redis

Redis is an **in-memory data structure** store that can be used as a **database, cache, and message broker.** It supports a variety of data structures such as strings, hashes, lists, sets, and more. One of the main features of Redis is its ability to **cache data in memory,** which allows it to achieve very fast read and write speeds. Redis can be used to improve the performance of web applications by **reducing the time it takes to access data from a database** or other slow storage layer. In addition to its caching capabilities, Redis also offers publish/subscribe messaging, transactions, and support for multiple data structures, making it a versatile tool for a variety of use cases.

How Redis cache works

Assume you have a web application running on a server using a database like MySQL. That web application needs to retrieve some records from this database. Such queries take some time to return the requested records. And, if the query is expensive, a user waiting for that data for more than one minute may have a bad experience.

However, Redis is made to make such processing faster and efficient. With it, it's possible to store data processed by a MySQL database query inside of a Redis cache instance. This allows data to be **retrieved directly from the server's memory**. This way, the application will not go all the way back to the database. Instead, the web server can check with Redis if it has the data it wants. So when another call is made and requires the same query transaction, instead of hitting the MySQL server again, the Redis object will serve the request from the object cache.

These commands will install the Redis server on your system and configure it to use a maximum of 256MB of memory. The "**maxmemory"** directive sets the maximum amount of memory that Redis is allowed to use. The "**maxmemory-policy"** directive specifies the policy that Redis should use when the maximum memory limit has been reached. In this case, the "**allkeys-lru**" policy will cause Redis to remove the least recently used keys in order to free up memory.

It's important to carefully set the maximum memory limit for your Redis instance, as using too much memory can cause performance issues or even crash the system. You should also consider the type of workload that your Redis instance will be handling, as different workloads may have different memory requirements.

o keep the redis CMD ["redis-server", "--protected-mode", "no"]

The **redis-server** command is used to start the Redis server, which is a persistent key-value store that can be used to store data structures such as strings, hashes,

lists, and sets.
The **--protected-mode** option specifies whether the Redis server should run in protected mode or not. Protected mode is a security feature that was introduced in Redis version 4.0. It prevents the Redis server from accepting connections from clients that are not running on the same host as the server.
By specifying the **no** argument, the **--protected-mode** option disables protected mode and allows the Redis server to accept connections from clients running on any host.
and add these line to wp-config.php
define('WP_REDIS_HOST', 'redis'); define('WP_REDIS_PORT', 6379);
define('WP_CACHE', true);

How to know your redis is installed on wordpress and running

1. **Check redis is properly installed on your redis image**
- docker exec -it redis redis-cli monitor, Then do ping and the answer should be PONG. Great your redis is installed.
2. **Check if the plugin is installed on wordpress**
    - Go to your wp-admin panel on wordpress
    - click on plugins on the left tab
    - If you see "Redis Object Cache", Congrats !, click on settings and you will see Status "Connected" in green
3. **To check the cache operation, run the following command:**
   docker exec -it redis redis-cli monitor
   If the output is OK, then everything is working for us