

When WordPress displays “Error establishing a database connection,” the message tells you that WordPress could not connect to its database—but not why. The cause may be an incorrect database password, an unavailable MySQL or MariaDB service, a full disk, or a hosting-side issue. The database error log can help identify the problem, but its location depends on your server and hosting provider.

Start with the fastest reversible checks. Record the exact message and time, check whether other sites or services on the same account are affected, and confirm database credentials before changing server settings. Don’t reinstall WordPress, add troubleshooting plugins, or edit several files at once. If you make a change, record what changed and how to roll back.



## First: Record the Error and Check the Scope

Before changing anything, make a brief incident scratchpad. Note the time, the affected URL, the exact error shown, and any recent changes—such as a password update, migration, hosting change, or WordPress deployment. Timestamps are especially useful because server logs are usually organized by date and time.

- Does the error affect every page, or only the WordPress admin?
- Are other websites in the same hosting account working?
- Did the issue begin after a database password or hosting change?
- Is the site intermittently unavailable, or consistently down?
- Do you see a different error, such as “Too many connections” or “Access denied”?

If multiple sites on the same server are down, or the [fantasticalawyers.com](https://fantasticalawyers.com) error appeared suddenly without a site change, a database service or hosting incident is more likely than a WordPress code problem. Check your hosting provider’s status page and account notices while you investigate.

## Where to Find the MySQL Error Log

The **mysql error log location** is not universal. It depends on the operating system, database configuration, and hosting setup. WordPress administrators on shared or managed hosting often cannot browse the server’s actual

MySQL or MariaDB error log directly. In that case, the hosting dashboard or support team is the right route to the relevant server-side information.

Hosting or server setup Where to look first What to know cPanel hosting Metrics → Errors, or the hosting log viewer This may show web-server errors, not the database service’s private error log. Plesk hosting Websites & Domains → Logs Use the log selector or filters to check the relevant site and time range. Managed WordPress hosting Hosting dashboard, site tools, or provider support Providers may expose PHP and web logs while restricting database logs. Linux VPS or dedicated server Database configuration and system journal Exact paths and service names vary by distribution and installation. Docker or cloud database service Container or cloud-provider logging console Logs may be collected centrally rather than written to a normal file.

On self-managed Linux servers, common places to investigate include `/var/log/mysql/` and `/var/log/mariadb/`. Depending on the distribution and configuration, the error log may instead be written to a file such as `/var/log/mysql/error.log`, recorded in the system journal, or sent elsewhere. These are examples, not guaranteed paths. The database configuration determines where the server writes its error output.

On a server you administer, check the MySQL or MariaDB configuration for the error-log setting, often named `log_error`. Configuration files may be under `/etc/mysql/` or `/etc/my.cnf`, but the precise layout varies. If you are not responsible for server administration, don’t edit these files just to find a log; ask the host where its **hosting log viewer** exposes database-service errors.

## How MySQL and MariaDB Logs Differ from WordPress Logs

MySQL and MariaDB are database servers. Their error logs can report startup failures, permission problems, corrupted data files, exhausted storage, or other service-level events. MariaDB is a common MySQL-compatible database, so you may see **mariadb logs** rather than files labeled “MySQL” even when WordPress simply calls the connection “MySQL.”

WordPress and PHP logs are different. A PHP log might reveal that WordPress could not authenticate or that a PHP process crashed, but it does not necessarily explain why the database service failed. Likewise, an Apache or Nginx error log is not the database log. When reviewing a hosting log viewer, confirm which log type you are looking at and match entries to the time the site failed.

## Check WordPress Database Credentials Safely

WordPress stores its database connection settings in the file `wp-config.php`, usually in the WordPress installation’s root directory. The relevant constants commonly look like this:

```
define( 'DB_NAME', 'database_name' ); define( 'DB_USER', 'database_user' ); define( 'DB_PASSWORD',  
'database_password' ); define( 'DB_HOST', 'localhost' );
```

The values must match the database and user configured by your host. In particular, `DB_HOST` is not always `localhost`. A host may specify a hostname, an IP address, or a hostname with a port. Use the exact value in the hosting dashboard or setup instructions rather than guessing.

1. **Make a backup first.** Download a copy of `wp-config.php` before editing it. Keep it private: it contains credentials that grant access to the database.
2. **Compare values with the hosting account.** Check the database name, database username, password, and host in the provider’s control panel or database settings.
3. **Check for recent changes.** A database password reset, migration, or account move can leave the site using old settings.

4. **Change only a confirmed mismatch.** If a value is wrong, update that value carefully and save the file.
5. **Retest the site and note the result.** If it remains down, restore the backup or revert the specific edit before moving to another hypothesis.

A typo, extra character, or incorrect database host can cause the same generic WordPress connection error as a database outage. Do not paste wp-config.php into a public forum or send its password through an insecure channel. If a credential may have been exposed, rotate it through the hosting panel and update WordPress using the provider's documented process.

## What Changed and How to Roll Back

**What changed:** If you correct a confirmed credential, record the exact constant changed and the time. Don't record the password itself in an incident note.

**How to roll back:** Restore the saved copy of wp-config.php if the edit makes the problem worse. If you changed the database password in the hosting panel, restoring the old file alone may not work; the database account and WordPress configuration must use the same current password.

## Check Hosting and Server Health

If the credentials look correct, check whether the database service is reachable and healthy. On shared and managed hosting, use the provider's dashboard, status notices, and log tools. A dashboard may show database resource limits or service availability; a hosting log viewer may show related PHP or web-server errors. If you cannot see database-service logs, ask support to check them for the incident window.

On a VPS or dedicated server that you administer, check the database service status using the system's normal service-management tools. Service names vary, commonly including mysql or mariadb. Review the service journal and database error log for the matching timestamp. Avoid restarting services as a reflex: a restart can briefly restore service while hiding the underlying cause, and a database that cannot start may need careful diagnosis.

Common server-side clues include:

- **"Access denied":** The database user, password, or permissions may be wrong.
- **"Unknown database":** The database name may be incorrect, or the database may not exist.
- **"Connection refused" or a timeout:** The database service may be stopped, unreachable, or listening on a different host or port.
- **"Too many connections":** The database may have reached a connection limit or be under unusual load.
- **Disk-space or write errors:** The server may be unable to write database files or logs.
- **Database startup or InnoDB errors:** The service may have a deeper storage or recovery issue that needs an administrator.

Do not delete database files, remove logs, or try improvised repair commands in response to these messages. Preserve the evidence and get a verified backup before any database repair or recovery operation.

## A Fast, Reversible Triage Sequence

1. **Record the incident.** Write down the exact WordPress error, URL, first observed time, and recent changes.
2. **Check scope and status.** See whether other sites or services are affected and check the host's service notices.

3. **Open the hosting log viewer.** Review the correct log type around the failure time; note that web and PHP logs are not necessarily database logs.
4. **Verify wp-config.php.** Back it up, then compare all four database settings with the hosting account.
5. **Ask for the database error log if needed.** Request the MySQL or MariaDB entries for the precise time window and the database service's status.
6. **Make one change at a time.** Retest after each confirmed correction and record the outcome.

## A Useful Script for Hosting Support

"My WordPress site at *[domain]* began showing 'Error establishing a database connection' at approximately *[date, time, and time zone]*. I checked the database settings in wp-config.php against the hosting panel, and they *[match / did not match]*. Could you check whether the MySQL or MariaDB service was available at that time, review its error log for this account or database, and confirm whether there were connection limits, disk-space issues, or a service incident? Please let me know which log entries or timestamps you find."

This request gives support a useful time window and asks for specific checks instead of a vague "site is down" report. Never include the database password in the message.

## After the Site Comes Back

Confirm that both the public site and WordPress admin load, then check whether pages that use the database behave normally. Save your incident notes: the original error, relevant timestamps, the log source, what changed, and how you rolled back or verified the fix. If the cause was a hosting outage, a recurring resource limit, or a database crash, ask the provider what corrective action will prevent a repeat.



The quickest route is usually simple: capture the exact error and time, verify WordPress's database settings against the hosting panel, and use the provider's hosting log viewer—or a targeted support request—to reach the right MySQL or MariaDB logs. Make reversible changes only after you have a backup and a clear reason for each one.