

How to Boost WordPress Performance with Vultr Managed Databases for Valkey

Learn how to enhance your WordPress site's speed and reliability by integrating Vultr Managed Databases for Valkey, optimizing performance and scalability.

Contents

01	Introduction	3
02	Prerequisites	3
03	Install the WordPress Redis® Plugin	3
04	Configure the Redis® Object Cache to work with Vultr Managed Databases for Valkey	4
05	Troubleshooting	8
06	Conclusion	9

Introduction

Valkey is an open-source NoSQL in-memory database that can work as an object caching solution for Content Management Systems (CMS) such as WordPress. Object caching stores database query results and delivers them faster the next time a user makes similar requests to the site. This caching boosts WordPress performance as the MySQL database response time commonly dictates how fast content gets served to the user.

Follow the steps in this article to boost your WordPress site performance with Vultr Managed Databases for Valkey.

Prerequisites

Before you begin:

- Deploy a Vultr Managed for Valkey Cluster.
- Make a [full Backup of your WordPress site and database](#).

Install the WordPress Redis® Plugin

Valkey Object Cache is a persistent object cache plugin that stores data directly in an existing Valkey database. To use Valkey on WordPress, install and activate the plugin as described in the following steps.

1. Using a web browser, log in to your WordPress backend dashboard with your administrative user account.

```
https://example.com/wp-admin
```

2. On the left navigation bar, find and click **Plugins**.
3. On the Plugins page, click **Add New**.
4. Enter the keyword `Redis® Object Cache` in the **Search Plugins** bar.

5. In the search results, find Redis® Object Cache, and click **Install Now**.
6. When the installation is complete, click **Activate** to make the plugin ready.

Configure the Redis® Object Cache to work with Vultr Managed Databases for Valkey

To integrate the WordPress Redis® Object Cache plugin with your Vultr Managed Database for Valkey, you need to edit your site's `wp-config.php` file and add new configurations as described in this section.

1. Use SSH to access your WordPress server as a [non-root user with sudo privileges](#). Replace `192.0.2.1` with your actual server's IP address.

```
$ ssh user@192.0.2.1
```

2. Install the PHP Redis® extension.

For Ubuntu or Debian:

```
$ sudo apt install php-redis
```

For CentOS or Rocky Linux:

```
$ dnf install php-redis
```

3. Switch to your WordPress web root directory. Usually, `/var/www/html` or `/var/www/public_html`.

```
$ cd /var/www/html
```

4. List files to verify that all WordPress files are available within the directory.

```
$ ls -l
```

5. Back up the `wp-config.php` file.

```
$ sudo cp wp-config.php wp-config.0RIG
```

6. Using a text editor of your choice, open and edit the `wp-config.php` file.

```
$ sudo nano wp-config.php
```

7. Add the following configurations at the top of the file just below `<? php`.

```
define('WP_VALKEY_SCHEME', tls );
define( 'WP_VALKEY_HOST', 'vultr-prod-a25-ab05-e71-vultr-
prod-8c01.vultrdb.com' );
define( 'WP_VALKEY_PORT', 16752 );
define( 'WP_VALKEY_PASSWORD', 'AVNS_A*****' );
define( 'WP_VALKEY_DATABASE', 0 );
define( 'WP_VALKEY_TIMEOUT', 1 );
define( 'WP_VALKEY_READ_TIMEOUT', 1 );
define( 'WP_VALKEY_RETRY_INTERVAL', 3 );
```

Edit all values to match your actual Vultr Managed Database for Valkey Cluster details. Below is what each of the configuration lines does:

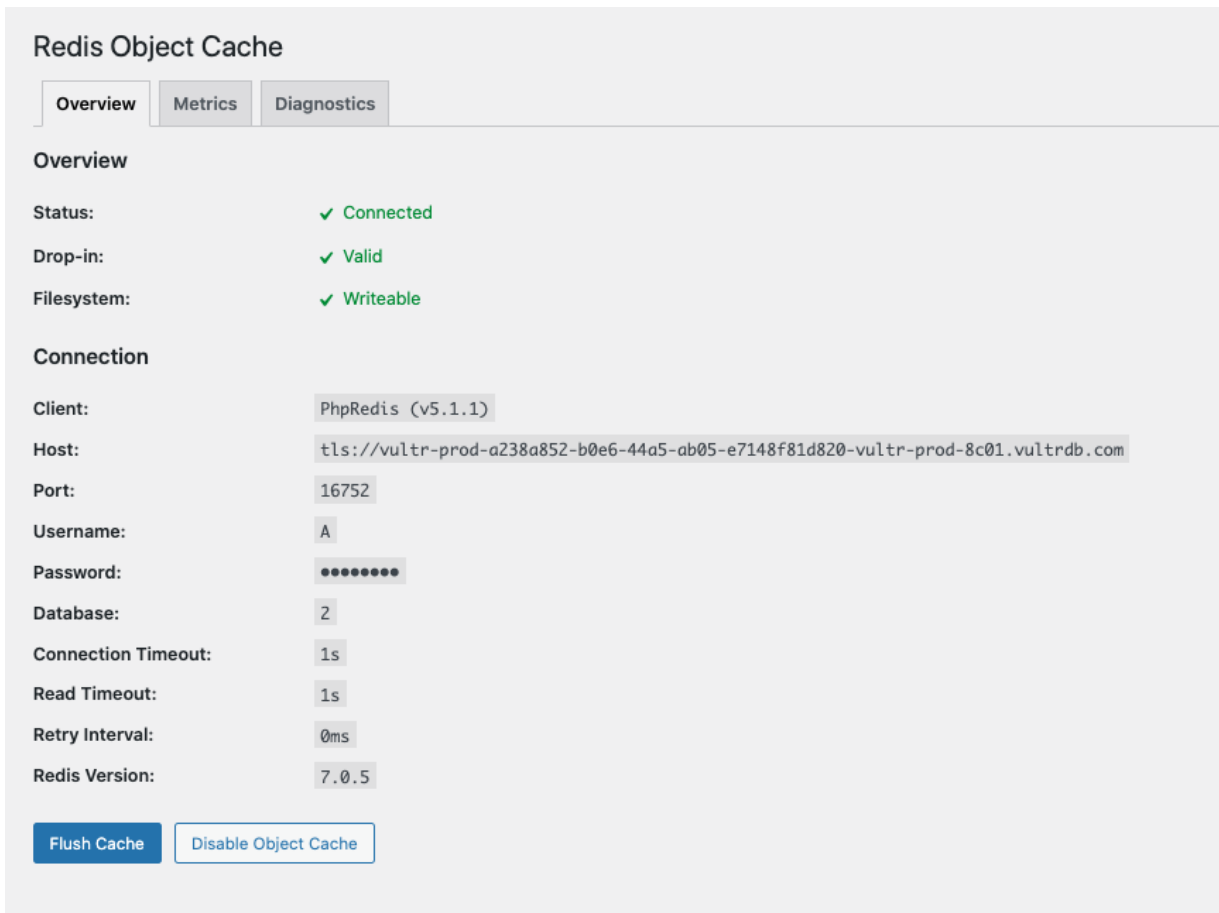
- **WP_VALKEY_SCHEME:** Specifies the connection protocol for communications with the Valkey Database.
- **WP_VALKEY_HOST:** The Vultr Managed Database for Valkey Host URL.
- **WP_VALKEY_PORT:** Specifies the Valkey cluster port.
- **WP_VALKEY_PASSWORD:** Valkey password to authenticate with the cluster.
- **WP_VALKEY_DATABASE:** Specifies the Valkey database to associate with your WordPress. Database values range between 0 to 15.
- **WP_VALKEY_TIMEOUT:** The amount of time in seconds a connection to the Valkey database should wait before timing out (failing).
- **WP_VALKEY_READ_TIMEOUT:** Amount of time in seconds the plugin should read data from the Valkey database before timing out.
- **WP_VALKEY_RETRY_INTERVAL:** The time (in milliseconds) the plugin should retry a failed database connection.

Save and close the file.

8. In a web browser, open your WordPress dashboard.
9. Navigate to **Settings**, and click **Valkey** from the list of options.
10. Click **Diagnostics** and verify that your `wp-config.php` changes display in the log output as below.

```
Status: Drop-in not installed
PhpRedis: 5.1.1
Metrics recorded: 0
Filesystem: Working
WP_VALKEY_SCHEME: "tcp"
WP_VALKEY_HOST: "vultr-prod-a205-e71-vultr-prod-8c01.vultrdb.com"
WP_VALKEY_PORT: 16752
WP_VALKEY_DATABASE: 0
WP_VALKEY_TIMEOUT: 1
WP_VALKEY_READ_TIMEOUT: 1
WP_VALKEY_PASSWORD: .....
Drop-ins: []
```

11. Navigate to the **Overview** tab, and click **Enable Object Cache** to establish a connection to your Vultr Managed Database for Valkey.
12. When successful, the plugin status should change to **Connected**, and the connection information is displayed in the overview tab.



The screenshot displays the 'Redis Object Cache' configuration page. It features three tabs: 'Overview', 'Metrics', and 'Diagnostics'. The 'Overview' tab is active, showing the following status: Status: ✓ Connected, Drop-in: ✓ Valid, and Filesystem: ✓ Writeable. Below this, the 'Connection' section lists various settings: Client: PhpRedis (v5.1.1), Host: tls://vultr-prod-a238a852-b0e6-44a5-ab05-e7148f81d820-vultr-prod-8c01.vultrdb.com, Port: 16752, Username: A, Password: [REDACTED], Database: 2, Connection Timeout: 1s, Read Timeout: 1s, Retry Interval: 0ms, and Redis Version: 7.0.5. At the bottom, there are two buttons: 'Flush Cache' and 'Disable Object Cache'.

If the plugin status is **Not Connected**, navigate to **Diagnostics** for full details on why the connection fails.

13. To verify that your WordPress requests get cached to the Valkey database. Log in to your Valkey cluster.

```
$ redis-cli -u rediss://default:[YOUR_PASSWORD]@[ -vultrdb.com]:16752
```

14. Select the WordPress Valkey Database displayed in the plugin connection information. For this article, database `0`.

```
> select 0
```

15. View all keys.

```
> keys *
```

All WordPress request results get cached to the Valkey database and flushed each time you make changes to the website.

Troubleshooting

Depending on your configuration, the Redis® Object Cache plugin may return errors on the Diagnostics page. Below is how you can fix them to connect successfully.

If you encounter this error:

```
Ping:
Connection Exception: read error on connection to vultr-prod-EXAMPLE.vultrdb.com:16752
(RedisException)
Errors: [
    "read error on connection to vultr-prod-EXAMPLE.vultrdb.com:16752"
```

- Verify that you are using the correct connection scheme. By default, Vultr Managed Database for Valkey requires **TLS/SSL** connections.
- Verify that you are using the `tls` connection scheme in your `wp-config.php` file.

If you encounter this error:

```
Ping:
Connection Exception: read error on connection to vultr-prod-a238a852-b0e6-44a5-ab05-
e7148f81d820-vultr-prod-8c01.vultrdb.com:16752 (RedisException)
Errors: [
    "Connection refused"
]
```

- Verify that your Valkey cluster details are correct and all `wp-config.php` configurations are above any other file configurations just below the `<? php` declaration.

If you encounter this error:

```
Ping:  
Connection Exception: Redis server went away (RedisException)  
Errors: [  
Redis server went away"  
]
```

- Check that all Valkey configurations are at the top of the `wp-config.php` file.

Conclusion

You have successfully integrated your WordPress site to a Vultr Managed Database for Valkey cluster. Your website performance will improve as users make requests and MySQL database results get cached to your Valkey database.



VULTR

