**How to Setup Smartsite (Application) in Docker Compose**

**By Michael Thompson (2/21)**

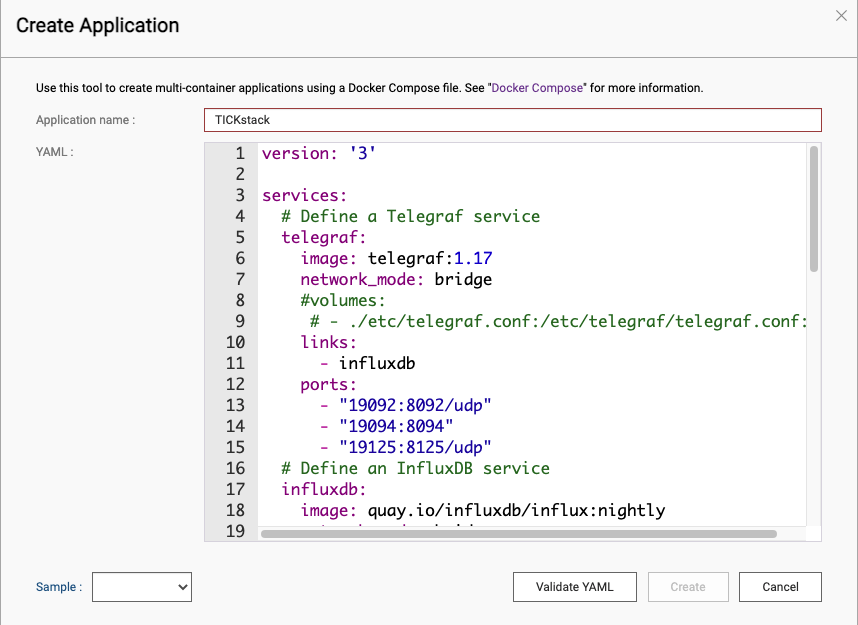
Prerequisites:

* Docker
* Docker-compose.yml containing latest versions of:
  + Telegraf
  + InfluxDB 2
  + Grafana
  + NodeRED (optional)

1. Go to Container Station -> Create -> Create Application



1. Name the application and paste docker-compose.yml into the YAML section



1. In the overview tab, you should now see your application with web and console links



1. TICKstack is now ready to be configured

* Log into influxdb, set up an organisation, bucket and generate a read/write token
* Use this token to connect mqtt to influxdb via Node-Red or Telegraf and to display data in Grafana
* Relevant guides can be found online

Tips:

* Ensure the latest versions are being used for each component of the application
* Ensure network bridges are set up in your docker-compose.yml or you may have connection issues between containers on the same machine
* Ensure volumes/bind mounts are used to avoid losing data

Further Reading:

* <https://www.influxdata.com/blog/tips-for-running-the-tick-stack-using-docker/>

**Sample docker-compose.yml (volumes)**

version: '3'

services:

# Define a Telegraf service

telegraf:

image: telegraf:1.17

network\_mode: bridge

#volumes:

# - ./etc/telegraf.conf:/etc/telegraf/telegraf.conf:ro

links:

- influxdb

ports:

- "19092:8092/udp"

- "19094:8094"

- "19125:8125/udp"

# Define an InfluxDB service

influxdb:

image: quay.io/influxdb/influx:nightly

network\_mode: bridge

volumes:

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

ports:

- "19086:8086"

# Define a Chronograf service

chronograf:

image: chronograf:1.8.9.1

network\_mode: bridge

environment:

INFLUXDB\_URL: http://influxdb:8086

KAPACITOR\_URL: http://kapacitor:9092

ports:

- "19888:8888"

links:

- influxdb

- kapacitor

# Define a Kapacitor service

kapacitor:

image: kapacitor:latest

network\_mode: bridge

environment:

KAPACITOR\_HOSTNAME: kapacitor

KAPACITOR\_INFLUXDB\_0\_URLS\_0: http://influxdb:8086

links:

- influxdb

ports:

- "19092:9092"

grafana:

image: grafana/grafana:latest

network\_mode: bridge

links:

- influxdb

ports:

- 19300:3000

**Sample docker-compose.yml (bind mounts)**

version: '3'

services:

# Define a Telegraf service

telegraf:

image: telegraf:latest

network\_mode: bridge

volumes:

- /share/Container/persistent-data/telegraf/etc/telegraf:/etc/telegraf

user: root

links:

- influxdb

ports:

- "20092:8092/udp"

- "20094:8094"

- "20125:8125/udp"

# Define an InfluxDB service

influxdb:

image: quay.io/influxdb/influx:nightly

network\_mode: bridge

volumes:

- /share/Container/persistent-data/influxdb/root/.influxdbv2:/root/.influxdbv2

user: root

ports:

- "20086:8086"

grafana:

image: grafana/grafana:latest

network\_mode: bridge

links:

- influxdb

ports:

- "20300:3000"

environment:

- GF\_SECURITY\_ALLOW\_EMBEDDING=true

- GF\_AUTH\_ANONYMOUS\_ENABLED=true

volumes:

- /share/Container/persistent-data/grafana/var/lib/grafana:/var/lib/grafana

- /share/Container/persistent-data/grafana/etc/grafana:/etc/grafana

user: root

# Explicitly define the persistent volume for your data storage

#volumes:

# grafana-data:

# telegraf-data:

# influxdb-data: