1-How do I trigger a Prometheus alert?

1. Setup and configure AlertManager.
2. Configure the config file on Prometheus so it can talk to the AlertManager.
3. Define alert rules in Prometheus server configuration.
4. Define alert mechanism in AlertManager to send alerts via Slack and Mail

Ref : https://medium.com/devops-dudes/prometheus-alerting-with-alertmanager-e1bbba8e6a8e

---

2-What is the difference between node exporter and mysql exporter ?

Node-exporter:
- The Prometheus Node Exporter is an open-source time-series monitoring and alerting system for cloud-native environments.
- It can collect and store node-level metrics as time-series data, recording information with a timestamp, various server resources such as RAM, disk space, and CPU utilization.
- It can also collect and record labels, which are optional key-value pairs.
- It works on port 9100.

Mysql-exporter:
- MySQL Exporter is a client application used to get MySQL metrics and export to Prometheus server.
- SQL Exporter is a configuration driven exporter that exposes metrics gathered from DBMSs, for use by the Prometheus monitoring system.
- It works on port 9104.

3-what is the maximum retention period to save data in Prometheus and how to increase it ?

-By default the retention is configured to 15 days. The amounts of data stored on disk depends on retention, higher retention means more data on disk.
- To increase:
  1. On the management node, open the /etc/sysconfig/prometheus file to edit, set the needed retention period for the STORAGE_RETENTION option, and then save your changes. For example:
     `STORAGE_RETENTION="--storage.tsdb.retention.time=30d"`
  2. Restart the Prometheus service:

```
systemctl restart prometheus.service
```

Ref: https://prometheus.io/docs/prometheus/latest/storage/

Ref 2 : https://stackoverflow.com/questions/59298811/increasing-prometheus-storage-retention

---

4-What are the different PromQL data types available in Prometheus Expression language?

PromQL uses three main data types: **scalars, range vectors, and instant vectors, but other references could divide them into:**

1- Floats (mostly scalars)
2- Range vectors
3- Instant vectors
4- Time (though it's often not counted in this category)

Ref : https://grafana.com/blog/2020/02/04/introduction-to-promql-the-prometheus-query-language/

Ref 2 : https://logz.io/blog/promql-examples-introduction/#func

---

5-How To calculate the average request duration over the last 5 minutes from a histogram ?

```
rate(http_request_duration_seconds_sum[5m])

rate(http_request_duration_seconds_count[5m])
```

Ref : https://prometheus.io/docs/practices/histograms/

6-What is Thanos Prometheus?

hanos is **a set of components that can be composed into a highly available metric system with unlimited storage capacity**, which can be added seamlessly on top of existing Prometheus deployments. Thanos is a CNCF Incubating project

ref: https://github.com/thanos-io/thanos

7-what is promtool and how i can use it ?

- Prometheus ships with a very useful supporting command-line tool called promtool.
- This small Golang binary can be used to quickly perform several troubleshooting actions and is packed with helpful subcommands.
- Used as a Tool for the Prometheus monitoring system.

Ref : https://linuxcommandlibrary.com/man/promtool

8-What types of Monitoring can be done via Grafana?

- Grafana is used to monitor their **infrastructure and log analytics**, predominantly to improve their operational efficiency. Dashboards make tracking users and events easy as it automates the collection, management, and viewing of data.

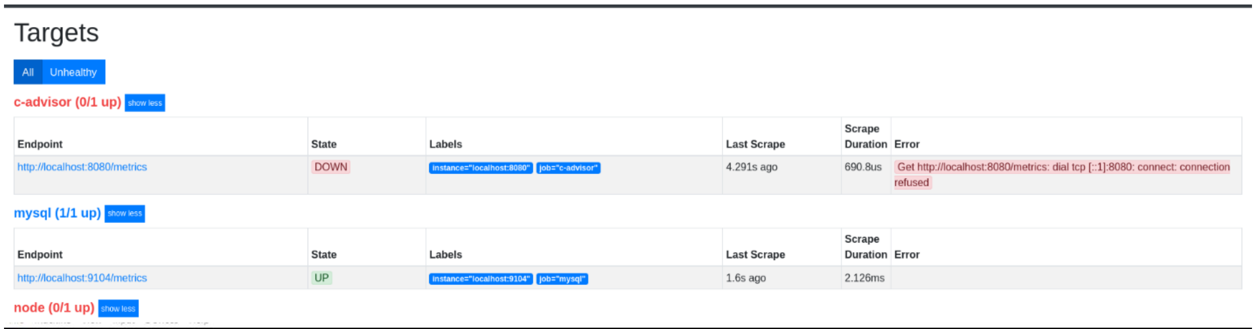
Ref : https://www.skedler.com/blog/everything-you-need-to-know-about-grafana/

9-Can we see different Servers CPU comparison in Grafana

- Yes, must making different queries for each server.

Ready dashboard to be implemented : https://grafana.com/grafana/dashboards/15334-server-metrics-cpu-memory-disk-network/

Screenshot from the lab 2



Targets

All | Unhealthy

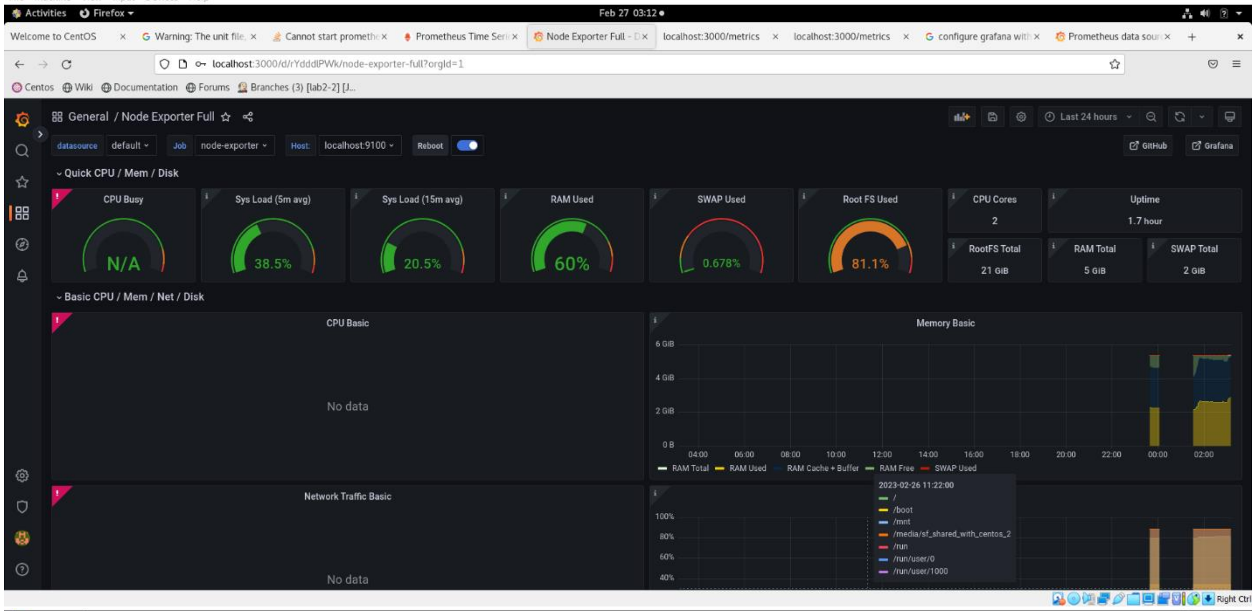
c-advisor (0/1 up) show less

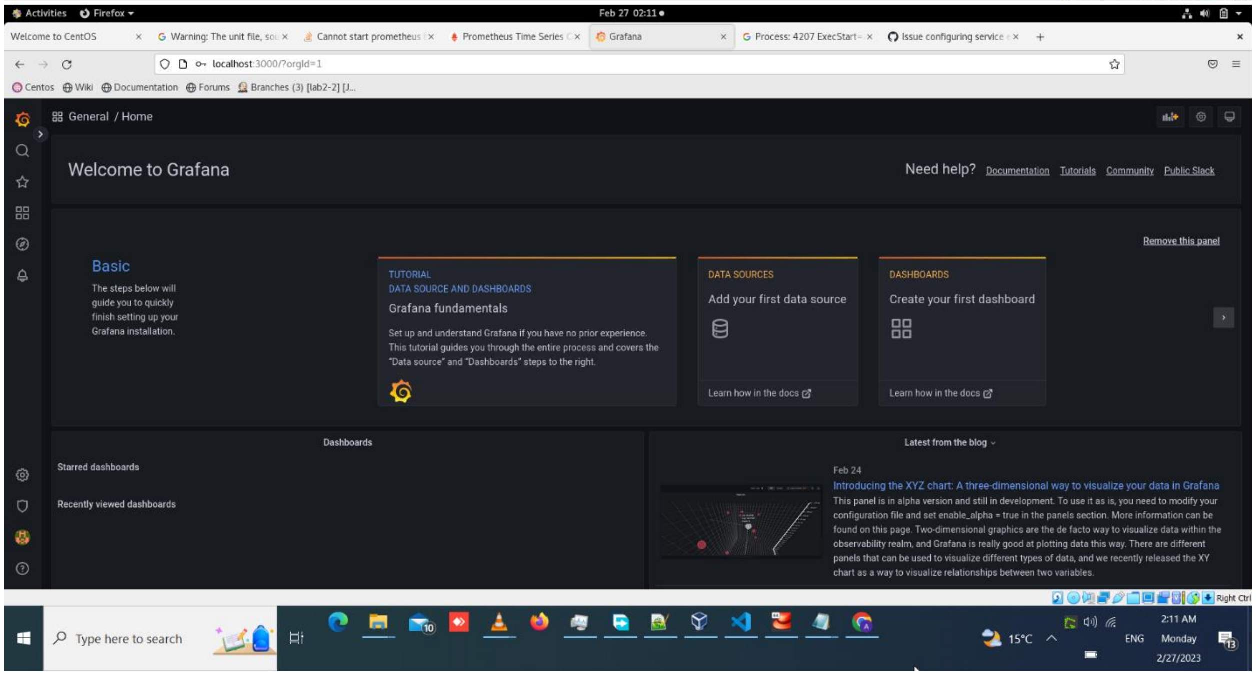
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8080/metrics	DOWN	instance="localhost:8080" job="c-advisor"	4.291s ago	690.8us	Get http://localhost:8080/metrics: dial tcp [::1]:8080: connect: connection refused

mysql (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9104/metrics	UP	instance="localhost:9104" job="mysql"	1.6s ago	2.126ms	

node (0/1 up) show less





Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Remove this panel

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

DATA SOURCES
Add your first data source

Learn how in the docs

DASHBOARDS
Create your first dashboard

Learn how in the docs

Dashboards

Starred dashboards

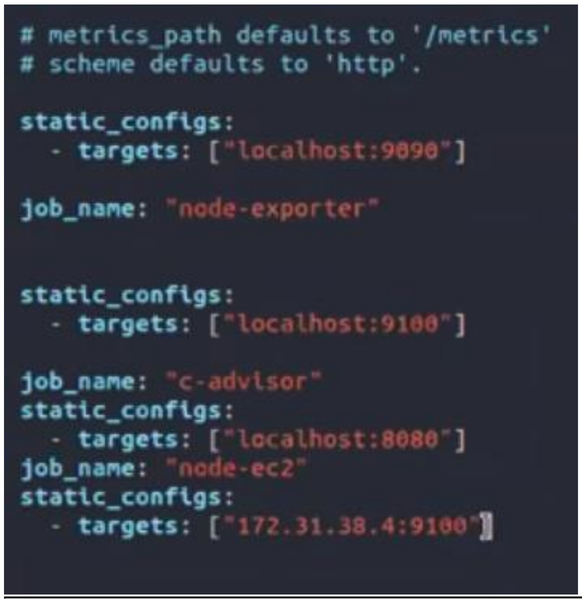
Recently viewed dashboards

Latest from the blog

Feb 24

Introducing the XYZ chart: A three-dimensional way to visualize your data in Grafana

This panel is in alpha version and still in development. To use it as is, you need to modify your configuration file and set enable_alpha = true in the panels section. More information can be found on this page. Two-dimensional graphics are the de facto way to visualize data within the observability realm, and Grafana is really good at plotting data this way. There are different panels that can be used to visualize different types of data, and we recently released the XY chart as a way to visualize relationships between two variables.



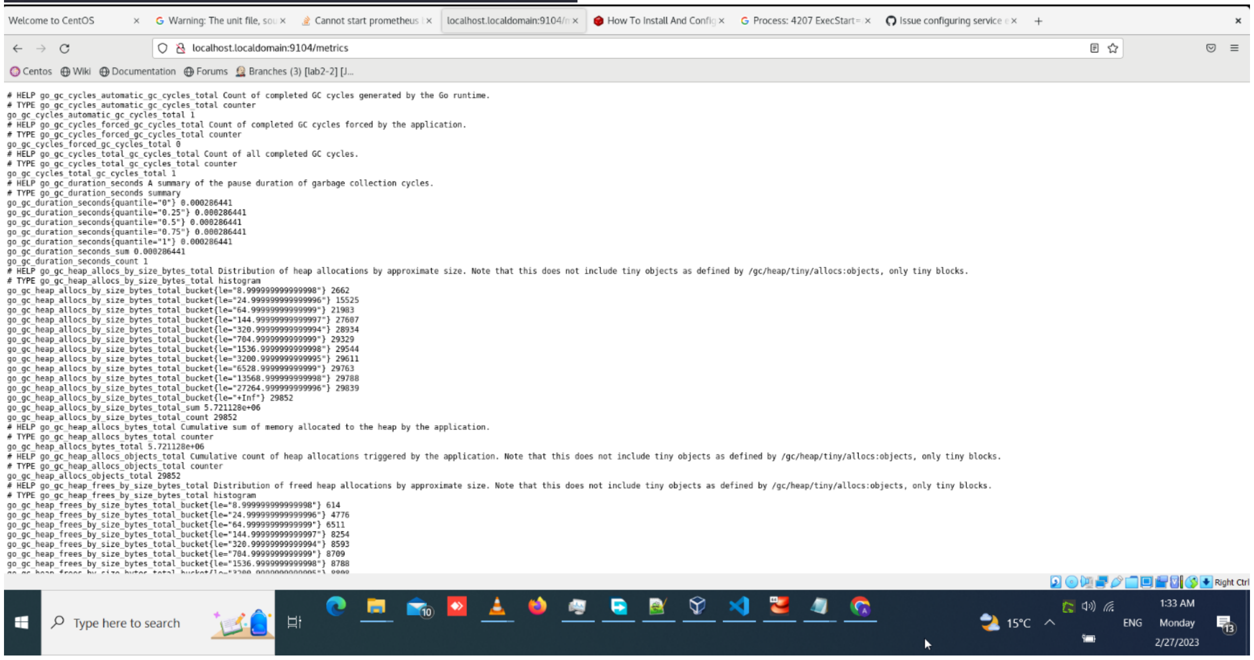
```
# metrics_path defaults to '/metrics'
# scheme defaults to 'http'.

static_configs:
  - targets: ["localhost:9090"]

job_name: "node-exporter"


static_configs:
  - targets: ["localhost:9100"]

job_name: "c-advisor"
static_configs:
  - targets: ["localhost:8080"]
job_name: "node-ec2"
static_configs:
  - targets: ["172.31.38.4:9100"]
```


Welcome to CentOS
Warning: The unit file, sou
Cannot start prometheus
localhost.localdomain:9104/m
How To Install And Config
Process: 4207 ExecStart=
Issue configuring service
localhost.localdomain:9104/metrics
Centos
Wiki
Documentation
Forums
Branches (3) [lab2-2] [J...
Type here to search
15°C
ENG
1:33 AM
Monday
2/27/2023

```
# 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 1
# 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 1
# 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"} 0.000286441
go_gc_duration_seconds{quantile="0.25"} 0.000286441
go_gc_duration_seconds{quantile="0.5"} 0.000286441
go_gc_duration_seconds{quantile="0.75"} 0.000286441
go_gc_duration_seconds{quantile="1"} 0.000286441
go_gc_duration_seconds_sum 0.000286441
go_gc_duration_seconds_count 1
# HELP go_gc_heap_allocs_by_size_bytes_total Distribution of heap allocations by approximate size. 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_total histogram
go_gc_heap_allocs_by_size_bytes_total_bucket{le="8.999999999999998"} 2662
go_gc_heap_allocs_by_size_bytes_total_bucket{le="24.999999999999996"} 15525
go_gc_heap_allocs_by_size_bytes_total_bucket{le="64.99999999999999"} 21983
go_gc_heap_allocs_by_size_bytes_total_bucket{le="144.99999999999997"} 27607
go_gc_heap_allocs_by_size_bytes_total_bucket{le="320.99999999999994"} 28934
go_gc_heap_allocs_by_size_bytes_total_bucket{le="704.9999999999999"} 29329
go_gc_heap_allocs_by_size_bytes_total_bucket{le="1536.9999999999998"} 29544
go_gc_heap_allocs_by_size_bytes_total_bucket{le="3200.9999999999995"} 29611
go_gc_heap_allocs_by_size_bytes_total_bucket{le="6528.999999999999"} 29763
go_gc_heap_allocs_by_size_bytes_total_bucket{le="13568.999999999998"} 29788
go_gc_heap_allocs_by_size_bytes_total_bucket{le="27264.999999999996"} 29839
go_gc_heap_allocs_by_size_bytes_total_bucket{le="+Inf"} 29852
go_gc_heap_allocs_by_size_bytes_total_sum 5.721128e+06
go_gc_heap_allocs_by_size_bytes_total_count 29852
# HELP go_gc_heap_allocs_bytes_total Cumulative sum of memory allocated to the heap by the application.
# TYPE go_gc_heap_allocs_bytes_total counter
go_gc_heap_allocs_bytes_total 5.721128e+06
# HELP go_gc_heap_allocs_objects_total Cumulative count of heap allocations triggered by the application. Note that this does not include tiny objects as defined by /gc/heap/tiny/allocs:objects, only tiny blocks.
# TYPE go_gc_heap_allocs_objects_total counter
go_gc_heap_allocs_objects_total 29852
# HELP go_gc_heap_frees_by_size_bytes_total Distribution of freed heap allocations by approximate size. Note that this does not include tiny objects as defined by /gc/heap/tiny/allocs:objects, only tiny blocks.
# TYPE go_gc_heap_frees_by_size_bytes_total histogram
go_gc_heap_frees_by_size_bytes_total_bucket{le="8.999999999999998"} 614
go_gc_heap_frees_by_size_bytes_total_bucket{le="24.999999999999996"} 4776
go_gc_heap_frees_by_size_bytes_total_bucket{le="64.99999999999999"} 6511
go_gc_heap_frees_by_size_bytes_total_bucket{le="144.99999999999997"} 8254
go_gc_heap_frees_by_size_bytes_total_bucket{le="320.99999999999994"} 8593
go_gc_heap_frees_by_size_bytes_total_bucket{le="704.9999999999999"} 8709
go_gc_heap_frees_by_size_bytes_total_bucket{le="1536.9999999999998"} 8788
```