

Grafana, Prometheus Continuous Monitoring on Docker

- Install docker and Docker-composer on Ubuntu-18.04 – WSL2
- ✓ Follow docker instructions at <https://docs.docker.com/engine/install/ubuntu/#prerequisites>

```
cris@DESKTOP-PM304DL:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:10d7d58d5ebd2a652f4d93fdd86da8f265f5318c6a73cc5b6a9798ff6d2b2e67
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

```
cris@DESKTOP-PM304DL:~$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
```

- Create the Docker Compose file (Included in the folder)
- Create Prometheus config file (Included in the folder)
- Pull docker images

✓ docker-compose pull

```
cris@DESKTOP-PM304DL:~/cmdocker_workspace$ docker-compose pull
Pulling grafana      ... done
Pulling prometheus  ... done
Pulling node_exporter ... done
Pulling cadvisor    ... done
Pulling app_example ... done
```

- Start container

✓ docker-compose up -d

```
cris@DESKTOP-PM304DL:~/cmdocker_workspace$ docker-compose up -d
Creating network "cmdocker_workspace_default" with the default driver
Creating volume "cmdocker_workspace_grafana-data" with default driver
Creating volume "cmdocker_workspace_prometheus-data" with default driver
Creating node_exporter ... done
Creating cadvisor      ... done
Creating prometheus    ... done
Creating grafana        ... done
Creating app_example    ... done
```

- Review each endpoint is working properly using the browser

✓ Example app

← → ↺ ⓘ localhost/metrics

```
# HELP http_request_duration_seconds Duration of all HTTP requests
# TYPE http_request_duration_seconds histogram
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="0.005"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="0.01"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="0.025"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="0.05"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="0.1"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="0.25"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="0.5"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="1"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="2.5"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="5"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="10"} 4
http_request_duration_seconds_bucket{code="200",handler="found",method="get",le="+Inf"} 4
http_request_duration_seconds_sum{code="200",handler="found",method="get"} 0.00025439999999999995
http_request_duration_seconds_count{code="200",handler="found",method="get"} 4
# HELP http_requests_total Count of all HTTP requests
# TYPE http_requests_total counter
http_requests_total{code="200",method="get"} 4
# HELP version Version information about this binary
# TYPE version gauge
version{version="v0.3.0"} 1
```

✓ Prometheus

Prometheus Alerts Graph Status ▾ Help Classic UI					
Targets					
All Unhealthy Collapse All					
app_example (1/1 up) Show logs					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://app_example:8080/metrics	UP	instance="app_example:8080" job="app_example"	15.62s ago	1.639ms	
cadvisor (1/1 up) Show logs					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://cadvisor:8080/metrics	UP	instance="cadvisor:8080" job="cadvisor"	12.80s ago	63.280ms	
node_exporter (1/1 up) Show logs					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://node_exporter:9100/metrics	UP	instance="node_exporter:9100" job="node_exporter"	8.20s ago	29.036ms	
prometheus (1/1 up) Show logs					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://prometheus:9090/metrics	UP	instance="prometheus:9090" job="prometheus"	20.82s ago	23.392ms	

✓ Node-exporter

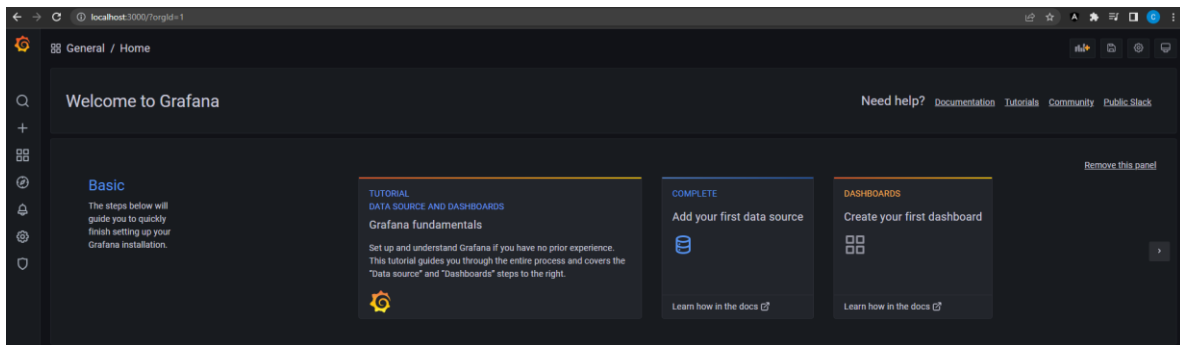
```

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"} 6.1e-05
go_gc_duration_seconds{quantile="0.25"} 0.0003179
go_gc_duration_seconds{quantile="0.5"} 0.0005318
go_gc_duration_seconds{quantile="0.75"} 0.0007144
go_gc_duration_seconds{quantile="1"} 0.0056074
go_gc_duration_seconds_sum 0.0809157
go_gc_duration_seconds_count 128
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 9
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.17.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 3.46344e+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 2.08937704e+08
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.487359e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 1.454587e+06
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.

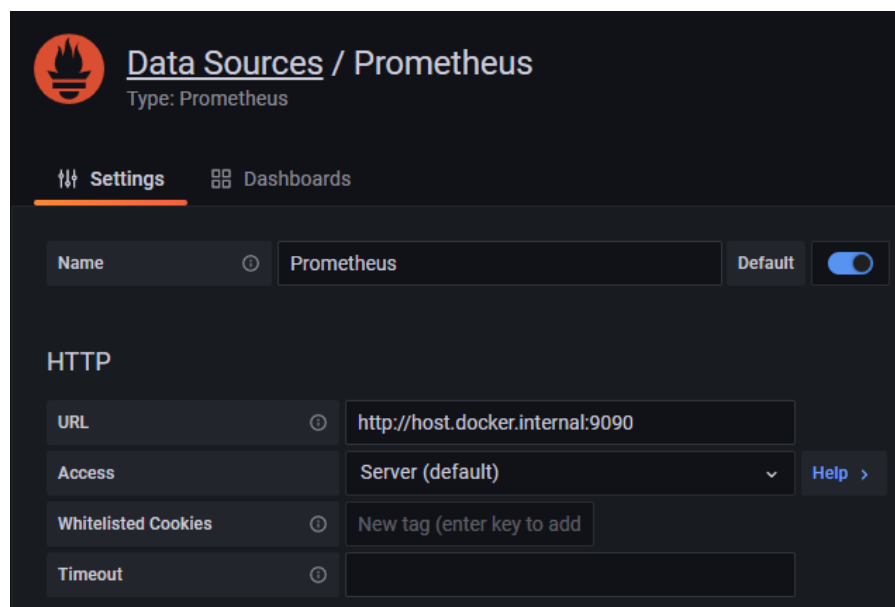
```

✓ Grafana

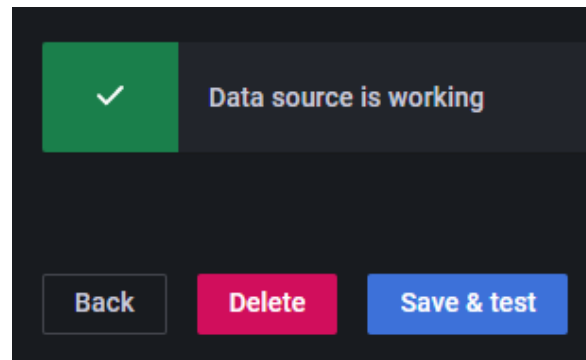


- Configure a new data source

✓ Add the host.docker.internal parameter so it can address the internal IP address



- ✓ Save and test the configuration



- Import a dashboard for data visualization
- ✓ Import existing dashboard using ID 1860 and add Prometheus as data source

Importing dashboard from [Grafana.com](#)

Published by	rfraile
Updated on	2022-02-08 14:51:42

Options

Name

Folder


General

Unique identifier (UID)

The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

Change uid

Prometheus

 Prometheus

Import

Cancel

✓ Check the node-exporter data

