📊 Analysis Report

≡ JVM Heap Size

Generation	Allocated ❓	Peak ❓
Young Generation	885 mb	884.4 mb
Old Generation	2.59 gb	2.38 gb
Meta Space	1.01 gb	3.8 mb
Young + Old + Meta space	4.89 gb	2.82 gb

JVM Heap size - Allocated vs Peak (mb)

	Young Gen	Old Gen	Meta Space
allocated	885mb	2,655mb	1,032mb
peak usage	884.4mb	2,436.66mb	3.8mb

Young Gen ■ Old Gen ■ Meta Space
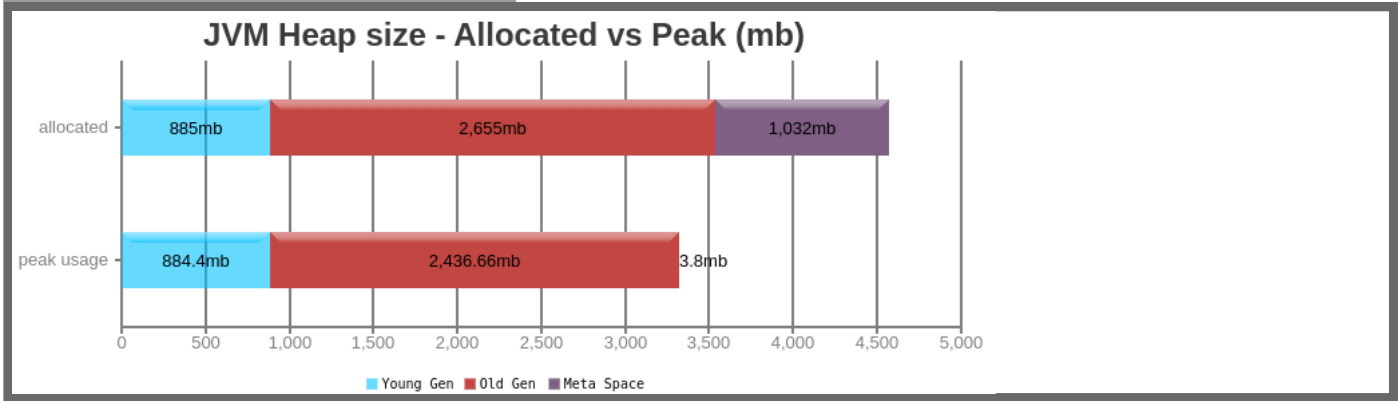
🔍 Key Performance Indicators

(Important section of the report. To learn more about KPIs, click here (https://blog.gceasy.io/2016/10/01/garbage-collection-kpi/))

1 Throughput ❓ : 42.588%

2 Latency:

Avg Pause GC Time ❓	**123 ms**
Max Pause GC Time ❓	**380 ms**

GC **Pause** Duration Time Range ❓:

Duration (secs)	No. of GCs	Percentage
0 - 0.1	32	42.105%
0.1 - 0.2	36	89.474%
0.2 - 0.3	4	94.737%
0.3 - 0.4	4	100.0%

GC Duration Time Range

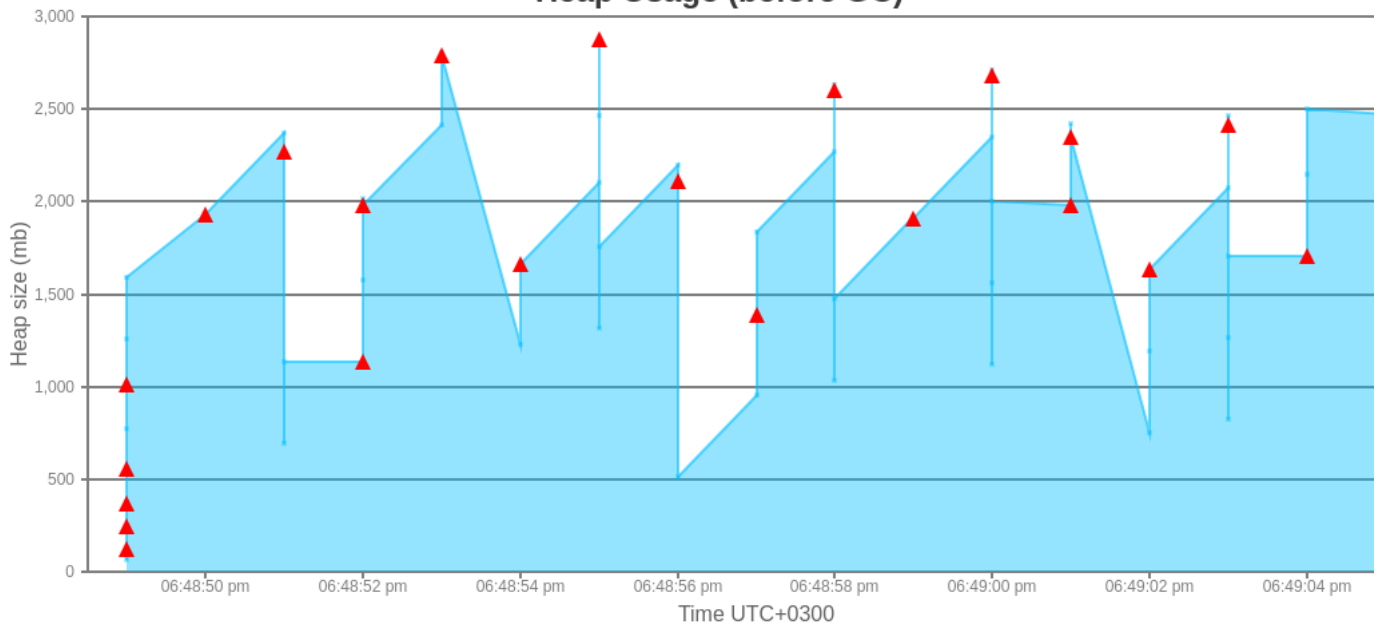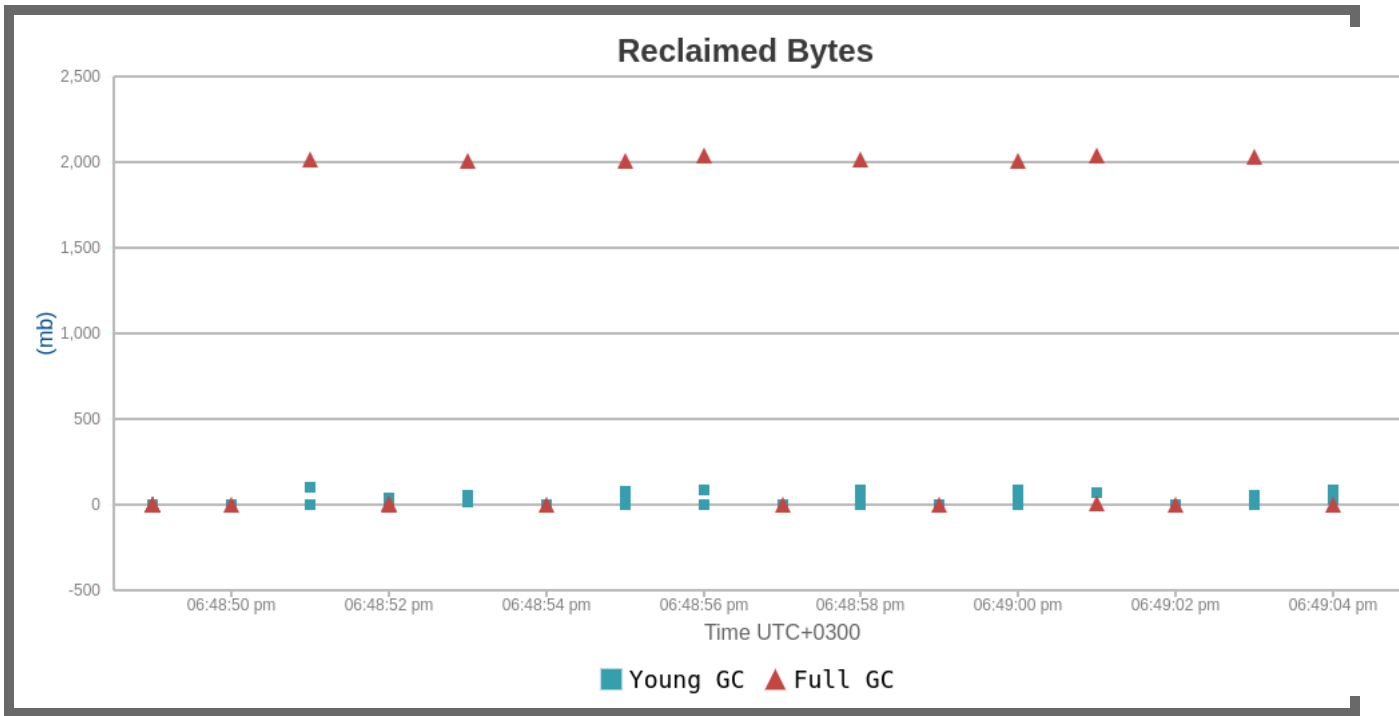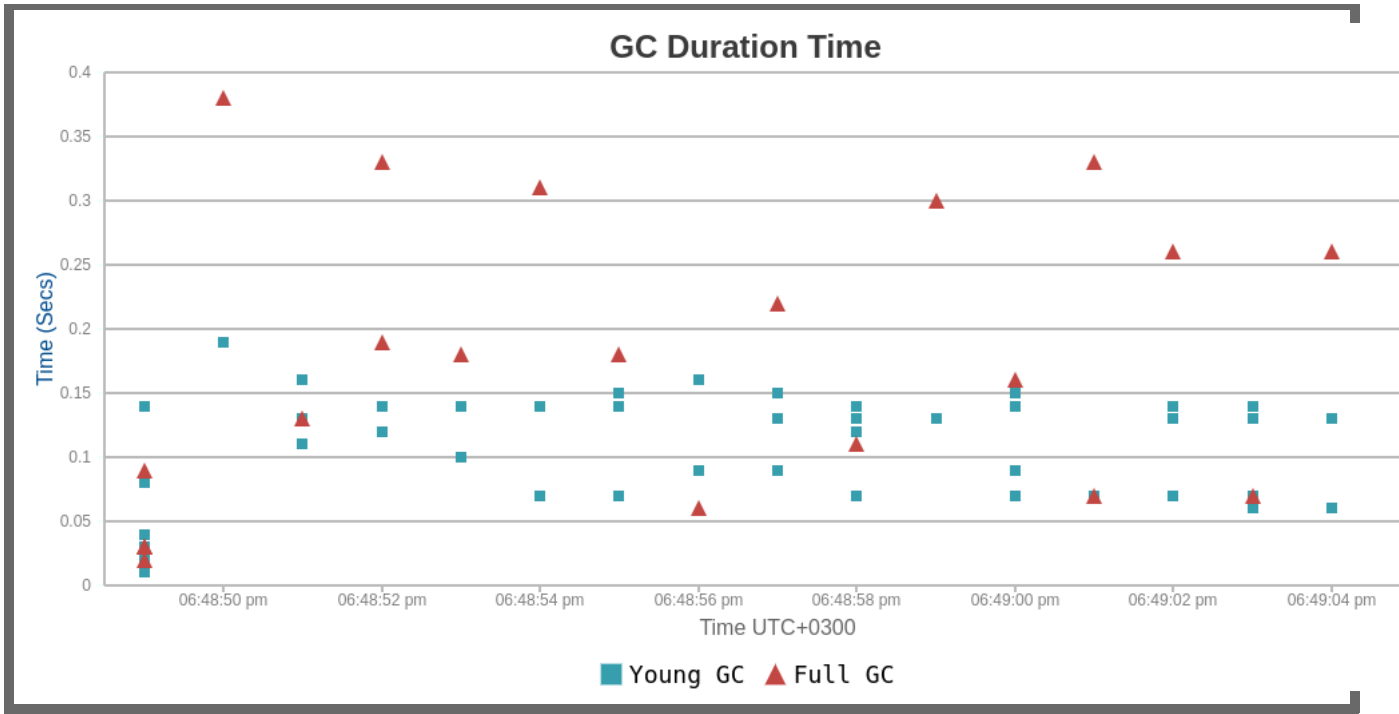|	
0.3 - 0.4 secs	5.26%
0.2 - 0.3 secs	5.26%
0.1 - 0.2 secs	47.37%
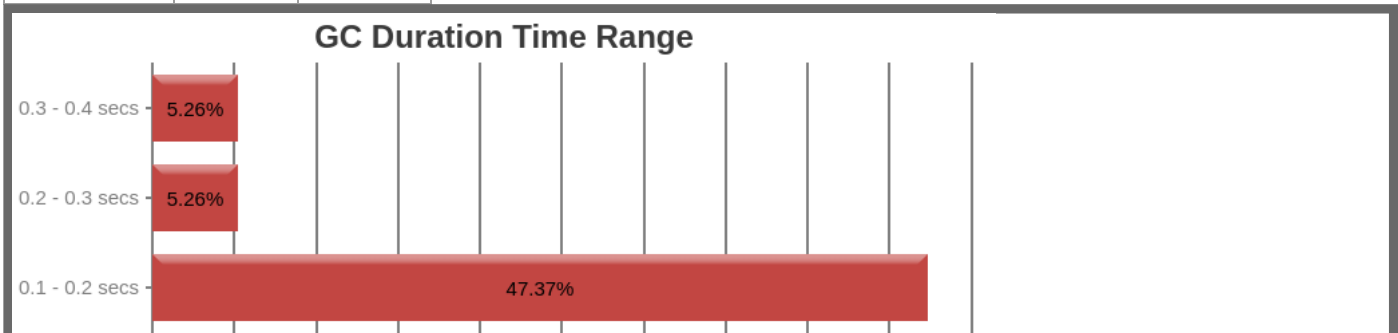
▁▂▃▅ Interactive Graphs
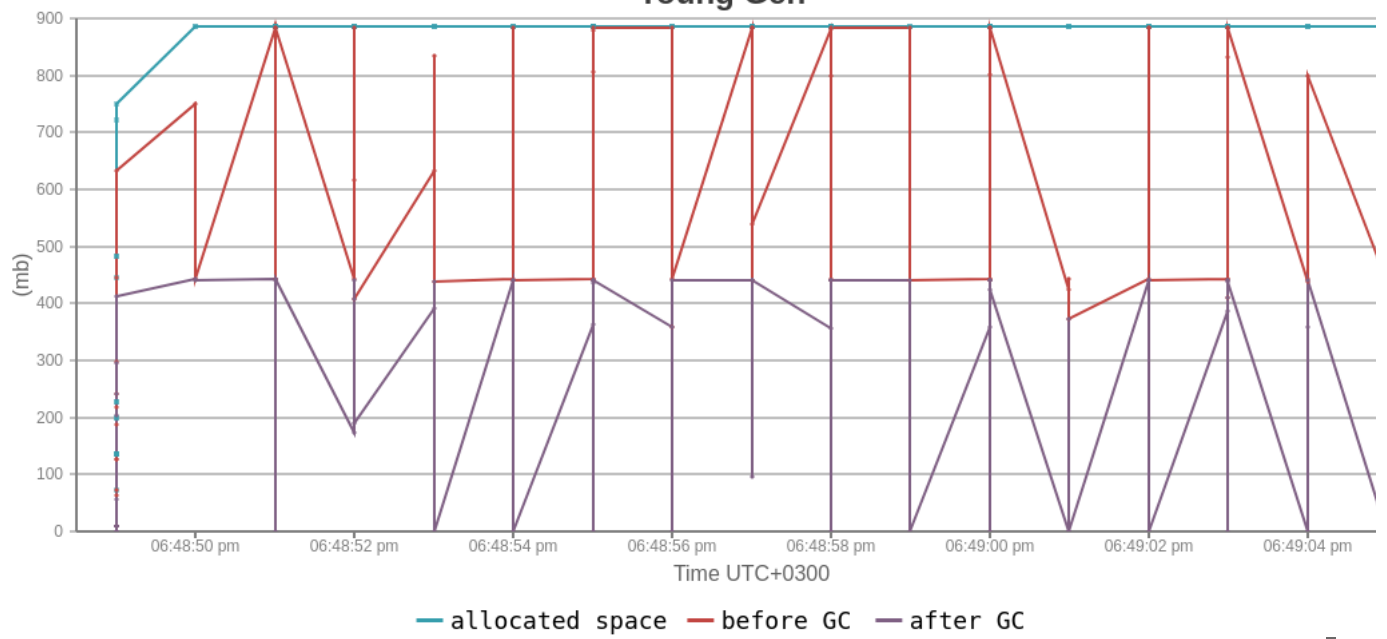
(All graphs are zoomable)
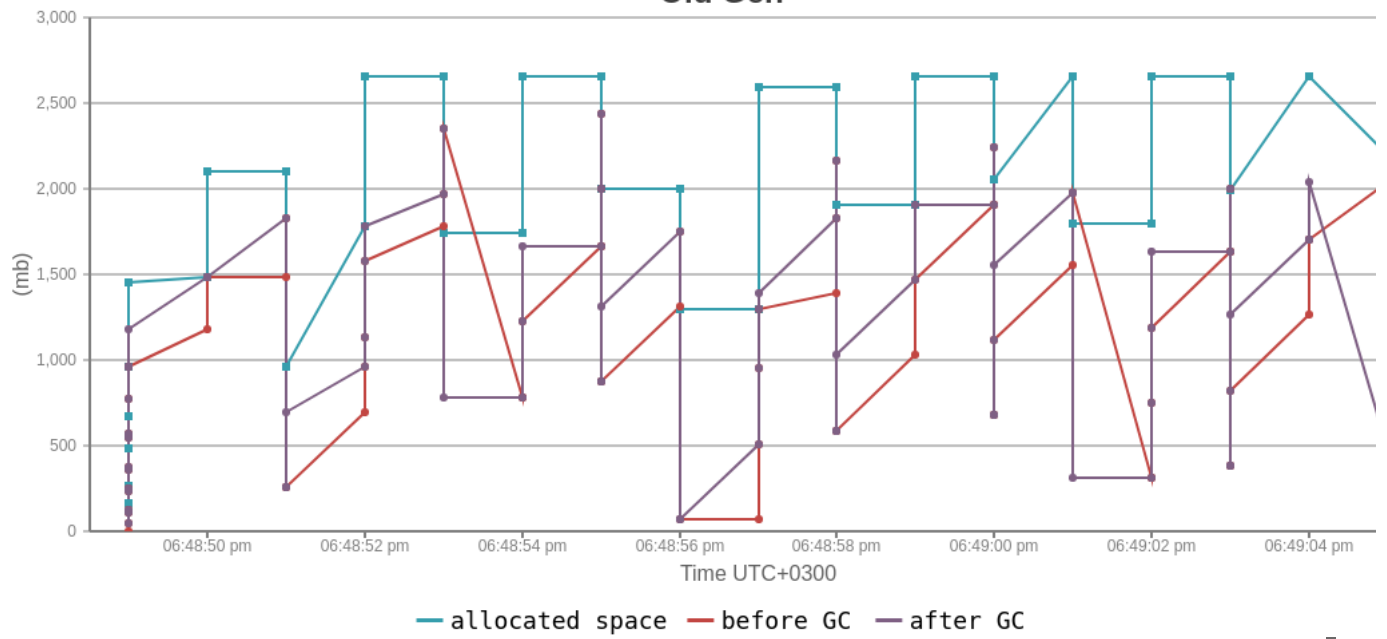
Heap Usage (after GC)
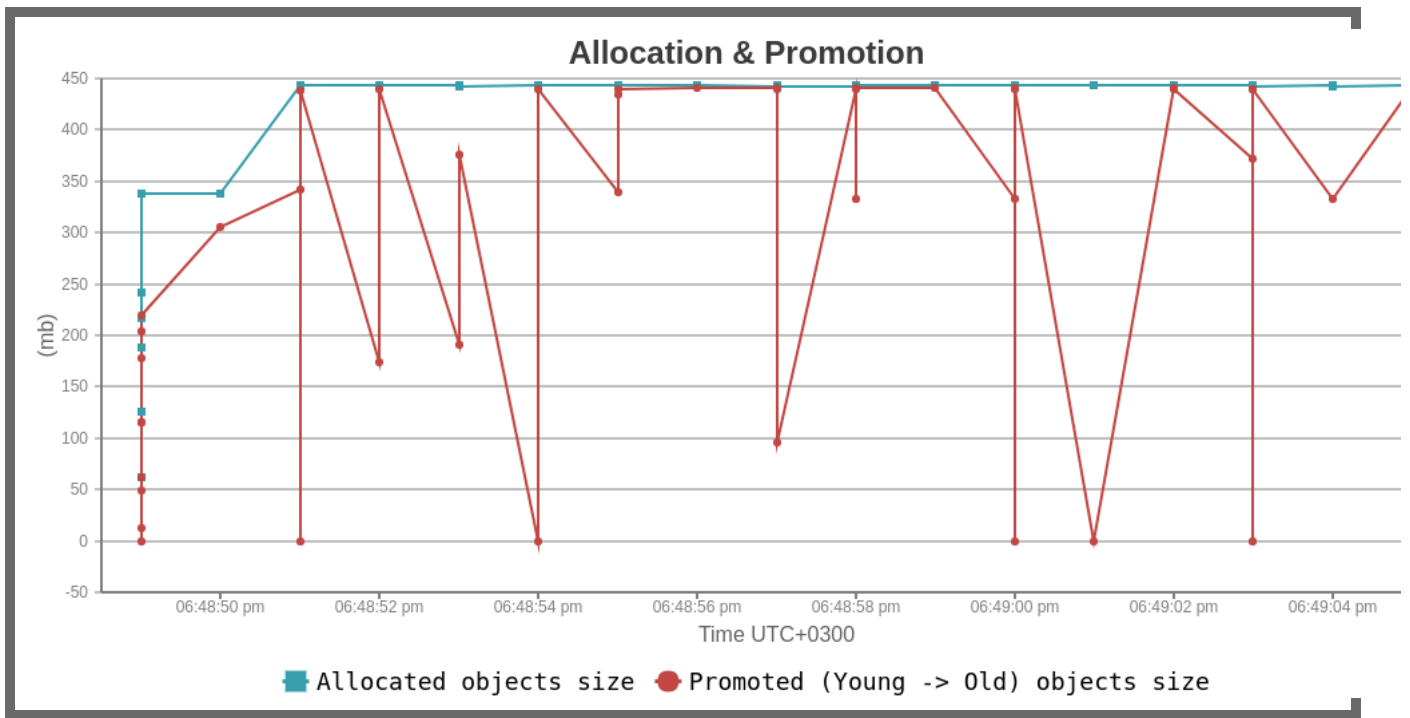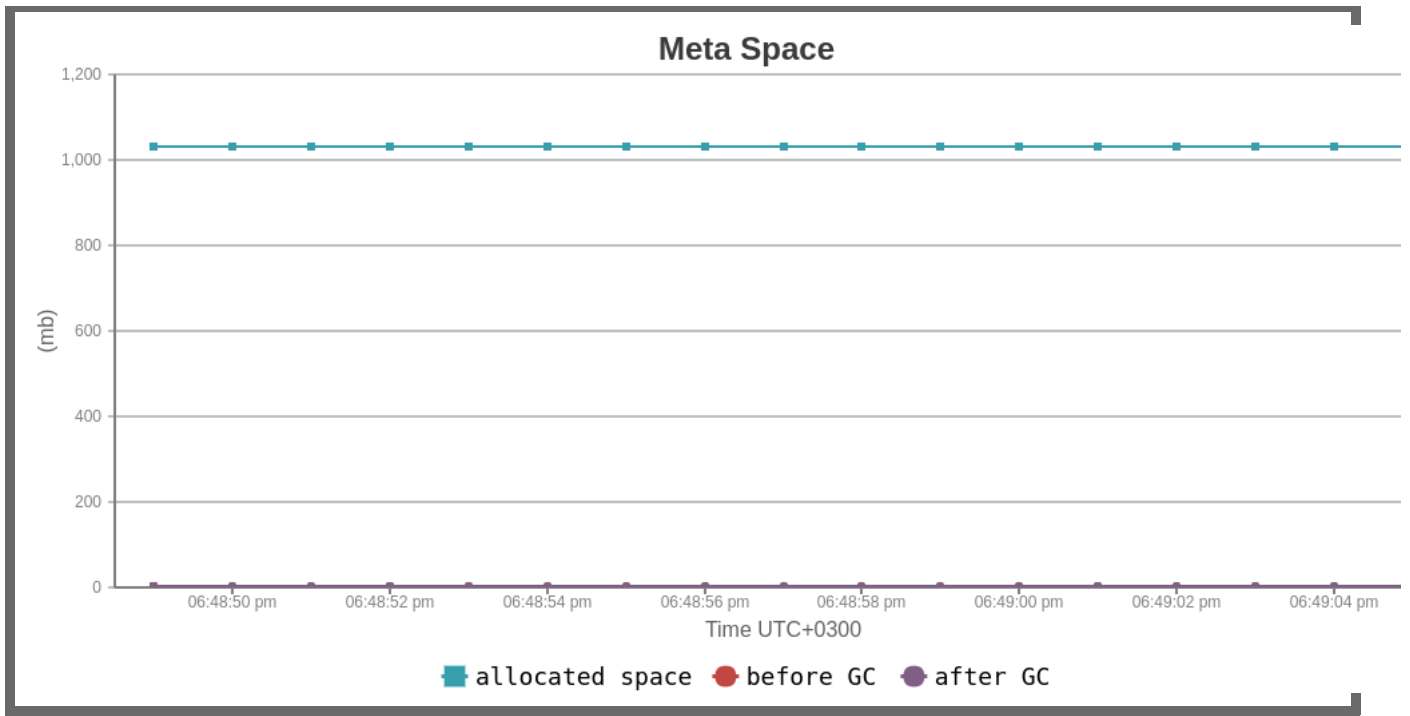

Heap Usage (before GC)

GC Duration Time

Reclaimed Bytes

Young Gen

— allocated space — before GC — after GC

Old Gen

— allocated space — before GC — after GC

Meta Space

Allocation & Promotion
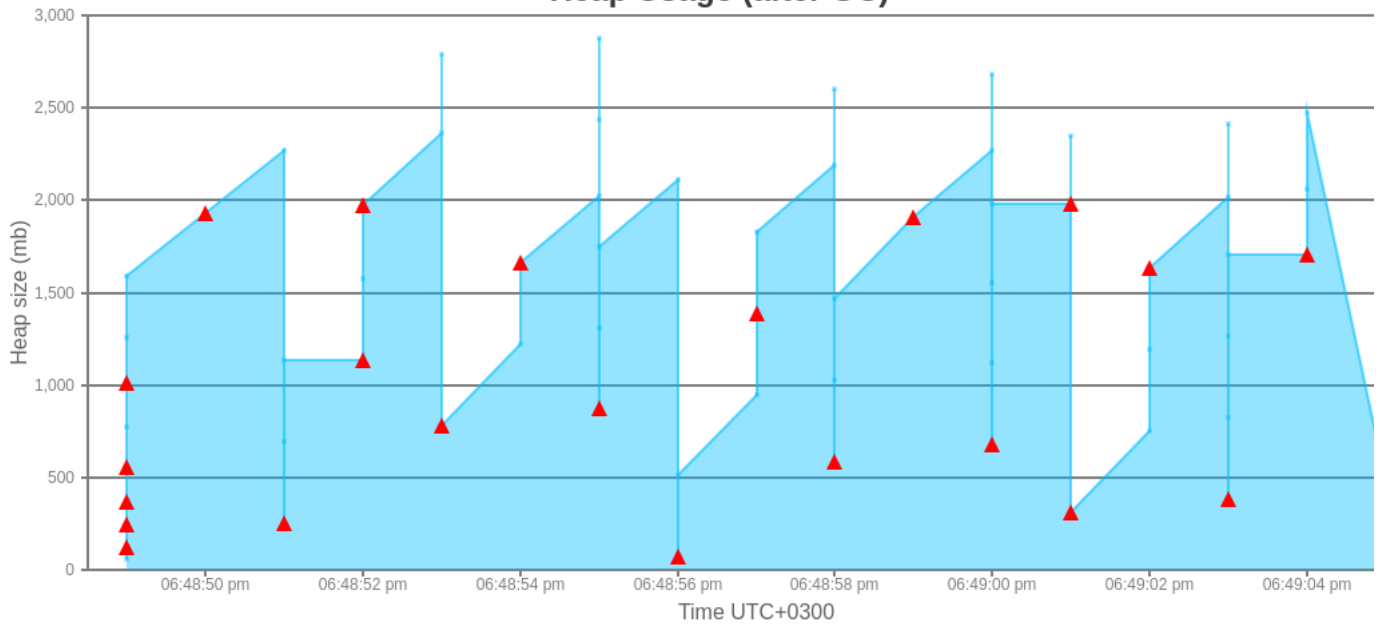
⊘ GC Statistics ❷
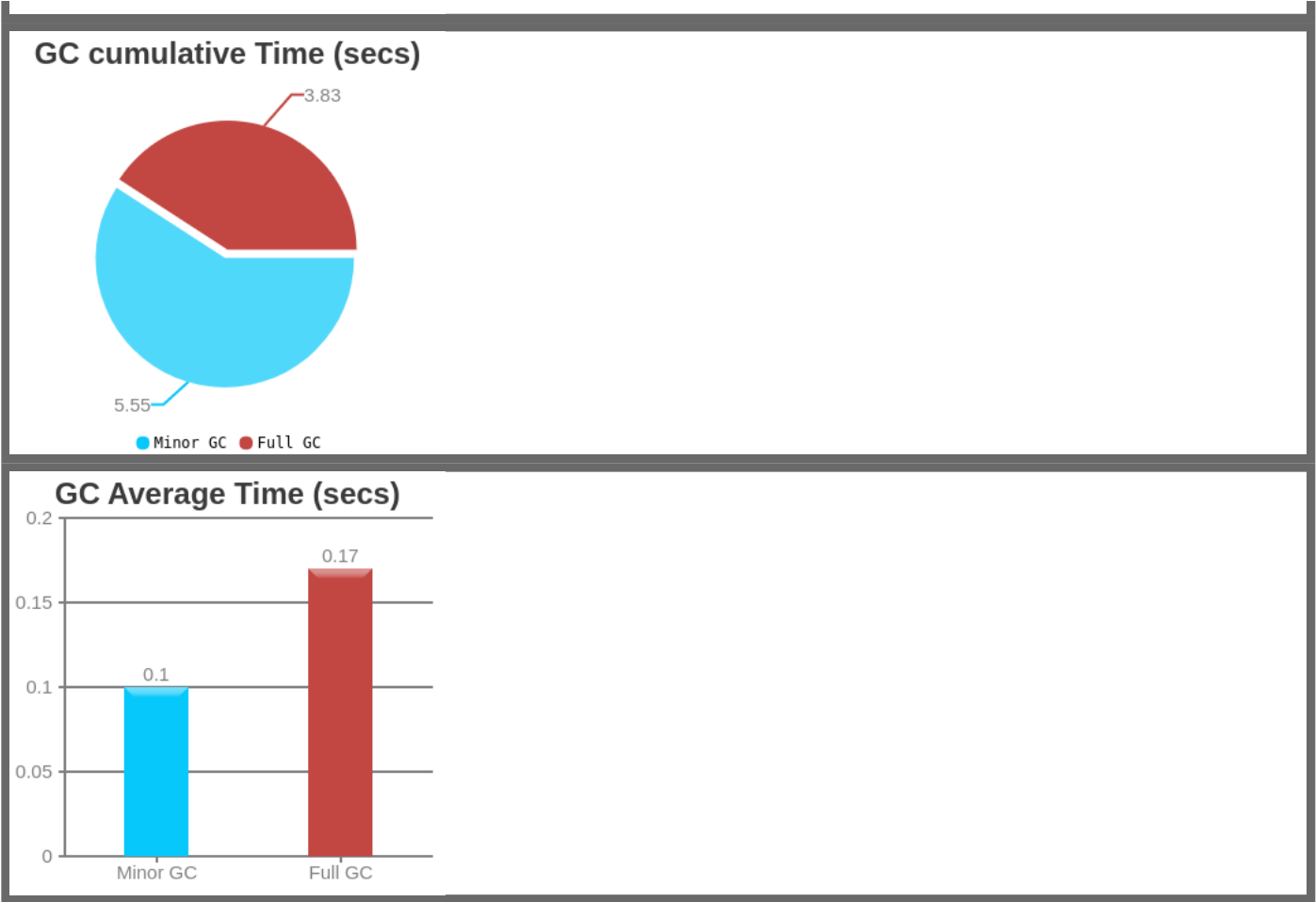
Reclaimed Bytes (mb)

Total GC stats

Total GC count ❓	76
Total reclaimed bytes ❓	18.74 gb
Total GC time ❓	9 sec 380 ms
Avg GC time ❓	123 ms
GC avg time std dev	77 ms
GC min/max time	10 ms / 380 ms
GC Interval avg time ❓	217 ms

Minor GC stats

Minor GC count	53
Minor GC reclaimed ❓	1,020.66 mb
Minor GC total time	5 sec 550 ms
Minor GC avg time ❓	105 ms
Minor GC avg time std dev	44 ms
Minor GC min/max time	10 ms / 190 ms
Minor GC Interval avg ❓	314 ms

Full GC stats

Full GC Count	23
Full GC reclaimed ❷	17.74 gb
Full GC total time	3 sec 830 ms
Full GC avg time ❷	167 ms
Full GC avg time std dev	111 ms
Full GC min/max time	20 ms / 380 ms
Full GC Interval avg ❷	719 ms

GC Pause Statistics

Pause Count	76
Pause total time	9 sec 380 ms
Pause avg time ❷	123 ms
Pause avg time std dev	0.0
Pause min/max time	10 ms / 380 ms

✿ Object Stats

(These are perfect micro-metrics (https://blog.gceasy.io/2017/05/30/improving-your-performance-reports/) to include in your performance reports)

Total created bytes ❷	20.49 gb
Total promoted bytes ❷	12.21 gb
Avg creation rate ❷	1.25 gb/sec
Avg promotion rate ❷	765.24 mb/sec

◗ Memory Leak ❷

No major memory leaks.

(**Note:** there are 8 flavours of OutOfMemoryErrors (https://tier1app.files.wordpress.com/2014/12/outofmemoryerror2.pdf). With GC Logs you can diagnose only 5 flavours of them(Java heap space, GC overhead limit exceeded, Requested array size exceeds VM limit, Permgen space, Metaspace). So in other words, your application could be still suffering from memory leaks, but need other tools to diagnose them, not just GC Logs.)

⬇Ξ Consecutive Full GC ❷

None.

❚❚ Long Pause ❷

None.