Sri Lanka Institute of Information Technology

Visual Analytics and User Experience Design

(IT4031)

4th Year 1st Semester

Assignment 2

Group ID: 2024_A2_G12

Gunawardana K.P - IT20298876

Wickramarathna N.A.N.D -IT20658540

Maddumage P.W - IT21007538

Hewage R.P - IT21054686

Kiriella K.G.A.K - IT21035876

Table of Contents

Table of Contents 2
List of Tables 3
Project Links 1
Introduction 2
Architecture Diagram 3
Docker Hub configurations, Docker hub repository 4
5
Configure Prometheus 7
  Installing Prometheus 7
Retrieving Metrics 8
Grafana Dashboard 14
  Install and Configure Grafana on Docker 14
Used Metrics 15
Member Contributions to the Project 17
References 18

Table of Figures

Figure 1: System Architecture 3
Figure 2: Prometheus monitoring targets 10
Figure 3: Node Exporter metrics 11
Figure 4: nodeJS metrics 11
Figure 5: Prometheus metrics 12
Figure 6: Grafana Dashboard Creation 14
Figure 7: Setting Prometheus as Data source 14
Figure 8: Grafana Dashboard Visualizing Metrics 18
Figure 9: Grafana Dashboard Visualizing System Metrics 18
Figure 10: Grafana Dashboard Visualizing Prometheus Metrics 19

List of Tables

Table 1: Metrics 15
Table 2: Prometheus Metrics 16

Project Links

Below links can be used to access the project implementation of the group.

- Node exporter metrics – http://192.168.56.1:8000/metrics
- Prometheus - http://192.168.56.1:9090/
- Grafana – http://192.168.56.1:3000/dashboards
- Dashboard - http://192.168.56.1:3000/d/PTSqcpJWk/nodejs-application-dashboard?orgId=1
- Project folder drive link - https://mysliit-my.sharepoint.com/personal/it21035876_my_sliit_lk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fit21035876%5Fmy%5Fsliit%5Flk%2FDocuments%2FVaued&ga=1
- Drive link- https://drive.google.com/drive/folders/1aTEQbLwYolfWCAyMq4yothyGpIMLRXJx?usp=sharing

Introduction

Using Prometheus and Grafana, TechVista, a dynamic tech team known for their creative solutions, led the construction of a state-of-the-art metrics monitoring and visualization system. Their main objective? To revolutionize the analysis and monitoring of key performance indicators by capturing metrics from apps and displaying them through immersive dashboards, thus unleashing the potential of data-driven insights.

A watchful Prometheus server, a potent Grafana server, and an application that interacted fluidly with outside exporters formed the core of their implemented system. Carefully designed, the application provided metrics compatible with Prometheus, and Prometheus collected and stored these metrics for periodic examination. Grafana, the master of visualization, used these metrics to create dynamic dashboards that gave the data life.

TechVista, however, didn't stop there. Proactive alerting methods were integrated into Grafana to guarantee prompt warnings of any anomalies or problems in performance. Their methodical approach also included recording setups, deployment procedures, and problem-solving techniques, producing an invaluable guide for upcoming projects.

The use of Docker for containerization was later made public, demonstrating TechVista's dedication to innovation. They demonstrated the system's scalability and mobility using Docker, opening countless possibilities.

Ultimately, TechVista gained priceless insights into application performance using Prometheus and Grafana in the deployment of the metrics monitoring and visualization system. The way in which these technologies were seamlessly integrated demonstrated the revolutionary potential of metrics-driven monitoring and visualization. This report acts as a guide for you at each turn in their journey.

Architecture Diagram

- **Objective** - Implement a dashboard using Grafana to visualize the metrics of NodeJS application, captured from Prometheus:
- **Application** – NodeJS
- **Language Used** - NodeJS
- **Exporters** – Node Exporter

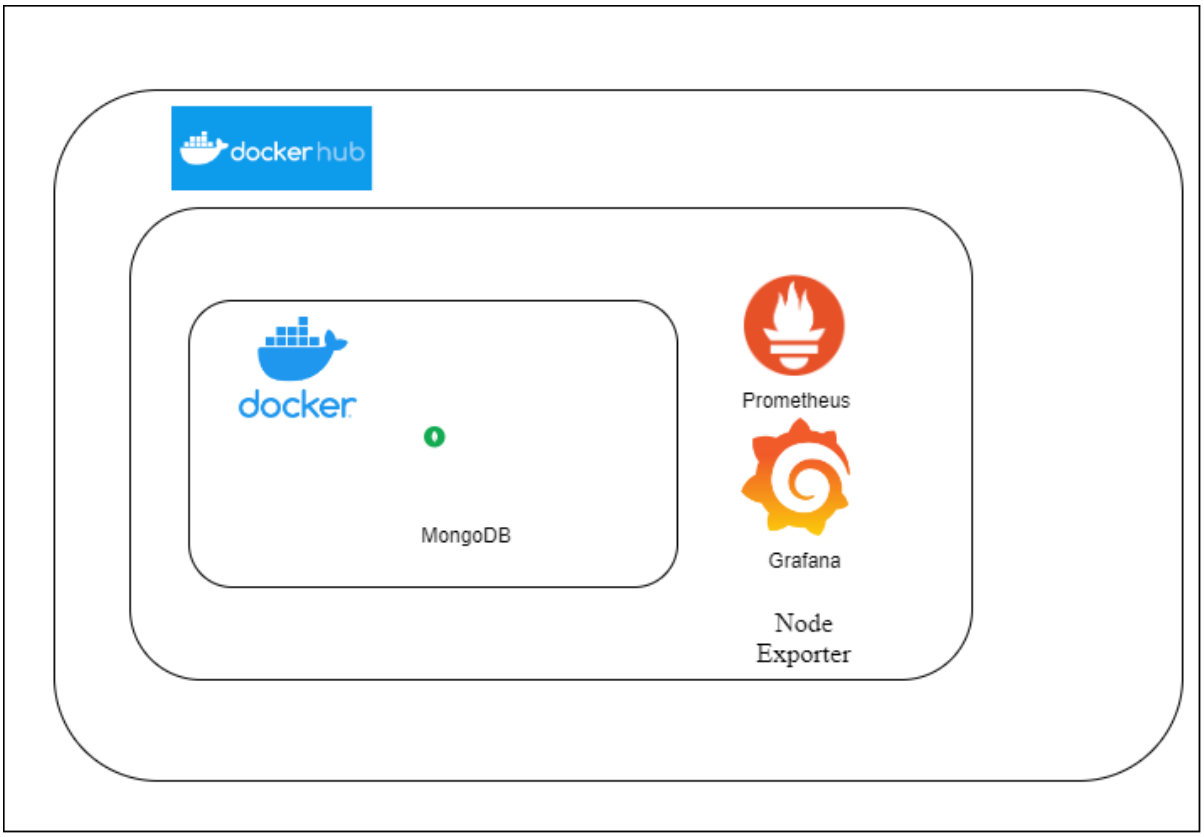
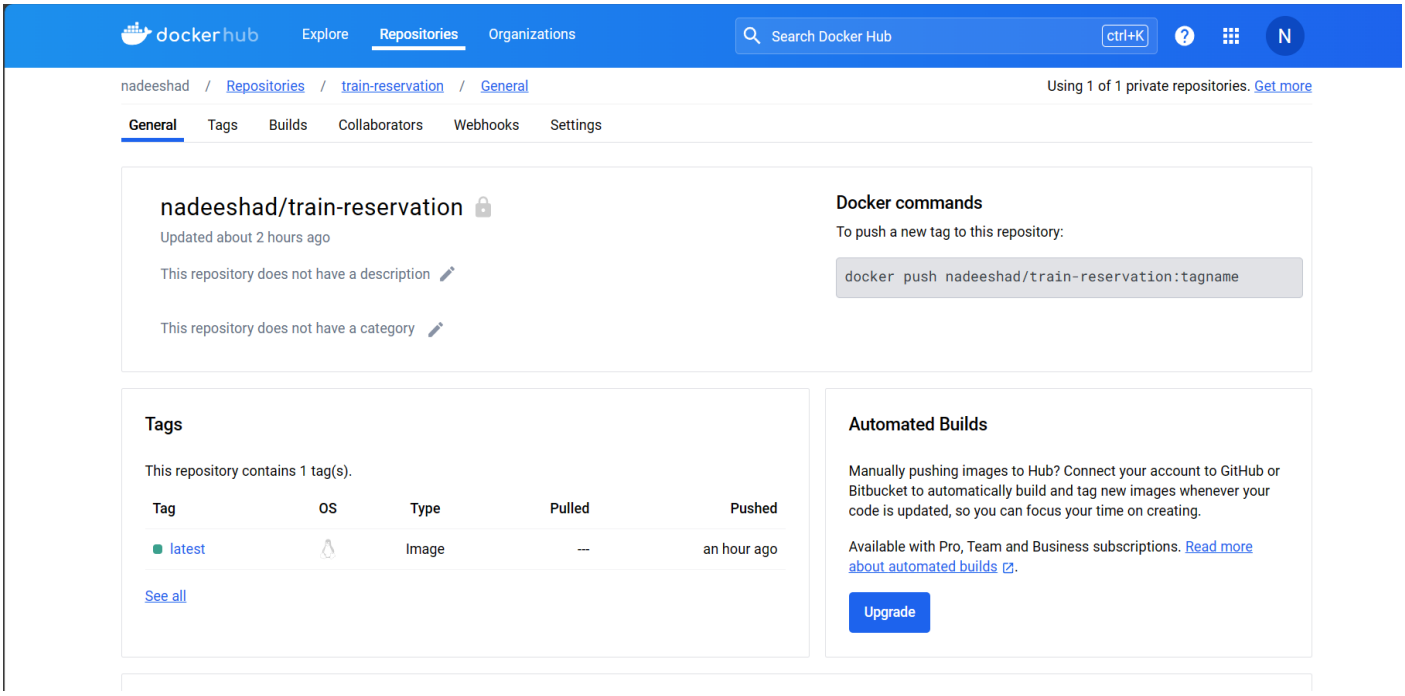
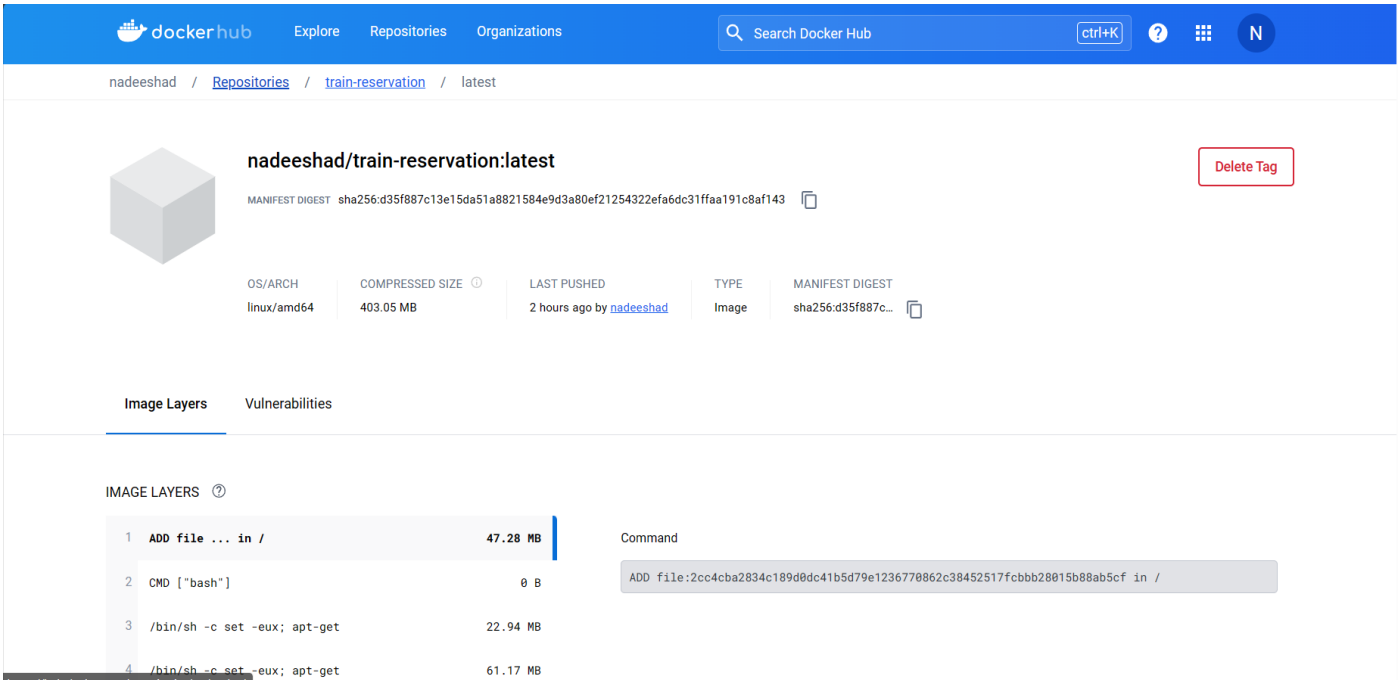


Figure 1: System Architecture

Docker Hub configurations, Docker hub repository





```
PS D:\SLIIT\SLIIT Y4S2\VAUED New sem\project\a-Nadeesha-Project\a-Nadeesha-Project> docker compose up
time="2024-05-09T23:51:17+05:30" level=warning msg="D:\\SLIIT\\SLIIT Y4S2\\VAUED New sem\\project\\a-Nadeesha-Project\\a-Nadeesha-Project\\docker-co
mpose.yml: `version` is obsolete"
[+] Running 1/0
 ✓ Container a-nadeesha-project-prom-server-1  Created
 0.0s
Attaching to prom-server-1
prom-server-1  | ts=2024-05-09T18:21:17.882Z caller=main.go:573 level=info msg="No time or size retention was set so using the default time retentio
n" duration=15d
prom-server-1  | ts=2024-05-09T18:21:17.882Z caller=main.go:617 level=info msg="Starting Prometheus Server" mode=server version="(version=2.52.0, br
anch=HEAD, revision=879d80922a227c37df502e7315fad8ceb10a986d)"
prom-server-1  | ts=2024-05-09T18:21:17.882Z caller=main.go:622 level=info build_context="(go=go1.22.3, platform=linux/amd64, user=root@1b4f4c206e41
, date=20240508-21:56:43, tags=netgo,builtinassets,stringlabels)"
prom-server-1  | ts=2024-05-09T18:21:17.882Z caller=main.go:623 level=info host_details="(Linux 5.15.146.1-microsoft-standard-WSL2 #1 SMP Thu Jan 11
 04:09:03 UTC 2024 x86_64 c357ef3b0293 )"
prom-server-1  | ts=2024-05-09T18:21:17.882Z caller=main.go:624 level=info fd_limits="(soft=1048576, hard=1048576)"
prom-server-1  | ts=2024-05-09T18:21:17.882Z caller=main.go:625 level=info vm_limits="(soft=unlimited, hard=unlimited)"
prom-server-1  | ts=2024-05-09T18:21:17.887Z caller=web.go:568 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
prom-server-1  | ts=2024-05-09T18:21:17.882Z caller=main.go:625 level=info vm_limits="(soft=unlimited, hard=unlimited)"
prom-server-1  | ts=2024-05-09T18:21:17.887Z caller=web.go:568 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
prom-server-1  | ts=2024-05-09T18:21:17.889Z caller=main.go:1129 level=info msg="Starting TSDB ..."
prom-server-1  | ts=2024-05-09T18:21:17.897Z caller=tls_config.go:313 level=info component=web msg="Listening on" address=[::]:9090
prom-server-1  | ts=2024-05-09T18:21:17.898Z caller=tls_config.go:316 level=info component=web msg="TLS is disabled." http2=false address=[::]:9090
prom-server-1  | ts=2024-05-09T18:21:17.900Z caller=head.go:616 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
```

```
prom-server-1  | ts=2024-05-09T18:21:17.900Z caller=head.go:711 level=info component=tsdb msg="Replaying WAL, this may take a while"
prom-server-1  | ts=2024-05-09T18:21:17.902Z caller=head.go:783 level=info component=tsdb msg="WAL segment loaded" segment=0 maxSegment=1
prom-server-1  | ts=2024-05-09T18:21:17.908Z caller=head.go:783 level=info component=tsdb msg="WAL segment loaded" segment=1 maxSegment=1
prom-server-1  | ts=2024-05-09T18:21:17.908Z caller=head.go:820 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=43.2
83µs wal_replay_duration=7.673026ms wbl_replay_duration=271ns chunk_snapshot_load_duration=0s mmap_chunk_replay_duration=26.401µs total_replay_durat
ion=7.776906ms
prom-server-1  | ts=2024-05-09T18:21:17.919Z caller=main.go:1150 level=info fs_type=EXT4_SUPER_MAGIC
prom-server-1  | ts=2024-05-09T18:21:17.919Z caller=main.go:1153 level=info msg="TSDB started"
prom-server-1  | ts=2024-05-09T18:21:17.919Z caller=main.go:1335 level=info msg="Loading configuration file" filename=/etc/prometheus/prometheus.yml

prom-server-1  | ts=2024-05-09T18:21:17.928Z caller=main.go:1372 level=info msg="Completed loading of configuration file" filename=/etc/prometheus/p
rometheus.yml totalDuration=8.817729ms db_storage=10.941µs remote_storage=6.442µs web_handler=2.696µs query_engine=5.52µs scrape=927.22µs scrape_sd=
103.81µs notify=2.986µs notify_sd=1.814µs rules=4.679µs tracing=87.12µs
prom-server-1  | ts=2024-05-09T18:21:17.928Z caller=main.go:1114 level=info msg="Server is ready to receive web requests."
prom-server-1  | ts=2024-05-09T18:21:17.928Z caller=manager.go:163 level=info component="rule manager" msg="Starting rule manager..."
prom-server-1  | ts=2024-05-09T23:50:45.640Z caller=compact.go:576 level=info component=tsdb msg="write block" mint=1715278726775 maxt=1715284800000
 ulid=01HXFW8DEQEYP0NY0MHDFWVWEG duration=176.446562ms ooo=false
.320075ms
prom-server-1  | ts=2024-05-09T23:50:45.728Z caller=compact.go:576 level=info component=tsdb msg="write block" mint=1715284800996 maxt=1715292000000
 ulid=01HXFW8DMZHG3S4668Z1D7S3BM duration=64.547056ms ooo=false
prom-server-1  | ts=2024-05-09T23:50:45.732Z caller=head.go:1345 level=info component=tsdb msg="Head GC completed" caller=truncateMemory duration=2.
168982ms
prom-server-1  | ts=2024-05-10T17:04:44.238Z caller=compact.go:576 level=info component=tsdb msg="write block" mint=1715295045442 maxt=1715299200000
 ulid=01HXHQDNQ5Z9G27X9H3VJHW554 duration=488.739305ms ooo=false
prom-server-1  | ts=2024-05-10T17:04:44.274Z caller=head.go:1345 level=info component=tsdb msg="Head GC completed" caller=truncateMemory duration=33
```

```
PS D:\SLIIT\SLIIT Y4S2\VAUED New sem\project\a-Nadeesha-Project\a-Nadeesha-Project> docker build -t nadeeshad/train-reservation:latest
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Start a build
```

```
PS D:\SLIIT\SLIIT Y4S2\VAUED New sem\project\a-Nadeesha-Project\a-Nadeesha-Project> docker build -t nadeeshad/train-reservation .
[+] Building 0.0s (0/0)  docker:default
[+] Building 3.7s (12/12) FINISHED                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 434B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/node:20                                                                    2.8s
 => [auth] library/node:pull token for registry-1.docker.io                                                                   0.0s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [1/6] FROM docker.io/library/node:20@sha256:5e362bbb5ef4c6f6e2c86a27b7269b3b3e4bd8dba16be18037ee7ee4caa8afc1              0.0s
 => [internal] load build context                                                                                             0.8s
 => => transferring context: 226.10kB                                                                                         0.7s
 => CACHED [2/6] WORKDIR /app                                                                                                 0.0s
 => CACHED [3/6] COPY package*.json ./                                                                                        0.0s
 => CACHED [4/6] RUN npm install                                                                                              0.0s
 => CACHED [5/6] COPY . .                                                                                                     0.0s
 => CACHED [6/6] RUN npm rebuild bcrypt --build-from-source                                                                   0.0s
 => exporting to image                                                                                                        0.0s
 => => exporting layers                                                                                                       0.0s
 => => writing image sha256:465b7e167aba12cbf88e86f7e28484d014cdf44e9606d454ee43922507ac4a85                                  0.0s
 => => naming to docker.io/nadeeshad/train-reservation                                                                        0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

```
PS D:\SLIIT\SLIIT Y4S2\VAUED New sem\project\a-Nadeesha-Project\a-Nadeesha-Project> docker images
REPOSITORY                  TAG       IMAGE ID       CREATED          SIZE
train-reservation           latest    465b7e167aba   23 minutes ago   1.19GB
nadeeshad/train-reservation   latest    465b7e167aba   23 minutes ago   1.19GB
prom/prometheus             latest    ecb74a3b23a9   43 hours ago     272MB
grafana/grafana-oss         latest    679e4be9f918   4 weeks ago      429MB
```

```
PS D:\SLIIT\SLIIT Y4S2\VAUED New sem\project\a-Nadeesha-Project\a-Nadeesha-Project> docker push nadeeshad/train-reservation
Using default tag: latest
The push refers to repository [docker.io/nadeeshad/train-reservation]
368369d5ac3f: Pushed
f2c14bd043c3: Pushed
272c8a265eb3: Pushed
94244c76e661: Pushed
5b0e5f10feba: Mounted from library/node
1b19565527ac: Mounted from library/node
c5dad4bd965d: Mounted from library/node
6309473771d3: Mounted from library/node
83db175c22e2: Mounted from library/node
c5d13b2949a2: Mounted from library/node
7e43f593c900: Mounted from library/node
072686bcd3db: Mounted from library/node
latest: digest: sha256:d35f887c13e15da51a8821584e9d3a80ef21254322efa6dc31ffaa191c8af143 size: 3053
PS D:\SLIIT\SLIIT Y4S2\VAUED New sem\project\a-Nadeesha-Project\a-Nadeesha-Project>
 *  History restored

PS D:\SLIIT\SLIIT Y4S2\VAUED New sem\project\a-Nadeesha-Project\a-Nadeesha-Project>
```

Configure Prometheus

Installing Prometheus

npm i prom-client

Docker compose prom

docker-compose.yml

```yaml
version: "3"

services:
  prom-server:
    image: prom/prometheus
    ports:
      - 9090:9090
    volumes:
      - ./prometheus-config.yml:/etc/prometheus/prometheus.yml
```

Prom-configuration.yml

prometheus-config.yml

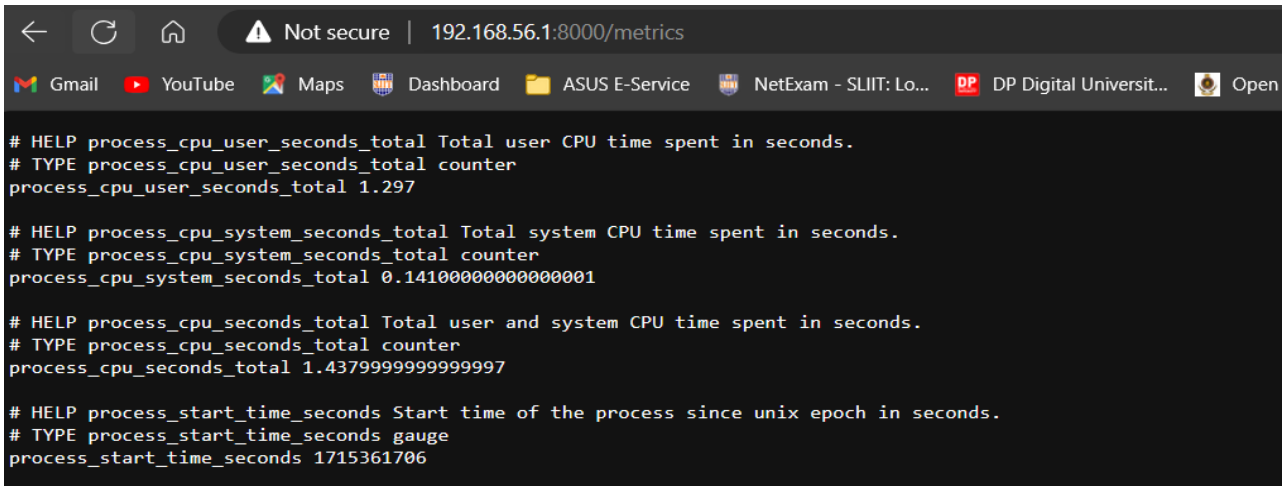
```yaml
global:
  scrape_interval: 4s

scrape_configs:
  - job_name: prometheus
    static_configs:
      - targets: ["192.168.56.1:8000"]
```

Retrieving Metrics

- Metrics can be retrieved as follows.

Metrics ss




```
# HELP process_cpu_user_seconds_total Total user CPU time spent in seconds.
# TYPE process_cpu_user_seconds_total counter
process_cpu_user_seconds_total 1.297

# HELP process_cpu_system_seconds_total Total system CPU time spent in seconds.
# TYPE process_cpu_system_seconds_total counter
process_cpu_system_seconds_total 0.14100000000000001

# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 1.4379999999999997

# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1715361706

# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 42024960

# HELP nodejs_eventloop_lag_seconds Lag of event loop in seconds.
# TYPE nodejs_eventloop_lag_seconds gauge
nodejs_eventloop_lag_seconds 0.001523

# HELP nodejs_eventloop_lag_min_seconds The minimum recorded event loop delay.
# TYPE nodejs_eventloop_lag_min_seconds gauge
nodejs_eventloop_lag_min_seconds 0.014843904

# HELP nodejs_eventloop_lag_max_seconds The maximum recorded event loop delay.
# TYPE nodejs_eventloop_lag_max_seconds gauge
nodejs_eventloop_lag_max_seconds 0.016482303

# HELP nodejs_eventloop_lag_mean_seconds The mean of the recorded event loop delays.
# TYPE nodejs_eventloop_lag_mean_seconds gauge
nodejs_eventloop_lag_mean_seconds 0.015583834352941176

# HELP nodejs_eventloop_lag_stddev_seconds The standard deviation of the recorded event loop delays.
# TYPE nodejs_eventloop_lag_stddev_seconds gauge
nodejs_eventloop_lag_stddev_seconds 0.00034254259026803246

# HELP nodejs_eventloop_lag_p50_seconds The 50th percentile of the recorded event loop delays.
# TYPE nodejs_eventloop_lag_p50_seconds gauge
nodejs_eventloop_lag_p50_seconds 0.015589375
```

```
# HELP nodejs_eventloop_lag_stddev_seconds The standard deviation of the recorded event loop delays.
# TYPE nodejs_eventloop_lag_stddev_seconds gauge
nodejs_eventloop_lag_stddev_seconds 0.00034254259026803246

# HELP nodejs_eventloop_lag_p50_seconds The 50th percentile of the recorded event loop delays.
# TYPE nodejs_eventloop_lag_p50_seconds gauge
nodejs_eventloop_lag_p50_seconds 0.015589375

# HELP nodejs_eventloop_lag_p90_seconds The 90th percentile of the recorded event loop delays.
# TYPE nodejs_eventloop_lag_p90_seconds gauge
nodejs_eventloop_lag_p90_seconds 0.016064511

# HELP nodejs_eventloop_lag_p99_seconds The 99th percentile of the recorded event loop delays.
# TYPE nodejs_eventloop_lag_p99_seconds gauge
nodejs_eventloop_lag_p99_seconds 0.016424959

# HELP nodejs_active_resources Number of active resources that are currently keeping the event loop alive, grouped by async resource type.
# TYPE nodejs_active_resources gauge
nodejs_active_resources{type="TTYWrap"} 2
nodejs_active_resources{type="PipeWrap"} 1
nodejs_active_resources{type="TCPServerWrap"} 1
nodejs_active_resources{type="TCPSocketWrap"} 13
nodejs_active_resources{type="Timeout"} 3
nodejs_active_resources{type="Immediate"} 1

# HELP nodejs_active_resources_total Total number of active resources.
# TYPE nodejs_active_resources_total gauge
nodejs_active_resources_total 21

# HELP nodejs_active_handles Number of active libuv handles grouped by handle type. Every handle type is C++ class name.
# TYPE nodejs_active_handles gauge
nodejs_active_handles{type="WriteStream"} 2
nodejs_active_handles{type="Socket"} 3
nodejs_active_handles{type="Server"} 1
nodejs_active_handles{type="TLSSocket"} 11

# HELP nodejs_active_handles_total Total number of active handles.
# TYPE nodejs_active_handles_total gauge
nodejs_active_handles_total 17

# HELP nodejs_active_requests Number of active libuv requests grouped by request type. Every request type is C++ class name.
# TYPE nodejs_active_requests gauge
```

```
# HELP nodejs_active_handles_total Total number of active handles.
# TYPE nodejs_active_handles_total gauge
nodejs_active_handles_total 17

# HELP nodejs_active_requests Number of active libuv requests grouped by request type. Every request type is C++ class name.
# TYPE nodejs_active_requests gauge

# HELP nodejs_active_requests_total Total number of active requests.
# TYPE nodejs_active_requests_total gauge
nodejs_active_requests_total 0

# HELP nodejs_heap_size_total_bytes Process heap size from Node.js in bytes.
# TYPE nodejs_heap_size_total_bytes gauge
nodejs_heap_size_total_bytes 33366016

# HELP nodejs_heap_size_used_bytes Process heap size used from Node.js in bytes.
# TYPE nodejs_heap_size_used_bytes gauge
nodejs_heap_size_used_bytes 26314440

# HELP nodejs_external_memory_bytes Node.js external memory size in bytes.
# TYPE nodejs_external_memory_bytes gauge
nodejs_external_memory_bytes 19713812

# HELP nodejs_heap_space_size_total_bytes Process heap space size total from Node.js in bytes.
# TYPE nodejs_heap_space_size_total_bytes gauge
nodejs_heap_space_size_total_bytes{space="read_only"} 0
nodejs_heap_space_size_total_bytes{space="old"} 25661440
nodejs_heap_space_size_total_bytes{space="code"} 2568192
nodejs_heap_space_size_total_bytes{space="map"} 1318912
nodejs_heap_space_size_total_bytes{space="large_object"} 2768896
nodejs_heap_space_size_total_bytes{space="code_large_object"} 0
nodejs_heap_space_size_total_bytes{space="new_large_object"} 0
nodejs_heap_space_size_total_bytes{space="new"} 1048576

# HELP nodejs_heap_space_size_used_bytes Process heap space size used from Node.js in bytes.
# TYPE nodejs_heap_space_size_used_bytes gauge
nodejs_heap_space_size_used_bytes{space="read_only"} 0
nodejs_heap_space_size_used_bytes{space="old"} 19769392
nodejs_heap_space_size_used_bytes{space="code"} 2214720
nodejs_heap_space_size_used_bytes{space="map"} 942768
nodejs_heap_space_size_used_bytes{space="large_object"} 2685440
nodejs_heap_space_size_used_bytes{space="code_large_object"} 0
nodejs_heap_space_size_used_bytes{space="new_large_object"} 0
nodejs_heap_space_size_used_bytes{space="new"} 706528
```

```
nodejs_heap_space_size_available_bytes{space="code"} 189632
nodejs_heap_space_size_available_bytes{space="map"} 349744
nodejs_heap_space_size_available_bytes{space="large_object"} 0
nodejs_heap_space_size_available_bytes{space="code_large_object"} 0
nodejs_heap_space_size_available_bytes{space="new_large_object"} 1030976
nodejs_heap_space_size_available_bytes{space="new"} 324448

# HELP nodejs_version_info Node.js version info.
# TYPE nodejs_version_info gauge
nodejs_version_info{version="v18.18.2",major="18",minor="18",patch="2"} 1

# HELP nodejs_gc_duration_seconds Garbage collection duration by kind, one of major, minor, incremental or weakcb.
# TYPE nodejs_gc_duration_seconds histogram
nodejs_gc_duration_seconds_bucket{le="0.001",kind="minor"} 190
nodejs_gc_duration_seconds_bucket{le="0.01",kind="minor"} 281
nodejs_gc_duration_seconds_bucket{le="0.1",kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="1",kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="2",kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="5",kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="+Inf",kind="minor"} 283
nodejs_gc_duration_seconds_sum{kind="minor"} 0.3479883999973535
nodejs_gc_duration_seconds_count{kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="0.001",kind="incremental"} 7
nodejs_gc_duration_seconds_bucket{le="0.01",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="0.1",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="1",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="2",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="5",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="+Inf",kind="incremental"} 8
nodejs_gc_duration_seconds_sum{kind="incremental"} 0.00986159999668598
nodejs_gc_duration_seconds_count{kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="0.001",kind="major"} 0
nodejs_gc_duration_seconds_bucket{le="0.01",kind="major"} 3
nodejs_gc_duration_seconds_bucket{le="0.1",kind="major"} 4
nodejs_gc_duration_seconds_bucket{le="1",kind="major"} 4
nodejs_gc_duration_seconds_bucket{le="2",kind="major"} 4
nodejs_gc_duration_seconds_bucket{le="5",kind="major"} 4
nodejs_gc_duration_seconds_bucket{le="+Inf",kind="major"} 4
nodejs_gc_duration_seconds_sum{kind="major"} 0.024481199994683265
nodejs_gc_duration_seconds_count{kind="major"} 4

# HELP http_express_req_res_time This tells how much tme is taken by req and res
# TYPE http_express_req_res_time histogram
http_express_req_res_time_bucket{le="1",method="GET",route="/",status_code="304"} 0
http_express_req_res_time_bucket{le="50",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_bucket{le="100",method="GET",route="/",status_code="304"} 1
```

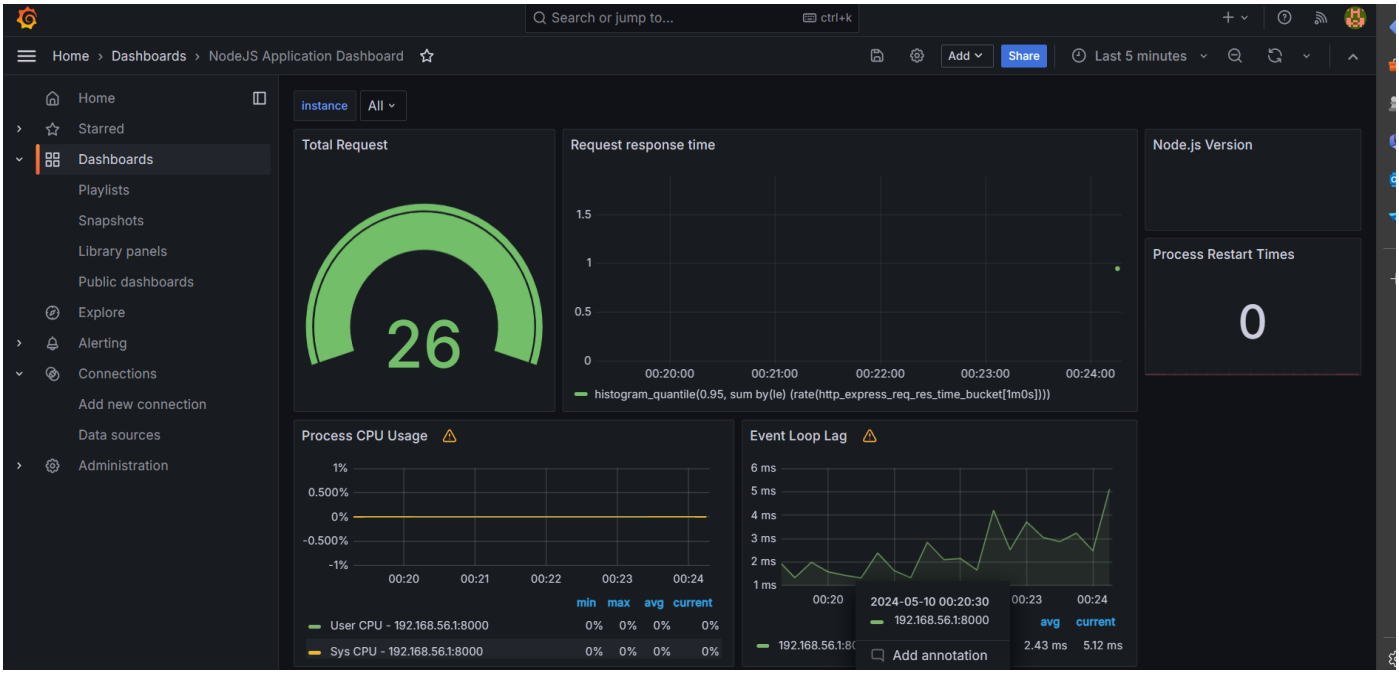
```
nodejs_gc_duration_seconds_bucket{le="0.001",kind="minor"} 190
nodejs_gc_duration_seconds_bucket{le="0.01",kind="minor"} 281
nodejs_gc_duration_seconds_bucket{le="0.1",kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="1",kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="2",kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="5",kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="+Inf",kind="minor"} 283
nodejs_gc_duration_seconds_sum{kind="minor"} 0.3479883999973535
nodejs_gc_duration_seconds_count{kind="minor"} 283
nodejs_gc_duration_seconds_bucket{le="0.001",kind="incremental"} 7
nodejs_gc_duration_seconds_bucket{le="0.01",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="0.1",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="1",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="2",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="5",kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="+Inf",kind="incremental"} 8
nodejs_gc_duration_seconds_sum{kind="incremental"} 0.00986159999668598
nodejs_gc_duration_seconds_count{kind="incremental"} 8
nodejs_gc_duration_seconds_bucket{le="0.001",kind="major"} 0
nodejs_gc_duration_seconds_bucket{le="0.01",kind="major"} 3
nodejs_gc_duration_seconds_bucket{le="0.1",kind="major"} 4
nodejs_gc_duration_seconds_bucket{le="1",kind="major"} 4
nodejs_gc_duration_seconds_bucket{le="2",kind="major"} 4
nodejs_gc_duration_seconds_bucket{le="5",kind="major"} 4
nodejs_gc_duration_seconds_bucket{le="+Inf",kind="major"} 4
nodejs_gc_duration_seconds_sum{kind="major"} 0.024481199994683265
nodejs_gc_duration_seconds_count{kind="major"} 4

# HELP http_express_req_res_time This tells how much tme is taken by req and res
# TYPE http_express_req_res_time histogram
http_express_req_res_time_bucket{le="1",method="GET",route="/",status_code="304"} 0
http_express_req_res_time_bucket{le="50",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_bucket{le="100",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_bucket{le="200",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_bucket{le="500",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_bucket{le="800",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_bucket{le="1000",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_bucket{le="2000",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_bucket{le="+Inf",method="GET",route="/",status_code="304"} 1
http_express_req_res_time_sum{method="GET",route="/",status_code="304"} 19.0518
http_express_req_res_time_count{method="GET",route="/",status_code="304"} 1

# HELP http_express_total_request This tells total request made to the server
# TYPE http_express_total_request counter
http_express_total_request 1
```

Grafana Dashboard

Install and Configure Grafana on Docker



Used Metrics

Table 1: Metrics

Metrics	
Metrics Used	**Justification**
process_cpu_user_seconds_total	Total user CPU time spent in seconds
process_cpu_system_seconds_total	Total system CPU time spent in seconds.
process_cpu_seconds_total	Total user and system CPU time spent in seconds.
process_start_time_seconds	Start time of the process since unix epoch in seconds.
process_resident_memory_bytes	Resident memory size in bytes.
nodejs_eventloop_lag_seconds	Lag of event loop in seconds.
nodejs_eventloop_lag_min_seconds	The minimum recorded event loop delay.
nodejs_eventloop_lag_max_seconds	The maximum recorded event loop delay.
nodejs_eventloop_lag_mean_seconds	The mean of the recorded event loop delays.
nodejs_eventloop_lag_stddev_seconds	The standard deviation of the recorded event loop delays.
nodejs_eventloop_lag_p50_seconds	The 50th percentile of the recorded event loop delays
nodejs_eventloop_lag_p90_seconds	The 90th percentile of the recorded event loop delays
nodejs_active_resources	Number of active resources that are currently keeping the event loop alive, grouped by async resource type.
nodejs_active_resources_total	Total number of active resources.
nodejs_active_handles	Number of active libuv handles grouped by handle type.
nodejs_active_handles_total	Total number of active handles.

nodejs_active_requests_total	Total number of active requests.
nodejs_heap_size_total_bytes	Process heap size from Node.js in bytes
nodejs_heap_size_used_bytes	Process heap size used from Node.js in bytes.
nodejs_external_memory_bytes	Node.js external memory size in bytes.
nodejs_heap_space_size_total_bytes	Process heap space size total from Node.js in bytes.
nodejs_heap_space_size_used_bytes	Process heap space size used from Node.js in bytes.
nodejs_version_info	Node.js version info.
nodejs_gc_duration_seconds	Garbage collection duration by kind.
http_express_req_res_time	Histogram of request and response time.
http_express_total_request	Total request made to the server.

Table 2: Prometheus Metrics

Prometheus Metrics	
Metrics Used & PromQL Queries	**Justification**
prometheus_tsdb_head_series	
sum(prometheus_tsdb_head_series)	Covers every series that has existed in the last 1-3 hours
prometheus_tsdb_head_chunks	
sum(prometheus_tsdb_head_chunks)	Total number of chunks in the head block.
prometheus_engine_query_duration_seconds_sum	
sum(prometheus_engine_query_duration_seconds_sum) by (slice)	The sum of the duration of all engine query processes.

Member Contributions to the Project

IT Number	Name	Contributions
IT20298876	Gunawardana K.P	• Setting up Grafana Dashboard. • Document creation.
IT20658540	Wickramarathna N.A.N.D	• Set up MongoDB Server and configure MongoExporter. • Document creation.
IT21007538	Maddumage P.W	• Configure Node exporter. • Document creation.
IT21054686	Hewage R.P	• Implementing Node JS application • Docker set up. • Document creation.
IT21035876	Kiriella K.G.A.K	• Installing and setting up Prometheus. • Document creation.

NOTE: All members contributed equally to each part of the project and the documentation by assisting in each part.

References

[1] https://docs.docker.com/reference/

[2] https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-onubuntu-22-04.

[3] https://prometheus.io/docs/guides/node-exporter/

[4] https://www.mongodb.com/lp/cloud/atlas/try4?utm_source=google&utm_campaign=search_gs_pl_evergreen_atlas_core_prosp-brand_gic-null_apac-lk_ps-all_desktop_eng_lead&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624368&adgroup=115749715143&cq_cmp=12212624368&gad_source=1&gclid=EAIaIQobChMIyoD9x-yEhgMVvtUWBR3QGgwiEAAYASAAEgLYC_D_BwE