Dokumentation Kafka Anwendungsszenario

Zum Starten des Anwendungsszenarios muss die Docker-Compose Datei ausgeführt werden (docker-compose up -d).
Anschließend sind der Kafka-Manager über localhost:9000 und Grafana über localhost:3000 verfügbar.

Mosquitto (Port:1883):

Um die Daten an den MQTT Broker senden zu können, muss die IP Adresse im Mita Code angepasst werden.
Nachrichten werden per default nach 30 Minuten gelöscht.

Connector:

Der Connector sendet alle Daten von Mosquitto zum Kafka Cluster.Beginnt das Topic einer Nachricht mit “mqtt.”, so wird der Präfix zu “kafka.” abgeändert.
Als Key bekommen die Records den aktuellen Zeitstempel. Key und Value werden als String serialisiert.

Kafka (Port:9092) & **Zookeeper** (Port:2181):

Die Kafka Topics werden automatisch erstellt, sobald ein Consumer ein Topic subscribed oder Records zu einem Topic gepublished werden.
Topics, die automatisch erstellt werden, haben eine Partition und einen Replikationsfaktor von 1.
Nachrichten werden per default nach 7 Tagen gelöscht.
Soll ein weiterer Kafka Broker hinzugefügt werden, so kann die Docker-Compose Datei um die entsprechenden Zeilen erweitert werden, allerdings dürfen sich Servicenamen nicht doppeln und ein Port kann nicht auf verschiedene Services gemappt werden. Der zweite Kafka Broker könne also beispielsweise kafka_1 genannt werden und durch “9093:9092” auf den Port 9093 gemappt werden.

Kafka-Manager (Port:9000):

Um das Kafka Cluster mit dem Kafka-Manager verbinden zu können muss dieser geöffnet werden. Beim hinzufügen des Clusters kann man diesem einen beliebigen Namen geben. Der Zookeeper Host ist für den Kafka-Manager über zookeeper:2181 verfügbar. JMX Polling kann aktiviert werden (funktioniert aber dennoch nicht).
Anschließend sollten Cluster-Informationen und -Aktionen verfügbar sein und funktionieren.

Telegraf:

Vor dem Starten der Anwendung muss die Config Datei für Telegraf entsprechend angepasst werden. Sollte die Anwendung bereits laufen, so haben Änderungen erst eine Auswirkung bei einem erneuten Start.
Für eine genauere Erklärung der Plugins und zugehöriger Parameter siehe https://docs.influxdata.com/telegraf/v1.14/.
Für diese Anwendung wurden alle nötigen Konfigurationen bereits getroffen, dazu gehört das Ändern der URL für die InfluxDB, der Name der Datenbank, die Kafka Broker (wird ein Broker hinzugefügt, so muss die Config Datei entsprechend erweitert werden), die zu subscribenden Kafka Topics und das Format der Daten (https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_INPUT.md, in diesem Fall InfluxDB Line Protocol).

InfluxDB (Port:8086):

Die Datenbank wird durch Telegraf automatisch erstellt, falls diese noch nicht vorhanden ist. Nachrichten werden per default nie automatisch gelöscht.

Grafana (Port:3000):

Wird der Grafana Container erstmals gestartet, so muss man sich mit dem Nutzernamen “admin” und gleichnamigen Passwort anmelden.
InfluxDB kann als Data Source hinzugefügt werden. Hierzu muss in Grafana für den HTTP URL “http://influxdb:8086” angegeben werden. Bei den InfluxDB Details muss der Name der gewünschten Datenbank angegeben werden. Dieser ist in der Telegraf Config auffindbar (die Datenbank wird durch Telegraf automatisch erstellt, falls noch nicht vorhanden) und lautet ursprünglich “Sensordata”.

Zum Beenden der Anwendung docker-compose stop ausführen. Sollen alle Container und das erstellte Netzwerk gelöscht werden, so kann docker-compose down ausgeführt werden.