

**MYSQL RDS SETUP GUIDE**

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

**Prerequisites**

Prerequisiteslink  
To connect your MySQL RDS database to Fivetran, you need:  
  
MySQL version 5.6.13 or above  
Your database host's IP (e.g., 1.2.3.4) or domain (your.server.com)  
  
Additionally, if you will be using binary logs as your incremental sync mechanism:  
  
If you upgrade your database version from 5.7 to 8.0, you do not need to re-sync your existing MySQL RDS connectors to continue syncing successfully. However, if your connector doesn't sync before the database's binary logs expire, you will need to re-sync.  
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 MySQL RDS connectors will fail and you may need re-sync them.  
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 incremental sync mechanism

* To keep your data up to date after the initial sync, we use one of the following incremental sync methods:
* Binary log
* Fivetran Teleport Sync PRIVATE PREVIEW
* Each of these methods keeps a record of recent data changes, which allows Fivetran to update only the data that has changed since our last sync.
* To learn the differences between the two mechanisms, see our incremental update documentation.

1. Choose connection method

* First, decide whether to connect Fivetran to your MySQL RDS master database or read replica 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 TLS setup instructions to enable TLS on your database.
* Fivetran connects directly to your MySQL RDS 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 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. Create read replica (optional)

* If you'd like, create a read replica for Fivetran's exclusive use. Using a read replica also allows us to integrate your data without putting unnecessary load on or interrupting the queries running on your primary server. We recommend that you connect a read replica to Fivetran, but it's not required.
* If you want to connect Fivetran to your master database or already have a read replica, skip ahead to Step 3.
* In your Amazon RDS dashboard, select the MySQL primary instance that you want to replicate.
* Click Actions, then select Create Read Replica.
* In the Settings section, enter your chosen instance ID.
* In the DB instance size section, specify the instance class for the read replica. It does not need to be as large as your master instance.
* In the Connectivity section, set the Public access setting to Publicly accessible to ensure that the read replica is accessible from outside your VPC.
* Click Create read replica.
* The replica's status should now be creating.
* It will take a few minutes for the read replica to finish being created. The status will change to available when it is done.

1. Enable database access

* Grant Fivetran's data processing servers access to your database server. How you grant access depends on whether or not your database instance is in a VPC.
* If your instance is in a VPC, you must configure the two mechanisms that control access: VPC security groups and network access control lists (ACLs). If your instance is not in a VPC, you only need to configure security groups.

1. Configure security group

* NOTE: These instructions assume that your database instance is in a VPC. If your database instance is not in a VPC, you can still use these instructions because configuring a non-VPC security group is an almost identical process.
* In your Amazon RDS dashboard, click on the database that you want to connect to Fivetran.
* In the Endpoint & port column, find the database's port number and make a note of it. You will need the port number to configure Fivetran.

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

* In the Security Group panel, click Actions, then select Edit inbound rules from the drop-down menu.
* Click Add Rule. This creates a new Custom TCP Rule at the bottom of the list with a blank space for a Port Range and a Source IP address.
* Fill in the new Custom TCP Rule.
* In the Port Range field, enter your database's port number that you wrote down in Step 2 of this section. The port number will be 3306 for direct connections, unless you changed the default.
* What you enter in the Custom IP field depends on whether you're connecting directly or using an SSH tunnel.
* 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.
* (Optional) Enter a brief description in the Description field.
* Click Save rules.

1. Configure Network ACLs (VPC only)

* If your database instance is not in a VPC, skip ahead to Step 4.
* Return to the RDS Dashboard.
* Click on your MySQL database.

1. In the Networking column, click the to the instance's VPC.

* Select the VPC.

1. In the Summary tab, click the Network ACL .

* On the Network ACLs page, click the Network ACL ID.
* You will see tabs for Inbound Rules and Outbound Rules. You must edit both.
* Edit inbound rules
* Go to the Inbound Rules tab.
* 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 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.) What you enter in the Source field depends on whether you're connecting directly or using an SSH tunnel.
* 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.
* For additional help, see Amazon's Network ACLs documentation.
* Edit outbound rules
* Go to the Outbound Rules tab.
* 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 MySQL primary database, create a database user for Fivetran's exclusive use. You cannot create a user in a read replica because it is read-only. Once you create the user in the primary database, it will automatically be replicated to the replica.
* How you create a user depends on which incremental update mechanism you are using. Follow the instructions below for your incremental update mechanism.
* WARNING: This user must be reserved for Fivetran use only and must be unique to your connector. For more information, see our MySQL documentation.

1. Binary log

* Open a connection to your primary 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 WITH mysql\_native\_password 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. Fivetran Teleport Sync PRIVATE PREVIEW

* Open a connection to your MySQL primary database using your favorite SQL tool (for example, MySQL Workbench or the mysql command in your operating system's terminal window).
* Create a Fivetran user and grant SELECT permissions by running the following SQL commands. Replace <username> and password with a username and password of your choice.
* CREATE USER <username>@'%' IDENTIFIED WITH mysql\_native\_password BY 'password';
* -- Option 1: Grant user SELECT permission on all tables and columns
* GRANT SELECT ON \*.\* TO <username>@'%';
* -- Option 2: Grant user SELECT permission on only specified table and columns
* GRANT SELECT ON <tables/columns> TO <username>@'%';
* content\_copy

1. Configure binary logging (binary log only)

* Update your database's default RDS configuration to enable binary logging. We need binary logs to perform incremental updates.

1. Change binary logging format

* Return to the RDS Dashboard.
* In the left menu, click Parameter groups.
* Click Create parameter group.
* On the Create parameter group page, enter the following information:
* In the Parameter group family drop-down menu, choose the most current major version for your database. For example, if your database version is 5.6.21, your parameter group family value would be mysql5.6.
* In the Group name field, enter a name for the parameter group.
* In the Description field, enter a brief description of the parameter group.
* Click Create. You will be redirected to the Parameter groups page.
* Select the new parameter group.
* In the Parameter group actions menu, click Edit.
* Select the binlog\_format parameter, then click Edit parameters.
* In the Values field, select ROW from the drop-down menu.
* Click Save Changes.
* If you have connected Fivetran to a read replica and plan to run your connector on history mode, check your read replica's slave\_parallel\_workers value.
* If the slave\_parallel\_workers value is 0, you do not need to do any additional configuration. Proceed to the Turn on automated backups section.
* If the slave\_parallel\_workers value is not 0, set the slave\_preserve\_commit\_order value to 1.
* NOTE: Setting the slave\_preserve\_commit\_order value to 1 does not preserve the order of non-transactional DML updates in your binlog. During incremental updates, Fivetran assumes that your binlog events are in sequential order; when they are not, we may encounter replication gaps that can cause data integrity problems in your destination. Learn more about gaps in MySQL’s Replication Inconsistencies documentation.
* Next, set the slave\_parallel\_type value to LOGICAL\_CLOCK.
* Connect to your master database, then set the binlog\_order\_commits value to 1.
* Learn more about these variables in MySQL's replica server variables documentation.

1. Turn on automated backups

* In the RDS dashboard, click Databases in the left menu.
* Select your MySQL database, then click Modify.
* In the Modify DB Instance screen, scroll down to find the Additional configurations section. Change the DB Parameter Group value to the new parameter group you created in the preceding section.
* Scroll down to Backup.
* Change the Backup Retention Period to 1 day.
* Click Continue.
* Select Apply Immediately.
* Click Modify DB Instance.
* The Parameter Group setting in the database details should now show the name of your new parameter group. The Parameter Group status will say "applying" at first. Wait until the status changes to "pending-reboot."
* To make your changes take effect, reboot your instance by clicking Actions > Reboot.
* Click Confirm to reboot the instance.
* Rebooting will take a few minutes. The configuration change is complete when the database's status changes to "available" and the parameter group status changes to "in-sync."

1. Set your binlog retention period (binary log only)

* Fivetran recommends that you set your binlog retention period to seven days (168 hours). To update your binlog retention period, run the following command on the MySQL RDS database that you want to connect to Fivetran:
* CALL mysql.rds\_set\_configuration('binlog retention hours', 168);

1. Grant Fivetran permission to check binlog retention period

* During the connector setup process, Fivetran can check your binlog retention period and alert you if you need to set a longer retention period. If you want Fivetran to check your binlog retention period, you must grant Fivetran permission to access the mysql.rds\_configuration table.
* Open a connection to your primary database in your favorite SQL tool (for example, MySQL Workbench or the "mysql" command in your operating system's terminal window).
* Run the following command to grant Fivetran permission to access the mysql.rds\_configuration table:
* GRANT SELECT ON mysql.rds\_configuration to <username>@'%'

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 your database host's IP (for example, 1.2.3.4) or domain (for example, your-database.cp0rdhwjbsae.us-east-1.rds.amazonaws.com)
* Enter your database instance's port number (usually 3306).
* Enter the Fivetran-specific user that you created in Step 5.
* Enter the password for the Fivetran-specific user that you created in Step 5.
* 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 2, set the Require TLS through tunnel toggle to ON.
* (Binary log only) 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 MySQL RDS 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 MySQL RDS 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.
* (Binary log only) 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 a generic MySQL connector with a MySQL RDS database.
* (Binary log only) 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.

