

**ORACLE SETUP GUIDE**

**The Below requirements are needed to establish a connection oracle.**

**Prerequisites**

Prerequisiteslink  
To connect your Oracle database to Fivetran, you need:  
  
Oracle 11g or above  
Your database host's IP (e.g., 1.2.3.4) or domain (e.g., your.server.com)  
Your database's port (usually 1521 for unencrypted connections and 2484 for encrypted connections using SSL/TLS)  
Your database's system identifier (SID)/service name

**Setup Guide**

1. Choose connection method

* First, decide whether to connect Fivetran to your generic Oracle database directly, using an SSH tunnel, or using AWS PrivateLink.

1. Connect directly (TLS required)

* IMPORTANT: You must have TLS enabled on your database to connect directly to Fivetran. Follow Oracle's TLS setup instructions to enable TLS on your database.
* Fivetran connects directly to your Oracle database. This is the simplest and most secure method.
* To connect directly, configure your firewall and/or other access control systems to allow incoming connections to your Oracle database host and port (usually 1521) from Fivetran's IPs for your database's region. How you do this will vary based on how your Oracle database is hosted (cloud platform, on-premises, etc.).

1. Connect via SSH (TLS optional)

* IMPORTANT: You must connect using an SSH tunnel if your Oracle database is version 12.1 or below.
* 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.
* To connect using SSH, follow our SSH connection instructions. If you want Fivetran to use end-to-end encryption using TLS, follow Oracle'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 user

* Create a database user for Fivetran's exclusive use.
* Connect to your Oracle database as an admin user.
* Execute the following SQL command to create a user for Fivetran and grant it permission to connect to your database. Replace <username> and <password> with a username and password of your choice. Follow the instructions below for your database type:
* NOTE: Usernames in Oracle are case sensitive. For example, fivetran is not the same user as FIVETRAN. Multitenant container database usernames must start with C## because they are common users.
* For standalone databases:
* CREATE USER <username> IDENTIFIED BY <password>;
* GRANT CREATE SESSION TO <username>;
* content\_copy
* For multitenant container databases Beta:
* You must create a common user at the container level. Create the user with the container as the active session and grant the following permissions:
* ALTER SESSION SET CONTAINER=CDB$ROOT;
* CREATE USER <username> IDENTIFIED BY <password> CONTAINER=ALL;
* GRANT CREATE SESSION, ALTER SESSION, SET CONTAINER TO <username> CONTAINER=ALL;
* ALTER USER <username> SET CONTAINER\_DATA=ALL CONTAINER=CURRENT;
* content\_copy

1. Grant read-only access

* Grant the Fivetran user read-only access to the data you want to sync.
* NOTE: Oracle database defaults to using upper case letters, unless the values are surrounded by double quotes.
* Grant the Fivetran user SELECT permission for each schema and table you want to sync. Follow the instructions below for your database type:
* For standalone databases:
* GRANT SELECT ON "<schemaA>"."<tableA>" TO <username>;
* GRANT SELECT ON "<schemaA>"."<tableB>" TO <username>;
* GRANT SELECT ON "<schemaB>"."<tableC>" TO <username>;
* content\_copy
* Alternatively, you can grant access to all tables.
* GRANT SELECT ANY TABLE TO <username>;
* content\_copy
* For multitenant container databases Beta:
* Execute the following command before granting permissions. Replace <PDB> with the name of the pluggable database (PDB) that you want to connect to Fivetran.
* ALTER SESSION SET CONTAINER=<PDB>;
* content\_copy
* GRANT SELECT ON "<schemaA>"."<tableA>" TO <username>;
* GRANT SELECT ON "<schemaA>"."<tableB>" TO <username>;
* GRANT SELECT ON "<schemaB>"."<tableC>" TO <username>;
* content\_copy
* Alternatively, you can grant access to all tables.
* GRANT SELECT ANY TABLE TO <username>;
* content\_copy
* Grant the Fivetran user access to the DBA\_EXTENTS, DBA\_TABLESPACES, DBA\_SEGMENTS, and TRANSACTION system views. We use these views to optimize our initial import queries.
* GRANT SELECT ON DBA\_EXTENTS TO <username>;
* GRANT SELECT ON DBA\_TABLESPACES TO <username>;
* GRANT SELECT ON DBA\_SEGMENTS TO <username>;
* GRANT SELECT ANY TRANSACTION TO <username>;
* content\_copy

1. Configure LogMiner

* To keep your data up to date after the initial sync, we use Oracle's built-in LogMiner technology. LogMiner uses archived redo log files to identify modified tables and determine which rows need to be updated, which allows Fivetran to update only the data that has changed since our last sync.
* To enable LogMiner, do the following:
* If ARCHIVELOG mode is not enabled on your database, enable ARCHIVELOG mode.
* NOTE: Enabling ARCHIVELOG mode requires the Oracle instance to be briefly taken offline. To learn more, see Oracle's archived redo log file documentation.
* SHUTDOWN IMMEDIATE;
* STARTUP MOUNT;
* ALTER DATABASE ARCHIVELOG;
* ALTER DATABASE OPEN;
* content\_copy
* Configure Oracle RMAN to retain backups and archive logs for at least 24 hours. We recommend retaining data for seven days.
* IMPORTANT: To sync your data, Fivetran must have a minimum of 3 hours' worth of log data to analyze. You cannot finish setting up your connector until 3 hours after you configure RMAN to retain archive logs.
* RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
* content\_copy
* (Recommended) Set the DB\_RECOVERY\_FILE\_DEST\_SIZE parameter to a value that matches your available disk space, because expired and obsolete log and backup files can quickly fill your disk. For more information, see Oracle's DB\_RECOVERY\_FILE\_DEST\_SIZE documentation.
* Enable supplemental logging. We require one of the following logging options:
* Database-level minimal supplemental logging with table level identification key logging. We recommend this option because it minimizes the overhead on your source database.
* Database-level identification key logging.
* Follow the instructions below for your chosen logging option.
* Database-level minimal supplemental logging with table-level identification key logging
* i. Enable minimal supplemental logging by executing the following SQL statement. Minimal supplemental logging ensures that LogMiner has sufficient information to process the redo operations associated with DML changes.
* NOTE: Per Oracle's database-level supplemental logging documentation, "minimal supplemental logging does not impose significant overhead on the database that generates the redo log files."
* ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
* content\_copy
* ii. Once minimal supplemental logging has been enabled at the database level, you then need to enable either primary key identification logging or all supplemental logging. You only need to use all supplemental logging for tables without primary keys.
* To enable primary key supplemental logging only for specific tables, run the following SQL statement for each table:
* ALTER TABLE "<schema>"."<table>" ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
* content\_copy
* To enable all supplemental logging only for specific tables, run the following SQL statement for each table:
* ALTER TABLE "<schema>"."<table>" ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS
* content\_copy
* NOTE: With all supplemental logging, if a row is updated, all of the columns associated with that row are placed in the redo log file.
* Database-level identification key logging
* Enable either primary key identification key logging or all supplemental logging at the database level if you want to enable logging for the entire database. You only need to use all supplemental logging for tables without primary keys.
* To enable primary key supplemental logging at the database level, run the following SQL statement:
* ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
* content\_copy
* To enable all supplemental logging at the database level, run the following SQL statement:
* ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
* content\_copy
* NOTE: If you don't configure the logging data correctly, you will receive a warning when Fivetran encounters a primary key change. The warning will give you customized instructions on how to fix the problem.
* Grant the Fivetran user permission to run LogMiner. Follow the instructions below for your database type:
* For standalone databases:
* GRANT SELECT ON SYS.V\_$DATABASE TO <username>;
* GRANT SELECT ON SYS.V\_$PARAMETER TO <username>;
* GRANT SELECT ON SYS.V\_$ARCHIVED\_LOG TO <username>;
* GRANT SELECT ON SYS.V\_$LOGMNR\_CONTENTS TO <username>;
* GRANT EXECUTE ON DBMS\_LOGMNR TO <username>;
* GRANT EXECUTE ON DBMS\_LOGMNR\_D TO <username>;
* GRANT SELECT ANY TRANSACTION TO <username>;
* GRANT EXECUTE\_CATALOG\_ROLE TO <username>;
* content\_copy
* For multitenant container databases Beta:
* Execute the following command before granting access to LogMiner.
* ALTER SESSION SET CONTAINER=CDB$ROOT;
* content\_copy
* GRANT SELECT ON SYS.V\_$DATABASE TO <username>;
* GRANT SELECT ON SYS.V\_$PARAMETER TO <username>;
* GRANT SELECT ON SYS.V\_$ARCHIVED\_LOG TO <username>;
* GRANT SELECT ON SYS.V\_$ARCHIVE\_DEST TO <username>;
* GRANT SELECT ON SYS.V\_$LOGMNR\_CONTENTS TO <username>;
* GRANT EXECUTE ON DBMS\_LOGMNR TO <username>;
* GRANT EXECUTE ON DBMS\_LOGMNR\_D TO <username>;
* GRANT SELECT ANY TRANSACTION TO <username>;
* GRANT EXECUTE\_CATALOG\_ROLE TO <username>;
* content\_copy
* (Oracle version 12 and above) Grant logmining permissions to the Fivetran user.
* GRANT LOGMINING TO <username>;
* content\_copy
* (Optional) Grant permissions to additional system tables.
* NOTE: We recommend granting these permissions to optimize the connector's performance and reliability.
* For standalone databases:
* GRANT SELECT ON DBA\_FREE\_SPACE TO <username>;
* GRANT SELECT ON SYS.V\_$LOG TO <username>;
* GRANT SELECT ON SYS.V\_$TEMPFILE TO <username>;
* GRANT SELECT ON SYS.V\_$DATAFILE TO <username>;
* content\_copy
* For multitenant container databases Beta:
* Execute the following command before granting access.
* ALTER SESSION SET CONTAINER=CDB$ROOT;
* content\_copy
* GRANT SELECT ON DBA\_FREE\_SPACE TO <username>;
* GRANT SELECT ON SYS.V\_$LOG TO <username>;
* GRANT SELECT ON SYS.V\_$TEMPFILE TO <username>;
* GRANT SELECT ON SYS.V\_$DATAFILE TO <username>;
* content\_copy

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 URL (for example, your.server.com).
* Enter your database's port number. The port number is usually 1521 for unencrypted connections and 2484 for encrypted connections using SSL/TLS.
* Enter the Fivetran-specific user that you created in Step 2.
* Enter the password for the Fivetran-specific user that you created in Step 2.
* Enter your database's SID/Service Name.
* (Multitenant container databases only) Enter your database's PDB Name.
* 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.
* Click Save & Test. Fivetran tests and validates our connection to your Oracle 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 generic Oracle 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 then checks that we can connect to your database using the SSH tunnel. (We skip this test if you aren't connecting using an SSH tunnel.)
* 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 Validating Database Version Test checks your database version. The test passes if your database version is 12.2 or above or if your database version is 12.1 or below and you are connecting using an SSH tunnel.
* The Validating Access to PDB Test checks that we have access to your pluggable database. We skip this test if your database is not a multitenant container database.
* The Validating System View Permission Test checks that we have permission to access the DBA\_EXTENTS, DBA\_TABLESPACES, and DBA\_SEGMENTS system views.
* The Validating Archive Log Access Test checks that we can access your archive log.
* The Validating Archive Log Retention Period Test verifies that your archive log is set to retain at least 24 hours' worth of changes.
* The Access to Database-Level Supplemental Logging Test verifies that supplemental logging is enabled on your database. If supplemental logging is not enabled, the test passes but generates a warning message in your dashboard.
* 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 less than 5MB per second.
* NOTE: The tests may take a few minutes to finish running.

