My github repository:

https://github.com/FahadUmair/nodeJS-todo-openTelemetry

About the App:

The app is written in Node.js and it is a simple todo app where you can add items, select, and remove then after you are done that item. The source code was taken from: https://github.com/missating/nodejs-todo

To run the app, simple type “node index.js”

To see the tracing, type “node --require ./tracing.js index.js”

To see the metrics, type “node --require ./instrumentation.js index.js”

Instrumenting with OpenTelemetry:

Tracing:

Each of the important functions have a tracer connected to them:

- Page load has tracer connected to it.
- Adding an item has a tracer with it.
- Selecting an item has a tracer with it.
- Removing an item has a tracer with it.

Now run the app with tracing which is

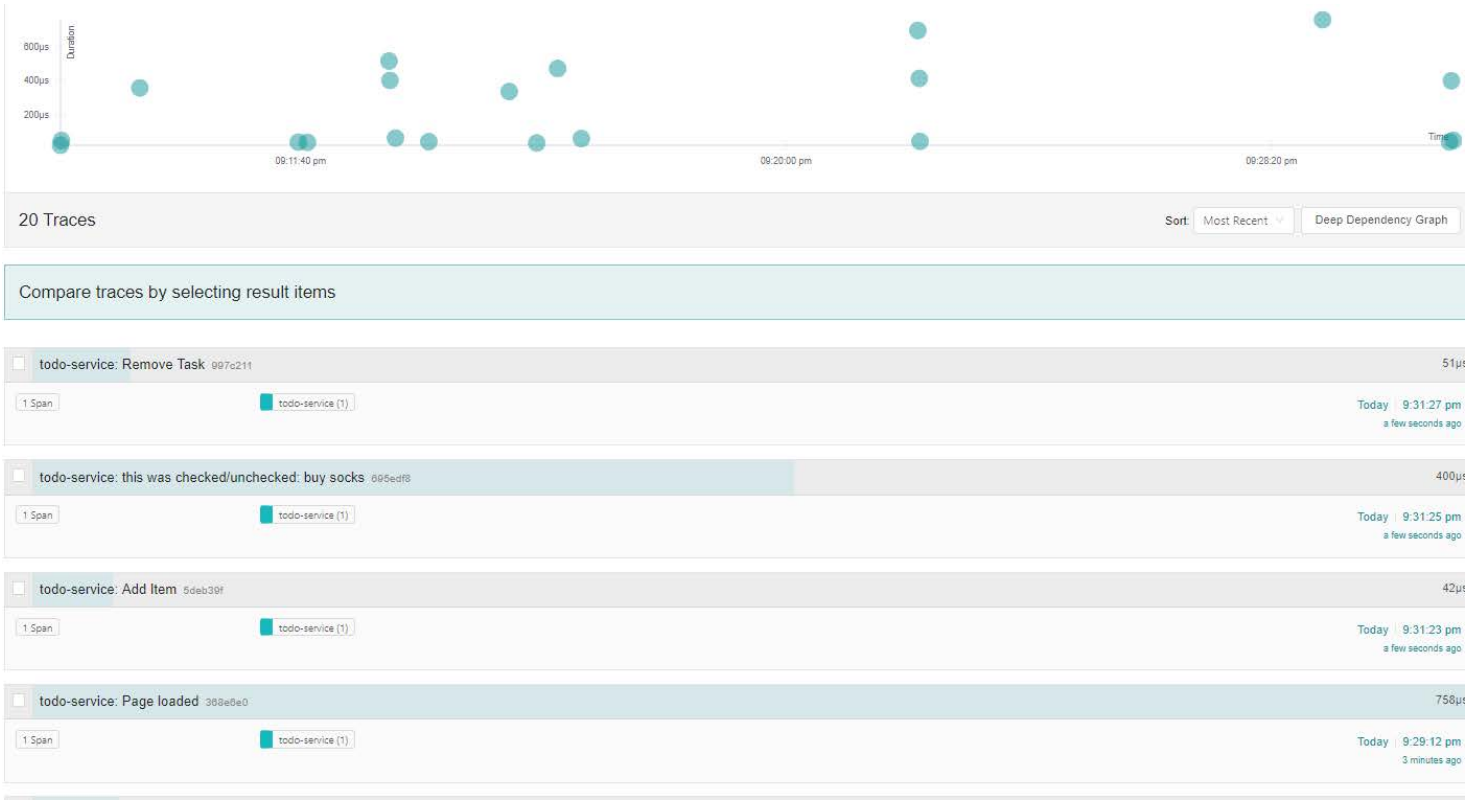
node --require ./tracing.js index.js

The page loads at localhost:3000

Let’s load the page, add an item, then select and then remove it:

Now go to **localhost:16686** to see jaeger’s visualization

This is what you’ll see there:



This clearly shows the traces of all the important functions of the app.

Metrics:

Now we use metrics to see the CPU utilization, counters, etc.

Run the app with this command:

node --require ./instrumentation.js index.js

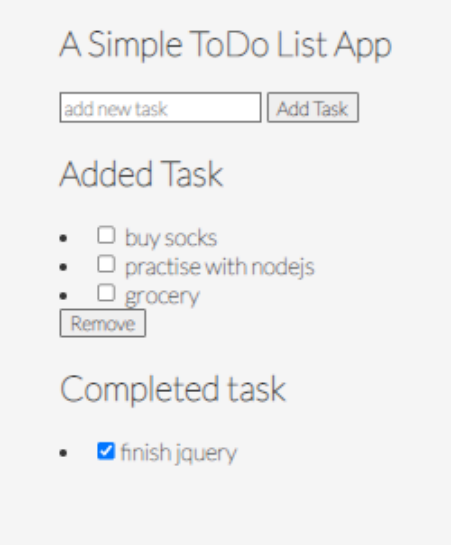
Now, you need to have Prometheus installed in order to see this,

We have three links now,

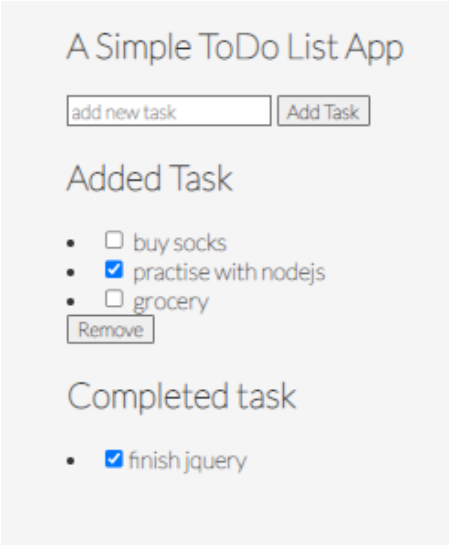
- **Localhost:3000** to run the app
- **Localhost: 9464/metrics** to see the metrics
- **Localhost:9090** to see the Prometheus's visualization for the metrics.
- Also run premetheus.exe inside the premetheus folder. Tou can run it by doing **./premetheus.exe**

Now we use counters and cpu_util to see the visualization:

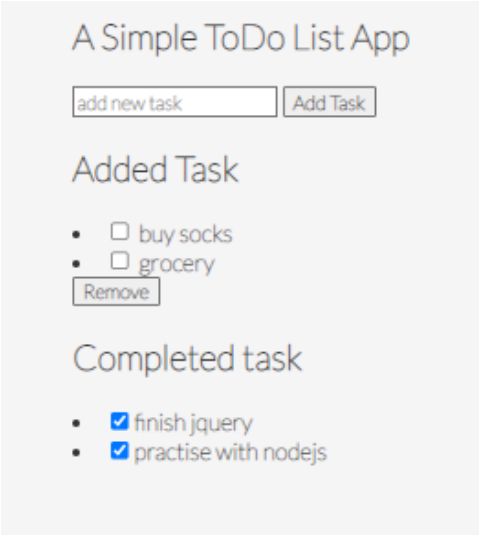
We add, select and remove items to test:



Item added “grocery”

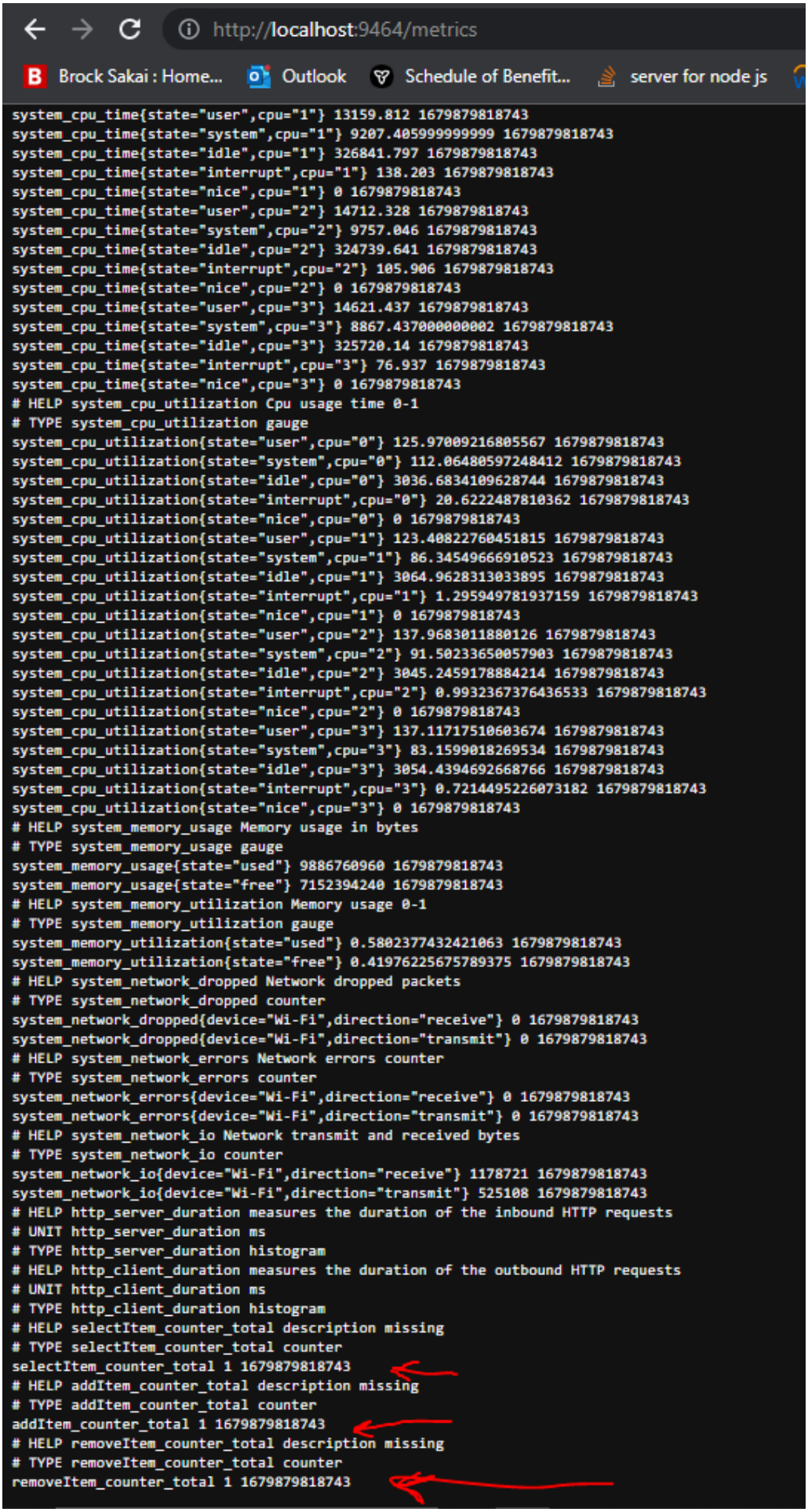


Item selected “practice with nodejs”



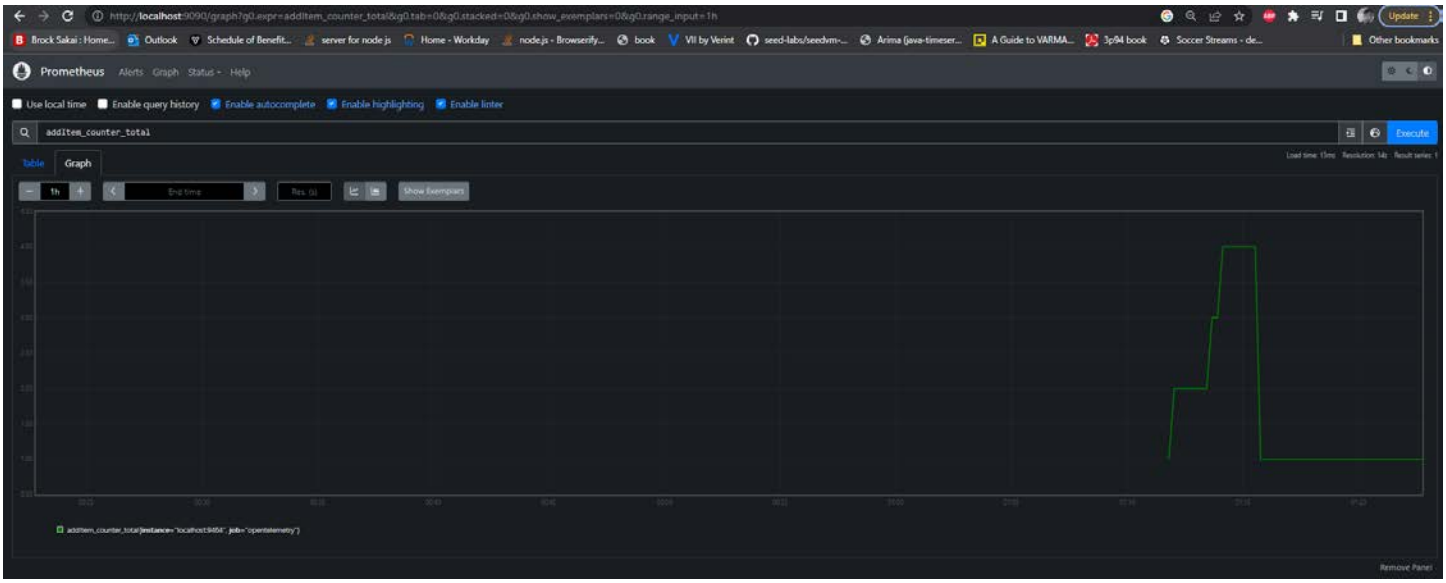
Remove button clicked

Now let’s go to **Localhost: 9464/metrics** to see the metrics:


http://localhost:9464/metrics
Brock Sakai : Home...
Outlook
Schedule of Benefit...
server for node js
system_cpu_time{state="user",cpu="1"} 13159.812 1679879818743
system_cpu_time{state="system",cpu="1"} 9207.405999999999 1679879818743
system_cpu_time{state="idle",cpu="1"} 326841.797 1679879818743
system_cpu_time{state="interrupt",cpu="1"} 138.203 1679879818743
system_cpu_time{state="nice",cpu="1"} 0 1679879818743
system_cpu_time{state="user",cpu="2"} 14712.328 1679879818743
system_cpu_time{state="system",cpu="2"} 9757.046 1679879818743
system_cpu_time{state="idle",cpu="2"} 324739.641 1679879818743
system_cpu_time{state="interrupt",cpu="2"} 105.906 1679879818743
system_cpu_time{state="nice",cpu="2"} 0 1679879818743
system_cpu_time{state="user",cpu="3"} 14621.437 1679879818743
system_cpu_time{state="system",cpu="3"} 8867.437000000002 1679879818743
system_cpu_time{state="idle",cpu="3"} 325720.14 1679879818743
system_cpu_time{state="interrupt",cpu="3"} 76.937 1679879818743
system_cpu_time{state="nice",cpu="3"} 0 1679879818743
HELP system_cpu_utilization Cpu usage time 0-1
TYPE system_cpu_utilization gauge
system_cpu_utilization{state="user",cpu="0"} 125.97009216805567 1679879818743
system_cpu_utilization{state="system",cpu="0"} 112.06480597248412 1679879818743
system_cpu_utilization{state="idle",cpu="0"} 3036.6834109628744 1679879818743
system_cpu_utilization{state="interrupt",cpu="0"} 20.6222487810362 1679879818743
system_cpu_utilization{state="nice",cpu="0"} 0 1679879818743
system_cpu_utilization{state="user",cpu="1"} 123.40822760451815 1679879818743
system_cpu_utilization{state="system",cpu="1"} 86.34549666910523 1679879818743
system_cpu_utilization{state="idle",cpu="1"} 3064.9628313033895 1679879818743
system_cpu_utilization{state="interrupt",cpu="1"} 1.295949781937159 1679879818743
system_cpu_utilization{state="nice",cpu="1"} 0 1679879818743
system_cpu_utilization{state="user",cpu="2"} 137.9683011880126 1679879818743
system_cpu_utilization{state="system",cpu="2"} 91.50233650057903 1679879818743
system_cpu_utilization{state="idle",cpu="2"} 3045.2459178884214 1679879818743
system_cpu_utilization{state="interrupt",cpu="2"} 0.9932367376436533 1679879818743
system_cpu_utilization{state="nice",cpu="2"} 0 1679879818743
system_cpu_utilization{state="user",cpu="3"} 137.11717510603674 1679879818743
system_cpu_utilization{state="system",cpu="3"} 83.1599018269534 1679879818743
system_cpu_utilization{state="idle",cpu="3"} 3054.4394692668766 1679879818743
system_cpu_utilization{state="interrupt",cpu="3"} 0.7214495226073182 1679879818743
system_cpu_utilization{state="nice",cpu="3"} 0 1679879818743
HELP system_memory_usage Memory usage in bytes
TYPE system_memory_usage gauge
system_memory_usage{state="used"} 9886760960 1679879818743
system_memory_usage{state="free"} 7152394240 1679879818743
HELP system_memory_utilization Memory usage 0-1
TYPE system_memory_utilization gauge
system_memory_utilization{state="used"} 0.5802377432421063 1679879818743
system_memory_utilization{state="free"} 0.41976225675789375 1679879818743
HELP system_network_dropped Network dropped packets
TYPE system_network_dropped counter
system_network_dropped{device="Wi-Fi",direction="receive"} 0 1679879818743
system_network_dropped{device="Wi-Fi",direction="transmit"} 0 1679879818743
HELP system_network_errors Network errors counter
TYPE system_network_errors counter
system_network_errors{device="Wi-Fi",direction="receive"} 0 1679879818743
system_network_errors{device="Wi-Fi",direction="transmit"} 0 1679879818743
HELP system_network_io Network transmit and received bytes
TYPE system_network_io counter
system_network_io{device="Wi-Fi",direction="receive"} 1178721 1679879818743
system_network_io{device="Wi-Fi",direction="transmit"} 525108 1679879818743
HELP http_server_duration measures the duration of the inbound HTTP requests
UNIT http_server_duration ms
TYPE http_server_duration histogram
HELP http_client_duration measures the duration of the outbound HTTP requests
UNIT http_client_duration ms
TYPE http_client_duration histogram
HELP selectItem_counter_total description missing
TYPE selectItem_counter_total counter
selectItem_counter_total 1 1679879818743
HELP addItem_counter_total description missing
TYPE addItem_counter_total counter
addItem_counter_total 1 1679879818743
HELP removeItem_counter_total description missing
TYPE removeItem_counter_total counter
removeItem_counter_total 1 1679879818743

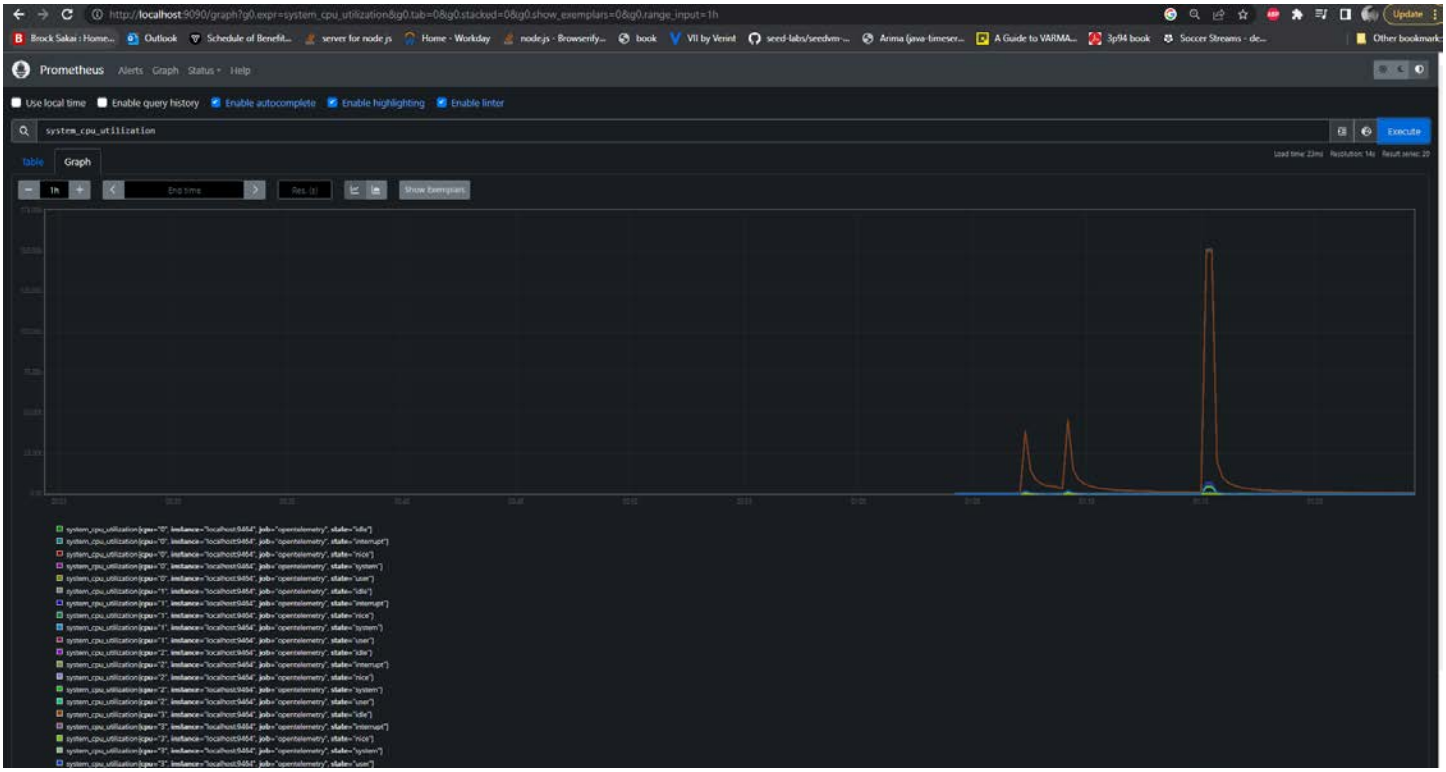
These are the counters working just fine.

Let’s look at one of the counters on Prometheus (we added 4 items):



So, we see the counter is at 4.

Now let’s look at the CPU utilization:



We see that adding, and removing creates a little spike in the utilization.

We can see all the other metrics in the same way as well.

to run jaeger through docker: docker run -d --name jaeger
-e COLLECTOR_ZIPKIN_HOST_PORT=:9411
-e COLLECTOR_OTLP_ENABLED=true
-p 6831:6831/udp
-p 6832:6832/udp
-p 5778:5778
-p 16686:16686
-p 4317:4317
-p 4318:4318
-p 14250:14250
-p 14268:14268
-p 14269:14269
-p 9411:9411
jaegertracing/all-in-one:latest