

# Prometheus Queries

App One:

Counter Metrics:

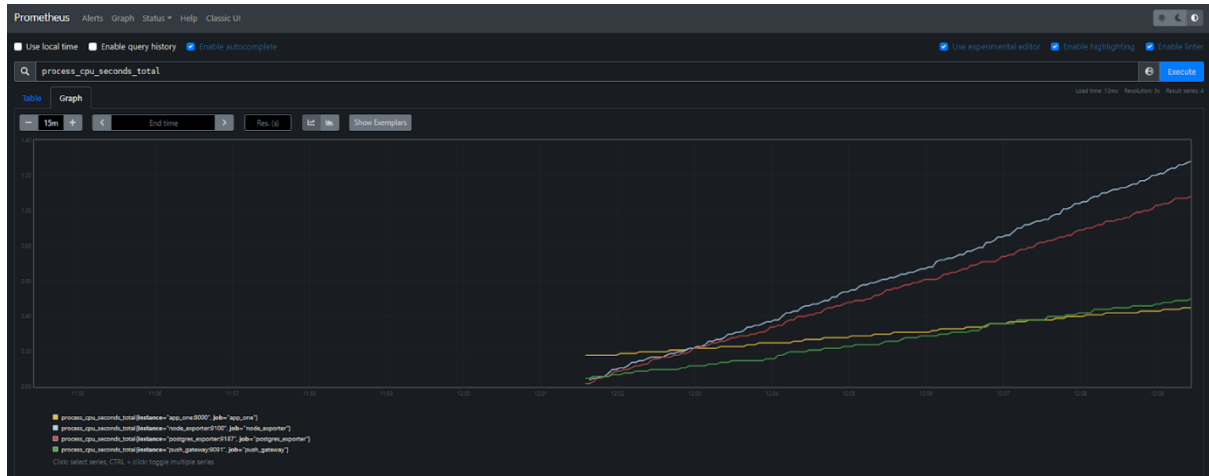


Figure 1: `"process_cpu_seconds_total"` measures the total amount of CPU time consumed by a process.



Figure 2: `"python_gc_objects_collected"` measures the number of objects collected by the garbage collector in Python

## Gauge Metrics:



Figure 3: "`process_virtual_memory_bytes`" tracks the total amount of virtual memory that the process is using.



Figure 4: "`process_max_fds`" indicates the maximum number of file descriptors a process can open.

## App Two:

### Counter Metrics for GoLang

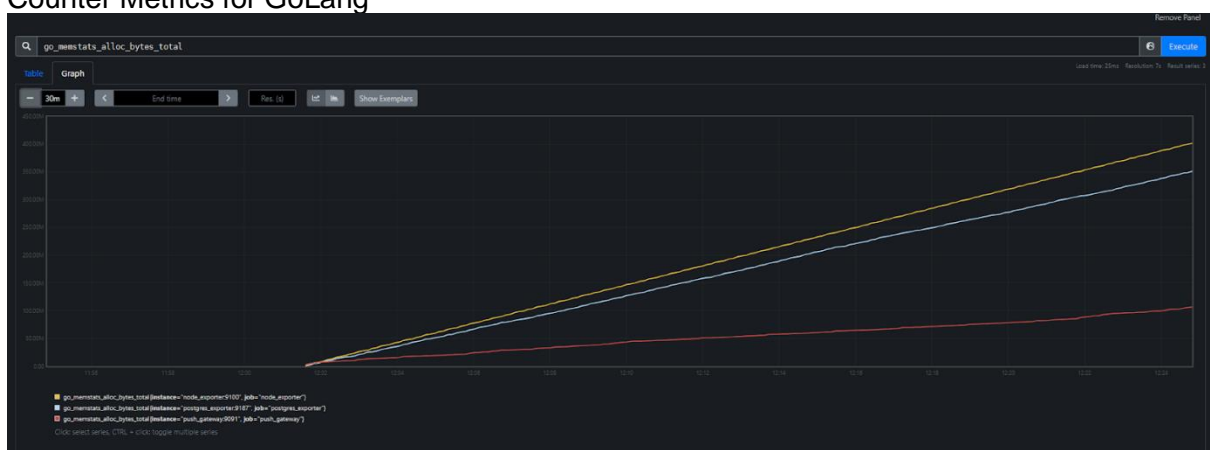


Figure 5: "`go_memstats_alloc_bytes_total`" tracks the cumulative number of bytes allocated by the Go runtime for memory.

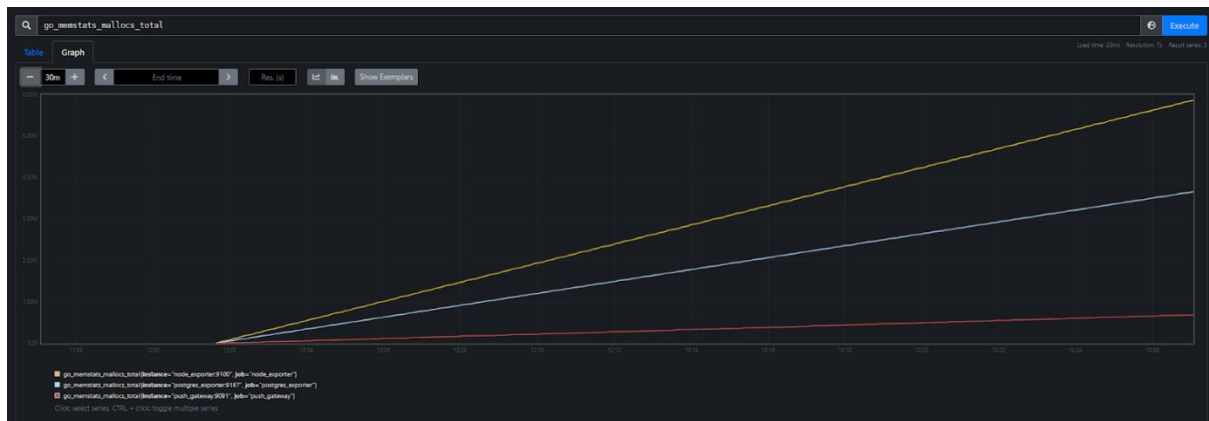


Figure 6: "go\_memstats\_malloc\_bytes\_total" tracks the cumulative number of bytes allocated with malloc by Go runtime since the application started.

## Counter metrics for nano

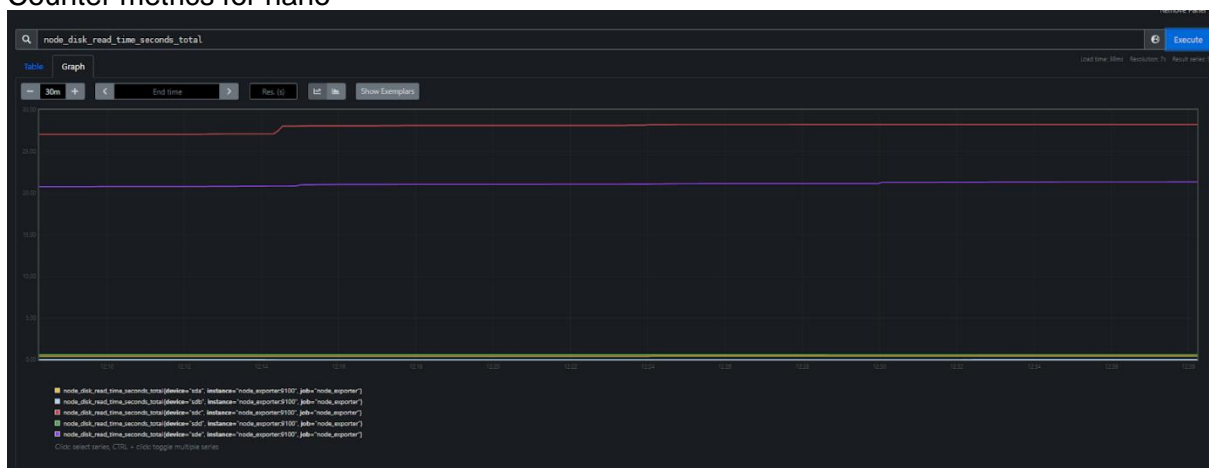


Figure 7: "node\_disk\_read\_time\_seconds\_total" measures the total time that the disk has spent reading data.

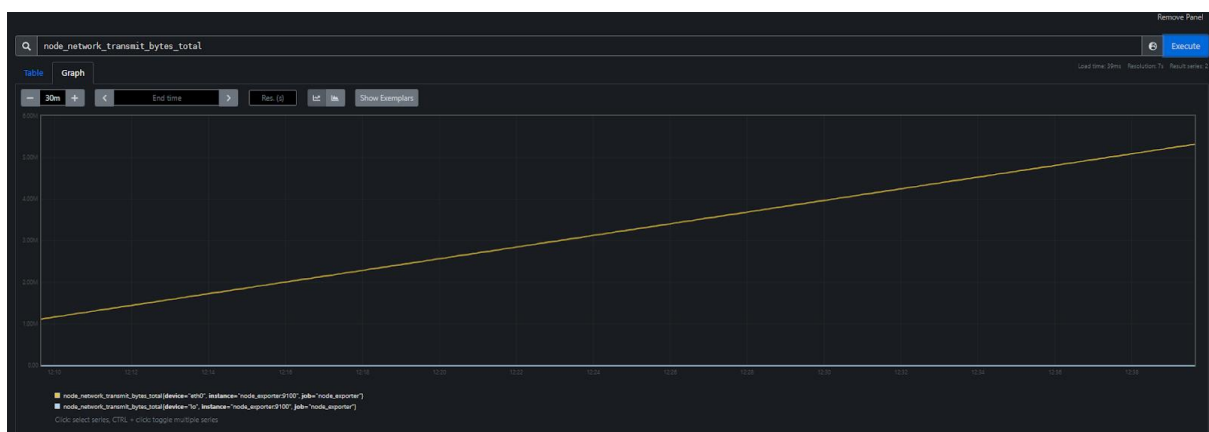


Figure 8: "node\_network\_transmit\_bytes\_total" measures the total number of bytes transmitted over the network interface since the system started.

## Gauge Metrics for GoLang

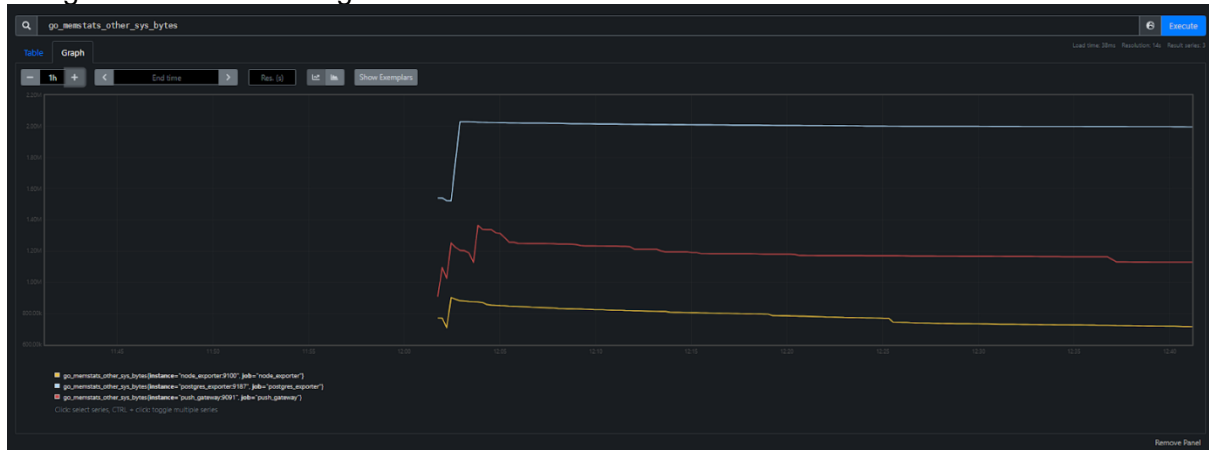


Figure 9: "go\_memstats\_other\_sys\_bytes" tracks the number of bytes allocated for miscellaneous purposes by the Go runtime.



Figure 10: "process\_virtual\_memory\_bytes" tracks the total amount virtual memory that a process is using.

## Gauge Metrics for Nano

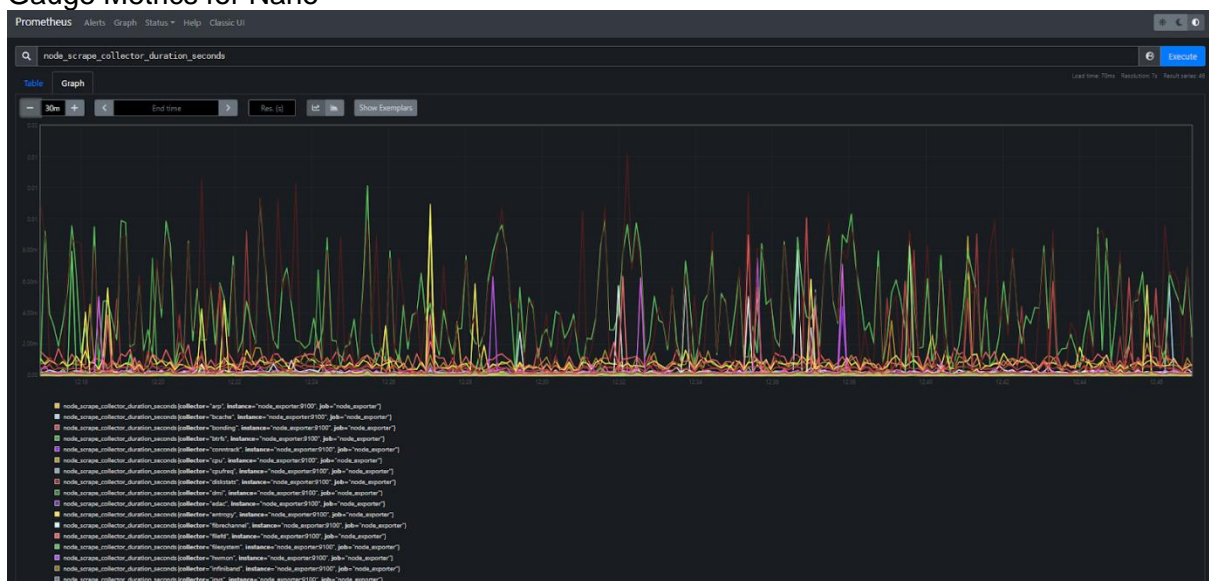


Figure 11: "node\_scrape\_collector\_duration\_seconds" measures the time taken by a specific collector to gather and return metrics during scrape.



Figure 12: "node\_time\_clocksource\_available\_info" provides information about the available clock sources on a machine.