

# Redis (Remote Dictionary Server)

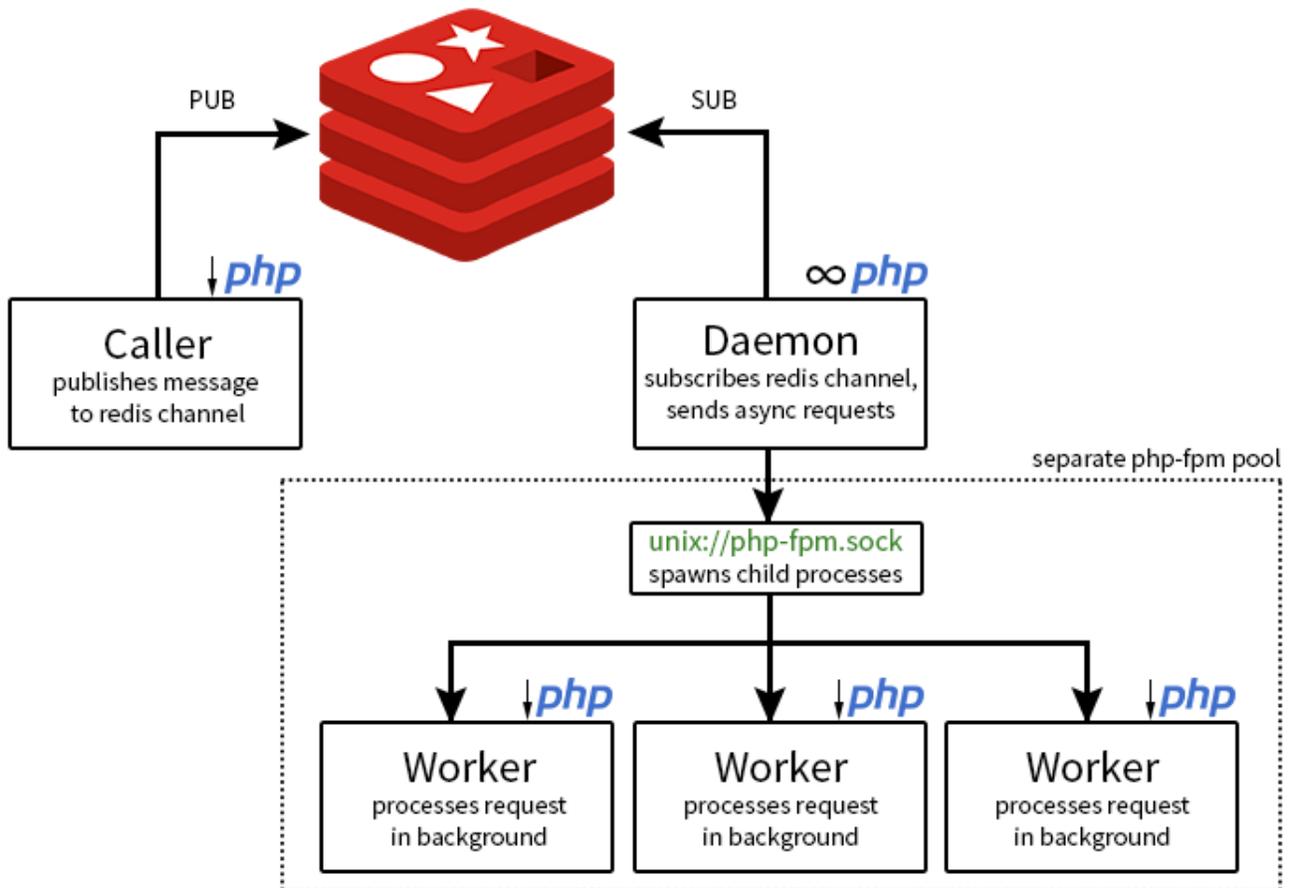


Redis is an open-source in-memory data structure project implementing a distributed, in-memory key-value database with optional durability. Redis supports different kinds of abstract data structures, such as strings, lists, maps, sets, sorted sets, hyperloglogs, bitmaps, streams and spatial indexes.

Redis typically holds the whole dataset in memory. Versions up to 2.4 could be configured to use what they refer to as virtual memory in which some of the dataset is stored on disk, but this feature is deprecated. Persistence is now achieved in two different ways: one is called snapshotting, and is a semi-persistent durability mode where the dataset is asynchronously transferred from memory to disk from time to time, written in RDB dump format. Since version 1.1 the safer alternative is AOF, an append-only file (a journal) that is written as operations modifying the dataset in memory are processed. Redis is able to rewrite the append-only file in the background in order to avoid an indefinite growth of the journal.

By default, Redis writes data to a file system at least every 2 seconds, with more or less robust options available if needed. In the case of a complete system failure on default settings, only a few seconds of data would be lost.

Below diagram shows a PHP daemon subscribing to redis channels and executing commands async on the php-fpm socket.



Reference: <https://hollo.me/php/experimental-async-php-volume-1.html>

- [Redis CLI\(Command Line Interface\) Examples](#) — redis-cli is a useful tool can handle Redis KEYs in the system. You can search / delete / get / clear / check its status simple on your terminal.
- [Redis Installation on CentOS 7](#) — Redis package is not included in the default CentOS repositories. It can be installed from the Remi repository.

- [Redis thread implementation in PHP without compilation on CentOS](#) — Redis is a useful solution when design systems on server and required the minimum latency between the client and server for handling message. We can easily build a thread application if we use shell\_exec based on & and wait. PHP also provides a solution named as pthreads. but unfortunately the pthreads extension cannot be used in a web server environment. Threading in PHP should therefore remain to CLI-based applications only.
- [Redis with High Availability by Replication](#)
- [Uncaught RedisException: Permission denied error in your PHP code](#) — When you face an error Uncaught RedisException: Permission denied in your PHP code, you will need to check if that is working normally in the command line but happening in the web page. If it is, that is 100% caused by your linux security policy. In that case, you will need to turn SELinux policy off that located at /etc/selinux/config.

Other references good to know

- <https://scotch.io/tutorials/getting-started-with-redis-in-php>
- [https://www.tutorialspoint.com/redis/redis\\_php.htm](https://www.tutorialspoint.com/redis/redis_php.htm)