

DevOps

Day 6

Date: 22.03.2025

Topics Covered: Prometheus and Java Application Minikube Deployment

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
```

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

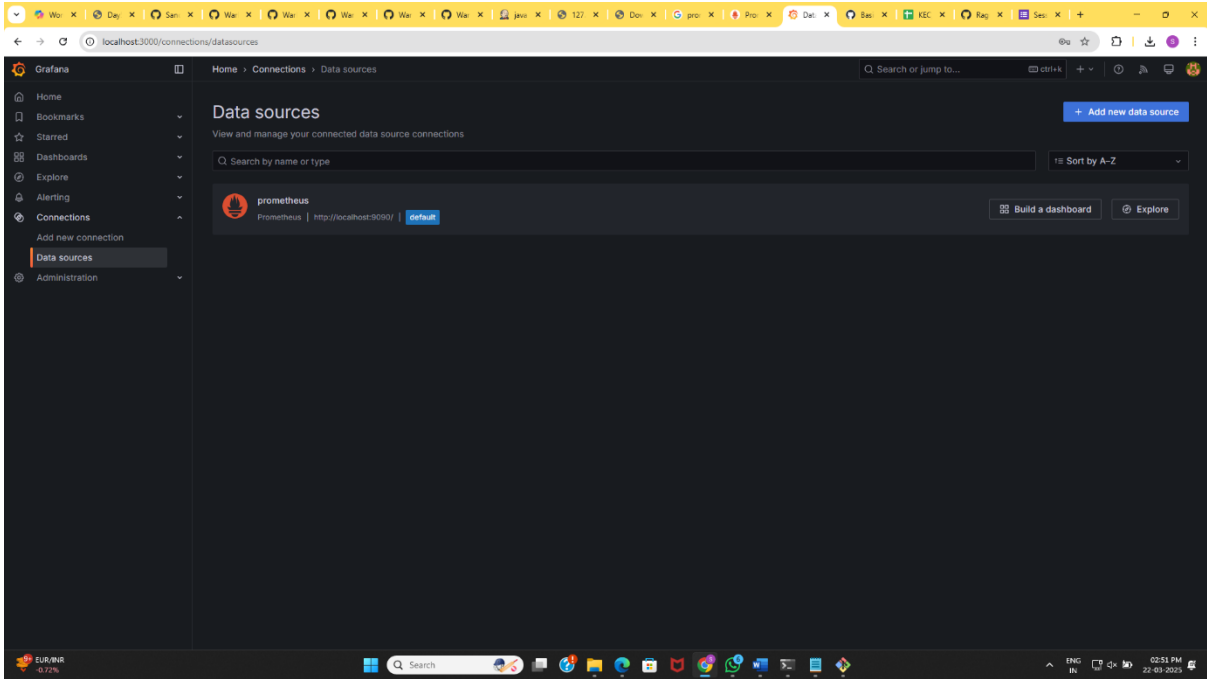
1960

The screenshot shows a web browser window with the address bar displaying `localhost:9090/metrics`. The page content lists various Go runtime metrics and their values:

- `# 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"} 5.601e-05`
- `go_gc_duration_seconds{quantile="0.25"} 0.000175586`
- `go_gc_duration_seconds{quantile="0.5"} 0.000381317`
- `go_gc_duration_seconds{quantile="0.75"} 0.00048761`
- `go_gc_duration_seconds{quantile="1"} 0.001078461`
- `go_gc_duration_seconds_sum 0.017467592`
- `go_gc_duration_seconds_count 50`
- `# 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_memstats_alloc_bytes Number of bytes allocated and still in use.`
- `go_memstats_alloc_bytes 3.0781232e+07`
- `# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.`
- `go_memstats_alloc_bytes_total 3.0914688e+08`
- `# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.`
- `go_memstats_buck_hash_sys_bytes 1.186247e+06`
- `# TYPE go_memstats_buck_hash_sys_bytes gauge`
- `go_memstats_free_total Total number of frees.`
- `go_memstats_free_total 1.157702e+08`
- `# TYPE go_memstats_free_total counter`
- `go_memstats_free_total 1.157702e+08`
- `# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.`
- `go_memstats_gc_sys_bytes 4.841264e+06`
- `# TYPE go_memstats_gc_sys_bytes gauge`
- `go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.`
- `go_memstats_heap_alloc_bytes 3.06868e+07`
- `# TYPE go_memstats_heap_alloc_bytes gauge`
- `go_memstats_heap_idle_bytes 1.0781232e+07`
- `# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.`
- `go_memstats_heap_idle_bytes 1.06408e+07`
- `# TYPE go_memstats_heap_idle_bytes gauge`
- `go_memstats_heap_inuse_bytes Number of heap bytes that are in use.`
- `go_memstats_heap_inuse_bytes 3.06868e+07`
- `# TYPE go_memstats_heap_inuse_bytes gauge`
- `go_memstats_heap_objects Number of allocated objects.`
- `go_memstats_heap_objects 146546`
- `# TYPE go_memstats_heap_objects gauge`
- `go_memstats_heap_released_bytes Number of heap bytes released to OS.`
- `go_memstats_heap_released_bytes 1.4508812e+07`
- `# TYPE go_memstats_heap_released_bytes gauge`
- `go_memstats_heap_sys_bytes Number of heap bytes obtained from system.`
- `go_memstats_heap_sys_bytes 3.1210236e+07`
- `# TYPE go_memstats_heap_sys_bytes gauge`
- `go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.`
- `go_memstats_last_gc_time_seconds 1.742635183541763e+09`
- `# TYPE go_memstats_last_gc_time_seconds gauge`
- `go_memstats_lookups_total Total number of pointer lookups.`
- `go_memstats_lookups_total 0`
- `# TYPE go_memstats_lookups_total counter`
- `go_memstats_malloc_total Total number of mallocs.`
- `go_memstats_malloc_total 2.384288e+06`
- `# TYPE go_memstats_malloc_total counter`
- `go_memstats_mcache_inuse_bytes Number of bytes in use by mcache structures.`
- `go_memstats_mcache_inuse_bytes 14400`
- `# TYPE go_memstats_mcache_inuse_bytes gauge`
- `go_memstats_mcache_sys_bytes Number of bytes used for mcache structures obtained from system.`
- `go_memstats_mcache_sys_bytes 14400`
- `# TYPE go_memstats_mcache_sys_bytes gauge`

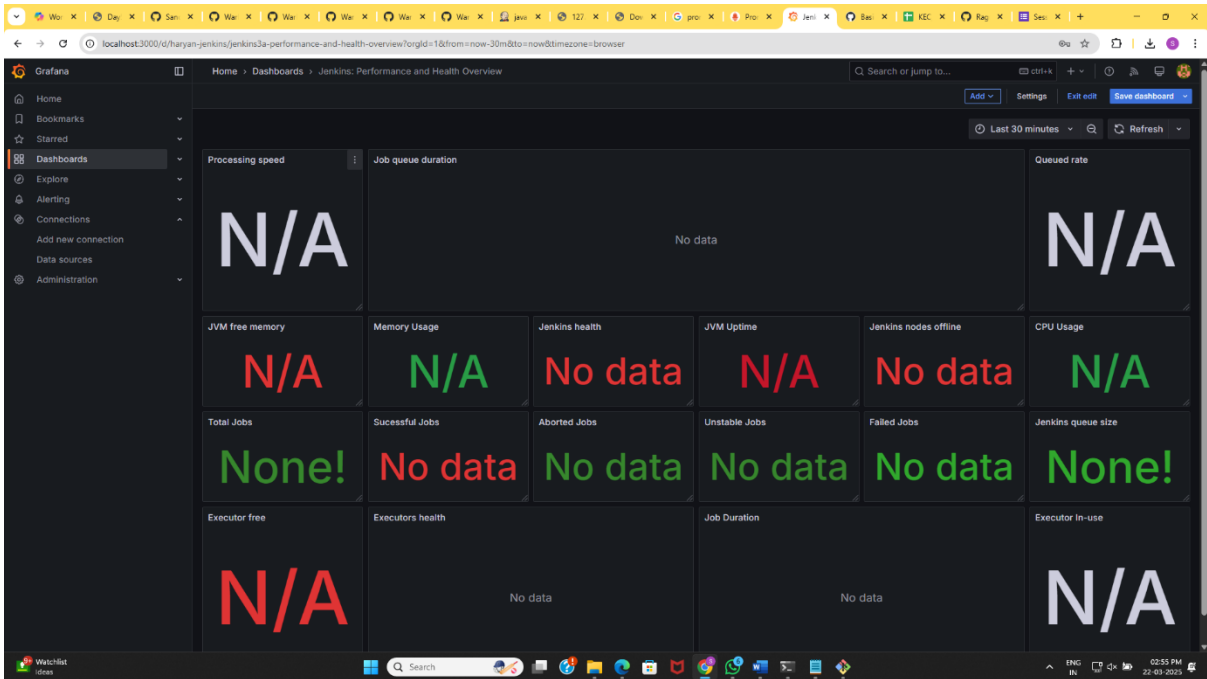
[illegible]

Creating Connections for Prometheus



Connections

Jenkins Dashboard



Grafana Dashboard