Troubleshoot PostgreSQL data source issues

- Connection errors
  - Failed to connect to PostgreSQL
  - Grafana Cloud can't reach a private database
  - Request timed out
  - Host not found
- Authentication errors
  - No PostgreSQL user name specified
  - Password authentication failed
  - Permission denied
  - No pg_hba.conf entry
- TLS and certificate errors
  - Certificate verification failed
  - SSL not supported
  - Client certificate error
- Database errors
  - Database does not exist
  - Relation does not exist
- Query errors
  - Query truncated by double-dash in string literals
  - Query syntax error
  - Column does not exist
  - No time column found
  - Macro expansion error
- Performance issues
  - Query timeout
  - Too many connections

- Slow query performance
- Provisioning errors
  - Invalid provisioning configuration
- Other common issues
  - Empty query results
  - TimescaleDB functions not available
  - Data appears delayed or missing recent points
- Get additional help

This document provides troubleshooting information for common errors you may encounter when using the PostgreSQL data source in Grafana.

Connection errors

The following errors occur when Grafana cannot establish or maintain a connection to PostgreSQL.

Failed to connect to PostgreSQL

Error message: `failed to connect to ... : connect: connection refused` or `dial tcp: connect: connection refused`

Cause: Grafana cannot establish a network connection to the PostgreSQL server.

Solution:

1. Verify that the Host URL is correct in the data source configuration.
2. Check that PostgreSQL is running and accessible from the Grafana server.
3. Verify the port is correct (the PostgreSQL default port is `5432`).
4. Ensure there are no firewall rules blocking the connection.
5. Check that PostgreSQL is configured to accept connections from the Grafana server in `pg_hba.conf`.
6. For Grafana Cloud, ensure you have configured Private data source connect if your PostgreSQL instance is not publicly accessible.

Grafana Cloud can't reach a private database

Error message: `dial tcp: connect: connection refused`, `i/o timeout`, or `context deadline exceeded` when using Grafana Cloud with a database on a private network.

Cause: Grafana Cloud runs in a hosted environment and can't directly reach databases on `localhost`, `127.0.0.1`, or private IP ranges (`10.x`, `172.16.x`, `192.168.x`). This is the most common issue when migrating from self-managed Grafana to Grafana Cloud.

Solution:

1. Set up Private data source connect (PDC) to create a secure tunnel between Grafana Cloud and your private network.
2. Install the PDC agent on a machine that has network access to your PostgreSQL instance.
3. If you experience intermittent connection drops with the Docker-based PDC agent, try the binary-based agent instead. This has resolved stability issues in some environments.
4. Update the **Host URL** in the data source settings to use the hostname as seen from the PDC agent's network (not `localhost`).

Request timed out

Error message: "context deadline exceeded" or "i/o timeout"

Cause: The connection to PostgreSQL timed out before receiving a response.

Solution:

1. Check the network latency between Grafana and PostgreSQL.
2. Verify that PostgreSQL is not overloaded or experiencing performance issues.
3. Increase the **Max lifetime** setting in the data source configuration under **Connection limits**.
4. Reduce the time range or complexity of your query.
5. Check if any network devices (load balancers, proxies) are timing out the connection.

Host not found

Error message: `failed to connect to ... : hostname resolving error` or `lookup hostname: no such host`

Cause: The hostname specified in the data source configuration can't be resolved.

Solution:

1. Verify the hostname is spelled correctly.
2. Check that DNS resolution is working on the Grafana server.

3. Try using the database's public IP address instead of a hostname. This is a useful diagnostic step and can serve as a workaround if DNS resolution is the issue.
4. Ensure the PostgreSQL server is accessible from the Grafana server's network.
5. For Grafana Cloud, DNS resolution behavior can differ between stack regions and cloud providers. If a hostname resolves from one stack but not another, try the direct IP address and contact Grafana Support.

Authentication errors

The following errors occur when there are issues with authentication credentials or permissions.

No PostgreSQL user name specified

Error message: `FATAL: no PostgreSQL user name specified in startup packet (SQLSTATE 28000)`

Cause: The connection to PostgreSQL was attempted without a username. This typically means the **Username** field in the data source configuration is empty or was cleared.

Solution:

1. Open the data source settings and verify the **Username** field contains a valid PostgreSQL user.
2. Click **Save & test** and confirm the connection succeeds.
3. If the username disappears after saving, check your Grafana version. A bug in Grafana v13.1 on the fast release channel caused the username field to be cleared on save. Upgrading to a patched release or switching to the steady release channel resolves this issue.

Password authentication failed

Error message: `failed to connect to ... : server error: FATAL: password authentication failed for user "username" (SQLSTATE 28P01)`

Cause: The username or password is incorrect.

Solution:

1. Verify that the username and password are correct in the data source configuration.
2. Check that the user exists in PostgreSQL.
3. Verify the password has not expired.
4. If no password is specified, ensure a PostgreSQL password file is configured.

Permission denied

Error message: `ERROR: permission denied for table table_name (SQLSTATE 42501)` or `ERROR: permission denied for schema schema_name (SQLSTATE 42501)`

Cause: The database user does not have permission to access the requested table or schema.

Solution:

1. Verify the user has `SELECT` permissions on the required tables.
2. Grant the necessary permissions:

   ```
   GRANT USAGE ON SCHEMA schema_name TO grafanareader;
   GRANT SELECT ON schema_name.table_name TO grafanareader;
   ```

3. Check that the user has access to the correct database.
4. Verify the search path includes the schema containing your tables.

No pg_hba.conf entry

Error message: `failed to connect to ... : server error: FATAL: no pg_hba.conf entry for host "ip_address", user "username", database "database_name" (SQLSTATE 28000)`

Cause: PostgreSQL is not configured to accept connections from the Grafana server.

Solution:

1. Edit the `pg_hba.conf` file on the PostgreSQL server.
2. Add an entry to allow connections from the Grafana server:

   ```
   host    database_name    username    grafana_ip/32    md5
   ```

3. Reload PostgreSQL configuration: `SELECT pg_reload_conf();`
4. If using SSL, ensure the correct authentication method is specified (for example, `hostssl` instead of `host`).

TLS and certificate errors

The following errors occur when there are issues with TLS configuration.

Certificate verification failed

Error message: "x509: certificate signed by unknown authority" or "certificate verify failed"

Cause: Grafana cannot verify the TLS certificate presented by PostgreSQL.

Solution:

1. Set the **TLS/SSL Mode** to the appropriate level (`require`, `verify-ca`, or `verify-full`).
2. If using a self-signed certificate, add the CA certificate in **TLS/SSL Auth Details**.
3. Verify the certificate chain is complete and valid.
4. Ensure the certificate has not expired.
5. For testing only, set **TLS/SSL Mode** to `disable` (not recommended for production).

SSL not supported

Error message: `failed to connect to ... : server refused TLS connection` or `server does not support SSL`

Cause: The PostgreSQL server is not configured for SSL connections, but the data source requires SSL.

Solution:

1. Set **TLS/SSL Mode** to `disable` if SSL is not required.
2. Alternatively, enable SSL on the PostgreSQL server by configuring `ssl = on` in `postgresql.conf`.
3. Ensure the server has valid SSL certificates configured.

Client certificate error

Error message: "TLS: failed to find any PEM data in certificate input" or "could not load client certificate"

Cause: The client certificate or key is invalid or incorrectly formatted.

Solution:

1. Verify the certificate and key are in PEM format.
2. Ensure the certificate file path is correct and readable by the Grafana process.
3. Check that the certificate and key match (belong to the same key pair).
4. If using certificate content, ensure you've pasted the complete certificate including headers.

Database errors

The following errors occur when there are issues with the database configuration.

Database does not exist

Error message: `failed to connect to ... : server error: FATAL: database "database_name" does not exist (SQLSTATE 3D000)`

Cause: The specified database name is incorrect or the database doesn't exist.

Solution:

1. Verify the database name in the data source configuration.
2. Check that the database exists: `\l` in `psql` or `SELECT datname FROM pg_database;`
3. Ensure the database name is case-sensitive and matches exactly.
4. Verify the user has permission to connect to the database.

Relation does not exist

Error message: `ERROR: relation "table_name" does not exist (SQLSTATE 42P01)`

Cause: The specified table or view does not exist, or the user cannot access it.

Solution:

1. Verify the table name is correct and exists in the database.
2. Check the schema name if the table is not in the public schema.
3. Use fully qualified names: `schema_name.table_name`.
4. Verify the user has `SELECT` permission on the table.
5. Check the search path: `SHOW search_path;`

Query errors

The following errors occur when there are issues with SQL syntax or query execution.

Query truncated by double-dash in string literals

Error message: Unexpected syntax errors or truncated results when string values contain `--` (double dash).

Cause: In older Grafana versions, the SQL comment-stripping parser didn't correctly handle `--` inside single-quoted strings. A query like `WHERE name = 'value--suffix'` would be truncated at the `--`, causing the rest of the query to be silently dropped. This also affected strings with consecutive hyphens (for example, `'10YDE-VE-------2'` would be truncated to `'10YDE-VE`).

This was fixed with a quote-aware comment-stripping parser (PR #121772). The fix also handles PostgreSQL dollar-quoted strings (`$$...$$`). It is available in Grafana 13.1 and later, Grafana 13.0.2 and later, and these 12.x patch releases: 12.1.11, 12.2.9, 12.3.7, and 12.4.3.

Solution:

1. Upgrade to a Grafana release that includes the fix (13.1+, 13.0.2+, or one of the 12.x patch releases listed above).
2. If you can't upgrade immediately, work around the issue by using PostgreSQL string concatenation to avoid literal `--` in your queries:

   ```
   WHERE name = 'value' || '--' || 'suffix'
   ```

3. Alternatively, use a parameterized approach with a template variable for the value.

Query syntax error

Error message: `ERROR: syntax error at or near "keyword" (SQLSTATE 42601)`

Cause: The SQL query contains invalid syntax.

Solution:

1. Check your query syntax for typos or invalid keywords.
2. Verify column and table names are correctly quoted if they contain special characters or are reserved words.
3. Use double quotes for identifiers: `"column_name"`.
4. Test the query directly in a PostgreSQL client such as `psql`.

Column does not exist

Error message: `ERROR: column "column_name" does not exist (SQLSTATE 42703)`

Cause: The specified column name is incorrect or doesn't exist in the table.

Solution:

1. Verify the column name is spelled correctly.
2. Check that column names are case-sensitive in PostgreSQL when quoted.
3. Use the correct quoting for column names: `"Column_Name"` for case-sensitive names.
4. Verify the column exists in the table: `\d table_name` in `psql`.

No time column found

Error message: "no time column found" or time series visualization shows no data

Cause: The query result does not include a properly formatted time column.

Solution:

1. Ensure your query includes a column named `time` that returns a timestamp or epoch value.
2. Use an alias to rename your time column: `SELECT created_at AS time`.
3. Ensure the time column is of type `timestamp`, `timestamptz`, or a numeric epoch value.
4. Order results by the time column: `ORDER BY time ASC`.

Macro expansion error

Error message: "macro '$__timeFilter' not found" or incorrect query results with macros

Cause: Grafana macros aren't being properly expanded.

Solution:

1. Verify the macro syntax is correct, for example `$\_\_timeFilter(time_column)`.
2. Ensure the column name passed to the macro exists in your table.
3. Check that the macro isn't inside a SQL comment (`--` or `/* */`). Grafana strips comments before expanding macros, so macros inside comments are silently ignored.
4. Use the **Preview** toggle in Builder mode to see the expanded query.
5. Open the Query inspector to view the exact SQL sent to PostgreSQL after macro expansion.
6. For time-based macros, ensure the column contains timestamp data.

Performance issues

The following issues relate to slow query execution or resource constraints.

Query timeout

Error message: "canceling statement due to statement timeout" or "query timeout"

Cause: The query took longer than the configured timeout.

Solution:

1. Reduce the time range of your query.
2. Add indexes to columns used in WHERE clauses and joins.
3. Use the `$\_\_timeFilter` macro to limit data to the dashboard time range.
4. Increase the statement timeout in PostgreSQL if you have admin access.
5. Optimize your query to reduce complexity.

Too many connections

Error message: `failed to connect to ... : server error: FATAL: too many connections for role "username" (SQLSTATE 53300)` or `connection pool exhausted`

Cause: The maximum number of connections to PostgreSQL has been reached.

Solution:

1. Reduce the **Max open** connections setting in the data source configuration.
2. Increase `max_connections` in PostgreSQL's `postgresql.conf` if you have admin access.
3. Check for connection leaks in other applications connecting to the same database.
4. Enable **Auto max idle** to automatically manage idle connections.

Slow query performance

Cause: Queries take a long time to execute.

Solution:

1. Reduce the time range of your query.
2. Add appropriate indexes to your tables.
3. Use the `$\_\_timeFilter` macro to limit the data scanned.
4. Increase the **Min time interval** setting to reduce the number of data points.
5. Use `EXPLAIN ANALYZE` in PostgreSQL to identify query bottlenecks.
6. Consider using materialized views for complex aggregations.

Provisioning errors

The following errors occur when provisioning the data source via YAML.

Invalid provisioning configuration

Error message: "metric request error" or data source test fails after provisioning

Cause: The provisioning YAML file contains incorrect configuration.

Solution:

1. Ensure parameter names match the expected format exactly.
2. Verify the database name is **not** included in the URL.
3. Use the correct format for the URL: `hostname:port`.
4. Check that string values are properly quoted in the YAML file.
5. Refer to the provisioning example for the correct format.

Example correct configuration:

```
datasources:
  - name: Postgres
    type: postgres
    url: localhost:5432
    user: grafana
    secureJsonData:
      password: 'Password!'
    jsonData:
      database: grafana
      sslmode: 'disable'
```

Other common issues

The following issues don't produce specific error messages but are commonly encountered.

Empty query results

Cause: The query returns no data.

Solution:

1. Verify the time range includes data in your database.
2. Check that table and column names are correct.
3. Test the query directly in PostgreSQL.
4. Ensure filters are not excluding all data.
5. Verify the `$\_\_timeFilter` macro is using the correct time column.

TimescaleDB functions not available

Cause: TimescaleDB-specific functions like `time_bucket` are not available in the query builder.

Solution:

1. Enable the **TimescaleDB** toggle in the data source configuration under **PostgreSQL Options**.
2. Verify TimescaleDB is installed and enabled in your PostgreSQL database.
3. Check that the `timescaledb` extension is created: `CREATE EXTENSION IF NOT EXISTS timescaledb;`

Data appears delayed or missing recent points

Cause: The visualization doesn't show the most recent data.

Solution:

1. Check the dashboard time range and refresh settings.
2. Verify the **Min time interval** is not set too high.
3. Ensure data has been committed to the database (not in an uncommitted transaction).
4. Check for clock synchronization issues between Grafana and PostgreSQL.

Get additional help

If you continue to experience issues after following this troubleshooting guide:

1. Check the PostgreSQL documentation for database-specific guidance.
2. Review the Grafana community forums for similar issues.
3. Contact Grafana Support if you are a Cloud Pro, Cloud Contracted, or Enterprise user.
4. When reporting issues, include:
   - Grafana version

- PostgreSQL version
- Error messages (redact sensitive information)
- Steps to reproduce
- Relevant configuration such as data source settings, TLS mode, and connection limits (redact passwords and other credentials)

□□□ #2
□ □□□ □□□ 2026-10-31 16:01:42 UTC
□ □□□ □□□ 2026-10-31 16:01:45 UTC