RedisGears 1.2 release notes

Requirements

RedisGears v1.2.4 requires:

- Minimum Redis compatibility version (database): 6.0.0

v1.2.4 (May 2022)

This is a maintenance release for RedisGears 1.2.

Update urgency: LOW: No need to upgrade unless there are new features you want to use.

Details:

- Improvements:
  - #772 Added the ability to upgrade a dependency at runtime with FORCE_REQUIREMENTS_REINSTALLATION on RG.PYEXECUTE.
  - #765 Allow deactivating override Python allocators for performance improvements.

- Bug fixes:
  - #761, #760, #778 StreamReader fixes to pause and unregister stream processing.

v1.2.3 (April 2022)

This is a maintenance release for RedisGears 1.2.

Update urgency: LOW: No need to upgrade unless there are new features you want to use.

Details:

- Improvements:
  - #739 Added TLS support
  - #734 Pause/Unpause registrations
  - #741 Added build for Python-only artifacts

- Bug fixes:
  - #740 Fix -nan value on registration stats

v1.2.2 (February 2022)

This is the General Availability release of RedisGears 1.2.

Headlines

Plugins and JVM support

RedisGears 1.2 comes with a new plugin mechanism that allows you to decide which languages you want to load into RedisGears.
Currently, we support two languages: Python and Java (JVM languages). You can decide which language you want to use using the new Plugin configuration.

Full documentation for JVM support can be found on the Redis documentation website.

**Python async await support**

RedisGears provides support for Python coroutines. Each step of your gears function can now be a Python coroutine that will take the execution to the background or will wait for some event to happen. Refer to the following links for more information:

- Async Await Support
- Async Await Advanced Topics

**Override commands API**

You can override Redis vanilla commands with a function. For more information, refer to the RedisGears command hooks documentation.

**Key miss event for read-through pattern**

Requested by many users, RedisGears 1.2 allows you to register functions on key miss event. One use case for this is to implement a read-through caching pattern. For more information about this topic, refer to the following links:

- Key Miss Event in the RedisGears documentation.
- rghibernate recipe that leverages the key miss event to implement read-through from external databases.

**Better visibility and analyzing tools**

We improved the experience during the development phase by enabling better debugging and troubleshooting. There is still room for improvement but RedisGears 1.2 makes the first steps toward a simpler API that is easier to use. This new version allows you to name your code and upgrade it with a single Redis command. For more information, refer to the upgrade section of the RedisGears introduction documentation.

RedisGears now tracks the following new statistics to better analyze your registrations:

- lastRunDurationMS - duration in milliseconds of the last execution
- totalRunDurationMS - total runtime of all executions in milliseconds
- avgRunDurationMS - average execution runtime in milliseconds

For streams, RedisGears also tracks the following data:

- lastEstimatedLagMS - gives the last batch lag (the time difference between the first batch entry in the stream and the time the entire batch finished processing)
- avgEstimatedLagMS - average of the lastEstimatedLagMS field.

The RG.DUMPREGISTRATIONS command exposes these new statistics.

RedisGears 1.2 also adds support for a Python profiler, specifically cProfile. For more information, refer to the documentation for the following commands:

- RG.PYPROFILE STATS
- RG.PYPROFILE RESET

**RedisAI integration**
Although RedisAI integration was already supported in v1.0, RedisGears 1.2 adds official support for all capabilities in RedisAI v1.2. The API was extended to support RedisAI DAG and was combined with the new async await API to achieve the best performance possible.

**Details**

Bug fixes (since 1.0.9):

- #557, #554 RG . CONFIGGET returns user-defined configuration
- #572 Lock Redis GIL when creating RedisAI DAG
- #661, #536 Added RG . TRIGGERONKEY
- #650 Do not propagate MULTI EXEC on Redis 7
- #671, #558 Wait for cluster to be initialized when reading stream data
- #656 Stream reader creates more than one execution on a stream
- #676 Globals dictionary not set correctly after deserialization
- #665, #679 Allow case-insensitive event type on command reader
- #688, #545 hashtag() function for Redis Enterprise
- #688, #545 Check REDISMODULE_CTX_FLAGS_DENY_BLOCKING flag before blocking the client

---

**Note:**

- This is the first GA version of 1.2. The version inside Redis is 1.2.2 in semantic versioning. Since the version of a module in Redis is numeric, we could not add a GA flag.
- Minimum Redis version: 6.0.0

---

**Updated:** May 23, 2022