of your choice.
* CREATE USER <username>@'%' IDENTIFIED BY 'password';
* GRANT SELECT, REPLICATION CLIENT, REPLICATION SLAVE ON \*.\* TO <username>@'%';
* content\_copy
* NOTE: You must grant the Fivetran user SELECT permissions for all of the columns in the tables that you want to sync. When we do not have SELECT access to all columns in a table, we trigger a re-sync for that table, which slows down your syncs. If you don't want to sync certain columns, you can exclude them from your syncs in the Fivetran dashboard.

1. Configure binary logging

* Configure the master node in your Amazon Aurora cluster to output binary logs. We need binary logs to perform incremental updates.

1. Enable binary logging

* In your RDS dashboard, go to your Aurora cluster.
* On the Configuration tab, note the name of the DB cluster parameter group. If the name begins with default., it is a default parameter group. Otherwise, it is a custom group.
* In the left menu, go to the Parameter groups tab.
* Do one of the following to create a new parameter group:
* (Option 1) Click Create parameter group.
* On the Create parameter group page, ensure that the Parameter Group Family value matches the one in your existing default parameter group. Set the Type to "DB Cluster Parameter Group." Give the new group a name and description, then click Create.
* (Option 2) Select your existing cluster parameter group, click Parameter group actions, then select Copy.
* On the Copy DB parameter group page, give the new group a name and description, then click Copy.

1. Click on the to the new parameter group (whether you created a new group or copied an existing group). Make sure the Type is "DB Cluster Parameter Group." If not, revisit the previous step.

* Click on the name of the new parameter group.
* Select the binlog\_format parameter, then click Edit parameters.
* Change the binlog\_format value to ROW.
* Click Save changes.
* In the left menu, go to the Databases tab.
* Select your Amazon Aurora MySQL master/writer database, then click Modify.
* In the Modify DB Instance screen, scroll down to find the Database Option section. Change the DB cluster parameter group to the new group you created.
* At the bottom of the page, click Continue.
* Select Apply Immediately.
* Click Modify DB Instance.
* To make the changes to take effect, you must reboot the DB instance. Select your database, then click Actions > Reboot.
* Click Confirm to confirm that you want to reboot the instance.
* Wait for the master node to reboot. The instance's Status will change from "rebooting" to "available" when it is done.
* The configuration change will be complete when your database's DB Cluster Parameter Group reflects the new parameter group.

1. Set binary log retention period

* Update your binary log retention period. By default, Amazon Aurora databases cull binary log files as quickly as possible. However, Fivetran cannot perform incremental updates unless your database retains binary logs for at least 24 hours.
* Open a connection to your master database in your favorite SQL tool (for example, MySQL Workbench or the "mysql" command in your operating system's terminal window).
* Use the function below to view your current settings.
* CALL mysql.rds\_show\_configuration;
* content\_copy
* If your binlog retention period is less than 24 hours or if the result of this query is NULL, proceed to the next step to increase your retention period.
* NOTE: If the result of this query is NULL, your database has Aurora's default behavior and your database culls binary log files as quickly as possible. If we cannot sync a log before it's deleted, we will need to re-sync your database in full, which can cause significant downtime.
* Increase your binary log retention period. Your retention period must be at least 24 hours, though we recommend a retention period of 7 days (168 hours). Use the sample queries below to update your binary log retention period.
* IMPORTANT: Increasing the binary log retention period requires additional disk space in your source Amazon Aurora MySQL database, since it increases how many log files are stored at one time.
* Set your binlog retention period to 24 hours:
* CALL mysql.rds\_set\_configuration('binlog retention hours', 24);
* content\_copy
* or set your binary log retention period to 7 days (168 hours).
* CALL mysql.rds\_set\_configuration('binlog retention hours', 168);
* content\_copy

1. Create a daily event to rotate binary logs (optional)

* Binary logs must be rotated at least once per retention period. Binary log rotation keeps the latest row changes from being written into an expired binary log file that is scheduled for deletion. Your Amazon Aurora MySQL database automatically rotates a binary log when it reaches the maximum size; however, the log does not rotate if it does not reach the maximum size because the database has had too little write activity.
* To ensure that your logs rotate frequently enough, you can create an event that automatically rotates binary logs every 24 hours.
* Execute the following SQL statement to turn on the event scheduler.
* NOTE: The event scheduler runs custom events on an automatic schedule.
* SET GLOBAL event\_scheduler = ON;
* content\_copy
* Execute the following SQL statements to create a custom event that rotates (flushes) the binary logs and schedule it to run every 24 hours.
* DROP EVENT IF EXISTS rotate\_binlog\_24\_hrs\_event;
* CREATE EVENT rotate\_binlog\_24\_hrs\_event
* ON SCHEDULE EVERY 24 hour
* DO FLUSH BINARY LOGS;
* content\_copy
* NOTE: If you have a low-activity database, you must have at least two binary log files at all times. Having two binary log files ensures that your binary log data isn't rotated before Fivetran can perform an incremental update. Check your binary log files by looking at the output of SHOW BINARY LOGS in your Amazon Aurora MySQL database.

1. Finish Fivetran configuration

* In your connector setup form, enter a destination schema prefix. This prefix applies to each replicated schema and cannot be changed once your connector is created.
* In the Host field, enter the endpoint you found in Step 2. Alternatively, you can enter your database host's IP (for example, 1.2.3.4).
* Enter your database instance's port number that you found in Step 2. The port will be 3306, unless you changed the default.
* Enter the Fivetran-specific user that you created in Step 4.
* Enter the password for the Fivetran-specific user that you created in Step 4.
* Choose your connection method. If you selected Connect via an SSH tunnel, provide the following information:
* SSH hostname (do not use a load balancer's IP address/hostname)
* SSH port
* SSH user
* If you enabled TLS on your database in Step 1, set the Require TLS through tunnel toggle to ON.
* Enter a unique replica ID for Fivetran. We provide a random replica ID, but you can provide your own if you'd prefer or if the setup form's replica ID conflicts with one of your existing replica IDs.
* Click Save & Test. Fivetran tests and validates our connection to your Amazon Aurora MySQL database. Upon successful completion of the setup tests, you can sync your data using Fivetran.

1. Setup tests

* Fivetran performs the following tests to ensure that we can connect to your Amazon Aurora MySQL database and that it is properly configured:
* The Connecting to SSH Tunnel Test validates the SSH tunnel details you provided in the setup form. It generates a pop-up window where you must verify the SSH fingerprint. It then checks that we can connect to your database using the SSH Tunnel. (We skip this test if you are connecting directly.)
* The Connecting to Host Test 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 are connecting using an SSH tunnel and did not choose to require TLS.)
* The Validating Database User Test validates the database credentials you provided in the setup form.
* The Checking Database Configuration Test verifies that we can find your database's server ID. It then checks your binary log configuration and confirms that we can connect to the binary log.
* The Validating Database Type Test checks that your database type matches the connector type. For example, this test will generate a warning if you try to set up an Amazon Aurora MySQL connector with a MySQL RDS database.
* The Checking Binlog Retention Period Test verifies that your binary log is set to retain at least 1 day's worth of changes.
* The Validating Speed Setup test checks how quickly Fivetran can fetch data from your source database. The test will show a warning if the speed is low, that is less than 5MB/sec.
* NOTE: The tests may take a few minutes to finish running.

