Tutorial Prometheus and Grafana

Create a docker-compose file with Prometheus, Grafana and Node-exporter.

Create a docker-compose.yml to create a container for Grafana, Prometheus and Node-Exporter.

```
version: "3.8"

volumes:
    grafana-data:
    prometheus-data:

services:
    grafana:
    image: grafana/grafana:latest
    ... etc

prometheus:
    image: prom/prometheus:latest
    ... etc

node_exporter:
    image: quay.io/prometheus/node-exporter:latest
    ... etc
```

Create a prometheus.yml with the configurations to connect with Node-Eporter.

```
global:
scrape_interval: 15s
evaluation_interval: 15s

scrape_configs:
- job_name: 'prometheus'
static_configs:
- targets: ['prometheus:9090']

- job_name: 'node_exporter'
static_configs:
- targets: ['node_exporter'
```

Run docker-compose up --build -d to run the docker-compose.yml configurations

```
Liox@DESKTOP-SGIFBT8:~/jalasoft/jalasoft-bootcamp/devOps_1/JalaUniversity_DevOps/docker-compose$ docker-compose up -d
[+] Running 3/0

" Container prometheus Running 0.6s

" Container node_exporter Bunning 8.6s

" Container grafana Running 0.6s
```

Verify the containers docker-compose ps

```
compose$ docker-compose ps
NAME
                    COMMAND
                                               SERVICE
                                                                    STATUS
grafana
                    "/run.sh"
                                               grafana
                                                                                         0.0.0.0:3000->3000/tcp
                                                                    running
node_exporter
                    "/bin/node_exporter ..."
                                               node_exporter
                                                                    running
                                                                                         0.0.0.0:9100->9100/tcp
                    "/bin/prometheus --c.."
                                                                                         0.0.0.0:9090->9090/tcp
prometheus
                                               prometheus
                                                                    running
```

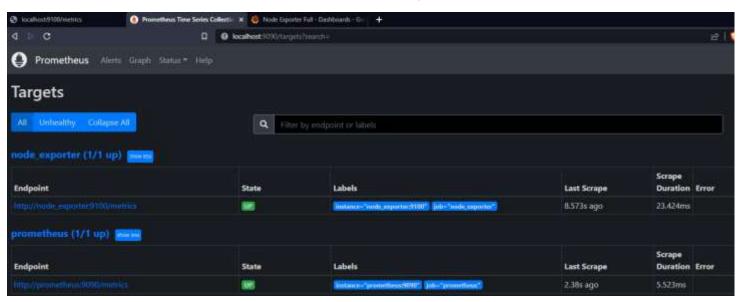
Go to http://localhost:9100/metrics to see node-exporter metrics

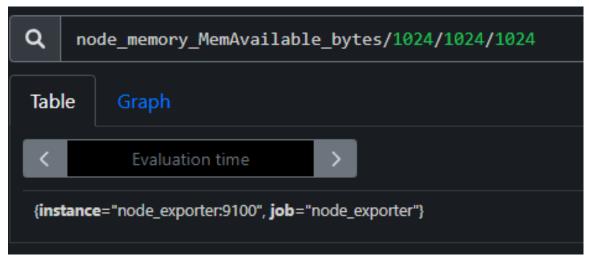
localhost:9100/metrics

```
4
    \triangleright \mathbf{c}
                                                       П
                                                             • localhost:9100/metrics
# 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"} 2e-05
go_gc_duration_seconds{quantile="0.25"} 3.68e-05
go_gc_duration_seconds{quantile="0.5"} 3.91e-05
go_gc_duration_seconds{quantile="0.75"} 4.38e-05
go gc duration seconds{quantile="1"} 4.44e-05
go gc duration seconds sum 0.0002624
go gc duration seconds count 7
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go goroutines gauge
go goroutines 8
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go info{version="go1.19.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go memstats alloc bytes gauge
go memstats alloc bytes 2.65084e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go memstats alloc bytes total 1.4250632e+07
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
```

Prometheus Time Series Collection and 6 Node Exporter Full - Dashboards

Go to http://localhost:9090 to see Prometheus targets and make a query.







Go to Grafana on http://localhost:3000.

Add a data source and import a dashboard to see the data.

