

DevOps

Day 6

Date: 22.03.2025

Topics Covered: Prometheus and Java Application Minikube Deployment

Prometheus and Grafana

Prometheus:

```
sudo useradd \  
    --system \  
    --no-create-home \  
    --shell /bin/false Prometheus  
  
wget https://github.com/prometheus/prometheus/releases/download/v2.47.1/prometheus-  
2.47.1.linux-amd64.tar.gz  
  
tar -xvf prometheus-2.47.1.linux-amd64.tar.gz  
  
sudo mkdir -p /data /etc/prometheus  
  
cd prometheus-2.47.1.linux-amd64/  
  
sudo mv prometheus promtool /usr/local/bin/  
  
sudo mv consoles/ console_libraries/ /etc/prometheus/  
  
sudo mv prometheus.yml /etc/prometheus/prometheus.yml  
  
sudo chown -R prometheus:prometheus /etc/prometheus/ /data/  
  
cd  
  
rm -rf prometheus-2.47.1.linux-amd64.tar.gz  
  
prometheus --version  
  
sudo vim /etc/systemd/system/prometheus.service  
  
[Unit]  
  
Description=Prometheus  
  
Wants=network-online.target  
  
After=network-online.target  
  
StartLimitIntervalSec=500  
  
StartLimitBurst=5  
  
[Service]
```

```
User=prometheus
Group=prometheus
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path=/data \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries \
  --web.listen-address=0.0.0.0:9090 \
  --web.enable-lifecycle
```

[Install]

```
WantedBy=multi-user.target sudo systemctl enable prometheus
```

```
sudo systemctl start prometheus
```

```
sudo systemctl status prometheus
```

```
journalctl -u prometheus -f --no-pager
```

```
sudo useradd \
```

```
  --system \
```

```
  --no-create-home \
```

```
  --shell /bin/false node_exporter
```

```
wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
```

```
tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz
```

```
sudo mv \
```

```
  node_exporter-1.6.1.linux-amd64/node_exporter \
```

```
  /usr/local/bin/
```

```
rm -rf node_exporter*
```

```
node_exporter --version
```

```
sudo vim /etc/systemd/system/node_exporter.service
```

[Unit]

Description=Node Exporter

Wants=network-online.target

After=network-online.target

StartLimitIntervalSec=500

StartLimitBurst=5

[Service]

User=node_exporter

Group=node_exporter

Type=simple

Restart=on-failure

RestartSec=5s

ExecStart=/usr/local/bin/node_exporter \

--collector.logind

[Install]

WantedBy=multi-user.target

sudo systemctl enable node_exporter

sudo systemctl start node_exporter

sudo systemctl status node_exporter

journalctl -u node_exporter -f --no-pager

sudo vim /etc/prometheus/prometheus.yml

- job_name: 'jenkins'

metrics_path: '/prometheus'

static_configs:

- targets: ['<jenkins-ip>:8080']

promtool check config /etc/prometheus/prometheus.yml

curl -X POST <http://localhost:9090/-/reload>

sudo apt-get install -y apt-transport-https software-properties-common

wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list

sudo apt-get update

```
sudo apt-get -y install grafana  
sudo systemctl enable grafana-server  
sudo systemctl start grafana-server  
sudo systemctl status grafana-server
```

Grafana:

Open source

Prometheus is an open-source system monitoring and alerting toolkit originally built at SoundCloud. It is now a standalone open source project . Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.

Features:

multi dimensional

Features

1. a multi-dimensional data model with time series data identified by metric name and key/value pairs
2. PromQL, a flexible query language to leverage this dimensionality
3. no reliance on distributed storage; single server nodes are autonomous
4. time series collection happens via a pull model over HTTP
5. pushing time series is supported via an intermediary gateway
6. targets are discovered via service discovery or static configuration
7. multiple modes of graphing and dashboarding support

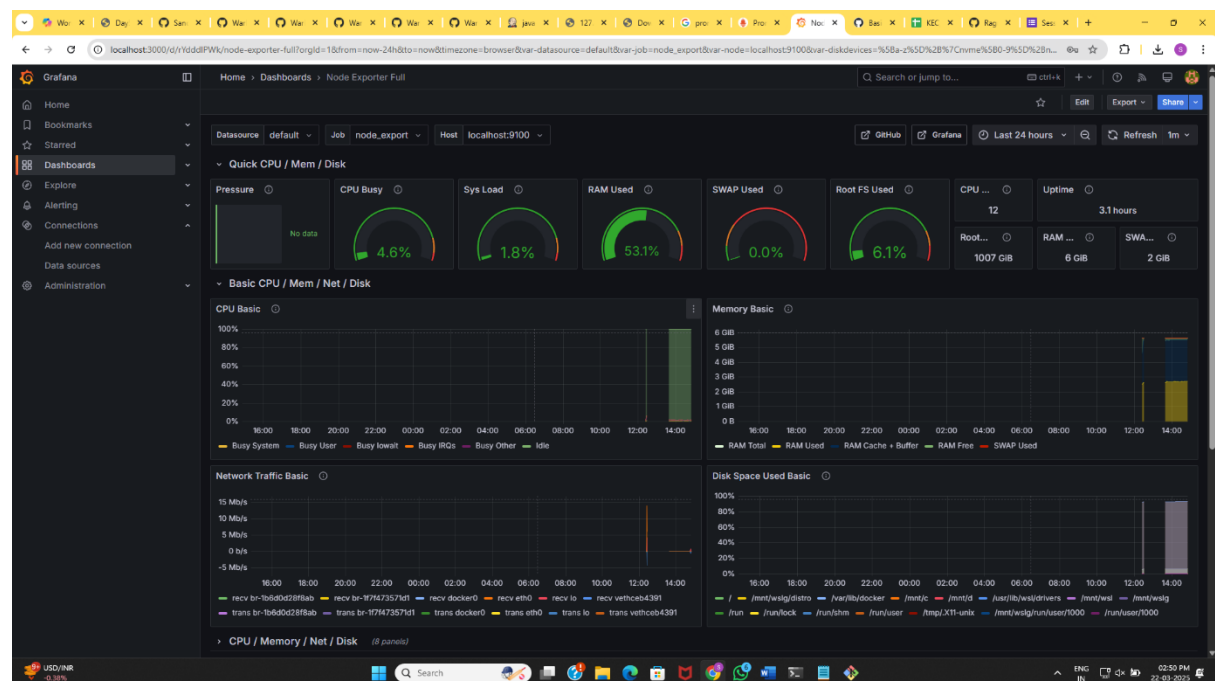
In Grafana new Dashboard

9964 - prometheus

1960

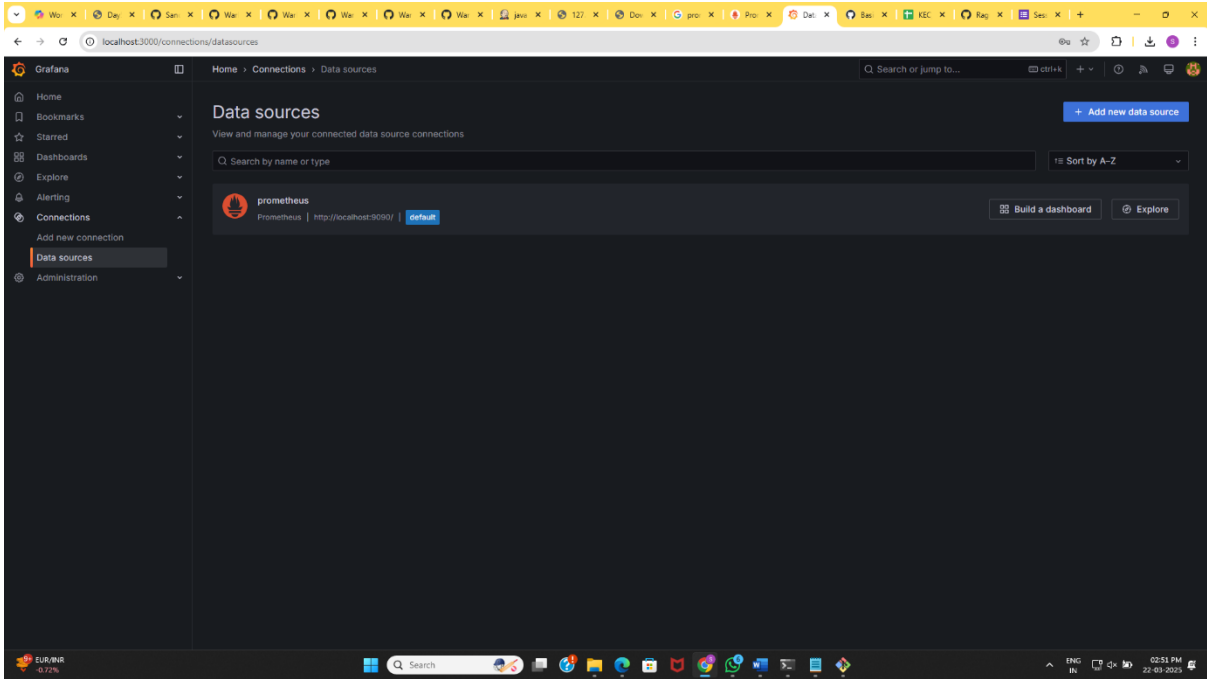
```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 5.8611e-05
go_gc_duration_seconds{quantile="0.25"} 0.000175586
go_gc_duration_seconds{quantile="0.5"} 0.000303127
go_gc_duration_seconds{quantile="0.75"} 0.00047361
go_gc_duration_seconds{quantile="1"} 0.001078461
# HELP go_gc_duration_seconds_sum Sum of go_gc_duration_seconds
go_gc_duration_seconds_sum 0.017440792
# HELP go_gc_duration_seconds_count Sum of go_gc_duration_seconds
# TYPE go_gc_duration_seconds_count gauge
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 41
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.21.1"} 1
# HELP go_memstat_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstat_alloc_bytes gauge
go_memstat_alloc_bytes 3.0781232e+07
# HELP go_memstat_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstat_alloc_bytes_total counter
go_memstat_alloc_bytes_total 3.0916168e+08
# HELP go_memstat_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstat_buck_hash_sys_bytes gauge
go_memstat_buck_hash_sys_bytes 1.30347e+06
# HELP go_memstat_free_total Number of free bytes.
# TYPE go_memstat_free_total counter
go_memstat_free_total 2.157702e+08
# HELP go_memstat_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstat_gc_sys_bytes gauge
go_memstat_gc_sys_bytes 4.84124e+06
# HELP go_memstat_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstat_heap_alloc_bytes gauge
go_memstat_heap_alloc_bytes 3.0781232e+07
# HELP go_memstat_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstat_heap_idle_bytes gauge
go_memstat_heap_idle_bytes 2.56480e+07
# HELP go_memstat_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstat_heap_inuse_bytes gauge
go_memstat_heap_inuse_bytes 3.56864e+07
# HELP go_memstat_heap_objects Number of allocated objects.
# TYPE go_memstat_heap_objects gauge
go_memstat_heap_objects 146546
# HELP go_memstat_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstat_heap_released_bytes gauge
go_memstat_heap_released_bytes 1.4588032e+07
# HELP go_memstat_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstat_heap_sys_bytes gauge
go_memstat_heap_sys_bytes 6.1210202e+07
# HELP go_memstat_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstat_last_gc_time_seconds gauge
go_memstat_last_gc_time_seconds 1.7423451835473e+09
# HELP go_memstat_lookups_total Total number of pointer lookups.
# TYPE go_memstat_lookups_total counter
go_memstat_lookups_total 0
# HELP go_memstat_malloc_total Total number of mallocs.
# TYPE go_memstat_malloc_total counter
go_memstat_malloc_total 2.384238e+06
# HELP go_memstat_mcache_inuse_bytes Number of bytes in use by mcache structures.
# TYPE go_memstat_mcache_inuse_bytes gauge
go_memstat_mcache_inuse_bytes 14488
# HELP go_memstat_mcache_sys_bytes Number of bytes used for mcache structures obtained from system.
# TYPE go_memstat_mcache_sys_bytes gauge
```

Grafana Prometheus Dashboard



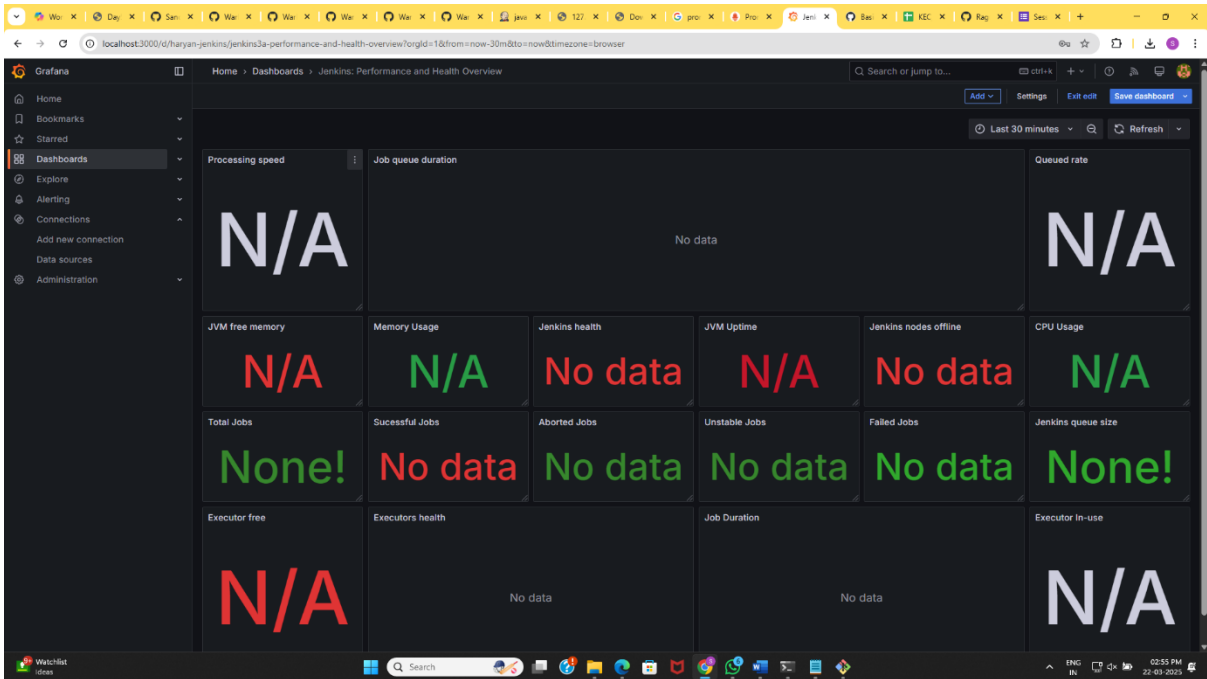
Grafana Dashboard

Creating Connections for Prometheus



Connections

Jenkins Dashboard



Grafana Dashboard

[illegible]

Prometheus

The screenshot shows the Prometheus web interface. At the top, there's a navigation bar with 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below this is a toolbar with checkboxes for 'Use local time', 'Enable query history', 'Enable autocomplete', 'Enable highlighting', and 'Enable linter'. The query bar contains the query: `rate(node_network_receive_bytes_total[1m])`. Below the query bar, there are tabs for 'Table' and 'Graph'. The 'Graph' tab is selected, showing a line graph of network receive bytes over time. The x-axis represents time from 09:10 to 10:00. The y-axis represents the rate of bytes received, ranging from 0.00 to 120.00k. The graph shows several peaks, with the highest peak reaching approximately 100.00k around 09:18. A legend at the bottom lists the series: `(device="bn-106d523f8ab", instance="localhost9100", job="node_export")`, `(device="bn-17f47271f4f", instance="localhost9100", job="node_export")`, `(device="docker0", instance="localhost9100", job="node_export")`, `(device="eth0", instance="localhost9100", job="node_export")`, `(device="lo", instance="localhost9100", job="node_export")`, and `(device="vethosb4391", instance="localhost9100", job="node_export")`. A note at the bottom says 'Click select series, CTRL + click to toggle multiple series'.

Prometheus

node_load15



Prometheus