z/TPF real-time insights dashboard starter kit readme

Copyright IBM Corporation 2019, 2024

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

NOTE: Before using this information and the product it supports, read the general information under "Notices" in this document.

Contents

This file includes the following information:

1.0 Introduction

The z/TPF real-time insights dashboard starter kit provides an example real-time analytics pipeline for data science analysis. With this pipeline, you can use statistical analysis, machine learning, and other forms of analysis to understand and diagnose system resource usage issues quickly. Understanding and diagnosing system resource usage more quickly can help limit impacts on service-level agreements and optimize business decisions. You also can use this analytics pipeline to feed data to monitors or databases for long-term analysis.

This starter kit includes a script for you to replay previously recorded real IBM test system data into the real-time analytics pipeline. This replay script, environment, and more can help you see the value of the solutions, experiment with various configuration and analysis, and inspire your future implementation of your real-time insights dashboard.

One of the primary goals of this starter kit is to make adoption and setup as easy as possible. Therefore, Docker is used extensively to install and configure the various components. You might need to make some modifications to the installation to suit your environment.

The starter kit also includes components required to collect, analyze and view the results of message analysis tool collections. The z/TPF message analysis tool provides you with the capability to capture and analyze the functions and macros that are used when the system processes a message. You can use this tool to determine where system resources are used when your message is being processed.

2.0 Change history

2019Dec18 Initial version.
2020Feb17 Corrections to readme.
2020Dec11 Enhanced for two server configurations and other improvements, including improved
           modeled CPU calculations with APAR PJ46295 applied to your z/TPF system.
2021Feb24 Added support for z/TPF system-wide JVM monitoring.
2021Jun30 Removal of Apache Spark.
2021Nov18 Message analysis tool initial support.
2022May31 Complex-wide dashboard support.
2022Sep30 Improve MySQL memory usage.
           Improve interface for setting up multiple tpf_zrtmc_analyzers.
2023Mar03 Runtime metrics collection Kafka encryption support
2023Jun30 Docker on Linux on IBM Z support. Use IBM Z and LinuxONE Container Registry.
           Easier to configure installation from trusted or local repositories.
           Provide IBM Semeru Runtime Open Edition 11 support.
           Provide sample pruning.
           Provide name-value pair collection starter kit support for all metrics.
2023Jul19 User-defined metrics support.
2023Oct04 PJ47156 tpf_prepare_configurations.sh might not work in all environments.
2023Oct25 PJ47175 Runtime metrics collection specified time zone support
2024Jun13 PJ47253 z/TPF real-time insights dashboard starter kit improvements
2024Jun28 Runtime metrics collection CDC support (APAR PJ47254)

3.0 Prerequisites

- Red Hat Enterprise Linux (RHEL) Version 7.4 for x86

- Docker 18.09.6 or later, which is available from:
  https://docs.docker.com/install/linux/docker-ce/binaries

- Docker-Compose 1.24.0 or later, which is available from:
  https://docs.docker.com/compose/install/

- A virtual machine (VM) with 2 cores, 4 GB memory, and 200 GB disk space

- Docker containers that use Java requires the following Java software:
  - Linux on an x86 system: IBM Semeru Runtime Open Edition 11
  - Linux on IBM Z: IBM Semeru Runtime Open Edition 11

IMPORTANT NOTE: These instructions and scripts were implemented and tested on a virtual machine that has no other Docker images or containers. If you use these instructions and scripts on a machine that has other Docker images and containers, carefully review each step, script, and other information to ensure no undesirable behaviors occur. Additionally, these instructions assume the ID that is used to implement the installation has "usermod -aG docker your_id" so that the sudo command does not have to be issued for every docker command manually entered or issued from the included scripts.

4.0 Installing the z/TPF real-time insights dashboard starter kit

The installation process uses Docker to download, install and configure the various open-source products, including Grafana, MariaDB or MySQL, Apache Kafka, and various Python packages.

You can also use your own MySQL database. You can find the database settings that are used during internal MySQL testing in the file tpf_db_docker_files/mysql.cnf.
If your MySQL database will be used to process name-value pair collection data, configure your database instance to use the jemalloc library. For more information, see https://www.ibm.com/docs/en/ztpf/latest?topic=collection-configuring-your-database-performance

You can configure the starter kit to use two servers or to use a single-server configuration.

In the z/TPF lab, testing indicated that it is best to split the components between two

servers. The two-server configuration is also the best configuration to use to run real-time runtime metrics collection in production.

The following information describes the two-server configuration:
* The first server is called the tpf_rtmc_server.
  The following docker-compose yaml files are required to start this server:
  * tpf_data_sci/Docker/tpf-insights-dashboard-network.yml
  * tpf_data_sci/Docker/tpf_rtmc_server.yml
  * tpf_data_sci/Docker/tpf_mariadb.yml or tpf_data_sci/Docker/tpf_mysql.yml
  * tpf_data_sci/Docker/tpf_kafka.yml

  The following components run on the tpf_rtmc_server:
  * MariaDB or MySQL
  * Kafka
  * tpfrtmc offline utility for processing real-time runtime metrics collection
  * tpfrtmc offline utility for processing name-value pair collection

  When you run real-time runtime metrics collection on your z/TPF system, configure the endpoint group descriptors to send results to the tpf_rtmc_server. In production environments, implement this server on a Linux on IBM Z system in the same complex as the z/TPF system.

* The second server is called the tpf_analytics_server.
  The following docker-compose yaml files are required to start this server:
  * tpf_data_sci/Docker/tpf-insights-dashboard-network.yml
  * tpf_data_sci/Docker/tpf_analytics_server.yml
  * tpf_data_sci/Docker/tpf_mariadb.yml or tpf_data_sci/Docker/tpf_mysql.yml

  The following components run on the tpf_analytics_server:
  * MariaDB or MySQL
  * Grafana
  * tpf_zrtmc_analyzer Python script
  * tpf_zmatc_analyzer Java package provided by IBM

  Name-value pair collection and message analysis tool results are written to the database on this server because Grafana is configured to view the content of the local database instance.

The single-server configuration is useful for experimentation and running the replay scripts. This configuration uses all of the following docker-compose yaml files:
* tpf_data_sci/Docker/tpf-insights-dashboard-network.yml
* tpf_data_sci/Docker/tpf_rtmc_server.yml
* tpf_data_sci/Docker/tpf_mariadb.yml or tpf_data_sci/Docker/tpf_mysql.yml
* tpf_data_sci/Docker/tpf_kafka.yml
* tpf_data_sci/Docker/tpf_analytics_server.yml

The single-server configuration installs all components on a single x86 Linux server.

The instructions for installing and using the starter kit provide details for how to install and use both server configurations.

The credentials used in various scripts are defined in the tpf_data_sci/tpf_default_credentials.txt file. Change the passwords to a value that is more secure for your environment. When you change the passwords, you must make updates to various files in the tpf_data_sci/Docker directory.

All scripts included in the starter kit are written to run in the bash shell.

4.1 Procedure for installing the starter kit

If you want to set up a two-server configuration, complete the following steps on both the tpf_rtmc_server and tpf_analytics_server, unless the step explicitly indicates to do the step only on one of the servers. Otherwise, if you want to set up a single-server configuration, complete the following steps on a single server.

1. Download or use FTP to transfer the z/TPF real-time insights dashboard starter kit tar file to your home directory on your Linux machine.

2. Extract the package: tar -xf tpf_realtime_insights_dashboard.tar

3. For the tpf_rtmc_server or single-server configurations, copy the base/tpfrtmc/bin/tpfrtmc.tar.gz file in binary format from your z/TPF source repository to the tpf_data_sci/Docker/tpf_rtmc_docker_files/ directory. Use the following command to extract the content from the tar file:

   ```
   tar -xf tpfrtmc.tar.gz
   ```

4. For the tpf_analytics_server or single-server configurations, copy the base/tpfrtmc/bin/tpf_zmatc_analyzer.tar.gz file in binary format from your z/TPF source repository to the tpf_data_sci/Docker/tpf_zmatc_analyzer_docker_files/ directory. Use the following command to extract the content from the tar file:

   ```
   tar -xf tpf_zmatc_analyzer.tar.gz
   ```

5. Define your Apache Kafka hosts, encryption settings, topic settings, and programmatic variables in the tpf_data_sci/user_files/kafka_hosts.yml file. For more information about how to configure this file, see the comments in the file.

6. If Python 3.8 and the pyyaml library, which are used by the tpf_prepare_configurations.sh script, are not installed on your system, enter the following commands to install them:
   ```
   sudo yum install python38
   sudo python3 -m pip install --upgrade pyyaml
   ```

7. Change your directory to the Docker directory: cd tpf_data_sci/Docker

8. For the tpf_rtmc_server and tpf_analytics_server, configure your server:

   The ./tpf_prepare_configurations.sh is used to configure your server. The settings that are used are set in the file: tpf_data_sci/user_files/tpf_prepare_configurations.yml. Modify the tpf_prepare_configurations.yml file with your desired settings. The tpf_prepare_configurations.yml file contains the following parameters:

   USER_ID
   This parameter is required and is used when logs, data, files and so on are created. This parameter lessens the requirements for root or sudo access.

   RTMC_SERVER
   This parameter is required. Specify the hostname of your single server or the hostname of the tpf_rtmc_server in dual server configurations.

   Note: You can use the hostname -f command to determine the full hostname of tpf_rtmc_server and tpf_analytics_server to pass as parameters to this script.

   ANALYTICS_SERVER
   This parameter is required. Specify the hostname of your single server or the hostname of the tpf_analytics_server in dual server configurations.

   Note: You can use the hostname -f command to determine the full hostname of tpf_rtmc_server and tpf_analytics_server to pass as parameters to this script.

   LOGGING
   This parameter is optional. Specify this parameter to set the finest debug logging level for the tpfrtmc and zmatc_analyzer offline utilities.

   DB_TYPE
   This parameter is required. Specify a value of mariadb or mysql to indicate whether your database on the server is a MariaDB or MySQL database.
   The default value is mariadb.

LINUX_PLATFORM
  This parameter is required. Specify a value of x86 if your platform is Linux on an x86
  system; specify a value of linux_on_ibm_Z if your platform is Linux on IBM Z.
  The default value is x86.

  For Linux on IBM Z servers:
    1. Follow these directions to sign up for an IBM Cloud ID and create an API key:
       https://ibm.github.io/ibm-z-oss-hub/main/main.html
    2. Before you issue a Docker-compose command, issue the Docker login command with
       your API key:
       docker login -u iamapikey icr.io

    Note: A MySQL configuration is not available for Linux on IBM Z because there is no
    viable container image provided by IBM or other vendors for the S/390 architecture.
    Also, MySQL does not work because the Rust compiler is required to build the
    cryptography dependency that is required by MySQL when building the
    tpf_zrtmc_analyzer python scripts. The Rust compiler for S/390 architectureis built
    for the GNU C library (glibc) and Python 3.8 is based on Alpine Linux, which uses
    the musl library. There is no Cargo/Rust dependency available for the musl library
    in the apk installer or the Rust websites. The MariaDB image that is used is pulled
    from Docker Hub because the IBM container registry does not provide one.

USE_TRUSTED_REPOSITORY
  This parameter is required. Specify this parameter to indicate whether your server
  installs open-source dependencies based on your default open-source configuration
  or from a local or trusted server.

  Specify a value of no or yes:
    no - Use default open-source configured locations. This is the default value.
    yes - Use a local or trusted repository. The starter kit relies on the apk and pip
          open-source dependency package installers. Modify the following files to
          point to your local or trusted repositories:
            apk: tpf_data_sci/user_files/local_apk_repository
            pip: tpf_data_sci/user_files/pip.conf
          For more information, see the following web pages:
            apk: https://wiki.alpinelinux.org/wiki/Alpine_Package_Keeper#Add_a_local_Package
            pip: https://pip.pypa.io/en/stable/topics/configuration/ and
                 https://pip.pypa.io/en/stable/cli/pip_install/#cmdoption-0

  Note: For MySQL configurations, the rpm package installer is used to install the
  jemalloc library. The tpf_data_sci/user_files/mysql_jemalloc_library.yml includes
  the default location and version of the jemalloc library. If you use MySQL and want
  to use a local or trusted provider instead, modify the
  tpf_data_sci/user_files/mysql_jemalloc_library.yml file to point to your local or
  trusted repositories. For more information see:
  https://yum.oracle.com/repo/OracleLinux/OL8/developer/EPEL/x86_64/index.html

DOCKER_REGISTRY
  This parameter is optional. Specify this parameter to indicate whether your server
  installs Docker containers from a local or trusted registry.
  Specify the hostname:port of your Docker registry.

  You can create a local Docker registry by the instructions on the following web page:
  https://docs.docker.com/registry/deploying/

IMPLEMENT_SAMPLE_PRUNING
  This parameter is optional. Specify this parameter to indicate whether to use the IBM
  pruning installation mechanism. Specify a value of yes or no. The default value is no,
  which means that the IBM sample pruning or the customer-defined pruning that is created
  based on the IBM sample pruning is not installed.

  IBM sample pruning provides the following capabilities:
    - The average, maximum and minimum per second for each one-minute time frame are
      created for all metrics in IBM provided tables.
    - Stored procedures are created to perform pruning.

- The tpf_sample_prune_daily event is created to run at midnight UTC time.
- You can modify pruning parameters in the
  tpf_data_sci/user_files/tpf_sample_pruning/tpf_create_prune_event.sql file.
  You can modify the X and Y parameters in the following stored procedure call:
    call tpf_sample_prune_data(X, Y); where:
  X is the number of days to keep per second data before the data is pruned at
    midnight UTC time.
  Y is the number of days to keep per minute data (average, maximum, and minimum
    per second in a one-minute time frame) before the data is pruned at midnight
    UTC time.
  After Y days, all data has been pruned.
- You can add average, maximum, and minimum metrics and additional processing for
  user-defined tables by modifying the files in the
  tpf_data_sci/user_files/tpf_sample_pruning directory. If you add additional SQL
  files, ensure that you modify the installation commands:
    tpf_data_sci/user_files/tpf_sample_pruning/tpf_sample_pruning_SQL_install_commands.sh

To use customer-defined pruning, follow the code structure in the
tpf_data_sci/user_files/tpf_sample_pruning file and modify the IBM sample pruning
to meet your needs. If you add additional SQL files, ensure that you modify the
installation commands:
  tpf_data_sci/user_files/tpf_sample_pruning/tpf_sample_pruning_SQL_install_commands.sh

CONFIGURE_KAFKA_CONTAINERS
  This parameter is required. Specify this parameter to indicate whether to configure
  various files to create the required Kafka containers. Specify a value of yes or no.
  For ease of use in testing configurations, the default value is yes. If your analytics
  pipeline uses an existing Kafka installation, set this parameter to no.

  Note: The provided Kafka container configurations are for test purposes only.
  The provided Kafka configuration does not provide high availability and might not
  conform to your corporate production standards. Do not use the provided Kafka
  configuration for production installations.

CONFIGURE_KAFKA_CONTAINER_PORT
  This parameter specifies the port to use for connecting to Kafka. If you specify yes
  for the CONFIGURE_KAFKA_CONTAINERS parameter, you must specify this parameter.
  The default value is 9093.

KAFKA_HOST_SETTING
  This parameter is required. Specify one of the following values for the kafka parameter:
  use   - Use this option if you modified the tpf_data_sci/user_files/kafka_hosts.yml
           file to specify your host and other configuration variables. This is the
           default option.
  update - Use this option if you did not modify the
           tpf_data_sci/user_files/kafka_hosts.yml file. You can use this option for
           small test configurations that use the included tpf_kafka.yml file to build
           Kafka containers.

IMPLEMENT_SAMPLE_UDM
  This parameter is optional. Specify this parameter to indicate whether to use the IBM
  sample user-defined metrics support on your server. The default is no.
  NOTE: No pruning sample is provided for the IBM sample user-defined metrics data.
  For an example of how to implement prining for your user-defined metric data, see the
  IBM sample pruning implementation. For more information about how to implement
  user-defined metrics, see
  https://www.ibm.com/docs/en/ztpf/latest?topic=metrics-user-defined-tutorial

HSC_ENCRYPTION_FILES_FOR_RTMC
  This parameter is optional. Specify the directory where the high speed connector
  encryption files are stored. These files are used to create an encrypted socket when
  the tpfrtmc offline utility receives data from the z/TPF system. The files in this
  directory are copied into the tpf_rtmc_realtime container. For more information on
  encrypting the data that is sent from the z/TPF system, see

https://www.ibm.com/docs/en/ztpf/latest?topic=encrypting-connections-between-ztpf-system-tpfrtmc-offline-utility

TIME_ZONE_TEXT
This parameter is optional. Specify this parameter only if you want the timestamps in all containers, logs, and so on to be set to a specified time zone. If this parameter is specified, the default time zone of the Grafana dashboards is also changed to the specified time zone.

The default value is UTC, which means that the docker containers will be created with a UTC timestamp and time zone. As such, logs will have UTC timestamps for the time zone of your server. Grafana dashboards will default to the time zone of the browser.

Specify a value such as America/New_York. You must specify a value that can be found in both of the following web pages:
https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/time/ZoneId.html
https://dev.mysql.com/downloads/timezones.html

GRAFANA_ORG_ID
This parameter is optional. Specify this parameter if you will install the sample analytics pipeline dashboards in an enterprise Grafana instance instead of using the Grafana container definition provided. Set the GRAFANA_ORG_ID parameter to the organization ID that is used for the IBM provided data sources and dashboards.

Run the tpf_prepare_configurations script to modify the files in the tpf_data_sci/Docker/tpf_grafana_docker_files/provisioning directory to have the desired organization ID.

For more information, see the Granfana provisioning documentation (https://grafana.com/docs/grafana/latest/administration/provisioning/).

Follow your administrator's guidance to install these dashboards and data sources. For example, you might perform the following steps to install the updated provisioning files in an enterprise Grafana installation.
1. Identify your desired organization ID from the Grafana -> Administration -> Organizations dashboard.
2. Set the GRAFANA_ORG_ID parameter to the desired organization ID.
3. Run the tpf_prepare_configurations.sh script to modify the dashboard provisioning files for the organization ID.
4. Stop your Grafana instance.
5. Copy the contents of tpf_data_sci/Docker/tpf_grafana_docker_files/provisioning directory into the provisioning directory of your Grafana instance.
6. Start your Grafana instance.

Container Repository Definitions
In this section, specify which versions of open-source components will be used for building containers.

After the tpf_prepare_configurations.yml is configured, you can issue the prepare script:

./tpf_prepare_configurations.sh

To view which files are edited and what changes are made to satisfy the settings that you want, see the tpf_prepare_configurations.sh script.

9. Use the docker-compose command to start the docker containers:

For the tpf_rtmc_server or single-server configurations, take one of the following actions:
* If you are using a MySQL database, enter the following command:
docker-compose --file tpf-insights-dashboard-network.yml --file tpf_mysql.yml --file tpf_kafka.yml up -d --build

* If you are using a MariaDB database, enter the following command:
  docker-compose --file tpf-insights-dashboard-network.yml --file tpf_mariadb.yml --file tpf_kafka.yml up -d --build

For the tpf_analytics_server, take one of the following actions:
* If you are using a MySQL database, enter the following command:
  docker-compose --file tpf-insights-dashboard-network.yml --file tpf_mysql.yml up -d --build
* If you are using a MariaDB database, enter the following command:
  docker-compose --file tpf-insights-dashboard-network.yml --file tpf_mariadb.yml up -d --build

For more information about managing containers and images, see the Docker and docker-compose documentation: https://docs.docker.com/

10. Set up the database tables and stored procedures by running the SQL script:

    For the tpf_rtmc_server, tpf_analytics_server and single-server configurations:

      ./tpf_setup_db.sh

11. Run the following script:

    For the tpf_rtmc_server or single-server configurations:
      ./tpf_create_kafka_topics.sh

    This script creates the Apache Kafka topics.

12. Run the following script:

    For the tpf_rtmc_server or single-server configurations:
      ./tpf_modify_kafka_topics.sh hostname:port
    where hostname:port is a host specified in the tpf_data_sci/user_files/kafka_hosts.yml in step 5.

    This script modifies the Apache Kafka topics based on the modify_script_variables settings that are specified for your host in the tpf_data_sci/user_files/kafka_hosts.yml file.

13. Use docker-compose to start the tpfrtmc docker containers:

    For the tpf_rtmc_server or single-server configurations:
      docker-compose --file tpf-insights-dashboard-network.yml --file tpf_rtmc_server.yml up -d --build

14. [Optional] Configure tpf_zrtmc_analyzer instances to support multiple z/TPF systems.

    If you plan to have only one tpf_zrtmc_analyzer, the tpf_prepare_configurations.sh already does the required setup and you can skip this step. Otherwise, complete this step on the tpf_analytics_server or single-server.

    For example, consider the following configuration:

    US z/TPF Complex > US tpfrtmc > US Kafka > US tpf_zrtmc_analyzer 1 > US MariaDB > US Grafana
    EU z/TPF Complex > EU tpfrtmc > EU Kafka > US tpf_zrtmc_analyzer 2 > US MariaDB > US Grafana

    With this configuration, you can use the US Grafana instance to see real-time data feeds from both the US and EU z/TPF complexes.

    You must decide if you want the EU and US data to appear on dashboards together. If so, the two tpf_zrtmc_analyzer profiles will use the same database name. Otherwise, they will use different database names and you must define an additional data source in Grafana for the EU database name. In this example, the US and EU data will appear on the dashboards together and will use the same database name.

Complete the following steps:

1. Make a US copy of the tpf_zrtmc_analyzer_profile.yml file.

   ```
   cp
   tpf_data_sci/Docker/tpf_zrtmc_analyzer_docker_files/profile/tpf_zrtmc_analyzer_profile.y
   ml

   tpf_data_sci/Docker/tpf_zrtmc_analyzer_docker_files/profile/tpf_zrtmc_analyzer_profile_U
   S.yml
   ```

   If the tpf_prepare_configurations.sh was run for the US system, no further changes are required to tpf_zrtmc_analyzer_profile_US.yml. However, you might want to change the logging > file to tpf_zrtmc_analyzer_US.log. For example, change the group ID to a unique value if the tpf_zrtmc_analyzer is running in the US for US data: tpf-zrtmc-analyzer-US.

2. Make an EU copy of the tpf_zrtmc_analyzer_profile.yml file.

   ```
   cp
   tpf_data_sci/Docker/tpf_zrtmc_analyzer_docker_files/profile/tpf_zrtmc_analyzer_profile.y
   ml

   tpf_data_sci/Docker/tpf_zrtmc_analyzer_docker_files/profile/tpf_zrtmc_analyzer_profile_E
   U.yml
   ```

3. Modify the Kafka host and other Kafka settings for the EU Kafka in the tpf_zrtmc_analyzer_profile_EU.yml. In this example, the US and EU data will appear on the dashboards together, so database settings are left unchanged. Change the logging > file to tpf_zrtmc_analyzer_EU.log. You can make additional changes to the database, desired analysis, and so on, as needed. For example, change the group ID to a unique value if the tpf_zrtmc_analyzer is running in the US for EU data: tpf-zrtmc-analyzer-US.

4. Modify tpf_data_sci/Docker/tpf_analytics_server.yml to have two tpf-zrtmc-analyzer services: one for US and one for EU.
   Modify the following values:
   - service name
   - container_name
   - hostname
   - ZRTMC_ANALYZER_PROFILE

   For example, US or EU are appended to each element below:

   ```
   tpf-zrtmc-analyzer-US:
      container_name: tpf-zrtmc-analyzer-US
      hostname: tpf-zrtmc-analyzer-US
      build:
         context: ./tpf_zrtmc_analyzer_docker_files
         dockerfile: Dockerfile
      image:
         tpf-zrtmc-analyzer_img
      networks:
         - tpf-insights-dashboard-network
      volumes:
         - ./tpf_zrtmc_analyzer_docker_files/profile:/tpf_zrtmc_analyzer/profile
         - ./tpf_zrtmc_analyzer_docker_files/logs:/tpf_zrtmc_analyzer/logs
      logging:
         driver: "json-file"
         options:
            max-size: "10m"
            max-file: "3"
      environment:
         -
         ZRTMC_ANALYZER_PROFILE=/tpf_zrtmc_analyzer/profile/tpf_zrtmc_analyzer_profile_US
         .yml
   ```

```
        tpf-zrtmc-analyzer-EU:
           container_name: tpf-zrtmc-analyzer-EU
           hostname: tpf-zrtmc-analyzer-EU
           build:
              context: ./tpf_zrtmc_analyzer_docker_files
              dockerfile: Dockerfile
           image:
              tpf-zrtmc-analyzer_img
           networks:
              - tpf-insights-dashboard-network
           volumes:
              - ./tpf_zrtmc_analyzer_docker_files/profile:/tpf_zrtmc_analyzer/profile
              - ./tpf_zrtmc_analyzer_docker_files/logs:/tpf_zrtmc_analyzer/logs
           logging:
              driver: "json-file"
              options:
                 max-size: "10m"
                 max-file: "3"
           environment:
              -
              ZRTMC_ANALYZER_PROFILE=/tpf_zrtmc_analyzer/profile/tpf_zrtmc_analyzer_profile_EU
              .yml
```

15. Use docker-compose to start the remaining docker containers:

    For tpf_analytics_server or single-server configurations:
      docker-compose --file tpf-insights-dashboard-network.yml --file tpf_analytics_server.yml
      up -d --build

    Note: The tpf_zrtmc_analyzer connects to both Kafka and database upon startup. Any data
    that is available on the configured Kafka topics will begin to be processed.
    The tpf_zmatc_analyzer performs analysis on any available message analysis tool results
    in the database on the tpf_analytics_server.

16. [Optional] If you have a firewall active, you might need to ensure that the ports
    specified in the yml files are open. For example, enter the following command for each
    port that is exposed by the yml files:
      sudo firewall-cmd --zone=public --add-port=<portID>/tcp --permanent

      where <portID> represents the following ports:
        For MariaDB or MySQL: 3306
        For Grafana: 3000
        For Kafka: 2181, 9092, 9093, 8082, 8000
        For tpfrtmc: 9090

    Reload the firewall by entering the following command:
      sudo firewall-cmd --reload

    Note: You can modify all of the ports before entering the reload command. Additionally,
    the tpf_data_sci/Docker/tpf_open_firewall_ports.sh script is provided to process all
    of these commands for you for the default ports.

After the previous steps are completed, the analytics pipeline is fully functional.

You can use the following instructions to test the installation on the tpf_analytics_server
or single-server configuration.
1. Test Apache Kafka by opening the KafkaUI in a browser at
http://your.rtmc.server.name.com:8000
2. Test the tpf_zrtmc_analyzer Python script by checking the logs for messages that
    indicate successful connections to both Kafka and the database. The default path of the
    log file is as follows:
       tpf_data_sci/Docker/tpf_zrtmc_analyzer_docker_files/logs/tpf_zrtmc_analyzer.log
3. Run the replay scripts as described in section 6.0.
4. Test the database and MySQL and Grafana by completing the following steps:
    a. Open Grafana in a browser at: http://your.analytics.server.name.com:3000

Grafana is configured to allow anonymous login so that no credentials are required to view the dashboards. If you want to make changes, you can log in as the admin using the sign-in button on the lower left. (Use the credentials for Grafana that are listed at the beginning of this section.)

b. Open a dashboard. For example, click the top left: Home > ZRTMC Results > 02. Correlation Analysis.

Grafana is configured to allow anonymous login so that no credentials are required to view the dashboards. If you want to make changes, you can log in as the admin using the sign-in button on the lower left. (Use the credentials for Grafana that are listed at the beginning of this section.)

The results of the analysis of the replay script data are displayed.

5.0 Customizing the z/TPF real-time insights dashboard starter kit

Grafana

Sign into Grafana as the admin by using the lower left button on the dashboard and the credentials that are listed in section 4.0. Grafana 6.5.2 will prevent you from modifying provisioned dashboards (even though the dashboards are marked as modifiable).
You can export a dashboard, import the dashboard, and modify the dashboard to your specifications. These displays are very closely tied to the stored procedures in the database. To save any changes you make to a dashboard, you must export the dashboard in JSON format to the appropriate dashboard file location in the Grafana Docker container volume. These files are located in the tpf_data_sci/Docker/tpf_grafana_docker_files/dashboards directory.

MariaDB and MySQL

You can modify the SQL stored procedures in the stored procedures sql files in the tpf_data_sci/Docker/tpf_db_docker_files directory or the table definitions in tpf_data_sci/Docker/tpf_db_docker_files/tpf_create_tables.sql file.
The stored procedures perform correlation and other forms of analysis.
For your convenience, the tpf_data_sci/Docker/tpf_db_docker_files/tpf_drop_tables.sql and tpf_data_sci/Docker/tpf_db_docker_files/tpf_delete_all_data.sql files are included.

Use the following commands to set up the database again.

* docker container stop tpf-db
* docker-compose --file tpf-insights-dashboard-network.yml --file tpf_mariadb.yml up -d --build tpf-db
  or
  docker-compose --file tpf-insights-dashboard-network.yml --file tpf_mysql.yml up -d --build tpf-db
* ./tpf_setup_db.sh

tpf_zrtmc_analyzer Python script

The tpf_zrtmc_analyzer Python script is responsible for processing CDC, JVM, and name-value pair collection data. This code is located in the tpf_data_sci/Docker/tpf_zrtmc_analyzer_docker_files directory. You can modify how the tpf_zrtmc_analyzer Python script handles the data that arrives from Kafka.
The configuration profile, profile/tpf_zrtmc_analyzer_profile.yml, specifies the Kafka and database connection parameters, the logging configuration, the active data sources for consumption, the active data types for z/TPF processing, and the application dimensions.
You can write code to do your own processing on the data that arrives from Kafka in addition to or instead of the z/TPF processing. Implement the user exit interface in tpf_user_exit/tpf_zrtmc_analyzer_user_exit.py and place all of your code in the tpf_user_exit directory or a subdirectory of the tpf_user_exit directory. For more details, see the comments in the tpf_zrtmc_analyzer_profile.yml and tpf_zrtmc_analyzer_user_exit.py files.

Run one of the following commands when your changes are complete. Include the "--build" flag if you make changes to the Python code. Otherwise, changes to the configuration profile only require a container to be started (or restarted).

* docker-compose --file tpf-insights-dashboard-network.yml --file tpf_analytics_server.yml up -d --build tpf-zrtmc-analyzer
* docker-compose --file tpf-insights-dashboard-network.yml --file tpf_analytics_server.yml up -d tpf-zrtmc-analyzer

6.0 Running the replay scripts

To run the replay script and see the z/TPF real-time insights dashboard in action, complete the following steps:

1. On the tpf_rtmc_server or single-server configuration, change your directory to the Docker directory:
   cd tpf_data_sci/Docker

2. On the tpf_rtmc_server or single-server configuration, start the replay script:
   ./tpf_start_replay_script.sh scenario_name [port]

   where scenario_name is the name of the scenario to run. Enter the following command to see which scenarios are available:
   ./tpf_start_replay_script.sh

   The following scenarios are provided with the starter kit.
   o scenario_lowVolTraffic:
   View the Grafana dashboard by clicking Home > ZRTMC Results > 02. Correlation Analysis. Select "Message Type, SubType, Origin" as the Application Dimension. After 15 minutes of baseline data, the message rate from the low volume message type increases slightly corresponding to a rise in CPU utilization. The "Message Type, SubType, Origin Rate Correlated to Actual System CPU" panel indicates that name-value pair collection data is insufficient for the [Shopping, Air, Terminal] horizontal name-value pair combination. Change the Analysis Type to Aggregate to see the correlation highlighted.

   The ./tpf_start_replay_script.sh script uses the tpf_data_sci/user_files/kafka_hosts.yml file to determine how to connect to Kafka. For example, what security settings to use. The ./tpf_start_replay_script.sh script looks for the name of the local machine as host:9093 in the tpf_data_sci/user_files/kafka_hosts.yml file. If you specify the optional [port] parameter, that port is used when searching for the correct host to use.

   For more information about running the replay script, see the ./tpf_data_sci/tpfReplayScript/README.txt file that is included in the starter kit download package.

3. Open Grafana in a browser at http://your.server.name.com:3000 or http://your.analytics.server.name.com:3000

4. Open a dashboard. For example, click the top left:
   Home > ZRTMC Results > 02. Correlation Analysis.
   Set the time picker to Last 15 minutes.

The following process occurs:
1. The tpf_data_sci/tpfReplayScript/tpf_ReplayDiskToKafka.jar file simulates real-time data arriving in Kafka by transferring data from file to Kafka in time sequence and simulating real-time collection durations.

2. The processing that runs in the tpf_zrtmc_analyzer Python script in tpf_data_sci/Docker/tpf_zrtmc_analyzer_docker_files/tpf_zrtmc_analyzer.py pulls the data in real time from Kafka, performs some calculations, and writes the results to the database.

3. The Grafana dashboards are set up to automatically refresh.
   When the dashboard refreshes, it will process a variety of analyses that are implemented in SQL and SELECT statements to display the analyzed data.

7.0 Running real-time runtime metrics collection

To process live data from the z/TPF system and see the z/TPF real-time insights dashboard in action, complete the following steps:

1. Define endpoint group descriptors.
   For more information, see the instructions in IBM Documentation at

   https://www.ibm.com/docs/en/ztpf/latest?topic=icrmc-defining-endpoint-groups-real-time-runtime-metrics-collection

2. [Optional] Start a Java application configured for monitoring.
   For more information, see the information in the IBM Documentation at

   https://www.ibm.com/docs/en/ztpf/latest?topic=guide-monitor-java-applications-across-ztpf-system

3. Enter the ZRTMC command to start real-time runtime metrics collection.
   For more information, see the ZRTMC command information in IBM Documentation at

   https://www.ibm.com/docs/en/ztpf/latest?topic=zz-zrtmc-manage-real-time-runtime-metrics-collection-processing

   Note: The following CDC data types that are used by the starter kit require a frequency
   of one second:

   - CDC_SYSTEM_BLOCK
   - CDC_SYSTEM_MESSAGE
   - CDC_TCPIP
   - CDC_ISTREAM
   - CDC_COMMON_DEPLOY_FILES
   - CDC_SERVICE

4. Open Grafana in a browser at http://your.server.name.com:3000 or
   http://your.analytics.server.name.com:3000

5. Open a dashboard. For example, click the top left:
   Home > ZRTMC Results > 02. Correlation Analysis.

6. If runtime metrics collection is sending JVM data from the z/TPF system, open a
   ZRTMC JVM dashboard. For example:
   Home > ZRTMC JVM > 01. JAM Summary.

Three Grafana dashboards are included that provide education about the analysis performed,
columns, mathematical references, and a typical user story of how to use the dashboards.
You can find these dashboards in the following locations:
Home > ZRTMC Results > Education 1: Basic usage guidance
Home > ZRTMC Results > Education 2: The details
Home > ZRTMC JVM > Education 1: JVM Monitoring

8.0 Running message analysis tool collection

To process live data from the z/TPF system and see the z/TPF real-time insights dashboard
in action, complete the following steps:

1. Define endpoint group descriptor.
   For more information, see the instructions in IBM Documentation at

   https://www.ibm.com/docs/en/ztpf/latest?topic=icrmc-defining-endpoint-groups-message-analysis-tool

2. Define a configuration file and enter the ZMATC command to start message analysis
   collection. For more information, see the ZMATC command information in IBM Documentation
   at https://www.ibm.com/docs/en/ztpf/latest?topic=zz-zmatc

3. Open Grafana in a browser at http://your.server.name.com:3000 or
   http://your.analytics.server.name.com:3000

4. Open a dashboard. For example, click the top left:
   Home > ZMATC > 01. Collections.

6. Select a target UOWID and click the dropdown menu in the upper-right corner of the dashboard entitled "Select Target Checkbox and Open Dashboard" to navigate to another dashboard to perform analysis of your target.

Grafana dashboards are included that provide education about the analysis performed, columns, mathematical references, and a typical user story of how to use the dashboards. You can find these dashboards in the following locations:
Home > ZMATC Results > 00. START HERE
Home > ZMATC Results > Education 1. Message Analysis Tool

9.0 Viewing name-value pair collection results

1. Capture name-value pair collection results.
   For more information, see the instructions in IBM Documentation at
   https://www.ibm.com/docs/en/ztpf/latest?topic=data-running-runtime-metrics-collection

2. On the tpf_rtmc_server or single-server configurations, create a subdirectory in the tpf_data_sci/Docker/tpf_rtmc_docker_files/volumes/tape/binary-tapes/ directory if one does not already exist.

   For example:
   mkdir tpf_data_sci/Docker/tpf_rtmc_docker_files/volumes/tape/binary-tapes/preprodtest

   For more information about collection group directories, see the runtime metrics collection information in IBM Documentation at
   https://www.ibm.com/docs/en/ztpf/latest?topic=data-running-runtime-metrics-collection

3. Copy name-value pair collection binary tape files into your subdirectory, such as tpf_data_sci/Docker/tpf_rtmc_docker_files/volumes/binary-tapes/preprodtest

   Runtime metrics collection automatically detects and processes your tape and creates the results in the TPFNVPCDB database on the tpf_analytics_server or the single server.

4. Open Grafana in a browser at http://your.server.name.com:3000 or http://your.analytics.server.name.com:3000

5. Open a dashboard. For example, click the top left:
   Home > ZCNPV Results > Trends

10.0 Known problems and workarounds

There are many components in use in this starter kit. The component versions that are included and used by this starter kit were stable at the time of release.
To use the latest versions of these components, remove or modify the version numbers throughout the files in the tpf_data_sci/Docker/* directory.

11.0 Other sources of information

https://grafana.com/docs/
https://mariadb.com/kb/en/library/documentation/
https://kafka.apache.org/documentation/

The JAR files for the replay script include the source in case you would like to modify it.

12.0 Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

12.1 Trademarks

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Red Hat®, JBoss®, OpenShift®, Fedora®, Hibernate®, Ansible®, CloudForms®, RHCA®, RHCE®, RHCSA®, Ceph®, and Gluster® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

12.2 Warranty

This package is provided on an "as is" basis. There are no warranties, express or implied, including the implied warranties of merchantability and fitness for a particular purpose. IBM has no obligation to provide service, defect correction, or any maintenance for the package. IBM has no obligation to supply any updates or enhancements for the package to you even if such are or later become available.

12.3 Third Party License and Terms

Apache Software License 2.0

This package includes some or all of the following software that IBM obtained under the Apache License Version 2.0:

- Apache Commons Lang
- Apache Kafka (kafka-clients)
- cryptography
- GSON
- Jackson Core
- Jackson Databind
- Python client for Apache Kafka (kafka-python)
- SnakeYAML

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

END OF APACHE SOFTWARE LICENSE 2.0 NOTICES AND INFORMATION
==========

MIT License

This package includes some or all of the following software that IBM obtained under the MIT License:

- PyMySQL: Copyright (c) 2010, 2013 PyMySQL contributors
- PyYAML: Copyright (c) 2017-2021 Ingy döt Net; Copyright (c) 2006-2016 Kirill Simonov
- SLF4J API Module (slf4j-api): Copyright (c) 2004-2022 QOS.ch Sarl (Switzerland)

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

END OF MIT LICENSE NOTICES AND INFORMATION
==========