

# Module 3: Troubleshooting Common Web Server Issues

Web server issues can disrupt website functionality and user experience.

This module explores common web server issues, their causes, and step-by-step troubleshooting strategies.

---

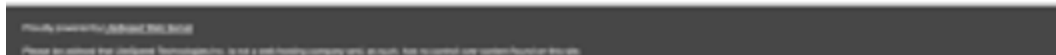
## HTTP Status Codes

Understanding HTTP status codes is fundamental to diagnosing web server and client communication issues.

Below are some critical status codes, their meanings, and how to troubleshoot them:

### 1. HTTP Status Codes Overview

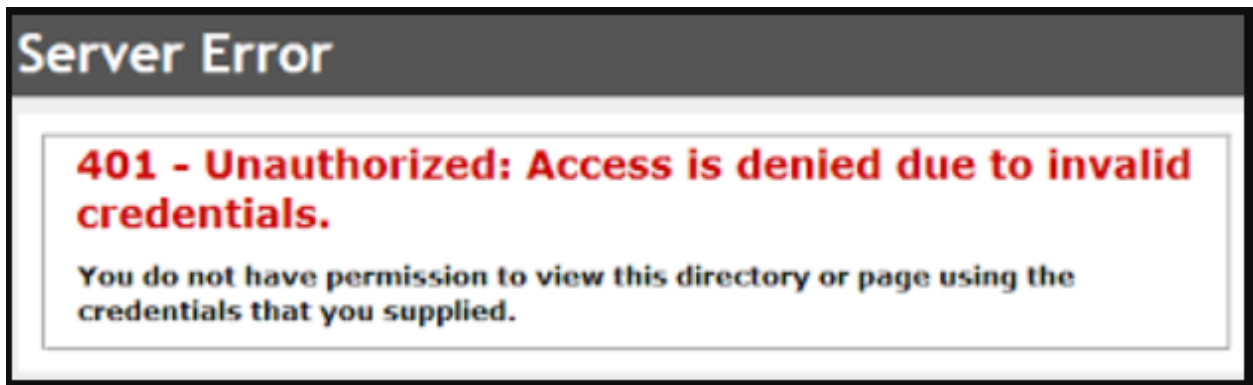
- **404 Not Found:** The server cannot find the requested resource.



- **403 Forbidden:** Access to the resource is restricted.



- **401 Unauthorized:** Authentication is required but not provided or incorrect.



## 401 Authorization Required

nginx



## This page isn't working

If the problem continues, contact the site owner.

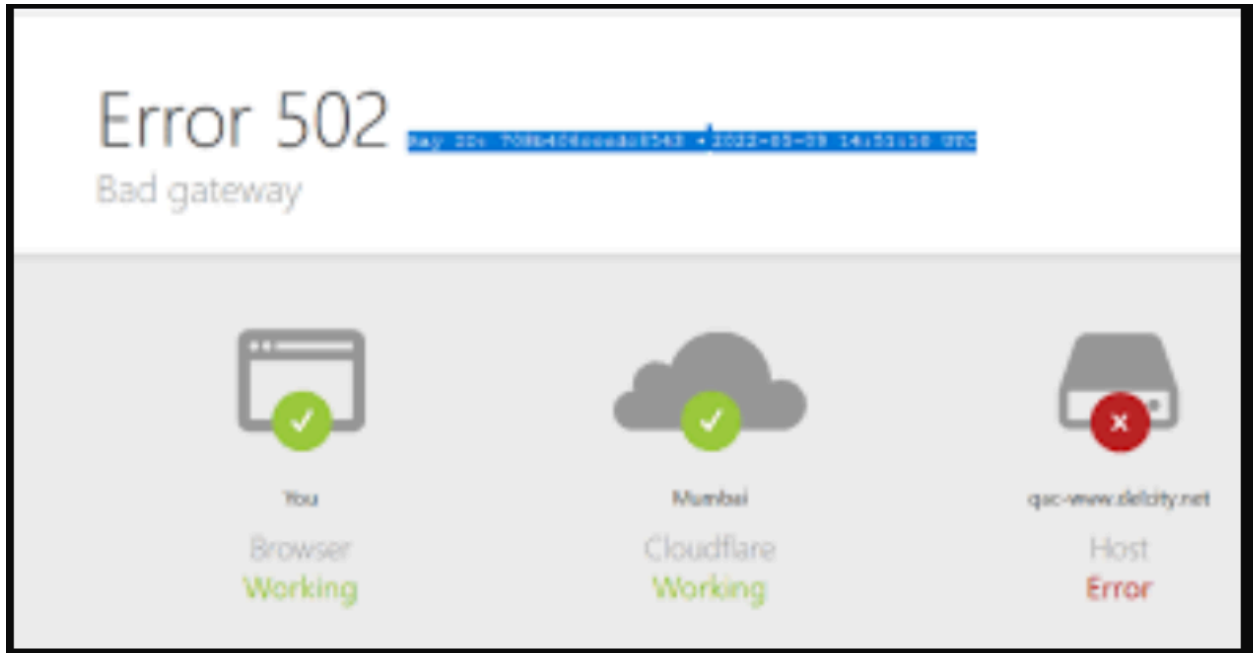
HTTP ERROR 401

- **500 Internal Server Error:** The server encountered an unexpected condition.

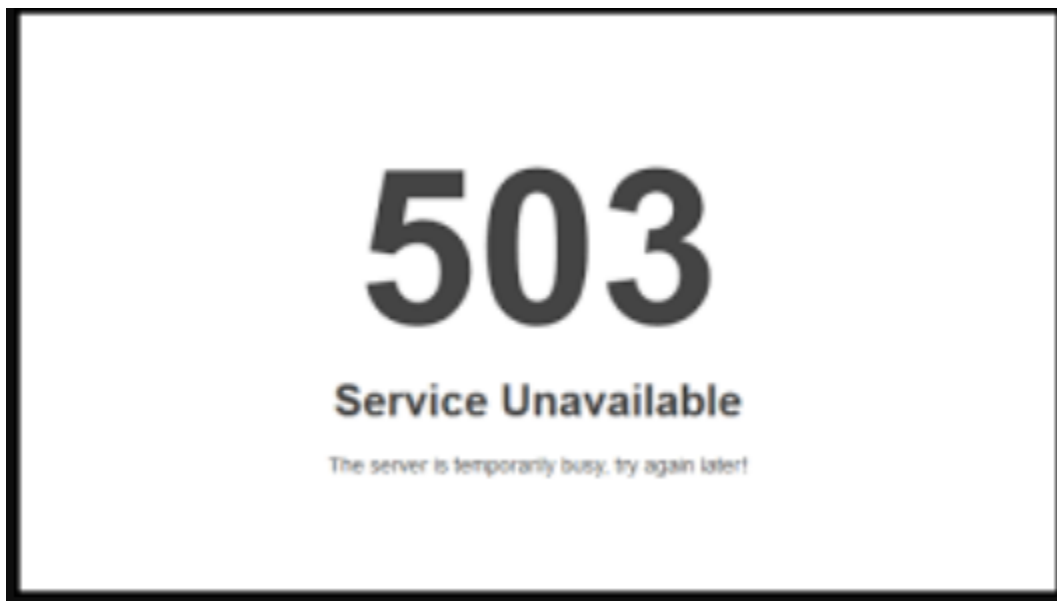


- **502 Bad Gateway:** A server acting as a gateway received an invalid response.





- **503 Service Unavailable:** The server is overloaded or under maintenance.



- **509 Bandwidth Limit Exceeded:** The server has exceeded its bandwidth allocation.



- **525 SSL Handshake Failed:** This indicates that the SSL handshake between Cloudflare and the origin web server failed. This only occurs when the domain is using Cloudflare Full or Full (Strict) SSL mode



---

## 2. Troubleshooting 404 Not Found Errors

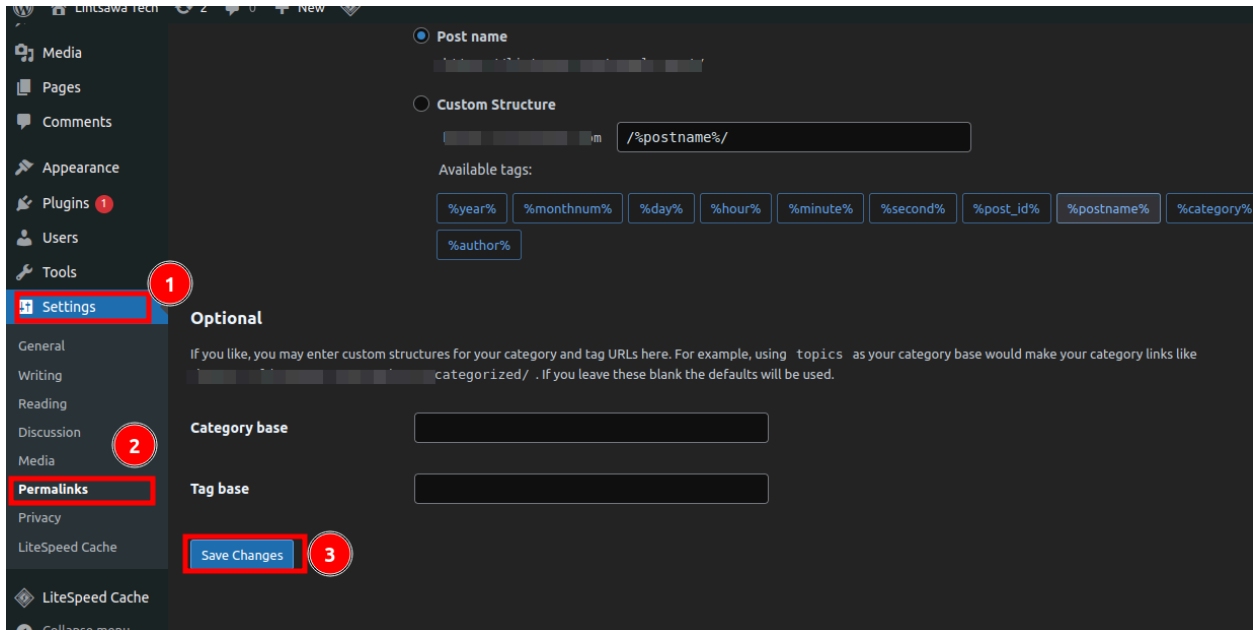
### Common Causes:

- Incorrect URLs.
- Missing files or directories.
- Misconfigured routes or permalinks.

### Platform-Specific Steps:

#### 1. WordPress:

- Check permalink settings.
- Ensure the **.htaccess** file is present and configured correctly.
- Go to Wordpress -> **Admin Dashboard > Settings > Permalinks > Save Changes.**



## 2. Laravel:

- Verify **routes/web.php** contains the correct routes.
- Check the **public** directory for the correct index file.
- Check you have added the rewrite rules in the **.htaccess** file in the root directory of your site. Sample is as below.

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^$ public/index.php [L]
RewriteRule ^((?!public/).*)$ public/$1 [L,NC]
</IfModule>
```

## 3. PHP Custom Sites:

- Confirm file paths and directory structures.

- Check URL rewriting rules in `.htaccess`.
- Sample is as below.

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)    /index.php
</IfModule>
```

#### 4. React Sites:

- Ensure proper configuration for SPA routing (e.g., fallback to `index.html`).
- Verify the build directory is deployed.
- Add the `.htaccess` code as sample below.
- NB: Modify as needed to match your entry file name.

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)    /index.html
</IfModule>
```

#### 5. Flask/Django:

- Confirm route mappings in `app.py` (Flask) or `urls.py` (Django).
- Check for missing static files.
- You can use the command below to locate static files issues.

`Python manage.py findstatic /path/to/static/files`

---

### 3. Troubleshooting 500 Internal Server Errors

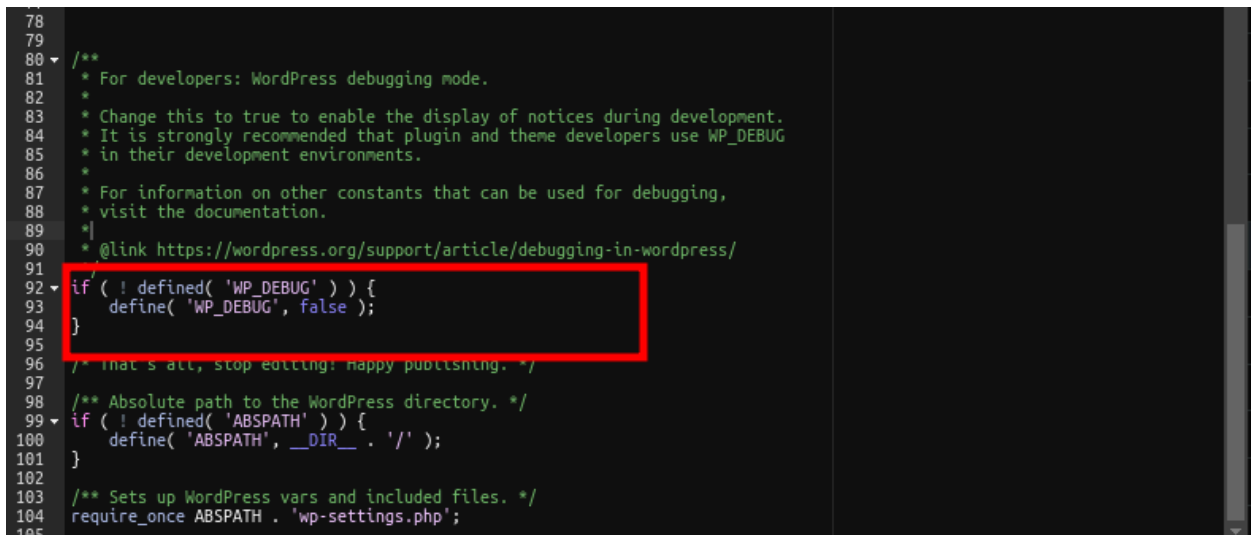
#### Common Causes:

- Syntax errors in scripts or configuration files.
- Permissions issues.
- Missing dependencies or modules.

#### Platform-Specific Steps:

##### 1. WordPress:

- Enable Debug mode to assist in troubleshooting. This will cause your site to display error messages logs on the website when you load the domain on the browser.
- Enable debug on **wp-config.php** file and turn it from **false** to **true**.
- Deactivate plugins and switch to the default theme.

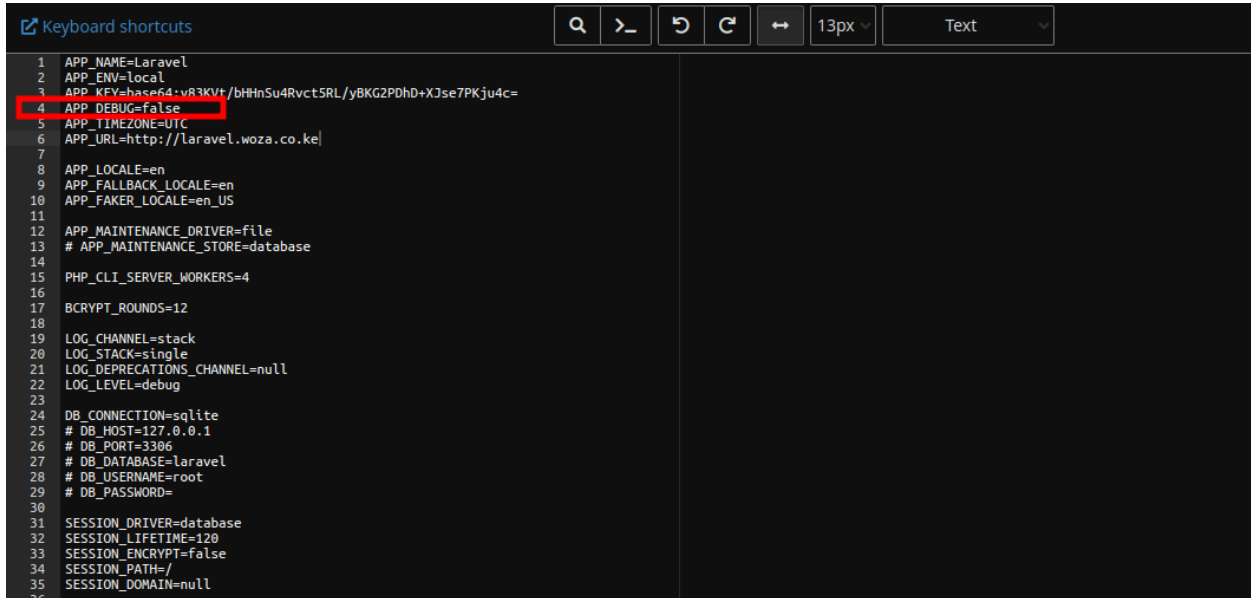


```
78
79
80 /**
81  * For developers: WordPress debugging mode.
82  *
83  * Change this to true to enable the display of notices during development.
84  * It is strongly recommended that plugin and theme developers use WP_DEBUG
85  * in their development environments.
86  *
87  * For information on other constants that can be used for debugging,
88  * visit the documentation.
89  *
90  * @link https://wordpress.org/support/article/debugging-in-wordpress/
91
92 if ( ! defined( 'WP_DEBUG' ) ) {
93     define( 'WP_DEBUG', false );
94 }
95
96 /* That's all, stop editing! Happy publishing. */
97
98 /** Absolute path to the WordPress directory. */
99 if ( ! defined( 'ABSPATH' ) ) {
100     define( 'ABSPATH', __DIR__ . '/' );
101 }
102
103 /** Sets up WordPress vars and included files. */
104 require_once ABSPATH . 'wp-settings.php';
105
```

##### 2. Laravel:

- Clear the cache using **php artisan cache:clear**.
- Check **.env** for database or app configuration errors.
- Enable Debug to assist in troubleshooting.

- For Laravel, you can enable debug in the .env file on in the config/app.php file
- In the .env file adjust **APP\_DEBUG=false** to **APP\_DEBUG=true**



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64-v83Kvt/bHHnSu4Rvct5RL/yBKG2PDhD+XJse7PKju4c=
4 APP_DEBUG=false
5 APP_TIMEZONE=UTC
6 APP_URL=http://laravel.woza.co.ke|
7
8 APP_LOCALE=en
9 APP_FALLBACK_LOCALE=en
10 APP_FAKER_LOCALE=en_US
11
12 APP_MAINTENANCE_DRIVER=file
13 # APP_MAINTENANCE_STORE=database
14
15 PHP_CLI_SERVER_WORKERS=4
16
17 BCRYPT_ROUNDS=12
18
19 LOG_CHANNEL=stack
20 LOG_STACK=single
21 LOG_DEPRECATIONS_CHANNEL=null
22 LOG_LEVEL=debug
23
24 DB_CONNECTION=sqlite
25 # DB_HOST=127.0.0.1
26 # DB_PORT=3306
27 # DB_DATABASE=laravel
28 # DB_USERNAME=root
29 # DB_PASSWORD=
30
31 SESSION_DRIVER=database
32 SESSION_LIFETIME=120
33 SESSION_ENCRYPT=false
34 SESSION_PATH=/
35 SESSION_DOMAIN=null
36
```

- In the config/app.php file, adjust 'debug' => (bool) env('APP\_DEBUG', false), to 'debug' => (bool) env('APP\_DEBUG', true),

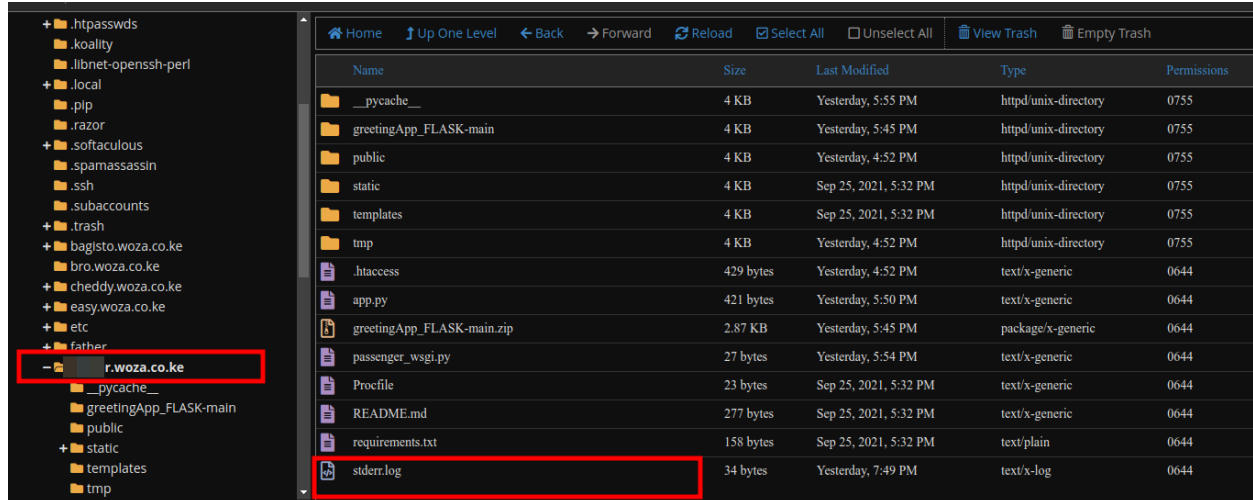
```
10 | This value is the name of your application, which will be used when the
11 | framework needs to place the application's name in a notification or
12 | other UI elements where an application name needs to be displayed.
13 |
14 | */
15 |
16 | 'name' => env('APP_NAME', 'Laravel'),
17 |
18 | /*
19 | -----
20 | Application Environment
21 | -----
22 |
23 | This value determines the "environment" your application is currently
24 | running in. This may determine how you prefer to configure various
25 | services the application utilizes. Set this in your ".env" file.
26 |
27 | */
28 |
29 | 'env' => env('APP_ENV', 'production'),
30 |
31 | /*
32 | -----
33 | Application Debug Mode
34 | -----
35 |
36 | When your application is in debug mode, detailed error messages with
37 | stack traces will be shown on every error that occurs within your
38 | application. If disabled, a simple generic error page is shown.
39 |
40 | */
41 |
42 | 'debug' => (bool) env('APP_DEBUG', false),
43 |
44 | /*
45 | -----
46 | Application URL
47 | -----
48 |
49 | This URL is used by the console to properly generate URLs when using
```

### 3. Custom PHP Sites:

- Verify proper file permissions (e.g., 755 for directories, 644 for files).
- Check for missing **php.ini** settings.
- Enable Debug on cPanel using the Multi PHP INI Editor.
- 

### 4. Flask, Django, Nodejs:

- Review deployment logs.
- Validate environment variables and dependencies.
- On cPanel, the logs are located in **App Root Directory**
- **The log file is called stderr.log**



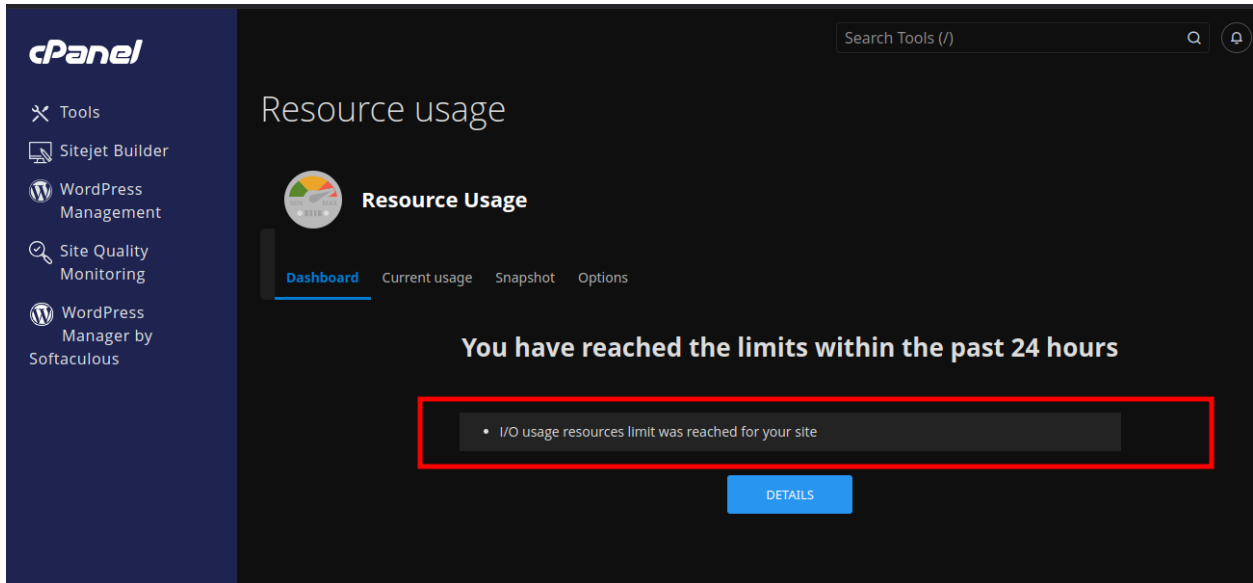
## 4. Troubleshooting 503 Service Unavailable Errors

### Common Causes:

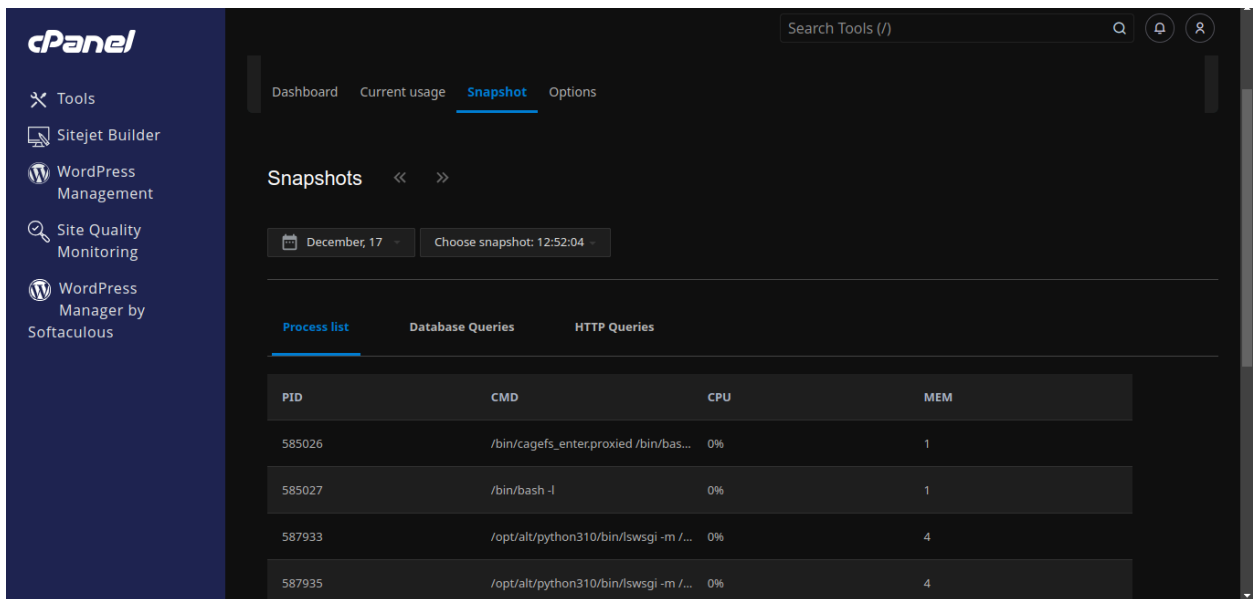
- Server overload or downtime.
- Resource overuse or limits reached.
- Long running scripts or intensive plugins/themes or poorly configured code.
- Wrong application versions or missing modules or dependencies especially for Nodejs, Django and Flask applications.

### Steps:

- Check server load using tools like **top** or **htop**.
- Check resource usage on the cPanel Resource Usage tab.



- Check the snapshot to see a detailed breakdown of resource usage.



- For Nodejs, Django, Flask, check the stderr.log file for more details.

## 5. Troubleshooting 403 Access to Resource is Forbidden/Denied Errors

### Common Causes:

- Server Security settings such as Mod Security.

- Incorrect File permissions. Files should have file permissions of **0644** and Folders should have permissions of **0755** unless stated otherwise.
- Incorrect runtime version, i.e PHP version, Python version, Nodejs Version.
- Data Not yet uploaded on said path (especially on Plesk)
- Security Plugins for Wordpress.
- Security Rules in the **.htaccess** file (Here you may notice **deny** rules added)

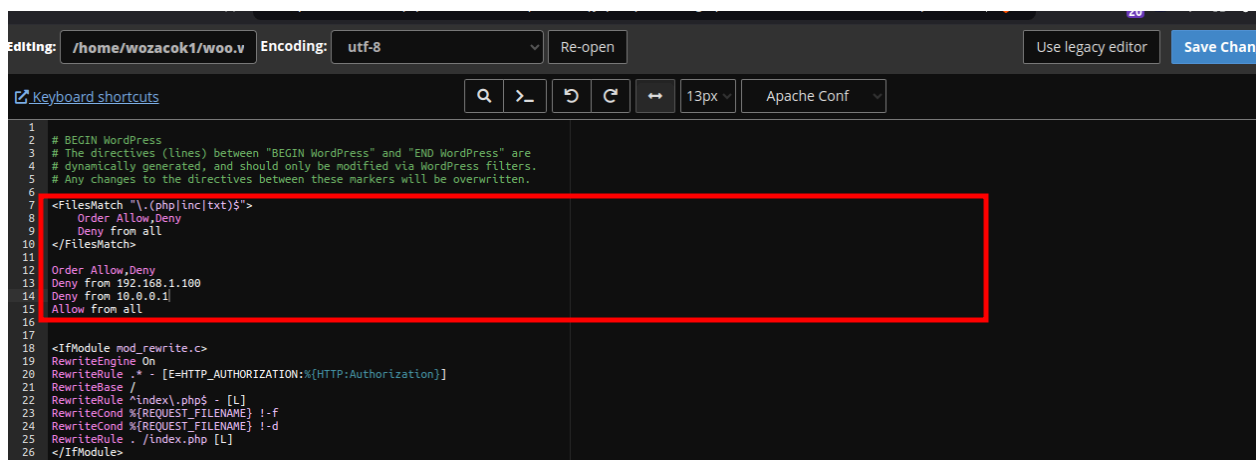
**Example** snippet below denies access to all files having the **.php, inc, txt** extension.

**<FilesMatch "\.(php|inc|txt)\$">**

**Order Allow,Deny**

**Deny from all**

**</FilesMatch>**



```
1 # BEGIN WordPress
2 # The directives (lines) between "BEGIN WordPress" and "END WordPress" are
3 # dynamically generated, and should only be modified via WordPress filters.
4 # Any changes to the directives between these markers will be overwritten.
5
6
7 <FilesMatch "\.(php|inc|txt)$">
8     Order Allow,Deny
9     Deny from all
10 </FilesMatch>
11
12 Order Allow,Deny
13 Deny from 192.168.1.100
14 Deny from 10.0.0.1
15 Allow from all
16
17
18 <IfModule mod_rewrite.c>
19 RewriteEngine On
20 RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
21 RewriteBase /
22 RewriteRule ^index\.php$ - [L]
23 RewriteCond %{REQUEST_FILENAME} !-f
24 RewriteCond %{REQUEST_FILENAME} !-d
25 RewriteRule . /index.php [L]
26 </IfModule>
27
```

More info:

<https://truehost.com/support/knowledge-base/how-to-resolve-error-403-on-cpanel/>

---

## Web Server Errors

Web server-specific errors often require direct log inspection and configuration adjustments.

### 1. Apache Error Logs in CWP

- Access logs from the cPanel dashboard.
- Path: **`/usr/local/apache/logs/error_log`**

```
[root@srv2 ~]# cd /usr/local/apache/logs/
[root@srv2 logs]# ls
access log error log httpd.pid phpmail.log
[root@srv2 logs]# cat /usr/local/apache/logs/error_log
[Wed Dec 18 00:37:35.961433 2024] [mpm_event:notice] [pid 22325:tid 22325] AH00489: Apache/2.4.62 (Unix) configured -- resuming normal
operations
[Wed Dec 18 00:37:35.961617 2024] [core:notice] [pid 22325:tid 22325] AH00094: Command line: '/usr/local/apache/bin/httpd'
[Wed Dec 18 00:37:38.418832 2024] [mpm_event:notice] [pid 22325:tid 22325] AH00492: caught SIGWINCH, shutting down gracefully
[Wed Dec 18 00:37:38.521517 2024] [mpm_event:notice] [pid 22690:tid 22690] AH00489: Apache/2.4.62 (Unix) configured -- resuming normal
operations
[Wed Dec 18 00:37:38.521657 2024] [core:notice] [pid 22690:tid 22690] AH00094: Command line: '/usr/local/apache/bin/httpd'
[Wed Dec 18 00:37:48.503096 2024] [mpm_event:notice] [pid 22690:tid 22690] AH00492: caught SIGWINCH, shutting down gracefully
```

### 2. LiteSpeed Error Logs in cPanel and Cyberpanel

- Path: **`/usr/local/lsws/logs/error.log`**
- Check for issues with LiteSpeed-specific modules or configurations.

```
[root@srv ~]# cat /usr/local/lsws/logs/error.log
2024-10-24 06:47:36.279842 [INFO] [config:server:basics] httpdWorkers: 1, Num of Processors: 1
2024-10-24 06:47:36.279889 [INFO] [config:server:basics] enableLVE: 0
2024-10-24 06:47:36.279895 [INFO] [config:server:basics] bubbleWrap: 0, cmd: '(null)'
2024-10-24 06:47:36.279912 [NOTICE] [testAndFixDirs] "/usr/local/lsws//tmp/" exist and changed mod to 0755.
2024-10-24 06:47:36.279927 [NOTICE] [testAndFixDirs] "/usr/local/lsws//tmp/ocspcache/" exist and change own to 65534:65534
.
2024-10-24 06:47:36.279960 [NOTICE] [testAndFixDirs] "/usr/local/lsws//conf/" own or permission need to be changed.
2024-10-24 06:47:36.307299 [INFO] [PlainConf] [httpServerConfig:] start parsing file /usr/local/lsws/conf/httpd_config.con
```

### 3. Common Web Server Configuration Issues

- Missing or misconfigured **.htaccess** files.
  - Incorrect **DocumentRoot** settings in the web server configuration.
  - Outdated server software.
- 

## Other Common Issues

### 1. WordPress Troubleshooting

- **Critical Errors:** Review the debug log or enable debug.
- **Error Establishing Database Connection:** Verify database credentials in **wp-config.php**.
- **Cross-check the database server is running.**
- **White Screen of Death:** Disable plugins and themes; enable debugging in **wp-config.php**.
- **Do a clean up of your site as per guide below.**

<https://truehost.com/support/knowledge-base/how-to-clean-up-a-wordpress-website/>

### 2. Database-Related Issues

- Ensure the database service is running.
- Repair tables using **phpMyAdmin** or the command line.
- Optimize database queries to reduce load.

### 3. Slow-Loading Websites

- Analyze performance using tools like Google PageSpeed Insights or Gtmetrix
- Optimize images and use caching mechanisms. Example, use litespeed cache for wordpress.
- Check your website traffic against allocated resources/ resource limits.

### 4. Site Can't Be Reached Errors

- Confirm the domain is registered and the Whois status must be active or ok.
- Verify DNS settings and domain propagation.
- Check server firewalls and ensure ports 80/443 are open.
- Test connectivity using **ping** or **traceroute**.

## 5. WooCommerce Errors

- Check WooCommerce logs in the Status section in WordPress admin panel.
- Go to **WooCommerce > Status > Logs**

The screenshot displays the WordPress admin interface. On the left sidebar, the 'Status' menu item is highlighted with a red circle containing the number '1'. In the main content area, the 'Status' page is open, and the 'Logs' tab is selected and highlighted with a red circle containing the number '2'. Below the tabs, there is a search bar and a table of log files. The table has columns for 'Source', 'Date created', 'Date modified', and 'File size'. Two log entries are visible:

<input type="checkbox"/>	Source ↕	Date created ↕	Date modified ↕	File size ↕
<input type="checkbox"/>	plugin-woocommerce	2024-12-19	2024-12-19 06:19:58	3 KB
<input type="checkbox"/>	wc-image-regeneration	2024-12-19	2024-12-19 06:16:32	218 B

The second row, corresponding to 'wc-image-regeneration', is highlighted with a red box and a red circle containing the number '3'. Below the table, there are bulk action controls and a '2 items' indicator.

The screenshot shows the WordPress Status page. The left sidebar has a red circle '1' next to the 'Status' menu item. The main content area has a red circle '2' above the 'Logs' tab. Below the tabs, the text 'Viewing log file plugin-woocommerce-2024-12-19' is displayed. A table of log entries follows, with a red circle '3' next to the first entry. The table contains the following data:

Line	Timestamp	Severity	Message
1	2024-12-19T06:17:57+00:00	Warning	Empty patterns received from the PTK Pattern Store
2	2024-12-19T06:17:58+00:00	Warning	Empty patterns received from the PTK Pattern Store
3	2024-12-19T06:18:02+00:00	Warning	Empty patterns received from the PTK Pattern Store
4	2024-12-19T06:18:03+00:00	Warning	Empty patterns received from the PTK Pattern Store
5	2024-12-19T06:18:06+00:00	Warning	Empty patterns received from the PTK Pattern Store
6	2024-12-19T06:18:46+00:00	Warning	Empty patterns received from the PTK Pattern Store
7	2024-12-19T06:18:48+00:00	Warning	Empty patterns received from the PTK Pattern Store
8	2024-12-19T06:18:58+00:00	Warning	Empty patterns received from the PTK Pattern Store
9	2024-12-19T06:18:59+00:00	Warning	Empty patterns received from the PTK Pattern Store

- Ensure compatibility of plugins with WooCommerce.
  - Verify payment gateway configurations.
-