Exercise 3.-

Firstly, I create the **docker-compose.yml** version 3.3 as you can see below:

Here it starts the **docker-compose.yml**

# This first part defines the telegraf service:

version: "3.3"

services:

telegraf:

image: telegraf:1.4.0

hostname: "{{.Node.ID}}"

volumes:

- /var/run/docker.sock:/var/run/docker.sock

configs:

- source: telegraf.conf

target: /etc/telegraf/telegraf.conf

deploy:

mode: global

configs:

telegraf.conf:

file: ./telegraf/telegraf.conf

#This second part defines:

services:

influxdb:

image: influxdb:1.3.5-alpine

configs:

- source: influxdb.conf

target: /etc/influxdb/influxdb.conf

volumes:

- /data/influxdb:/var/lib/influxdb

deploy:

placement:

constraints:

- node.labels.influxdb == true

configs:

influxdb.conf:

file: ./influxdb/influxdb.conf

#This 3 part is about Grafana

services:

grafana:

image: grafana/grafana:4.5.2

ports:

- 3000:3000

volumes:

- /data/grafana:/var/lib/grafana

deploy:

placement:

constraints:

- node.labels.grafana == true

#This last part, show how to execute connectivity\_check.sh, inside Grafana service

command: sh -c "/usr/local/bin/init.sh && exec connectivity\_check.sh"

Now we have to get the following config files:

1) I get the telegraf.conf file that I get from the following instructions:

docker pull telegraf:1.4.0-alpine

docker run --rm telegraf:1.4.0-alpine telegraf config > telegraf.conf

I get as I show below:

[[inputs.net]]

interfaces = ["eth0,eth1,lo"]

[[inputs.cpu]]

percpu = true

totalcpu = true

collect\_cpu\_time = false

[[inputs.disk]]

ignore\_fs = ["tmpfs", "devtmpfs"]

[[inputs.diskio]]

[[inputs.kernel]]

[[inputs.mem]]

[[inputs.processes]]

[[inputs.swap]]

[[inputs.system]]

[[inputs.netstat]]

[[inputs.docker]]

endpoint = "unix:///var/run/docker.sock"

container\_names = []

timeout = "5s"

perdevice = true

total = false

docker\_label\_include = []

docker\_label\_exclude = []

[[outputs.influxdb]]

urls = ["http://influxdb:8086"]

database = "telegraf"

retention\_policy = ""

write\_consistency = "any"

timeout = "5s"

2)I get the influxdb.conf file that I get from the following instructions:

docker pull influxdb:1.3.5-alpine

docker run --rm influxdb:1.3.5-alpine influxd config > influxdb.conf

I get as I show below:

[meta]

dir = "/var/lib/influxdb/meta"

retention-autocreate = true

logging-enabled = true

[data]

dir = "/var/lib/influxdb/data"

index-version = "inmem"

wal-dir = "/var/lib/influxdb/wal"

wal-fsync-delay = "0s"

query-log-enabled = true

cache-max-memory-size = 1073741824

cache-snapshot-memory-size = 26214400

cache-snapshot-write-cold-duration = "10m0s"

compact-full-write-cold-duration = "4h0m0s"

max-series-per-database = 1000000

max-values-per-tag = 100000

max-concurrent-compactions = 0

trace-logging-enabled = false

[http]

enabled = true

bind-address = ":8086"

auth-enabled = false

log-enabled = true

write-tracing = false

pprof-enabled = true

https-enabled = false

https-certificate = "/etc/ssl/influxdb.pem"

https-private-key = ""

max-row-limit = 0

max-connection-limit = 0

shared-secret = ""

realm = "InfluxDB"

unix-socket-enabled = false

bind-socket = "/var/run/influxdb.sock"

3) Grafana as you can see above has not config file.

Secondly, we have to active the the monitoring stack:

1) To Active Influxdb in the node we need influxdb rus

docker node update --label-add influxdb=true <NODE-ID>

2) To active grafana in the node we want grafana runs

docker node update --label-add grafana=true <NODE-ID>

Lastly, we have to deploy our monitoring stack by docker stack

docker stack deploy -c docker-compose.yml MONITORING

and now we can see the containers as the command below show as:

docker stack services MONITORING

To the end, we have to configure grafana in this order:

1) Configure a Data Source going to grafana admin page and create a new Data Source. We have to connect [http://localhost:3000](http://localhost:3000/)and configure the following fields:

* Name: InfluxDB
* Type: InfluxDB
* Http settings:
  + Url: http://influxdb:8086 (Swarm provides a DNS for us)
  + Access: proxy
* InfluxDB Details:
  + Database: telegraf

2) Create our own dashboards and add data to them. Here we have to configure the results that our bash script connectivity\_check.sh has gotten of its execution.

* https://grafana.com/dashboards/1443
* <https://grafana.com/dashboards/1150>