RedisTimeSeries 1.4 release notes

Requirements
RedisTimeSeries v1.4.14 requires:

- Minimum Redis compatibility version (database): 5.0.0

v1.4.14 (February 2022)
This is a maintenance release for RedisTimeSeries 1.4.
Update urgency: MODERATE: Program an upgrade of the server, but it’s not urgent.
Bug fixes:

- #891, #892 Fixed memory leak in parseCreateArgs when parsing error occurs (MOD-1958)

v1.4.13 (November 2021)
This is a maintenance release for RedisTimeSeries 1.4.
Update urgency: MODERATE: Program an upgrade of the server, but it’s not urgent.
Bug fixes:

- #881 Replicate only successful insertion of TS. MADD

v1.4.11 (November 2021)
This is a maintenance release for RedisTimeSeries 1.4.
Update urgency: MODERATE: Program an upgrade of the server, but it’s not urgent.
Bug fixes:

- #862 Index dictionary should be freed on removing the last element

v1.4.10 (July 2021)
This is a maintenance release for RedisTimeSeries 1.4.
Update urgency: HIGH: There is a critical bug that may affect a subset of users. Upgrade!
Bug fixes:

- #760 Avoid closing the same key twice, causing server crash on RENAME of other keys

v1.4.9 (May 2021)
This is a maintenance release for version 1.4.

Update urgency: Medium

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Bug fixes:

- #712 Missing keytype check on TS.INCRBY/DECRBY causes shards to crash
- #719 Support for renaming time series keys

v1.4.8 (March 2021)

This is a maintenance release for version 1.4.

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Bug fixes:

- #612 Crash on MGET/MRANGE
- #606 Memory leak when key loaded from RDB
- #624 Uninitialised memory access on log

v1.4.7 (December 2020)

This is a maintenance release for version 1.4.

Headlines:

- This release improves overall stability, and provides fixes for issues found after the previous release.

Bug fixes:

- #581 Misaligned allocators usage might crash Redis.
- #588 ON_DUPLICATE min/max rules not working for negative value.

v1.4.6 (November 2020)

This is a maintenance release for version 1.4.

Headlines:

- This release improves overall stability and provides fixes for issues found after the previous release.

Minor enhancements:

- #565 duplicate policy: add SUM option: If a previous sample exists, add the new sample to it so that the updated value is equal to (previous + new). If no previous sample exists, set the updated value equal to the new value. (PR #565
- #559 Compressed chunk will be the default global option
Bug fixes:

- #528 Out of order insert might crash Redis if there's an update to an empty downsampled key
- #561 TS.MRANGE command might crash if there's an expired key that was deleted in the result set

v1.4 GA (September 2020)

This is the General Availability release for RedisTimeSeries 1.4.

Highlights:

**Ability to backfill time series!** You can now add samples to a time series where the time of the sample is older than the newest sample in the series. This enables:

- Adding out of order of samples to time series.
- Batch loading of historical samples into an existing series.
- Updating existing samples (for example for compliance reasons).

This has been the most requested feature for RedisTimeSeries. We look forward to your feedback so we can move to a general availability release soon.

Details:

- Added functionality:
  - #254 TS.REVRANGE and TS.MREVRANGE [commands] allow for querying in descending order of Timestamps. ([https://oss.redislabs.com/redistimeseries/1.4/commands/#tsmrangetsmrevrange](https://oss.redislabs.com/redistimeseries/1.4/commands/#tsmrangetsmrevrange))
  - #503 - RDB saves the whole chunk instead of individual samples giving a speed and space improvement when saving or loading an RDB file
  - #502 - The ability to set, at creation time, the data section size of each chunk using flag `CHUNK_SIZE`. TS.INFO uses chunkSize instead of `maxSamplesPerChunk`.
  - #437 Allow backfilling of samples and updating of existing samples
    - Works with compressed and uncompressed series.
    - This comes with a performance hit when a sample is written out-of-order. We will publish numbers once we are generally available, but are still considering optimisations.
  - #521 `DUPLICATE_POLICY` allows to configure on module, series and sample level how to handle duplicate samples. A duplicate sample is a sample for which the series holds already a sample on the same timestamp. Note that the default behaviour is equal to v1.2: BLOCK

Notes: The version inside Redis will be 10405 or 1.4.5 in semantic versioning. Since the version of a module in Redis is numeric, we could not add an GA flag.

**Updated:** February 8, 2022