vmware®

VMware Smart Assurance K4M 1.0.0

Installation and Administration Guide

VMware Smart Assurance K4M 1.0.0

Installation and Administration Guide

Published in the USA April 2019 VMware Smart Assurance K4M 1.0.0 Installation and Administration Guide

Abstract

This user guide describes installation and administration of the VMware Smart Assurance K4M product.

REV 190422

Table of Contents

List of Figures

Chapter 1 Introduction

About this document

This document describes the installation, administration and troubleshooting details for the VMware Smart Assurance K4M product. The product will be referred to as VMware K4M or simply K4M hereafter.

Related documents include:

- For details regarding the K4M product please refer to the following documents:
  - **VMware Smart Assurance K4M 1.0.0 User Guide** – Contains several topics including:
    - Features
    - Concepts
    - Architecture
    - REST API.
  - **VMware Smart Assurance K4M 1.0.0 KPI Designer User Guide** – This document describes how to create and deploy KPI definitions and KPI streams via a Graphical User Interface.
- Installation and operation details of the KPI Designer are available in the **VMware Smart Assurance UI documentation.** Though part of the K4M product, the KPI Designer is installed as part of the VMware Smart Assurance UI.

NOTE: This guide assumes the reader is familiar with K4M's, concepts and architecture.

Chapter 2 Installation and Verification

Installation Components

K4M comprises the following components:

- K4M REST Service
- KPI Engine
- KPI Configuration Store
- KPI Designer Graphical User Interface (GUI)

Installation, configuration and verification of all above components **except** the KPI Designer GUI is described here.

The KPI Designer GUI is included in the VMware Smart Assurance UI. Please refer to the **VMware Smart Assurance UI documentation** for details.

K4M will integrate with VMware Smart Assurance (VSA) via a Kafka broker.

System Requirements

VMware K4M should be installed on a single host/node. We will refer to this host as **K4M_HOST** and the host the K4M Designer GUI is installed as the **KPI_GUI_HOST.**

1. Minimum System Requirements
    i. Operating system: Red Hat Enterprise Linux (RHEL) or CentOS 6.9, 6.10, 7.5 or 7.6. A fresh installation of the operating system is recommended to avoid conflict with previously installed packages.
    ii. CPU: 2.0 GHz.
    iii. Memory: 16GB
    iv. Hard disk space recommended: 100GB
2. MongoDB version **4.0.4**. This guide assumes that MongoDB is running on the same host as VMware K4M. Modify the instructions, if it is on a different host.
3. Internet connectivity, to download dependent packages and software.
4. Connectivity to Kafka broker used by VMware Smart Assurance. Ensure that DNS is configured in your environment, or that you have statically configured the Fully Qualified Domain Name (FQDN) of the Kafka broker on your K4M host (in the /etc/hosts file).

5. VMware K4M creates separate user account during installation. This should be used primarily for all operations. Root access required for few steps, will be specified.

Installation Steps

To install K4M:

1. Running installer – copies and expands software on host.
2. Set up, configuration and launch - Includes downloading open-source dependencies.
3. Verify installation – Automated verification of all services.

Running the Installer

1. Login as root

2. Copy installer file “vmware-k4m-VERSION-nogui-installer.bin” to host. This location is INSTALLER_DIR.

3. Change to INSTALLER_DIR. Modify permissions
   ```
   $ chmod +x vmware-k4m-VERSION-nogui-installer.bin
   ```

4. Launch the installer and follow the prompts on the terminal till the installer completes installation
   ```
   $ ./vmware-k4m-VERSION-nogui-installer.bin
   ```

5. By default, VMware K4M is installed under /opt. Can be modified, must enter the absolute path to your location AND modify scripts as mentioned in subsequent steps. Recommend to use default.

6. Installer finishes with messages below.

   ```
   Congratulations. VMware K4M 1.0.0 has been successfully installed
   at:
   /opt/vmware-k4m
   PRESS <ENTER> TO EXIT THE INSTALLER:
   ```

   Product will be extracted to /opt/vmware-k4m (or your_location/vmware-k4m). This location is VMWARE_K4M_HOME.

7. VMWARE_K4M_HOME contains:
   - config
   - examples
   - k4m-rest-VERSION.tar.gz
   - license
   - scripts

   Installation logs

   Under INSTALLER_DIR/_VMware K4M_installer/logs. The .log file has the installation process details and the .txt file has information about the installer.

Set up, configuration and launch

1. Modify files **(Optional)**:
   If VMware K4M has been installed under default location i.e. /opt, you may skip this step. Otherwise replace occurrence of /opt with the full path to the parent folder of your VMware K4M installation in the following files:

   i. All scripts under VMWARE_K4M_HOME/scripts

   ii. VMWARE_K4M_HOME/config/k4m-rest-application.properties

2. Configure system settings, download system and open-source packages:
   Change to VMWARE_K4M_HOME/scripts. Run the script "step-1-as-root.sh" and follow prompts. This will:

   i. Create the user **k4m** and give it ownership of the installation under VMWARE_K4M_HOME. The default password is "**k4m**".

   ii. Download and install system packages, OpenJDK

   iii. Configure SSHD

   ```
   $ ./step-1-as-root.sh
   ```

   NOTE: On RHEL and CentOS 6.9 and 6.10, when the script finishes you will see the following. This can be ignored as necessary configuration is made by the OS by default.

   ```
   ./step-1-as-root.sh: line 64: sshd-keygen: command not found
   ```

3. Configure MongoDB [1]
   NOTE: It is assumed the reader is familiar with MongoDB administration.
   Create a database “k4m-rest” and user “k4m” in your MongoDB instance. A script has been provided with the configuration - VMWARE_K4M_HOME/config/mongo-config.js. The MongoDB Admin command line tool can be used to execute the script and configure the database and user.
   NOTE: The database connection details are captured in the “spring.data.mongodb.uri” property in the k4m-rest-application.properties file. This file must be updated to reflect the database's hostname/IP, username, and password configured for the service.

4. Install Flink and the K4M REST Service:
   Become k4m user

   ```
   $ su k4m
   ```

   Run the script "step-2-as-k4m.sh". This will

   i. Configure SSH keys

ii. Download and install Flink. Flink is extracted into folder "flink-VERSION", created under the same directory that vmware-k4m is installed. By default, it is under /opt or the path specified by you when modifying the scripts in step. We will refer to this complete path as FLINK_HOME (also set as environment variable).

iii. Install the K4M REST Service

```
$ ./step-2-as-k4m.sh
```

5. Start the Flink cluster and K4M REST Service:
   Run the script "step-3-as-k4m.sh"

```
$ ./step-3-as-k4m.sh
```

Verify Installation

Confirms all services are installed correctly and communicate with each other including VSA Kafka broker.
Monitors status of following services

i. K4M Rest Service

ii. KPI Engine (based on Flink)

iii. KPI Configuration store (MongoDB)

Also monitors VSA Kafka broker status. Details exposed via **http://K4M_HOST_IP:8083/v1/health** endpoint.

Verification by script:

i. Checks message flow to Kafka broker by publishing and subscribing to test message. For this, the script downloads a Kafka distribution under the /scripts folder and runs the console producer and consumer.
ii. Query the VMware K4M /health endpoint to verify all services are running.

Prerequisite: Ensure Kafka broker for the VSA solution is running and accessible.

NOTE: In this release, the verification script checks one Kafka broker and VMware K4M monitors its status. This is applicable for either a single-broker or a Kafka cluster. For a Kafka cluster please provide any one broker and port as mentioned in the steps below, so the script can verify connectivity to at least one broker in the cluster.

1. Become k4m user. Modify the script "step-4-as-k4m.sh" by setting values for the variables shown below:

```
# Set to Kafka broker
KAFKA_HOST=kafka_broker_ip
KAFKA_PORT=kafka_broker_port

KAFKA_USER=kafka_user
KAFKA_PASSWORD=kafka_password

# Set to Zookeeper
ZOOKEEPER_HOST=zookeeper_ip
```

```
ZOOKEEPER_PORT=zookeeper_port
```

NOTE: Use the advertised Kafka broker IP found in your broker's server.properties.

If secured access to Kafka broker is not required, the Kafka username and password should be left blank.

2. Kafka status is monitored via JMX so the Kafka broker must have JMX port enabled. To enable, set the following variable on the Kafka broker host and restart the broker:

```
$ export JMX_PORT=kafka_broker_jmx_port
```

3. Set the same JMX port from step 2 in the script "step-4-as-k4m.sh":
The script creates a service entry in VMware K4M for the Kafka broker using ip, port, JMX port. VMware K4M monitors Kafka and updates the status of the service.

```
KAFKA_JMX_PORT=kafka_broker_jmx_port
```

NOTE: Without enabling the JMX port on the Kafka host, VMware K4M will not be able to monitor Kafka. However, this does not affect the working of VMware K4M. The script will report that JMX is not enabled.

4. Run the script "step-4-as-k4m.sh" and if everything is running correctly you will see:

```
$ ./step-4-as-k4m.sh

Verifying installation of VMware K4M services....

.....

VMware K4M services ready
```

You can also verify that all the services are online using the REST API endpoint at **http://K4M_HOST_IP:8083/v1/health**.

List of Processes and Ports

Upon successful start VMWare K4M should have the following processes running. Processes can be listed using the ”jps” and “ps” system commands.

K4M Service	Process Name	Port
K4M REST Service	`k4m-rest-VERSION.jar`	8083
KPI Engine (Flink)	`TaskManagerRunner` `StandaloneSessionClusterEntrypoint`	8081
KPI Configuration Store	`mongod`	27017
KPI Designer	Refer to VMware Smart Assurance UI documentation.	Refer to VMware Smart Assurance UI documentation.

Chapter 3 Operation

Stopping VMware K4M services

Follow steps to stop VMware K4M:

1. Become k4m user
   ```
   $ su k4m
   ```

2. K4M REST Service

   Identify the process id using ‘jps’ or the ‘ps’ system command. The process name includes "k4m-rest". Terminate it using the system command (kill).

3. KPI Engine

   Change to FLINK_HOME/bin and run:
   ```
   $ ./stop-cluster.sh
   ```

4. KPI Configuration store (MongoDB)

   Please refer to the documentation [1].

Starting VMware K4M services

VMware K4M services are started automatically after installation. Follow the steps below to restart:

1. Become k4m user
   ```
   $ su k4m
   ```

2. K4M REST Service

   Run the command:

   ```
   java -jar VMWARE_K4M_HOME/k4m-rest/k4m-rest-VERSION.jar
   --spring.config.location=file://VMWARE_K4M_HOME/config/k4m-rest-
   application.properties &
   ```

3. KPI Engine

   Change to FLINK_HOME/bin and run:

```
$ ./start-cluster.sh
```

4. KPI Configuration store (MongoDB)
   Please refer to the documentation [1].

Chapter 4 Uninstallation

Uninstalling VMware K4M

No uninstaller, manual process.

1. Stop all VMware K4M services as described in the "Stopping VMware K4M" section.

2. K4M REST Service

    a. Note down the installation location of other open-source components present in the scripts under VMWARE_K4M_HOME/scripts. This will be needed if you wish to remove them as well.

    b. Delete the folder vmware-k4m from VMWARE_K4M_HOME.

3. For open-source services viz. Flink and MongoDB, installation folders can be removed from the locations noted in the step above.
   Please refer to their documentation to remove the artifacts completely:
   MongoDB [1]
   Apache Flink [2]

4. KPI Designer

   Please refer to the **VMware Smart Assurance UI documentation** for details.

Chapter 5 Deploying vIMS KPIs

List of vIMS KPIs computed and input metrics

1. List of vIMS KPIs computed. The complete KPI Definition is under VMWARE_K4M_HOME/examples/vims/vims-system-kpis.json

    - *scscf-initial-reg-success-rate*
    - *scscf-video-session-average-setup-time*
    - *scscf-audio-session-average-setup-time*
    - *icscf-network-session-establishment-success-rate*
    - *icscf-user-session-establishment-success-rate*

2. List of vIMS metrics used in computing above KPIs. The complete metric catalog with description of each metric is in VMWARE_K4M_HOME/examples/vims/vims-metrics-catalog.json.

    - *SCSCFInitialRegistrationSuccesses*
    - *SCSCFInitialRegistrationAttempts*
    - *SCSCFInitialRegistrationFailures*
    - *SCSCFInitialRegistrationSuccessPercent*
    - *SCSCFVideoSessionSetupTimeAverage*
    - *SCSCFAudioSessionSetupTimeAverage*
    - *ICSCFSessionEstablishmentSuccesses*
    - *ICSCFSessionEstablishmentFailures*
    - *ICSCFSessionEstablishmentAttempts*
    - *ICSCFSessionEstablishmentSuccessPercent*
    - *ICSCFSessionEstablishmentNetworkSuccesses*
    - *ICSCFSessionEstablishmentNetworkFailures*
    - *ICSCFSessionEstablishmentNetworkAttempts*
    - *ICSCFSessionEstablishmentNetworkSuccessPercent*

Deployment Steps

Main steps:

1. Deploy the following artifacts to VMware K4M:
    i. vIMS metrics catalog
    ii. vIMS KPI Definition
    iii. vIMS KPI Stream
2. Configure vIMS KPI Stream with Kafka broker details
3. Deploy the vIMS KPI Stream

Steps:

1. Become k4m user. Change to VMWARE_K4M_HOME/examples/vims. This contains a script to deploy the catalog, KPI Definition, KPI Stream using the corresponding JSON DSL files.

2. Run the script below. On successful creation the DSL is displayed on the terminal.
```
$ ./deploy-vims-kpis.sh
```

3. Launch the KPI Designer in a browser. Please refer to the **VMware Smart Assurance UI documentation** for details regarding its installation and operation.
Please refer to the **VMware Smart Assurance K4M 1.0.0 KPI Designer User Guide** for details regarding its usage viz. creating and computing KPIs.
Upon launching, you will see the KPI Definitions page with the vIMS KPI Definition (vims-system-kpis) as shown below.

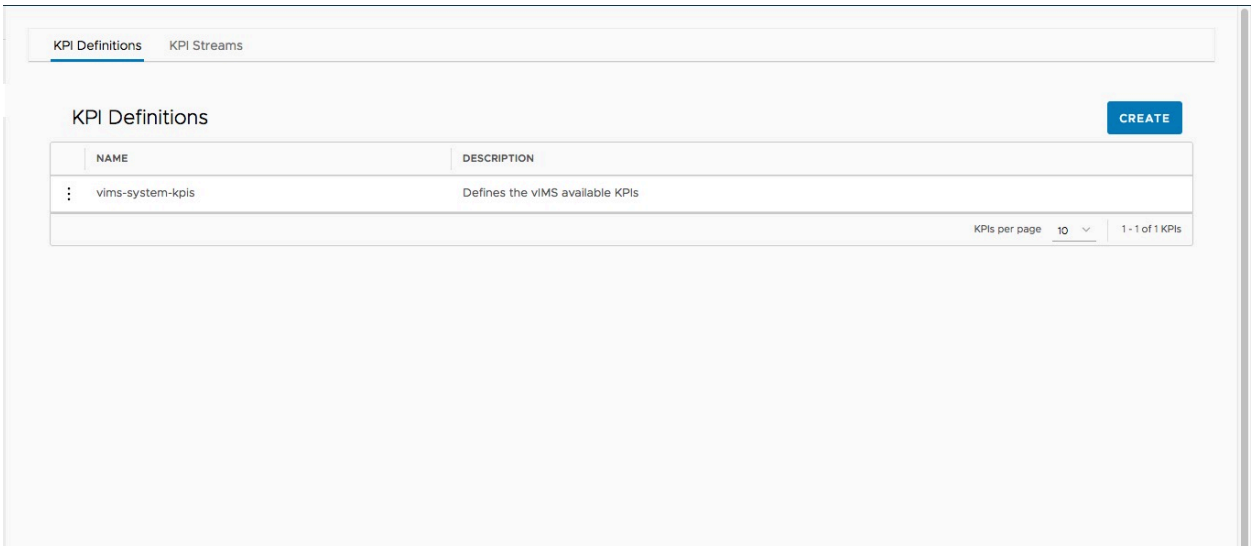

Figure 1 KPI Definitions page

4. Click on KPI Streams menu item to view the vims-kpi-stream listed.

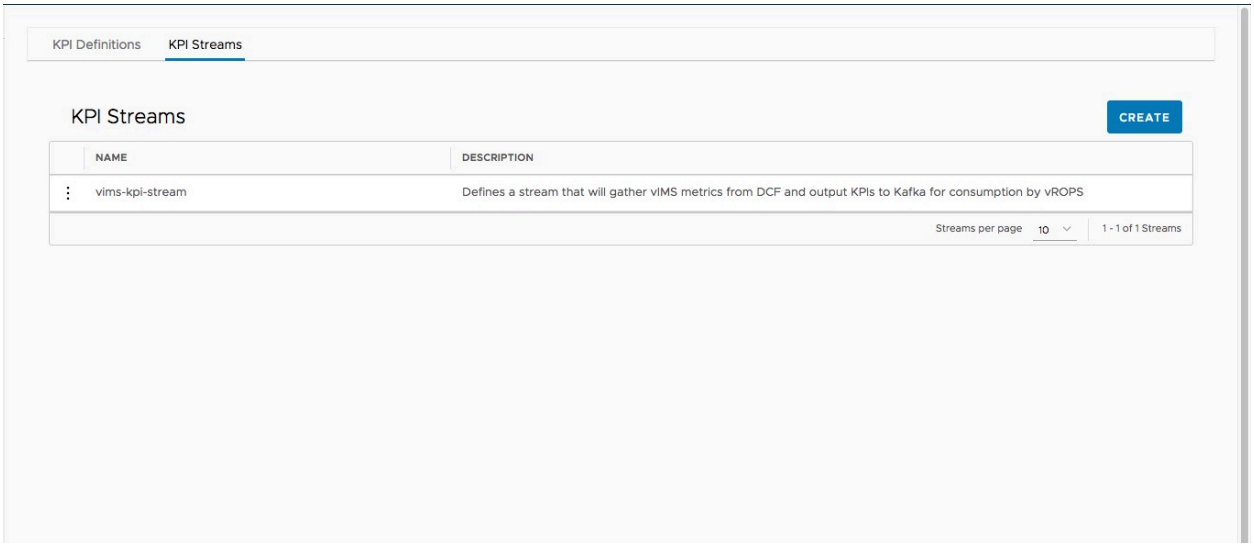

Figure 2 KPI Streams page

5. Click the dots before the vims-kpi-stream label and select Edit to open the Editor.

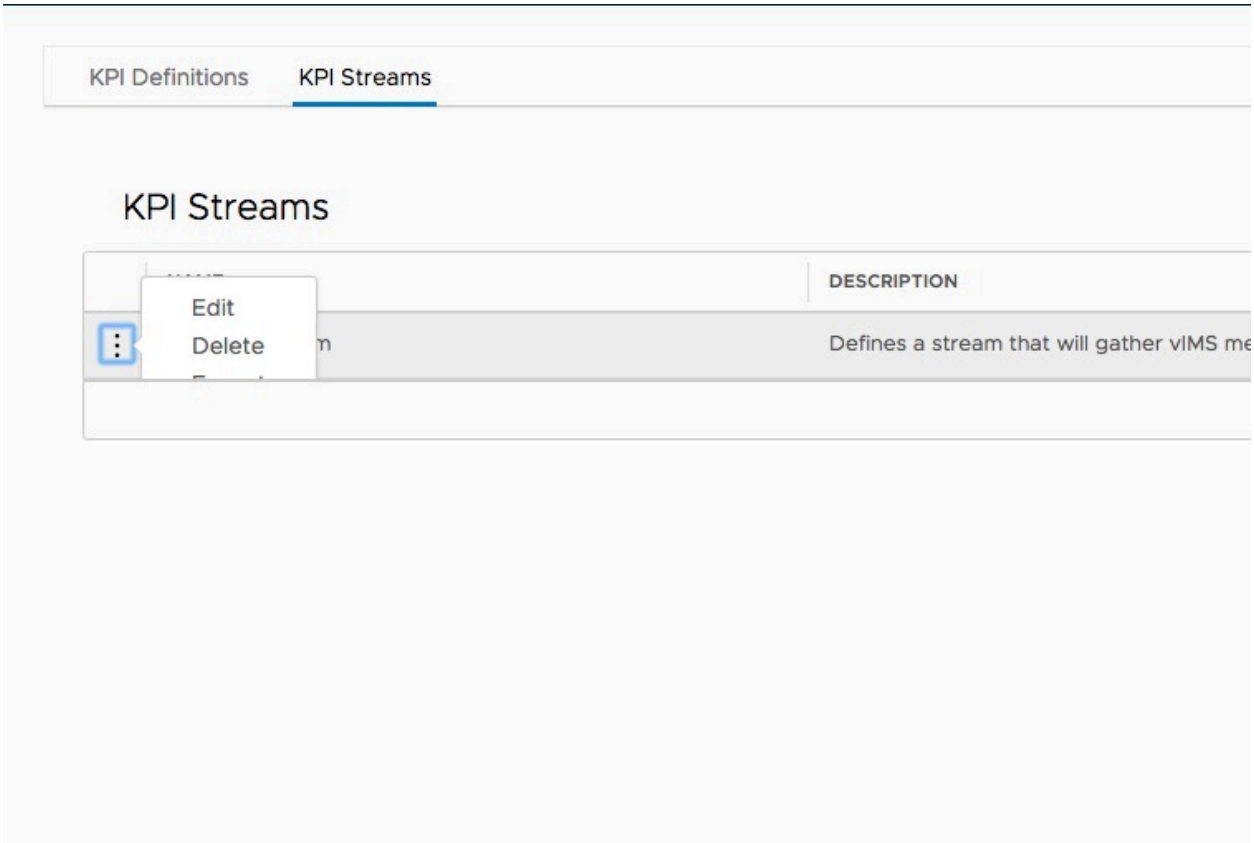

Figure 3 Edit KPI Stream

You will see the Input, KPI Definition and Output blocks on the canvas as shown below.

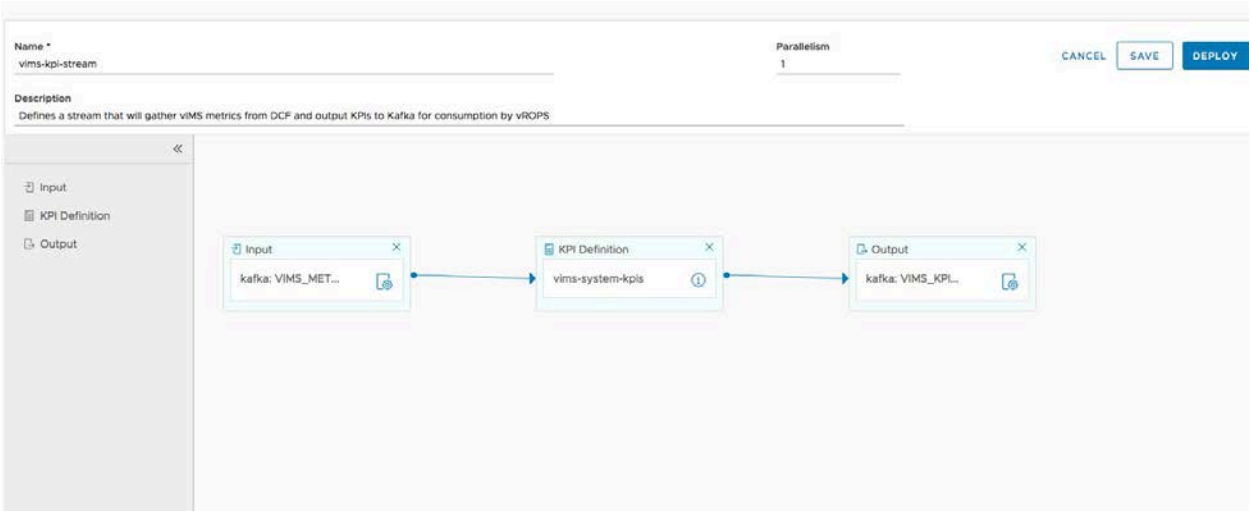

Figure 4 KPI Stream editor

6. Click the gear icon in the Input block for Kafka source. In the pop-up dialog as shown in the figure below, replace the placeholder strings as follows:

   - **Group ID:** consumer group id
   - **Bootstrap Servers:** `kafka_broker_ip:kafka_broker_port`
     For Kafka cluster enter list as:
     `kafka_broker_ip1:kafka_broker_port1,kafka_broker_ip2:kafka_broker_port2,…`
   - **Topic:** Topic name to subscribe for vIMS metrics.
   - **Username & Password :** User name and password if Kafka broker is configured for secure password authentication.
   - **Formatter:** Input metric format. Pre-filled to “DCF Metrics” i.e. to parse metrics published by VSA DCF.

   Click SAVE to close the dialog box.

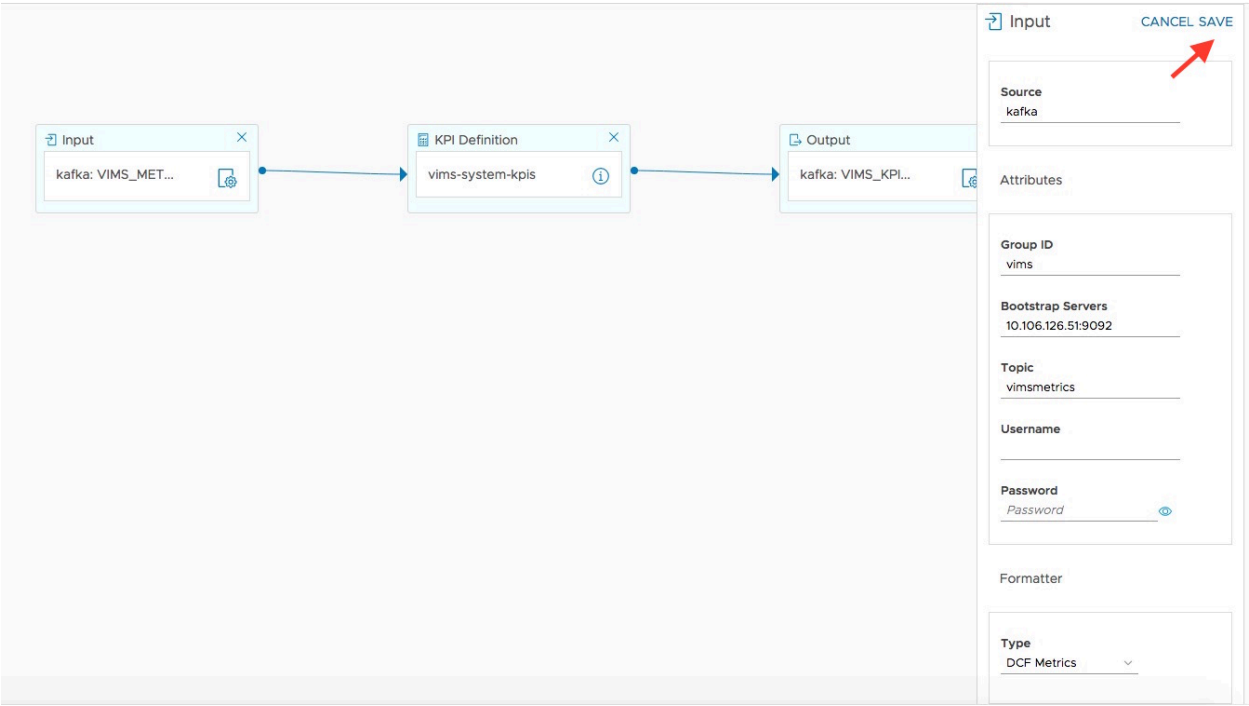

Figure 5 Input block configuration

7. Click on the gear icon in the Output block for Kafka destination. In the pop-up dialog, as shown in the figure below, replace the placeholder strings as follows:

   - **Bootstrap Servers:** `kafka_broker_ip:kafka_broker_port`
     For Kafka cluster enter list as:

`kafka_broker_ip1:kafka_broker_port1,kafka_broker_ip2:kafka_broker_port2,…`

- **Topic:** Topic name to publish vIMS KPIs.
- **Username & Password:** User name and password if Kafka broker is configured for secure password authentication.
- **Formatter:** Output KPI format. Pre-filled to “DCF Format”

Click SAVE to close the dialog box.

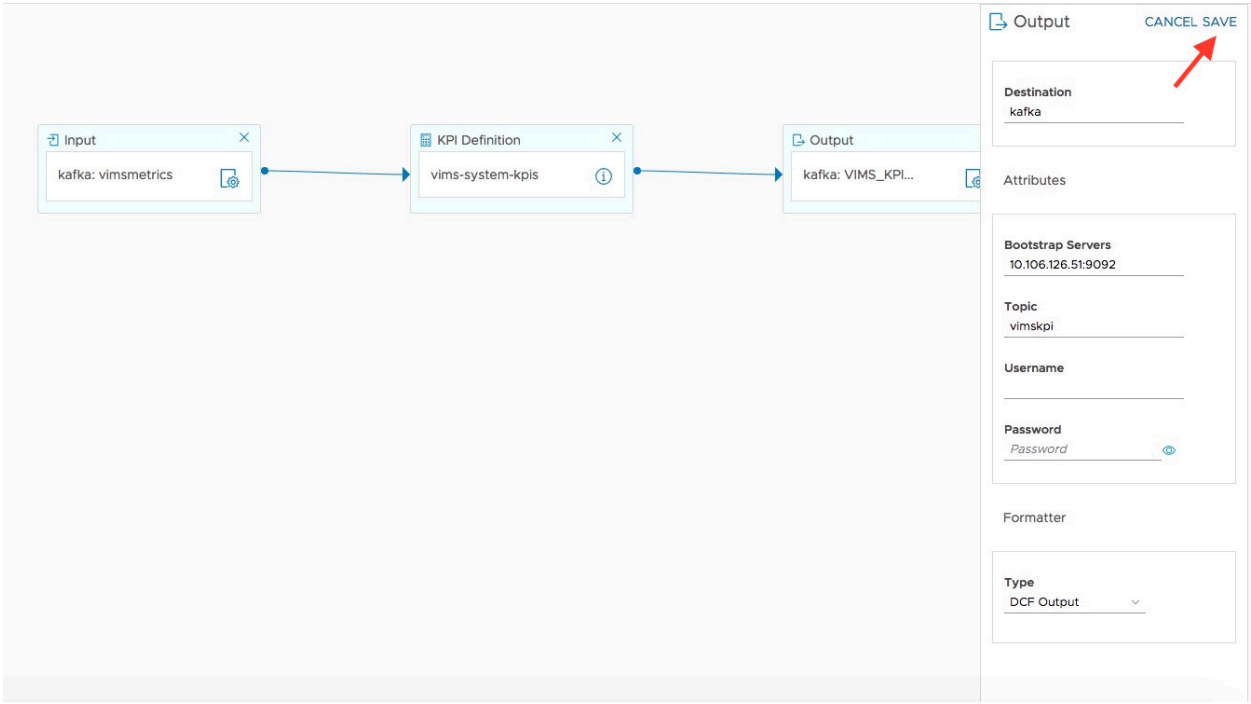

Figure 6 Output block configuration

NOTE: Clicking SAVE in the edit panel of the Input and Output block saves the block configuration in the client only. To save the configuration to the KPI Configuration Store, click the SAVE button in the workspace form (top right corner).

8. Click Deploy on the top right hand corner to save and deploy the stream.

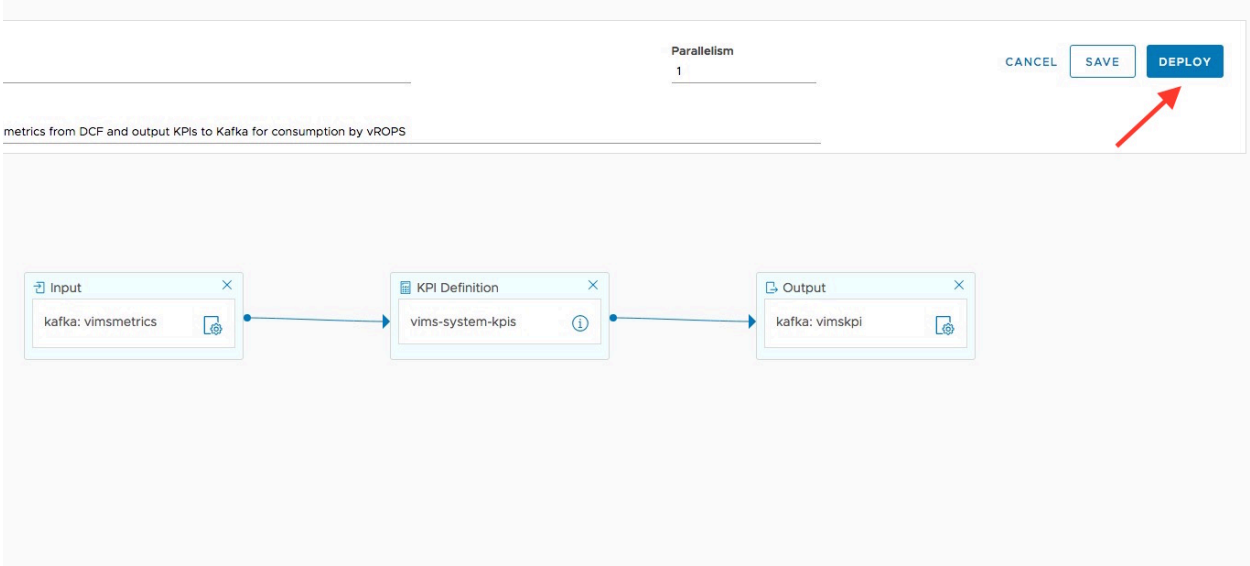

Figure 7 Deploy KPI Stream

You will see a confirmation dialog that the stream has been successfully deployed.

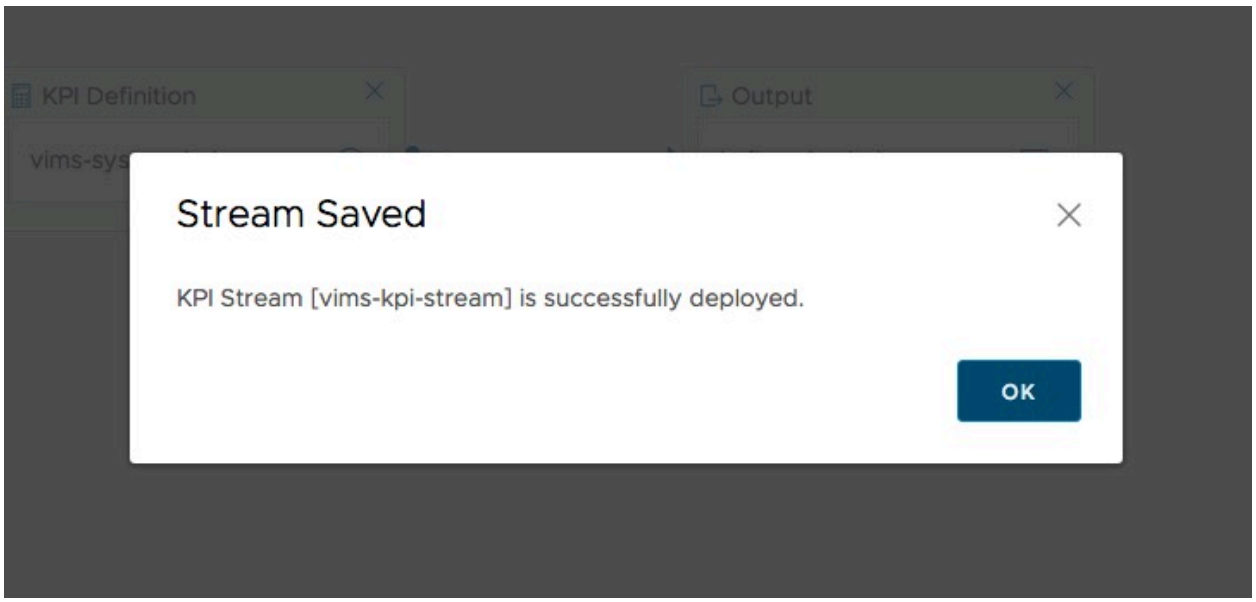

Figure 8 KPI Stream deployment confirmation

9. You can verify that KPIs are published to the topic configured using Kafka console consumer. For secure Kafka broker ensure consumer is configured for credentials.

Please refer to Apache Kafka documentation [3].

Chapter 6 Deploying new KPIs

The steps to create and deploy new KPIs are very similar to those for deploying the vIMS KPIs. Please refer to the **VMware Smart Assurance K4M 1.0.0 User Guide** for detailed instructions.

Chapter 7 Troubleshooting VMware K4M

NOTE:

1. For troubleshooting the KPI Designer please refer to the **VMware Smart Assurance UI** documentation for details.
2. The user is responsible for troubleshooting their MongoDB instance (KPI Configuration Store).

VMware K4M Logs

Application logs for VMware K4M services:

1. K4M REST Service
   Located at VMWARE_K4M_HOME/k4m-rest/log/k4m-rest.log

2. KPI Engine
   Logs are available at FLINK_HOME/log
   i. Jobmanager log is flink-k4m-standalonesession-NUMBER-HOSTNAME.log
   ii. Taskmanager log is flink-k4m-taskexecutor-NUMBER-HOSTNAME.log

3. KPI Configuration store
   Please refer to the documentation [1]

Examples of Success and Failure Log Entries

Some examples of log entries that you may see are as follows:

1. K4M REST Service

Successful Start

```
2019-04-17 10:07:35.032  INFO 12267 --- [main]
com.emc.k4m.Swagger2SpringBoot           : K4M-Rest version: v1.0.0.1
```

```
2019-04-17 10:07:35.035  INFO 12267 --- [main]
com.emc.k4m.Swagger2SpringBoot           : Started Swagger2SpringBoot
in 14.131 seconds (JVM running for 15.655)
```

Failure

Absence of the above message and/or any ERROR and exception stack traces, for example:

1.
```
2019-04-17 11:10:04.184 ERROR 12267 --- [http-nio-8083-exec-6]
com.emc.k4m.adapters.CatalogApiAdapter   : Cannot create a catalog
with existing name [vims-metrics-catalog]
```

2.
```
2019-04-17 11:18:11.775  INFO 12267 --- [http-nio-8083-exec-8]
com.emc.k4m.health.KafkaHealthProbe      : Connecting to kafka
broker [service:jmx:rmi:///jndi/rmi://strs-vm-
133.lss.emc.com:/jmxrmi]
2019-04-17 11:18:11.778 ERROR 12267 --- [http-nio-8083-exec-8]
com.emc.k4m.health.KafkaHealthProbe      : Failed to connect to
service 'KafkaHealth_strs-vm-133-lss-emc-com_9092_' on [strs-vm-
133.lss.emc.com:9092] jmx port []
```

2. KPI Engine (Flink)

Flink Taskmanager Successful Start

```
2019-04-17 09:59:25,247 INFO
org.apache.flink.runtime.taskexecutor.TaskExecutor           -
Successful registration at resource manager
akka.tcp://flink@localhost:6123/user/resourcemanager under
registration id 439d8c0d49f862d5fc6524dd0b4545af.
```

Failure

Absence of the above message and/or any ERROR and exception stack traces.

Troubleshooting Steps

The main steps are as follows:

1. Verify services are online:
   i. Check the http://K4M_HOST_IP:8083/v1/health endpoint. Status of all services should be “online”.

ii. Run the health check script “step-4-as-k4m.sh” as per the steps in the **Verify Installation** section.

One can also check individually each service as follows:

2. K4M REST Service and KPI Engine
   Become k4m user. Run the “jps” command to see running processes shown below:

```
$ jps
18385 TaskManagerRunner   (KPI Engine process)
18036 StandaloneSessionClusterEntrypoint (KPI Engine process)
18390 k4m-rest-0.0.300.jar    (K4M REST Service)
29468 Jps
```

3. KPI Configuration store
   Please refer to the MongoDB documentation [1].

4. Metric source and destination
   For Kafka, if you have run the script in the Verification Installation section, the status of the Kafka broker should be listed in the /health endpoint mentioned in step 1 above.
   For general information about monitoring Kafka please refer to the Kafka documentation [3] or other corresponding documentation.

Troubleshooting specific issues

Assumes KPI Designer is running. For troubleshooting the KPI Designer please refer to the **VMware Smart Assurance UI documentation** for details.

1. Unable to create/edit metric catalog, KPI Definition, KPI Stream

a. Verify that the K4M REST Service is available. See **Verify Services are online** section.

b. Check that Rest API endpoint (Swagger) is available as shown in the screen below (http://K4M_HOST_IP:8083/v1)

c. Verify that KPI Configuration Store is available. See **Verify Services are online** section.

d. Check the K4M REST Service log for any errors and exception stack traces. See **VMware K4M Logs** section.

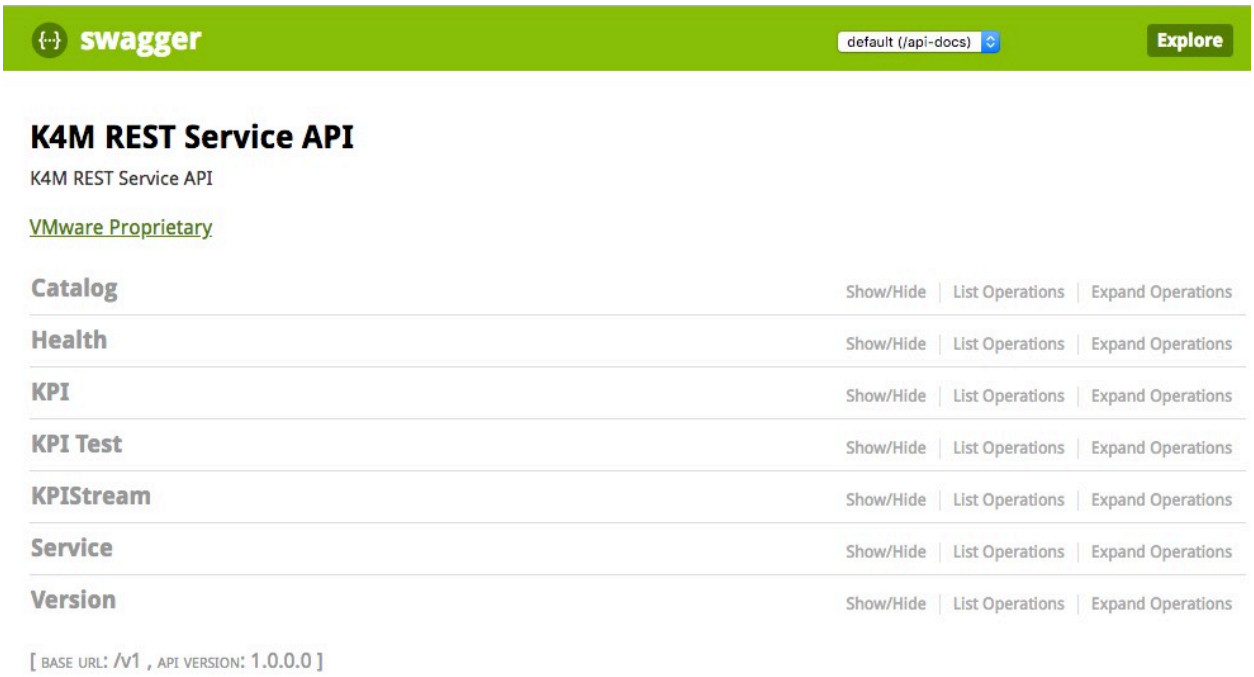

Figure 9 K4M REST Service API

2. KPI Stream does not deploy

a. Verify that K4M REST Service is available. See **Verify Services are online** section.

b. Verify that KPI Engine is available. See **Verify Services are online** section.

c. Check the K4M REST Service log and KPI Engine (Flink) logs for any errors and exception stack traces. See **VMware K4M Logs** section.

d. Using the KPI Designer, check the KPI Definition and KPI Stream configuration, including Input/Output block properties.

e. Check metric source and destination availability (for e.g. Kafka). See **Verify Services are online** section.

3. KPI Stream does not publish KPIs to Kafka topic

2. Check KPI Engine's Flink dashboard to see if stream has been deployed as shown below viz. http://K4M_HOST_IP:8081.
3. Check metric source and destination availability (for e.g. Kafka). See **Verify Services are online** section.
4. Check the KPI Engine (Flink) logs for any errors and exception stack traces. See **VMware K4M Logs** section.
5. Using the KPI Designer, check the KPI Definition and KPI Stream configuration, including Input/Output block properties.

    a.

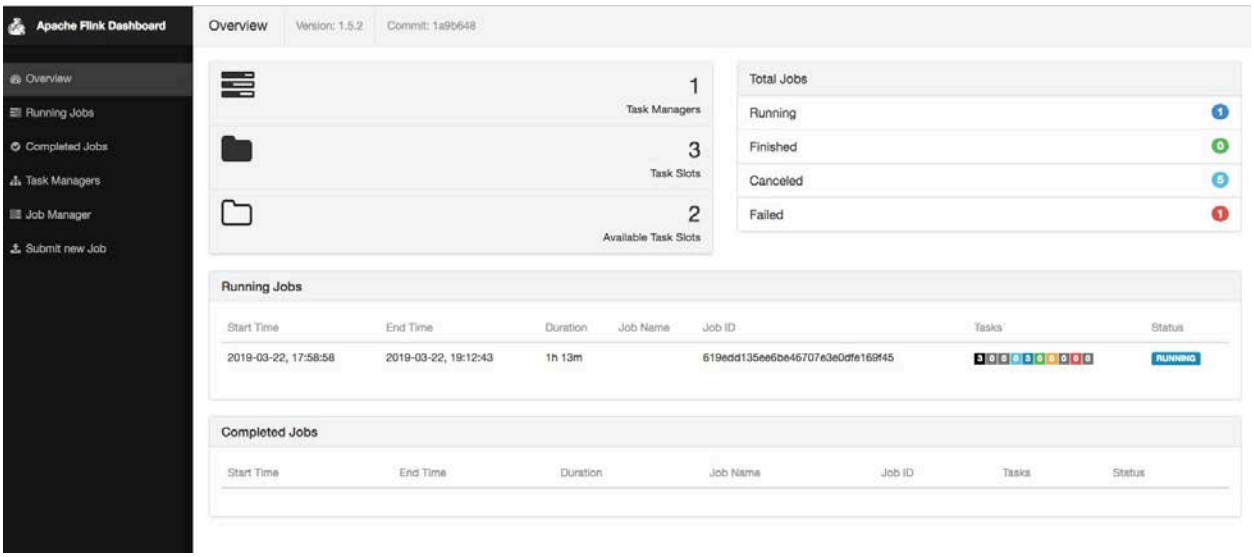

Figure 10 KPI Engine (Flink) Dashboard

Chapter 8 References

1. MongoDB - https://docs.mongodb.com/

2. Apache Flink - https://flink.apache.org/

3. Apache Kafka - https://kafka.apache.org/documentation/