Lesson 01 Demo 02

Configuring Prometheus to Scrape and Visualize the Metrics

Objective: To configure Prometheus to scrape and visualize metrics for monitoring system performance through its web interface for improved observability and real-time insights

Tools required: Linux operating system

Prerequisites: Basic understanding of Web Applications

Steps to be followed:

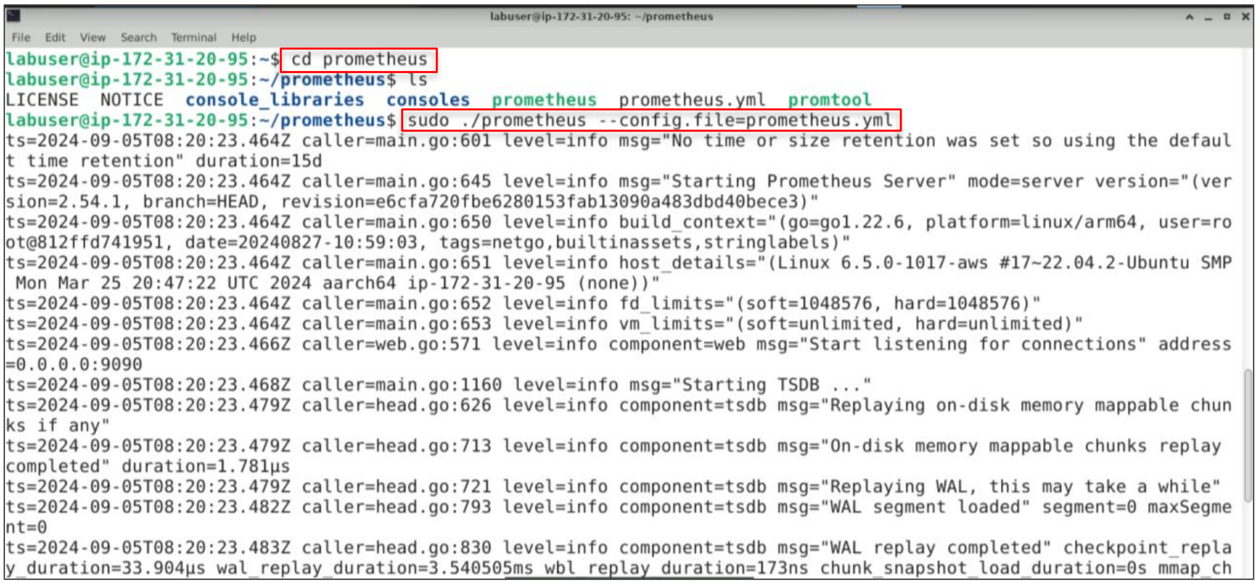
1. Start Prometheus binary
2. Explore Prometheus UI

Step 1: Start Prometheus binary

1.1 Run the following commands to change the current directory to the Prometheus directory and start the Prometheus server:

cd prometheus

sudo ./prometheus --config.file=prometheus.yml

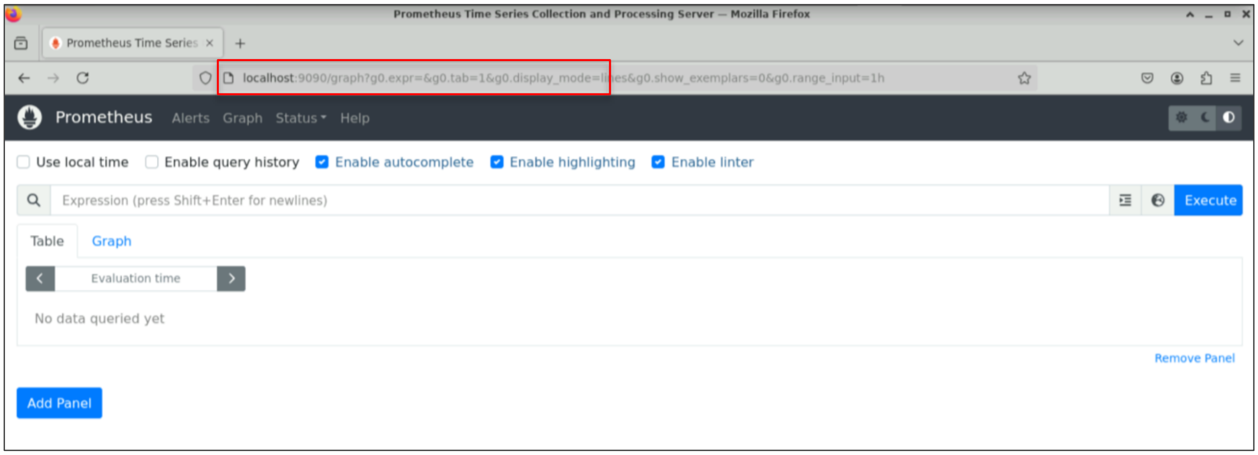


```
labuser@ip-172-31-20-95:~$ cd prometheus
labuser@ip-172-31-20-95:~/prometheus$ ls
LICENSE  NOTICE  console_libraries  consoles  prometheus  prometheus.yml  promtool
labuser@ip-172-31-20-95:~/prometheus$ sudo ./prometheus --config.file=prometheus.yml
ts=2024-09-05T08:20:23.464Z caller=main.go:601 level=info msg="No time or size retention was set so using the defaul
t time retention" duration=15d
ts=2024-09-05T08:20:23.464Z caller=main.go:645 level=info msg="Starting Prometheus Server" mode=server version="(ver
sion=2.54.1, branch=HEAD, revision=e6cfa720fbe6280153fab13090a483dbd40bece3)"
ts=2024-09-05T08:20:23.464Z caller=main.go:650 level=info build_context="(go=go1.22.6, platform=linux/arm64, user=ro
ot@812ffd741951, date=20240827-10:59:03, tags=netgo,builtinassets,stringlabels)"
ts=2024-09-05T08:20:23.464Z caller=main.go:651 level=info host_details="(Linux 6.5.0-1017-aws #17~22.04.2-Ubuntu SMP
 Mon Mar 25 20:47:22 UTC 2024 aarch64 ip-172-31-20-95 (none))"
ts=2024-09-05T08:20:23.464Z caller=main.go:652 level=info fd_limits="(soft=1048576, hard=1048576)"
ts=2024-09-05T08:20:23.464Z caller=main.go:653 level=info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2024-09-05T08:20:23.466Z caller=web.go:571 level=info component=web msg="Start listening for connections" address
=0.0.0.0:9090
ts=2024-09-05T08:20:23.468Z caller=main.go:1160 level=info msg="Starting TSDB ..."
ts=2024-09-05T08:20:23.479Z caller=head.go:626 level=info component=tsdb msg="Replaying on-disk memory mappable chun
ks if any"
ts=2024-09-05T08:20:23.479Z caller=head.go:713 level=info component=tsdb msg="On-disk memory mappable chunks replay
completed" duration=1.781µs
ts=2024-09-05T08:20:23.479Z caller=head.go:721 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2024-09-05T08:20:23.482Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=0 maxSegme
nt=0
ts=2024-09-05T08:20:23.483Z caller=head.go:830 level=info component=tsdb msg="WAL replay completed" checkpoint_repla
y_duration=33.904µs wal_replay_duration=3.540505ms wbl_replay_duration=173ns chunk_snapshot_load_duration=0s mmap_ch
```

.

Step 2: Explore Prometheus UI

2.1 Navigate to the browser and enter the URL **http://localhost:9090/** or **http://<public-ip>:9090/** to access the Prometheus console



Prometheus Time Series Collection and Processing Server — Mozilla Firefox

localhost:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h

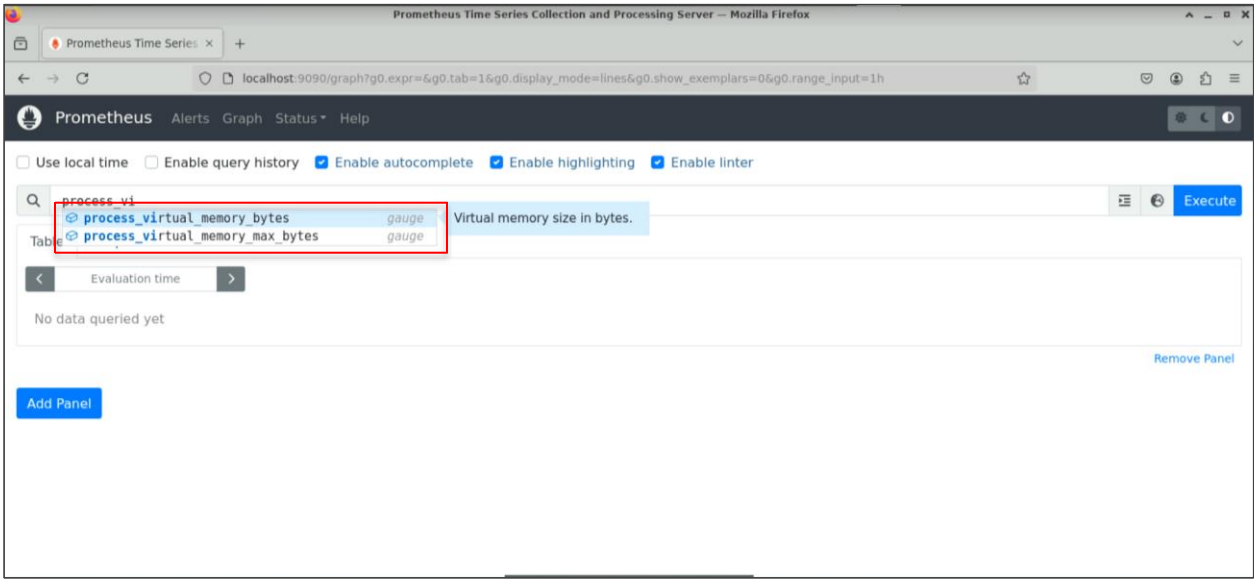
□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

Expression (press Shift+Enter for newlines)

No data queried yet

Note: The Prometheus server must run in the terminal to access the Prometheus UI.

2.2 Navigate to the **Expression Browser**, type **process_vi**, and then select the **process_virtual_memory_bytes** metric as shown



Prometheus Time Series Collection and Processing Server — Mozilla Firefox

localhost:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h

□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

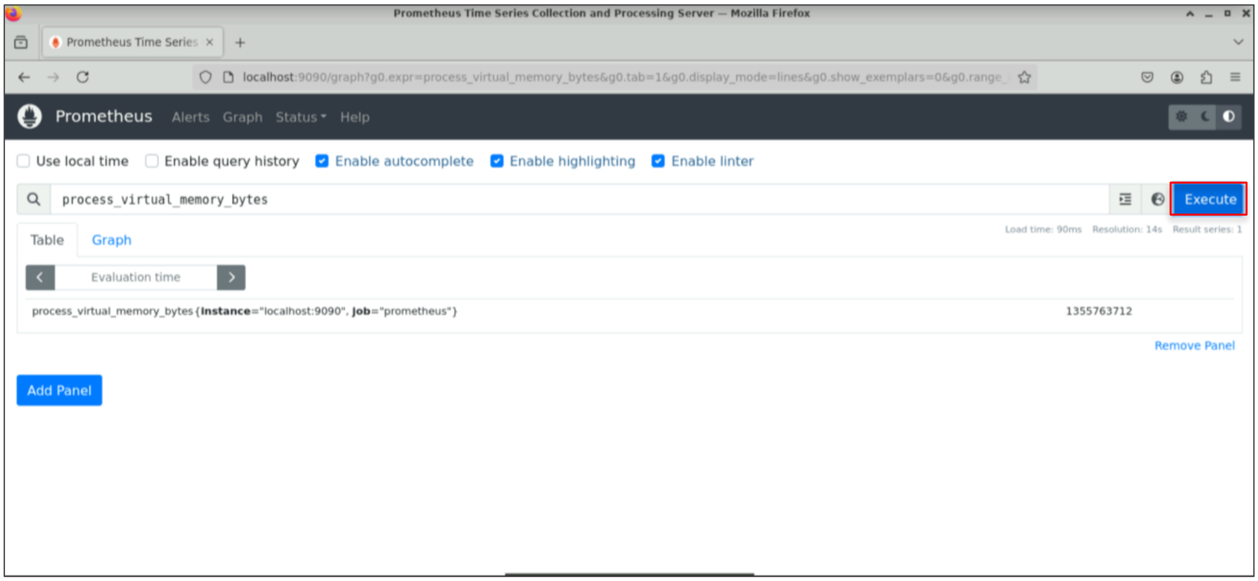
```
process_vi
process_virtual_memory_bytes       gauge
process_virtual_memory_max_bytes   gauge
```

Virtual memory size in bytes.

No data queried yet

.

2.3 Click on **Execute**



localhost:9090/graph?g0.expr=process_virtual_memory_bytes&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range

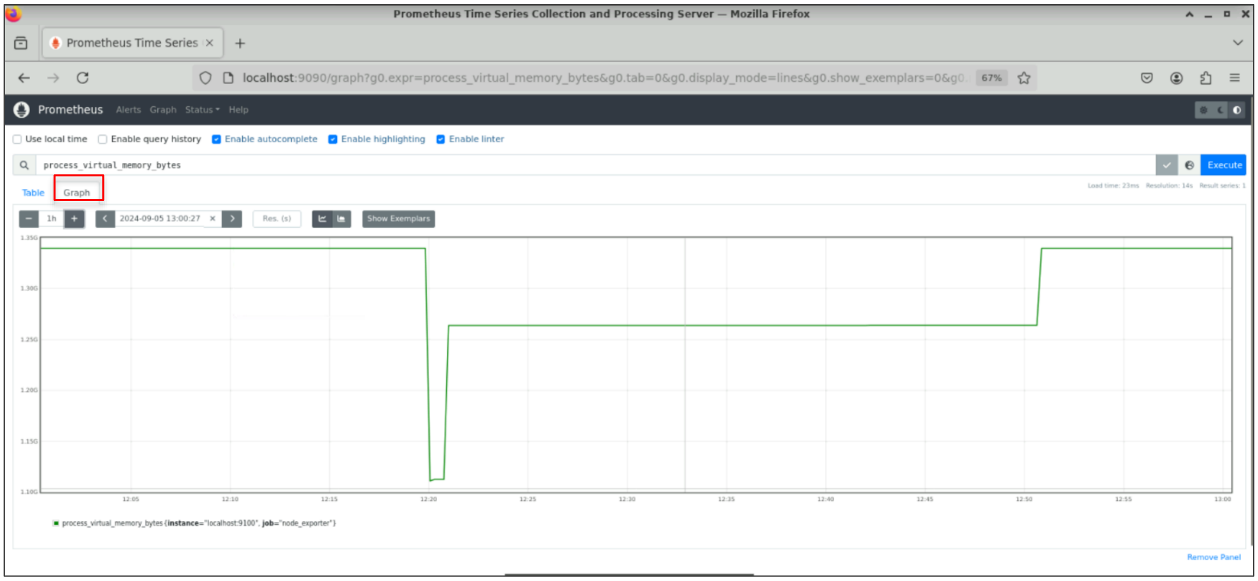
□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
process_virtual_memory_bytes
```

Load time: 90ms Resolution: 14s Result series: 1

process_virtual_memory_bytes{**instance**="localhost:9090", **job**="prometheus"} 1355763712

2.4 Click on the **Graph** tab to visualize **process_virtual_memory_bytes**



localhost:9090/graph?g0.expr=process_virtual_memory_bytes&g0.tab=0&g0.display_mode=lines&g0.show_exemplars=0&g0.

□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
process_virtual_memory_bytes
```

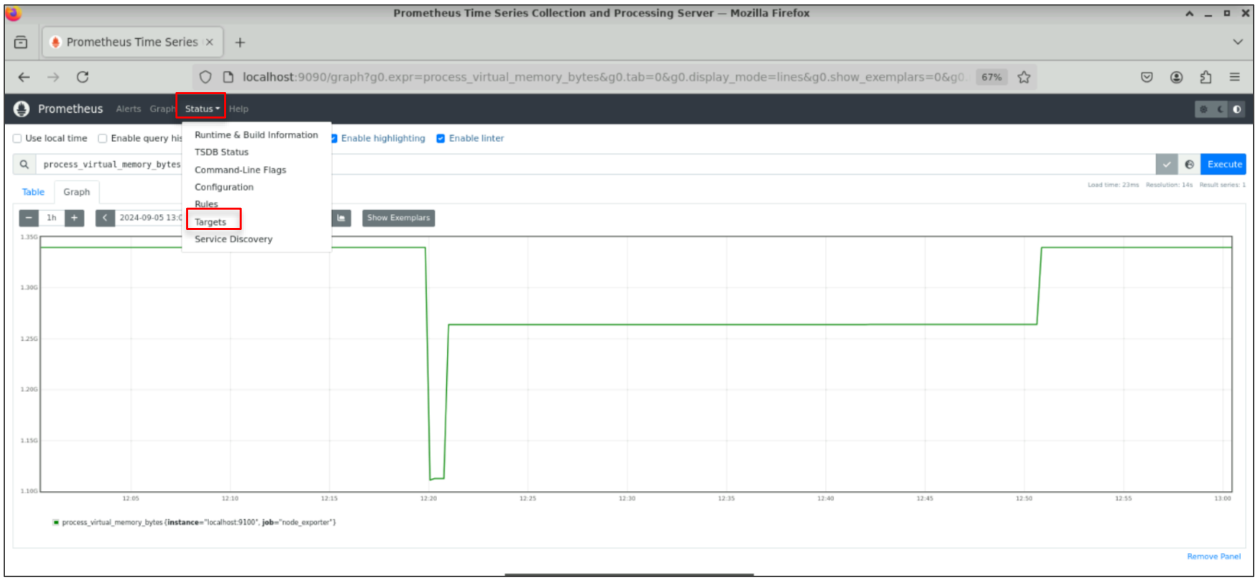
Load time: 23ms Resolution: 14s Result series: 1

2024-09-05 13:00:27

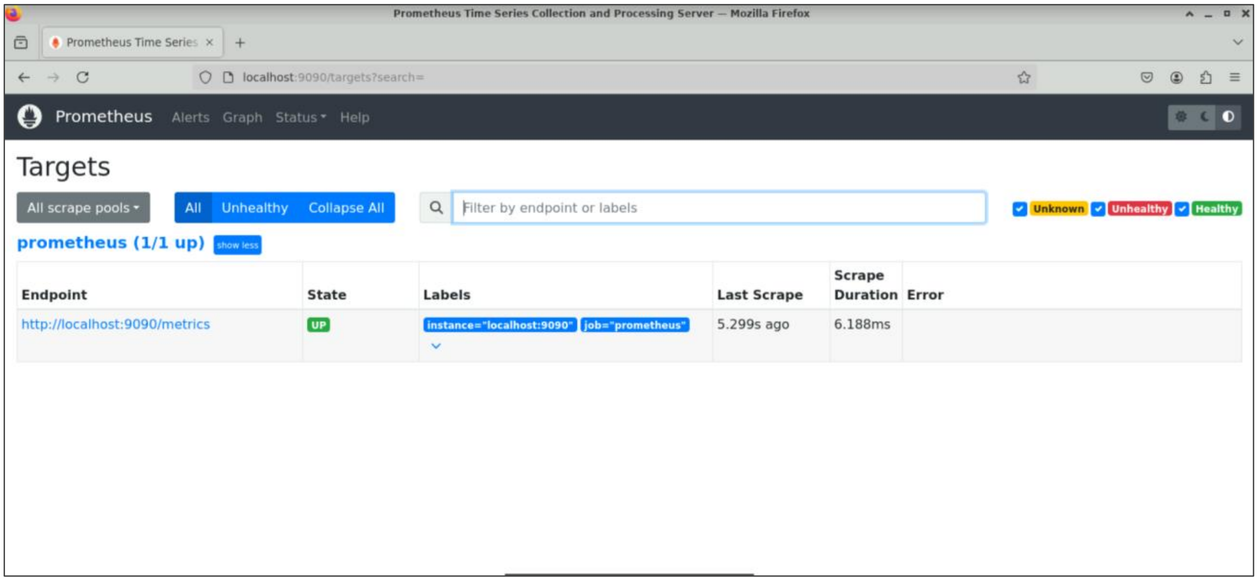
🟩 process_virtual_memory_bytes{**instance**="localhost:9100", **job**="node_exporter"}

.

2.5 Click on the **Status** section and select **Targets**


Prometheus Time Series Collection and Processing Server — Mozilla Firefox
Prometheus Time Series
localhost:9090/graph?g0.expr=process_virtual_memory_bytes&g0.tab=0&g0.display_mode=lines&g0.show_exemplars=0&g0
67%
Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query his
Enable highlighting
Enable linter
process_virtual_memory_bytes
Execute
Table
Graph
1h
2024-09-05 13:0
Show Exemplars
Runtime & Build Information
TSDB Status
Command-Line Flags
Configuration
Rules
Targets
Service Discovery
process_virtual_memory_bytes{instance="localhost:9100", job="node_exporter"}
Remove Panel

You will see the following interface:


Prometheus Time Series Collection and Processing Server — Mozilla Firefox
Prometheus Time Series
localhost:9090/targets?search=
Prometheus
Alerts
Graph
Status
Help

Targets

All scrape pools
All
Unhealthy
Collapse All
Filter by endpoint or labels
☑ Unknown
☑ Unhealthy
☑ Healthy

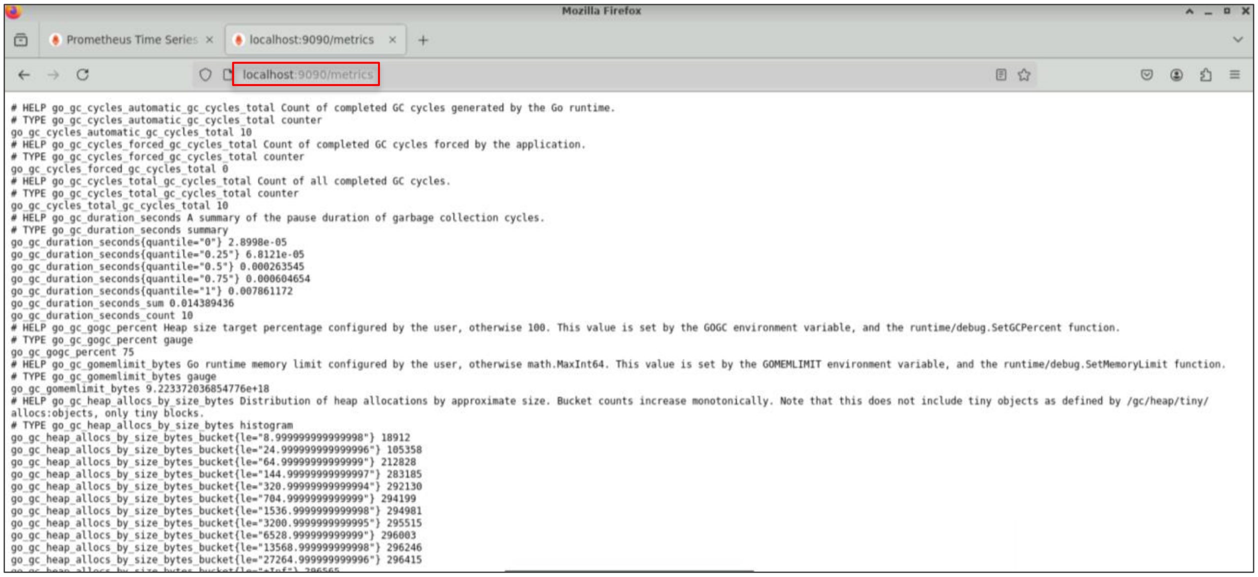
prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	5.299s ago	6.188ms	

Only a single Prometheus server is in the **UP** state on this page, indicating that the last scrape was successful. If there had been an issue with the previous scrape, an error message would appear in the **Error** field.

.

2.6 Copy the link **http://localhost:9090/metrics** from the **Targets** page and paste it into a new browser tab as shown:



```
# HELP go_gc_cycles_automatic_gc_cycles_total Count of completed GC cycles generated by the Go runtime.
# TYPE go_gc_cycles_automatic_gc_cycles_total counter
go_gc_cycles_automatic_gc_cycles_total 10
# HELP go_gc_cycles_forced_gc_cycles_total Count of completed GC cycles forced by the application.
# TYPE go_gc_cycles_forced_gc_cycles_total counter
go_gc_cycles_forced_gc_cycles_total 0
# HELP go_gc_cycles_total_gc_cycles_total Count of all completed GC cycles.
# TYPE go_gc_cycles_total_gc_cycles_total counter
go_gc_cycles_total_gc_cycles_total 10
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.8998e-05
go_gc_duration_seconds{quantile="0.25"} 6.8121e-05
go_gc_duration_seconds{quantile="0.5"} 0.000263545
go_gc_duration_seconds{quantile="0.75"} 0.000604654
go_gc_duration_seconds{quantile="1"} 0.007861172
go_gc_duration_seconds_sum 0.014389436
go_gc_duration_seconds_count 10
# HELP go_gc_gogc_percent Heap size target percentage configured by the user, otherwise 100. This value is set by the GOGC environment variable, and the runtime/debug.SetGCPercent function.
# TYPE go_gc_gogc_percent gauge
go_gc_gogc_percent 75
# HELP go_gc_gomemlimit_bytes Go runtime memory limit configured by the user, otherwise math.MaxInt64. This value is set by the GOMEMLIMIT environment variable, and the runtime/debug.SetMemoryLimit function.
# TYPE go_gc_gomemlimit_bytes gauge
go_gc_gomemlimit_bytes 9.223372036854776e+18
# HELP go_gc_heap_allocs_by_size_bytes Distribution of heap allocations by approximate size. Bucket counts increase monotonically. Note that this does not include tiny objects as defined by /gc/heap/tiny/
allocs:objects, only tiny blocks.
# TYPE go_gc_heap_allocs_by_size_bytes histogram
go_gc_heap_allocs_by_size_bytes_bucket{le="8.999999999999998"} 18912
go_gc_heap_allocs_by_size_bytes_bucket{le="24.999999999999996"} 105358
go_gc_heap_allocs_by_size_bytes_bucket{le="64.99999999999999"} 212828
go_gc_heap_allocs_by_size_bytes_bucket{le="144.99999999999997"} 283185
go_gc_heap_allocs_by_size_bytes_bucket{le="320.99999999999994"} 292130
go_gc_heap_allocs_by_size_bytes_bucket{le="704.9999999999999"} 294199
go_gc_heap_allocs_by_size_bytes_bucket{le="1536.9999999999998"} 294981
go_gc_heap_allocs_by_size_bytes_bucket{le="3200.9999999999995"} 295515
go_gc_heap_allocs_by_size_bytes_bucket{le="6528.999999999999"} 296003
go_gc_heap_allocs_by_size_bytes_bucket{le="13568.999999999998"} 296246
go_gc_heap_allocs_by_size_bytes_bucket{le="27264.999999999996"} 296415
go_gc_heap_allocs_by_size_bytes_bucket{le="+Inf"} 296565
```

This will display metrics for monitoring the application.

By following these steps, you have successfully configured Prometheus to scrape and visualize metrics for monitoring system performance through its web interface.