Drupal MySQL error fix

I am running a existing drupal project my ubuntu system. I am getting this error:

```
The website encountered an unexpected error. Please try again later.
PDOException: SQLSTATE[HY000] [2006] MySQL server has gone away in Drupal\Component\DependencyInjection\PhpArrayContainer->createService() (line 79 of /var/www/gpweb/web/core/lib/Drupal/Component/DependencyInjection/PhpArrayContainer.php).
Drupal\Component\DependencyInjection\PhpArrayContainer->createService() (Line: 176)
Drupal\Component\DependencyInjection\Container->get() (Line: 212)
Drupal\Component\DependencyInjection\PhpArrayContainer->resolveServicesAndParameters() (Line: 62)
Drupal\Component\DependencyInjection\PhpArrayContainer->createService() (Line: 176)
Drupal\Component\DependencyInjection\Container->get() (Line: 550)
Drupal\Core\DrupalKernel->getCachedContainerDefinition() (Line: 895)
Drupal\Core\DrupalKernel->initializeContainer() (Line: 472)
Drupal\Core\DrupalKernel->boot() (Line: 708)
Drupal\Core\DrupalKernel->handle() (Line: 28)
```

That error usually means Drupal tried to talk to MySQL while (a) the server dropped the connection or (b) MySQL rejected a too-large packet (very common with Drupal when it writes the compiled service container into the `cache_container` table). Do this sequence:

1) Make sure MySQL is up and you can log in

```bash
sudo systemctl status mysql
mysql -h 127.0.0.1 -u <DB_USER> -p <DB_NAME>
```

If that fails, fix credentials in `web/sites/default/settings.php`:

```php
$databases['default']['default'] = [
  'database' => 'YOUR_DB',
  'username' => 'YOUR_USER',
  'password' => 'YOUR_PASS',
  'host' => '127.0.0.1',   // prefer 127.0.0.1 over 'localhost' to force TCP
  'port' => '3306',
  'driver' => 'mysql',
  'prefix' => '',
  'collation' => 'utf8mb4_general_ci',
];
```

Tip: using `localhost` may try Unix sockets that don't match PHP's socket path; `127.0.0.1` avoids that.

2) Increase MySQL packet/timeout limits (fixes the common Drupal "gone away")

Check current values:

```bash
mysql -e "SHOW VARIABLES LIKE 'max_allowed_packet';"
mysql -e "SHOW VARIABLES LIKE 'wait_timeout';"
```

Set higher values (server side):

- Edit `/etc/mysql/mysql.conf.d/mysqld.cnf` (or `/etc/mysql/mariadb.conf.d/50-server.cnf` on MariaDB) and add under `[mysqld]`:

```ini
max_allowed_packet = 256M
wait_timeout       = 28800
interactive_timeout= 28800
```

Then restart MySQL:

```bash
sudo systemctl restart mysql
```

3) Clear Drupal's caches (forces a fresh container build)

From project root:

```bash
# If Drush is available
vendor/bin/drush cr

# If not, use Drupal script
php web/core/scripts/drupal cache:rebuild all
```

Also delete compiled PHP cache dir (it will regenerate):

```bash
rm -rf web/sites/default/files/php
```

4) Verify PHP extensions & limits

Make sure PDO MySQL is enabled and PHP can handle big requests:

```bash
php -m | grep -i pdo_mysql
```

In your PHP `php.ini` (or FPM pool):

```ini
memory_limit = 512M
```

Restart web server/PHP-FPM:

```bash
sudo systemctl restart apache2  # or nginx/php*-fpm
```

5) If you still get it…

- Confirm DB host/port/firewall again.
- If your DB is remote or in Docker, check container name/port mapping.
- If you see it only on long pages/operations, it's likely a timeout → raise `wait_timeout` (DB) and `max_execution_time` (PHP).

- If it happens right away on first load after code changes, it's almost always `max_allowed_packet` being too small.

Try the steps above (especially bumping `max_allowed_packet` + cache rebuild) and let me know what changes—if it persists, paste your `settings.php` DB array (redact creds) and the outputs of:

```nginx
mysql -e "SHOW VARIABLES LIKE 'max_allowed_packet';"
mysql -e "SHOW VARIABLES LIKE 'version','version_comment';"
```

what is the reason for getting this error:
Deprecated function: Creation of dynamic property Drupal\Core\DrupalKernel::$_serviceId is deprecated in Drupal\Core\DependencyInjection\Container->set() (line 20 of /var/www/gpweb/web/core/lib/Drupal/Core/DependencyInjection/Container.php).
Drupal\Core\DependencyInjection\Container->set('kernel', Object) (Line: 1192)
Drupal\Core\DrupalKernel->attachSynthetic(Object) (Line: 925)
Drupal\Core\DrupalKernel->initializeContainer() (Line: 472)
Drupal\Core\DrupalKernel->boot() (Line: 708)
Drupal\Core\DrupalKernel->handle(Object) (Line: 28)
Deprecated function: Creation of dynamic property Composer\Autoload\ClassLoader::$_serviceId is deprecated in Drupal\Core\DependencyInjection\Container->set() (line 20 of /var/www/gpweb/web/core/lib/Drupal/Core/DependencyInjection/Container.php).
Drupal\Core\DependencyInjection\Container->set('class_loader', Object) (Line: 1195)
Drupal\Core\DrupalKernel->attachSynthetic(Object) (Line: 925)
Drupal\Core\DrupalKernel->initializeContainer() (Line: 472)

```
Drupal\Core\DrupalKernel->boot() (Line: 708)
Drupal\Core\DrupalKernel->handle(Object) (Line: 28)
Deprecated function: Creation of dynamic property Drupal\Core\Http\RequestStack::$_serviceId is deprecated in Drupal\Component\DependencyInjection\Container->createService() (line 288 of /var/www/gpweb/web/core/lib/Drupal/Component/DependencyInjection/Container.php).
Drupal\Component\DependencyInjection\Container->createService(Array, 'request_stack') (Line: 176)
Drupal\Component\DependencyInjection\Container->get('request_stack', 2) (Line: 934)
Drupal\Core\DrupalKernel->initializeContainer() (Line: 472)
Drupal\Core\DrupalKernel->boot() (Line: 708)
Drupal\Core\DrupalKernel->handle(Object) (Line: 28)
```

Short answer: you're running Drupal code that isn't fully compatible with PHP 8.2.

PHP 8.2 deprecated "dynamic properties" (adding new properties to an object at runtime). Drupal's container used to tag service objects by setting a property like `$_serviceId` on them. On PHP 8.2 that triggers those "Deprecated function: Creation of dynamic property … is deprecated" messages. You'll also see it from Composer's ClassLoader and RequestStack because they're service objects too.

What to do (pick one):

1. Best fix — upgrade Drupal (and contrib)

- Make sure your project is on a Drupal core that supports PHP 8.2 (Drupal 10.1+; ideally 10.2/10.3) and update contrib modules/themes.
- In your project root:

```bash
composer outdated drupal/* symfony/*
composer require drupal/core-recommended:^10.2 drupal/core-composer-scaffold:^10.2 drupal/core-project-message:^10.2 --update-with-all-dependencies
composer update
vendor/bin/drush cr
```

(Adjust versions to what your project can accept.)

2. Quick workaround — use PHP 8.1
   If upgrading right now is painful, run the site on PHP 8.1 (fully supported by Drupal 9.5/10.0) and plan the upgrade.

```bash
php -v                    # confirm current
sudo apt install php8.1 php8.1-fpm php8.1-mysql
sudo a2dismod php8.2 && sudo a2enmod php8.1
sudo systemctl restart apache2
```

(Or switch FPM pools if using nginx.)

3. Don't do this as a "fix"
   Silencing `E_DEPRECATED` in `php.ini` or `settings.php` hides the warnings but leaves you on an incompatible stack. Use only temporarily if you must.

After upgrading/downgrading, clear caches:

```bash
vendor/bin/drush cr
```

If you tell me your Drupal core version (`vendor/bin/drush status` → "Drupal version") and your PHP version (`php -v`), I can say exactly which path is safest for you.

how to stop from drupal to show error on top of the website, literally before the nav starts it is showing error.

Got it—you're seeing PHP/Drupal warnings splashed at the top of every page. Hide them from visitors and keep them in logs instead:

Quick (UI) way

- Go to **Configuration → Development → Logging and errors**
  `/admin/config/development/logging`
- Set **Error messages to display** = **None**. Save.

Code way (recommended for production)

Add this to `web/sites/default/settings.php` (or your environment-specific `settings.prod.php`):

```php
$config['system.logging']['error_level'] = 'hide';  // 'hide', 'some', or 'all'
```

Then clear caches:

```bash
vendor/bin/drush cr
```

PHP runtime settings (so errors aren't echoed by PHP)

Edit your active `php.ini` (FPM or CLI as needed):

```ini
display_errors = Off
log_errors = On
; Optional: stop only deprecation spam while still logging other errors
error_reporting = E_ALL & ~E_DEPRECATED & ~E_USER_DEPRECATED
```

Restart PHP/web server:

```bash
sudo systemctl restart php*-fpm   # if using FPM
sudo systemctl restart apache2    # or nginx
```

One-liners with Drush (if you prefer)

```bash
vendor/bin/drush sset system.logging error_level hide
vendor/bin/drush cr
```

Notes

- Hiding errors doesn't fix the root cause (e.g., PHP 8.2 deprecations). Plan to update modules/core, but keep them off the page for users.

- Consider enabling **Database Logging** (dblog) or **Syslog** so you can review issues without exposing them to visitors.

If you tell me whether you're on Apache+mod_php or Nginx+PHP-FPM, I'll point you to the exact `php.ini` or pool file to edit.

but I am using php 8.1.33

Ah—that makes sense. You're probably seeing a **version mismatch**: your **CLI** is PHP 8.1.33, but your **web server (Apache/Nginx via FPM)** is likely running **PHP 8.2**, which is what triggers the "Creation of dynamic property … is deprecated" notices. Drupal then prints them at the top.

Do these checks/fixes:

1. Confirm the PHP version used by the web server

- Create `web/phpinfo.php` with:

  ```php
  <?php phpinfo();
  ```

  Visit `http://your-site/phpinfo.php` and check the PHP Version at the top. (Delete this file afterwards.)

2. If Apache with mod_php

```bash
a2query -m | grep php
# If 8.2 is enabled, switch to 8.1:
sudo a2dismod php8.2
sudo a2enmod php8.1
sudo systemctl restart apache2
```

3. If Nginx (or Apache) with PHP-FPM

```bash
# See which FPMs are running
systemctl status php8.2-fpm
systemctl status php8.1-fpm
```

```
# Stop 8.2 and use 8.1
sudo systemctl disable --now php8.2-fpm
sudo systemctl enable --now php8.1-fpm
sudo systemctl restart php8.1-fpm
sudo systemctl restart nginx  # or apache2 if that’s your web server
```

Make sure your vhost points to the **php8.1-fpm.sock** (or port) if you use sockets.

4. Composer (optional but helpful)
   Lock your dependency platform to 8.1 so Composer won’t pull PHP-8.2-only code paths:

```bash
 composer config platform.php 8.1.33
composer update
```

5. Hide messages from visitors (while you fix the stack)

- In Drupal: `/admin/config/development/logging` → **Error messages to display: None**, or:

  ```php
   $config['system.logging']['error_level'] = 'hide';
  ```

  Then `vendor/bin/drush cr`.

If phpinfo shows 8.1.33 **and** you still get those dynamic property deprecations, tell me your **Drupal core version** and **exact PHP SAPI** (from phpinfo: “Server API” row). But 9 times out of 10, it’s the web SAPI running 8.2 while CLI shows 8.1.

you were right, My apache server is running on php 8.3. But when I am trying to enable php8.1 I'm getting this:
ERROR: Module php8.1 does not exist!

Yep—that error means Apache doesn’t have a PHP 8.1 module installed on your box. Easy fix: install PHP 8.1 first, then switch Apache to use it. Here’s a clean, copy-paste path for Ubuntu.

0) Quick check: are you using mod_php or PHP-FPM?

```bash
a2query -m | grep -E 'php|proxy_fcgi|setenvif'
# If you see "php8.3 (enabled)" → you're on mod_php.
# If you see "proxy_fcgi" and maybe "php8.3-fpm" → you're using FPM.
```

1) Add the PHP packages repo (Ondřej's PPA)

(Ubuntu's default repos won't have multiple PHP versions.)

```bash
sudo apt update
sudo apt install -y software-properties-common
sudo add-apt-repository -y ppa:ondrej/php
sudo apt update
```

2A) If you use Apache mod_php (simple path)

Install PHP 8.1 + common extensions:

```bash
sudo apt install -y libapache2-mod-php8.1 php8.1 php8.1-common php8.1-mysql \
  php8.1-xml php8.1-gd php8.1-curl php8.1-mbstring php8.1-zip php8.1-opcache
```

Switch Apache to 8.1:

```bash
sudo a2dismod php8.3
sudo a2enmod php8.1
sudo systemctl restart apache2
```

2B) If you use PHP-FPM (recommended)

Install 8.1-FPM + extensions:

```bash
sudo apt install -y php8.1-fpm php8.1-mysql php8.1-xml php8.1-gd \
  php8.1-curl php8.1-mbstring php8.1-zip php8.1-opcache
```

Switch FPM service and Apache conf:

```bash
# Stop 8.3 FPM, start 8.1 FPM
sudo systemctl disable --now php8.3-fpm
sudo systemctl enable --now php8.1-fpm

# Ensure Apache is set up to talk to FPM
sudo a2enmod proxy_fcgi setenvif
sudo a2disconf php8.3-fpm 2>/dev/null || true
sudo a2enconf php8.1-fpm
sudo systemctl reload apache2
```

(If your vhost points to a socket/port explicitly, update it to `/run/php/php8.1-fpm.sock`.)

3) Make CLI match (optional but nice)

```bash
sudo update-alternatives --set php /usr/bin/php8.1
php -v
```

4) Verify

- Visit a `phpinfo()` page in the webroot (then delete it): it should show **PHP Version 8.1.x**.
- Or: `curl -s http://localhost/phpinfo.php | grep "PHP Version"`.
- In Drupal, clear caches:

```bash
vendor/bin/drush cr
```

Notes & gotchas

- On **Ubuntu 24.04/22.04**, PHP 8.1 isn't in the official repos anymore; the **ondrej/php** PPA above is the standard way to install parallel PHP versions.
- If Apache refuses to enable `php8.1`, it means `libapache2-mod-php8.1` isn't installed (step 2A takes care of that).
- If you just want to stop the warnings showing to users while you sort this out:

  ```bash
  ```

```
vendor/bin/drush sset system.logging error_level hide
vendor/bin/drush cr
```

If you tell me whether you’re on mod_php or FPM (and your Ubuntu version), I’ll tailor the exact commands to your setup.

↳ a2query -m | grep -E 'php|proxy_fcgi|setenvif'

when running this script I get this result:
php8.3 (enabled by site administrator)
setenvif (enabled by maintainer script)

Nice—thanks for running that.
`a2query` shows you’re on **Apache mod_php** (not FPM): `php8.3` is enabled, `proxy_fcgi` is not. So the fix is to **install the PHP 8.1 Apache module** and then switch Apache to it.

Do this:

1. Add the PHP repo (needed to get older PHP versions side-by-side)

```bash
sudo apt update
sudo apt install -y software-properties-common
sudo add-apt-repository -y ppa:ondrej/php
sudo apt update
```

2. Install PHP 8.1 for **Apache (mod_php)** + common extensions Drupal needs

```bash
sudo apt install -y libapache2-mod-php8.1 php8.1 php8.1-common php8.1-mysql \
  php8.1-xml php8.1-gd php8.1-curl php8.1-mbstring php8.1-zip php8.1-opcache php8.1-intl
```

3. Switch Apache from 8.3 → 8.1

```bash
```

```
sudo a2dismod php8.3
sudo a2enmod php8.1
sudo systemctl restart apache2
```

4. Verify the web SAPI is now 8.1
   Open your temporary `phpinfo.php` in the browser (or curl it) and confirm it shows **PHP Version 8.1.x**. Then delete that file. Also:

```bash
vendor/bin/drush cr
```

(Optional) Make CLI PHP match:

```bash
sudo update-alternatives --set php /usr/bin/php8.1
php -v
```

If you'd rather use **PHP-FPM** instead of mod_php (often better), say the word and I'll give you the FPM swap commands.