Redbooks
ibm.com/redbooks

Introduction to IBM Common Data Provider for z Systems

Michael Bonett

Domenico D'Alterio

Eric Goodson

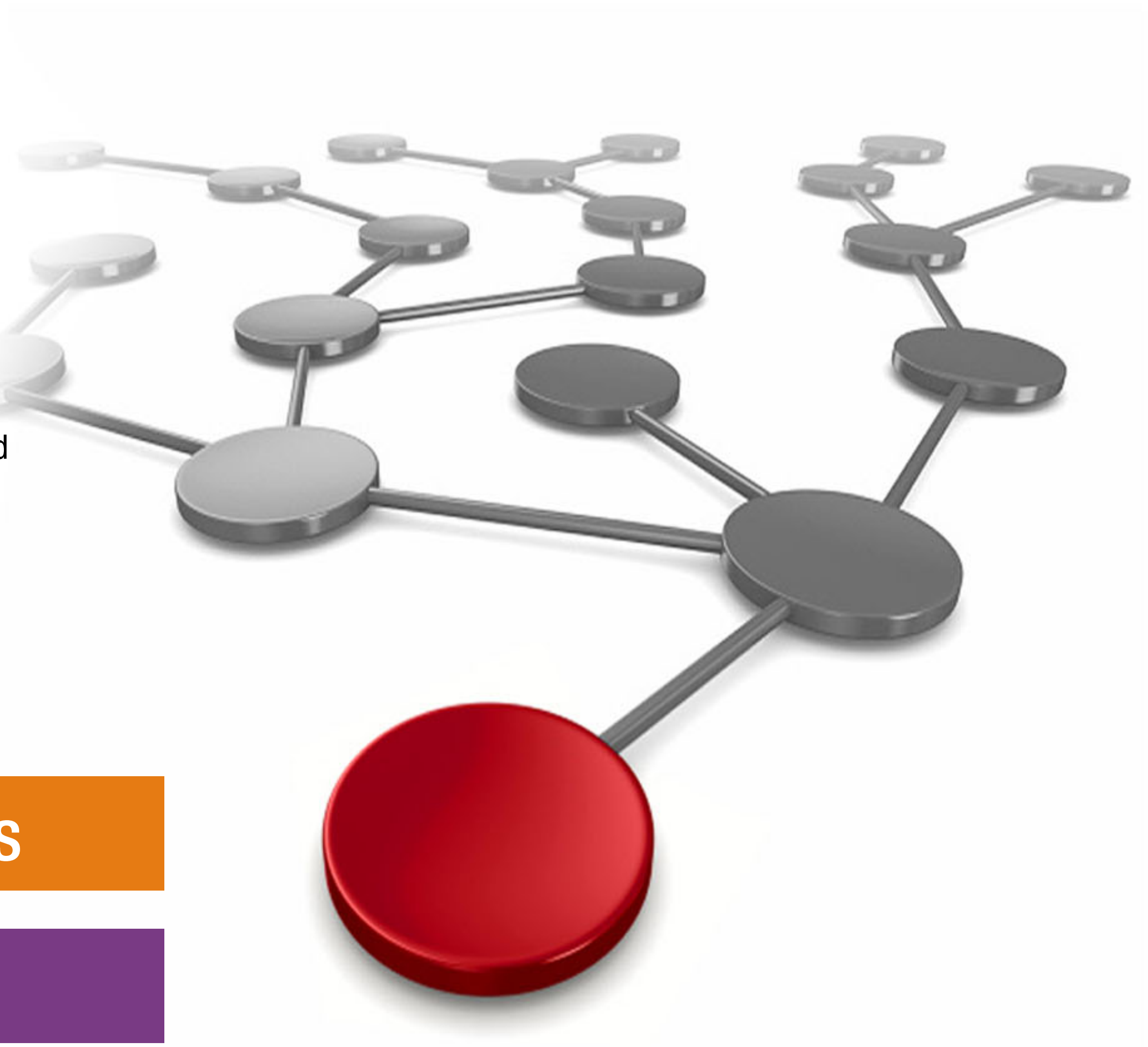
Matt Hunter

Keith Miller

Fabio Riva

Volkmar Burke Siegemund

John Strymecki



Analytics

z Systems

IBM

Redguide

Introduction to IBM Common Data Provider for z Systems

This IBM Redbooks® publication discusses the value of IBM Common Data Provider for z Systems, provides a high-level reference architecture for IBM Common Data Provider for z Systems, and introduces key components of the architecture. It shows how IBM Common Data Provider for z Systems provides operational data to various analytic solutions. The publication provides high-level integration guidance, preferred practices, tips on planning for IBM Common Data Provider for z Systems, and example integration scenarios.

The document is divided into these sections:

- “Part 1. Executive overview” on page 2
- “Part 2. IBM Common Data Provider for z Systems reference architecture and components” on page 9
- “Part 3. Practical integration scenarios” on page 18

Part 1. Executive overview

This part includes the following topics:

A recent survey of IBM® z Systems® clients concluded cost reduction and outage prevention were the top two focus items for their IT environments. Being able to analyze operational data quickly to identify the root cause of potential service impacts helps mitigate bottlenecks and problems. Also, the ability to observe negative trends and anomalies in your IT environment and infrastructure (based on operational data) makes it easier to perform proactive steps to avoid outages and performance issues.

IT operational data is continually growing in volume and complexity. This data holds the key to current and potential problem areas. The question becomes, *How does the IT organization collect operational data and make that data available for analysis, where the analysis results could identify ways to reduce costs, reduce bottlenecks, minimize outages, and eliminate blind spots?*

IBM Common Data Provider for z Systems collects, filters, and formats IT operational data in near real-time and provides that data to target analytics solutions. IBM Common Data Provider for z Systems enables authorized IT operations teams using a single web-based interface to specify the IT operational data to be gathered and how it needs to be handled. This data is provided to both on- and off-platform analytic solutions, in a consistent, consumable format for analysis.

IBM Common Data Provider for z Systems allows you to:

- Obtain access to analytics data within minutes. IBM Common Data Provider for z Systems collects operational data in near real time and streams that data to the analytic engine data lake. So, the data are available for analysis (in the analytic data lake).
- Collect data once and provide multiple different data subscribers (analytic solutions) with the data they need.
- Filter the data to target specific use cases and reduce data volumes and network traffic.
- In conjunction with IBM Tivoli® Decision Support for z/OS®, reduce operational and storage costs for System Management Facility (SMF) data by leveraging the IBM Db2® Analytics Accelerator.
- Collect data in batch mode to control CPU consumption.
- Share data both on- and off-platform in a consistent, consumable format.

IBM Common Data Provider for z Systems is the standard way to gather IT operations data for analytics solutions. The operational data that IBM Common Data Provider for z Systems collects is transformed into the formats required by analytic solutions executing on- and off-platform. IBM Common Data Provider for z Systems provides data for a wide variety of analytic solutions provided by either:

- Third-party vendors for example, Splunk and Elasticsearch
- IBM products such as, IBM Operations Analytics for z Systems and IBM Db2 Analytics Accelerator for IBM z/OS (IDAA) in conjunction with IBM Tivoli Decision Support for z/OS

Through proactive analysis of the IT operational data provided in near real-time these analytic solutions can produce insights that prevent impacts to business operations, optimize costs and efficiencies, and reduce risk. Plus, the IBM Common Data Provider for z Systems one-time charge licensing model means no data ingestion charges are applied.

For Splunk users, IBM Common Data Provider for z Systems provides sample dashboards available for free on Splunkbase (http://ibm.biz/CDPzSamples) to visualize **system health** as well as detailed dashboards for IBM CICS®, Db2, MQ, and z/OS.

For users of the Elastic Stack, IBM Common Data Provider for z Systems offers free sample dashboards from IBM developerWorks® (http://ibm.biz/CDPzSamplesElastic), providing insights equivalent to the Splunk sample dashboards.

Why is IBM Common Data Provider for z Systems important?

IBM Common Data Provider for z Systems enables you to leverage your IT operational data to gain actionable insights in near real-time. IBM Common Data Provider for z Systems is valuable for the following reasons:

- Gain fast access to insights

  You can access available operational data in near real-time for an ongoing health check of your IT environment and to establish an early warning system with predictive analytics tools. This approach enables you to respond quickly to changes in your environment before they become a real issue.

- Improve analytics flexibility

  You can choose the IBM analytics solutions or third-party analytics platform to analyze your data. Third-party analytic platform providers include Splunk and Elasticsearch. This approach gives you flexibility in choosing the analytic solution that fits your needs.

- Expand consumable options

  You can send IT operational data to multiple destinations in the required formats and filter the data to send only the required data to authorized consumers. This approach can speed analysis. Moreover, it can ensure that only “needed” data are sent to subscribers, thereby limiting the access to all operational data.

- Reduce cost and effort

  You can load your z Systems environment SMF data directly into IBM Db2 Analytics Accelerator to identify ways to reduce CPU processing and save storage of your IBM Tivoli Decision Support for z/OS installation. This approach enables you to balance CPU usage by using reclaimed space and space from non-production systems.

Use cases

This section presents typical use cases for the product.

Use case 1: Collect logs in a unique place for rapid problem solving

Modern applications span between mainframe and distributed environments. Most of the time, a customer has separate environments managed by separate technical teams analyzing different types of logs.

Collecting critical logs in a unique place allows the customer to:

- Perform cross platform searches, searching in mainframe and distributed logs.
- Perform searches in logs coming from different subsystems (CICS, Db2, IBM IMS™, applications) allowing fast problem resolution in the mainframe environment

- Correlate logs from different sources, filtering only the relevant information. Customer can also order them by time, obtaining the problem evolution flow in cross-platform/cross-silo mode.

The following graphic shows the IBM Operations Analytics for z Systems platform collecting logs related to z/OS.

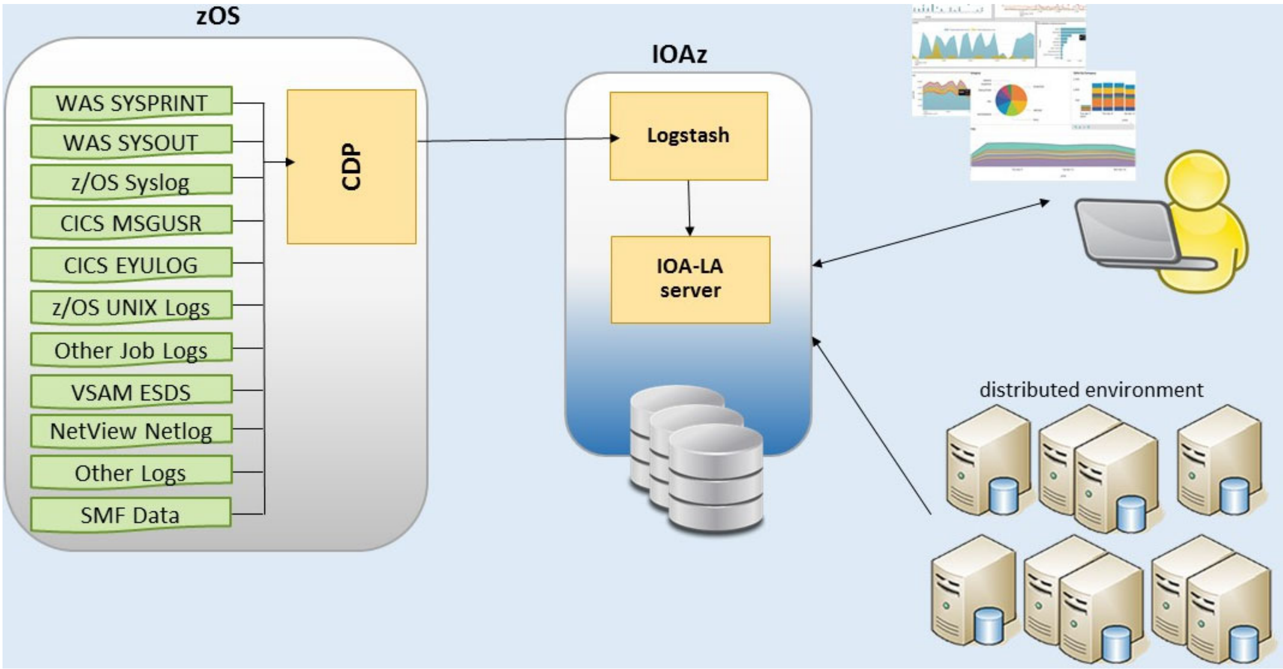


Figure 1   Logs collected in IBM Operations Analytics for z Systems platform

The following graphic shows the Splunk platform collecting logs related to z/OS.

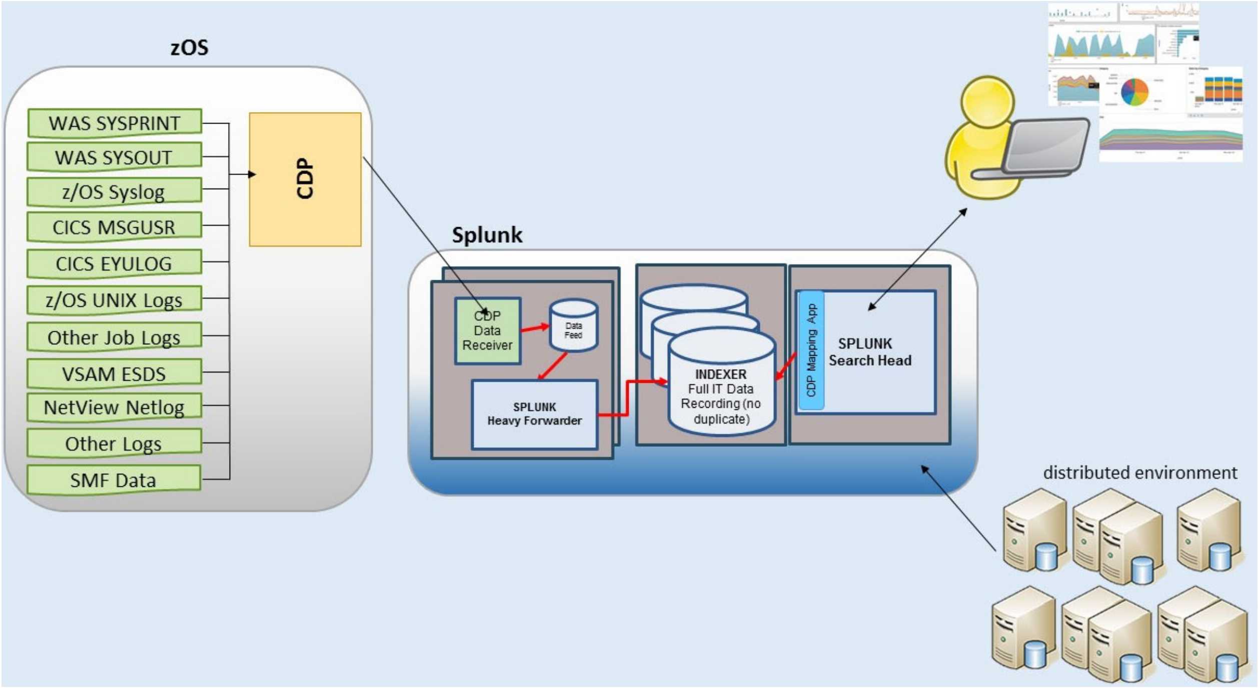


Figure 2   Logs collected in Splunk platform

The following graphic shows the ELK platform collecting logs related to z/OS.

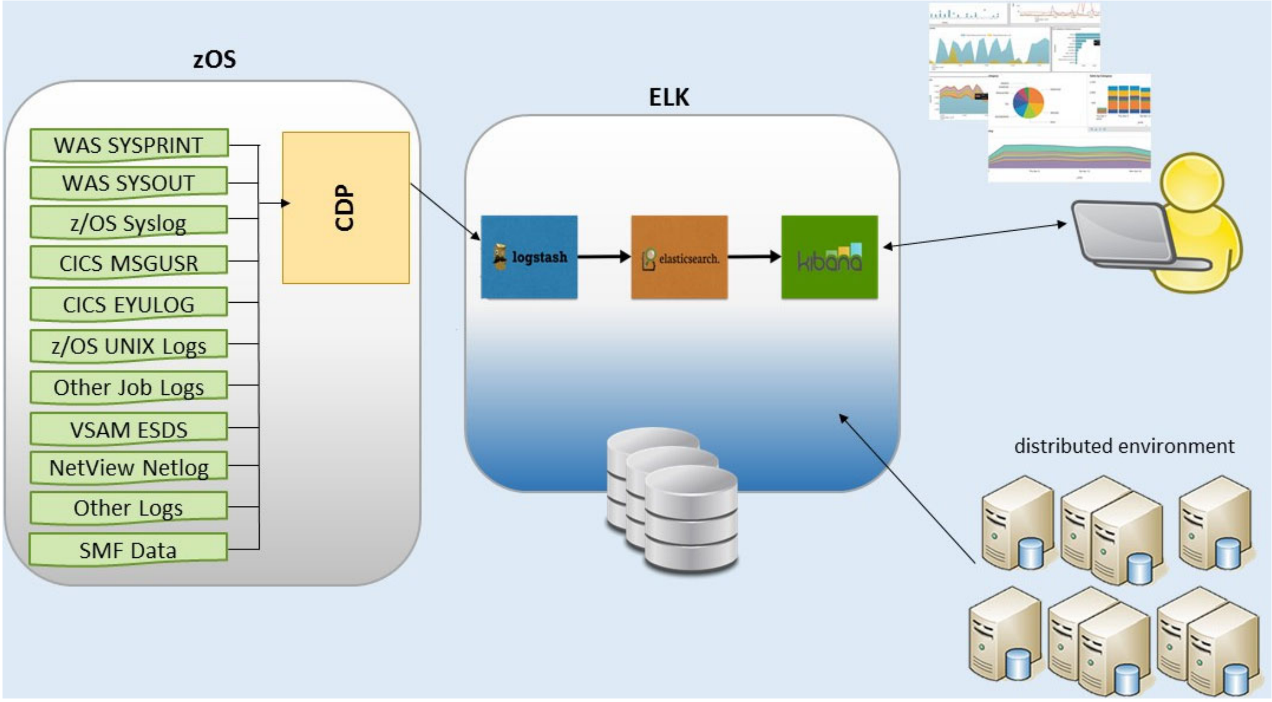


Figure 3 Logs collected in Elastic Stack platform

Use case 2: Collecting logs at enterprise level in a separate server

Some customers need to collect logs coming from different sources (mainframe and distributed) in a common place for the entire enterprise. This requirement might be due to regulations or due to the customer's needs, to access them even if there's platform failure.

CDP can play an important role in this scenario, as CDP can act as unique log collector for system, subsystem, and application logs on mainframe.

CDP supports the sending of data to most of platforms (through use of the **logstash** component), but also to specific platforms for log storage (such as syslog-NG server) or to generic subscribers (using the HTTP protocol).

The following graphic shows the collection of logs at the enterprise level.

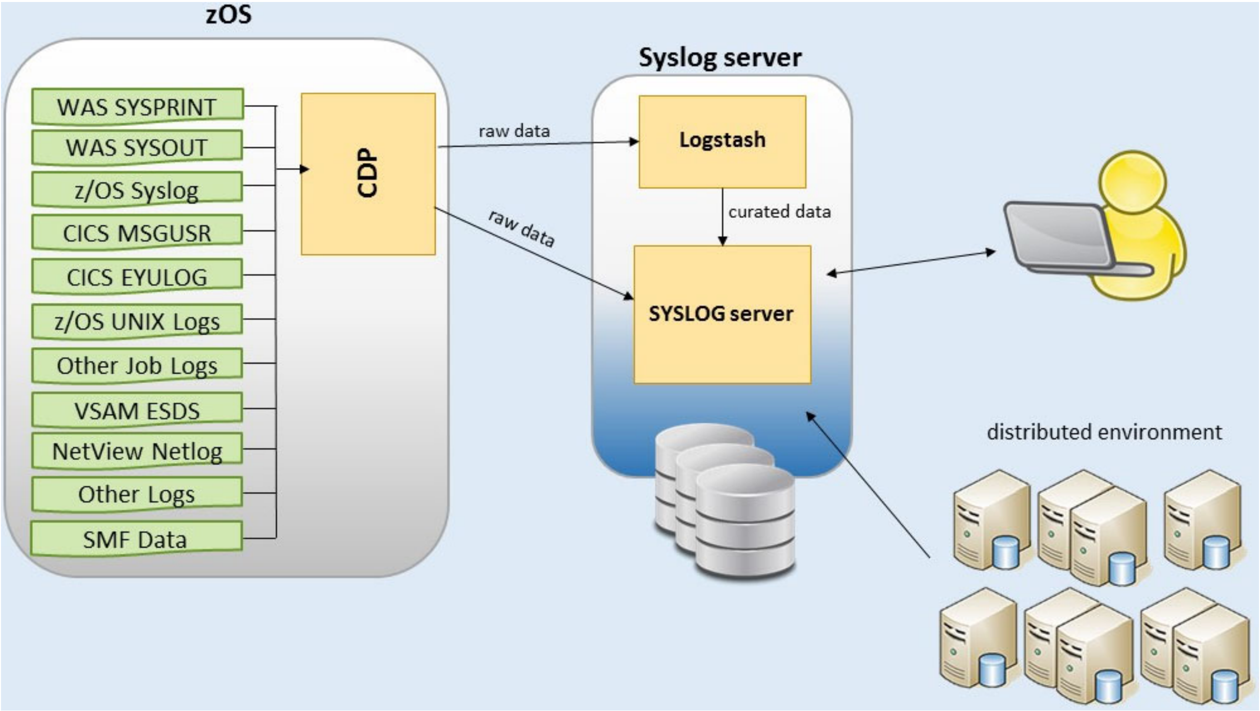


Figure 4 Collecting logs at enterprise level in a separate server

Use case 3: Collecting business critical applications metrics in a unique place, to control them end-to-end

It’s possible to collect transaction response times, time delays from infrastructure servers, and so on, in a unique place. As a result the customer has a complete, end-to-end view of the application for critical indicators.

Storing metrics in an analytic platform allows customers to perform deeper cross-platform analysis.

CDP can play an important role in this scenario, as CDP can act as unique log collector for system, subsystem, and application logs on mainframe.

The following graphic shows the collection of critical metrics in a unique platform. In this way, you control the metrics in an end-to-end fashion.

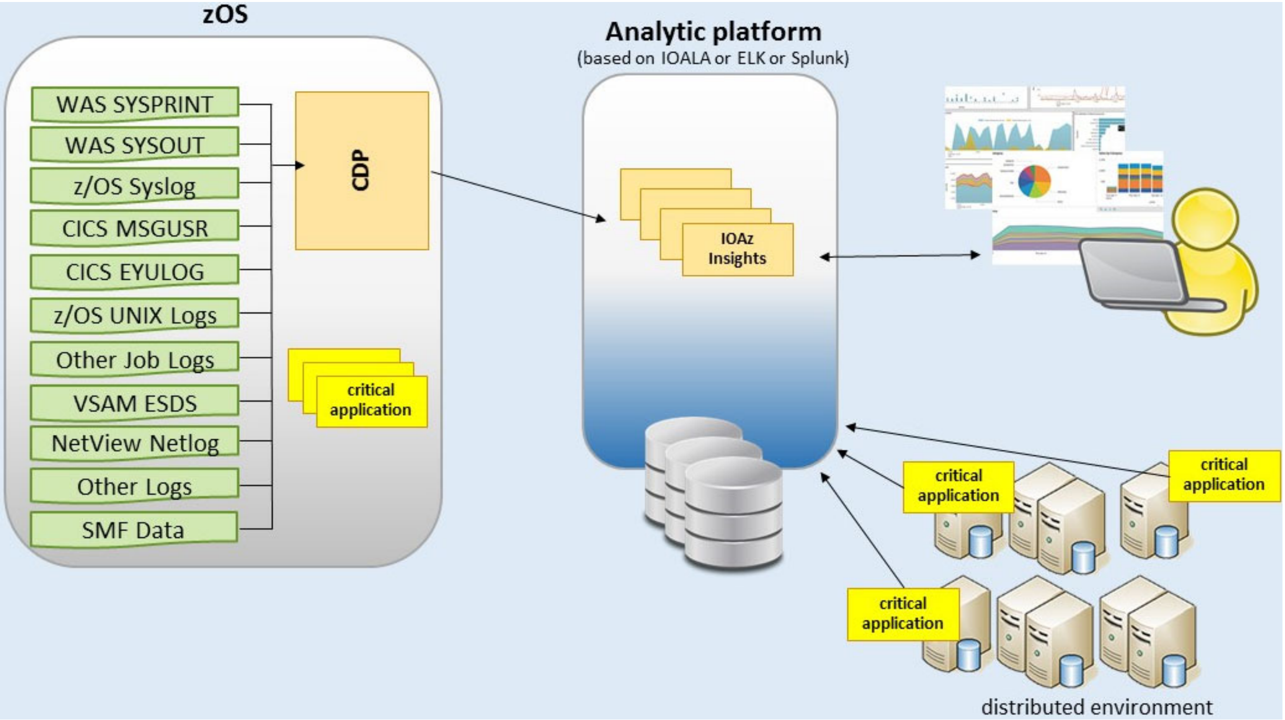


Figure 5 Metrics for business critical applications collected in a unique place, for end-to-end control

Use case 4: Integrating existing distributed performance/control of SLAs/monitoring from a unique place at the enterprise level

CDP can complement an existing distributed infrastructure designed to control performance and SLAs at the enterprise level. CDP can be used to collect almost all the metrics from mainframe, complementing data collected from distributed environments.

IT and LOB managers can access data through dashboards to show relevant data at the enterprise level. Dashboards are available for IBM Operations Analytics for z Systems, Splunk, and ELK platforms. CDP supports all of them.

The following graphic shows how an enterprise can use IBM Operations Analytics for z Systems, ELK, or Splunk as a common location to integrate any or all of the following information types:

- Distributed performance
- Control of SLAs
- Monitoring

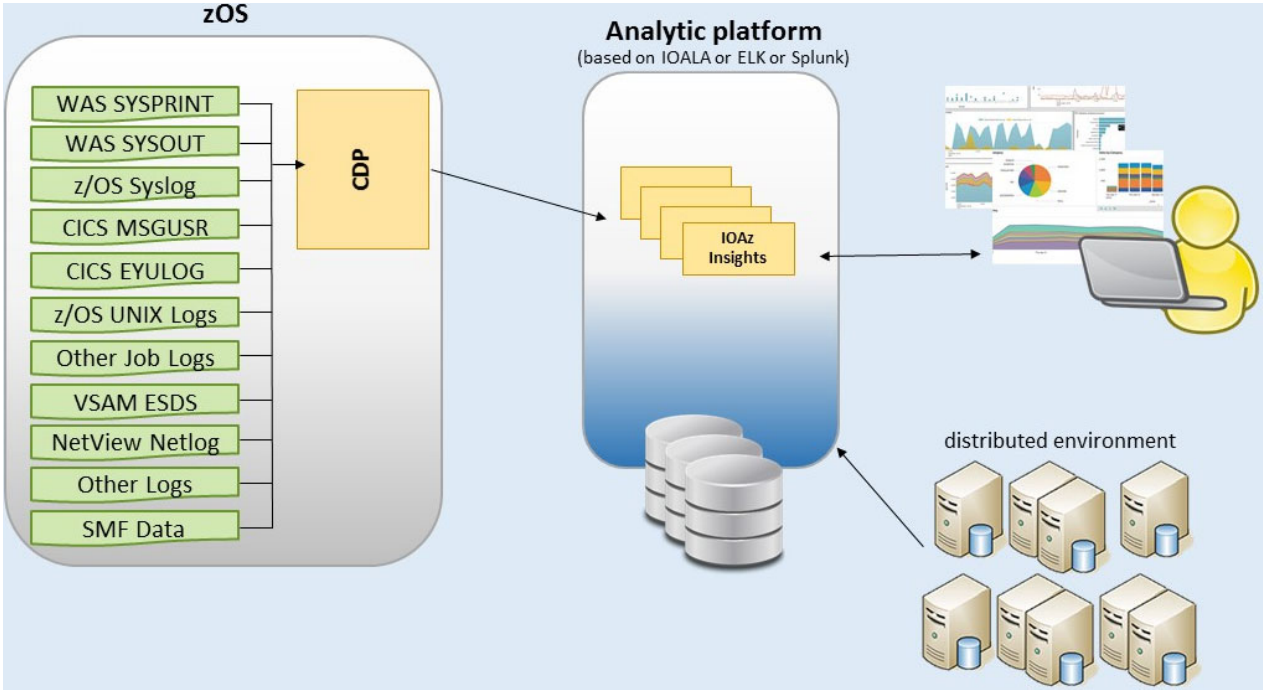


Figure 6   Existing distributed performance/control of SLAs/monitoring, integrated from a unique place at the enterprise level

Part 2. IBM Common Data Provider for z Systems reference architecture and components

This part includes the following topics:

- “How does IBM Common Data Provider for z Systems work?” on page 11
- “Subscribers flows” on page 13

IBM Common Data Provider for z Systems has various components that work together to identify, collect, organize, filter, and stream data to analytic solutions. Its components are grouped as follows:

- Configuration tool
- System Data Engine
- Log Forwarder
- Data Streamer
- Data Receiver

Figure 7 shows the IBM Common Data Provider for z Systems reference architecture.

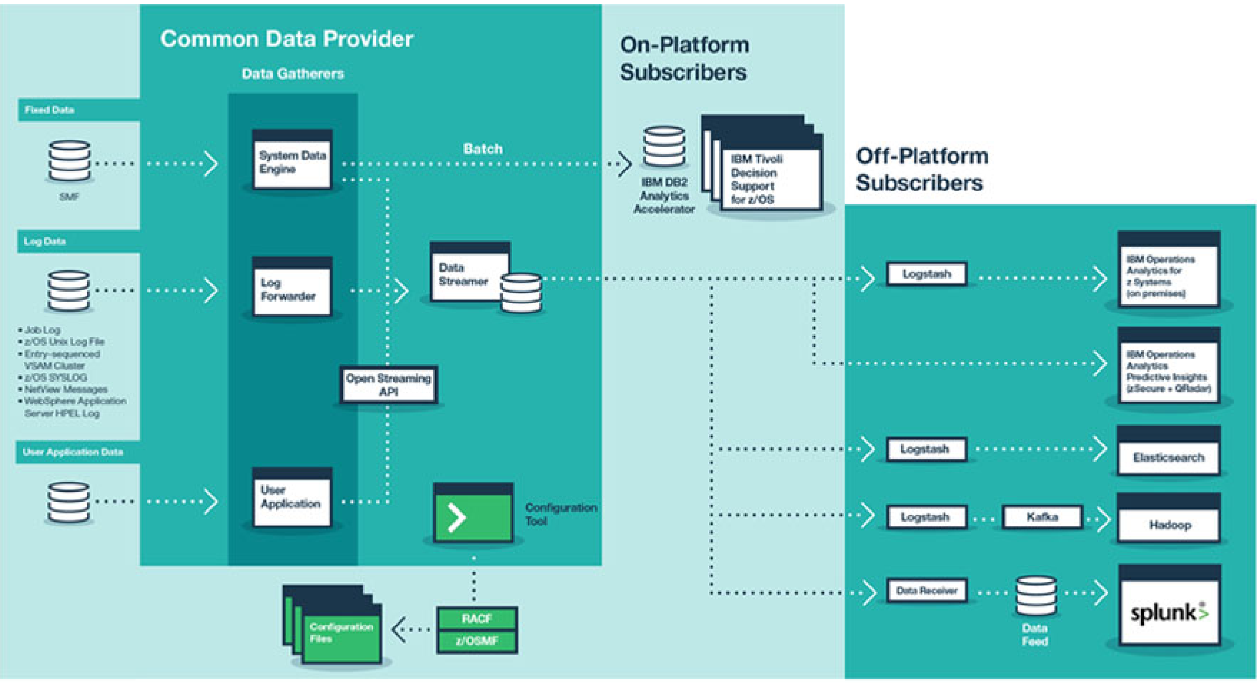


Figure 7 Common Data Provider for z Systems reference architecture

IBM Common Data Provider for z Systems consists of the following components (shown in Figure 7):

- Configuration Tool

  The IBM Common Data Provider for z Systems Configuration Tool is a web-based user interface that is provided as a plug-in to z/OSMF. You can use the Configuration Tool to perform the following activities:

  - Define which data sources to gather data from (such as SMF and log data)

  - Determine how to transform the data (such as splitting into individual messages or records, filtering fields or records or transforming to UTF-8 or)
  - Identify the subscribers that the data will be sent to.

  The Configuration Tool’s web interface is used to create policies that define all the data sources, transformations and subscribers. These policies reside on the host and can be secured by Resource Access Control Facility (IBM RACF®) or any other system authorization facility (SAF) product that you use.

  > **Note:** A policy is a group of configuration files for the various components. See the example in the section “Integration with Splunk” on page 19.

- Data Gatherers

  IBM Common Data Provider for z Systems provides several Data Gatherers which collect the specified operational data:

  - System Data Engine

    The System Data Engine gathers and processes SMF and IMS log data in near real-time. It can also gather and process SMF data in batch. All commonly used SMF types are supported and can be gathered from a number of sources:

    - SMF in-memory resource
    - SMF user exit HBOSMFEX
    - SMF log stream
    - SMF archive (which can only be processed in batch)

    SMF and IMS records are converted to comma separated values (CSV) for easy ingestion by subscribers.

  - Log Forwarder

    The Log Forwarder gathers a variety of log data, including:

    - Job log output written to a data definition (DD) by a running job, including (but not limited to):

      1. IBM CICS Transaction Server for z/OS logs

      2. IBM WebSphere® Application Server logs

    - z/OS UNIX log files, including (but not limited to):

      1. The UNIX System Services system log (syslogd)

    - z/OS system log (SYSLOG)
    - IBM Tivoli NetView® for z/OS messages
    - IBM WebSphere Application Server for z/OS High Performance Extensible Logging (HPEL) log

  - User applications

    The Open Streaming API enables your application to become a IBM Common Data Provider for z Systems data gather and supply operational data to IBM Common Data Provider for z Systems.

- Data Streamer

  The Data Streamer controls the format and destination of the operational data gathered by the data gatherers. It performs the following activities as needed:

  - Splits the data up into individual messages or records where required (for example, SYSLOG messages)

  - Transforms the data into the right format for the destination platform (such as UTF-8 encoding)
  - Sends the data to the subscriber for ingestion

  The Data Streamer can stream data to both on- and off-platform subscribers and can run on IBM z Integrated Information Processor (zIIP) processors to reduce general CPU usage and costs.

Let’s now look at the subscribers (shown in Figure 8 on page 14) that handle the data sent by the Data Streamer for IBM Common Data Provider for z Systems.

- Logstash

  Logstash is an open source, server-side data processing pipeline that ingests data from a multitude of sources, including IBM Common Data Provider for z Systems, performs additional data transformations as needed, and sends it to the destination of your choice, such as a data lake or analytics engine. Logstash is not provided with IBM Common Data Provider for z Systems.

- Data Receiver

  The Data Receiver is a component of IBM Common Data Provider for z Systems that acts as a target subscriber for the product. The receiver writes any data it receives out into files on disk sorted by the data source type. These files can then be ingested into analytics applications. The Data Receiver is used when the intended consumer of the stream is incapable of ingesting the data feed from IBM Common Data Provider for z Systems directly. It is typically run on a distributed platform, but it can also be run on z/OS.

- The analytic solutions are discussed in section “Analytic solutions” on page 12.

How does IBM Common Data Provider for z Systems work?

IBM Common Data Provider for z Systems provides IT operational data to analytic solutions enabling you to identify, isolate, and resolve problems across your enterprise operations environment. IBM Common Data Provider for z Systems is a collection of software programs that operate in near real-time or batch mode in order to centralize IBM z/OS operational data and feed all or filtered subsets of that data to a various analytic platforms for analysis. IBM Common Data Provider for z Systems supports all standard SMF records and a broad set of log data (such as syslog, syslogd, and job logs).

Let's take a broad view of IBM Common Data Provider for z Systems in an IT environment to get a sense of how it fits. The mechanisms are organized into these areas:

Setup and maintain IBM Common Data Provider for z Systems configuration files; the runtime environment; and analytic solutions that IBM Common Data Provider for z Systems feeds.

Setup and maintain IBM Common Data Provider for z Systems

Authorized IT operations team members use the Common Data Provider for z Systems Configuration Tool with its intuitive web interface to establish policies, which are stored in configuration files. This tool enables you to specify the types of data you need from your z/OS systems, specify where you would like the data to be sent, and select the format you would like it to arrive in. The tool is designed to help you create and manage these policies in a simple and intuitive manner. The tool is implemented as a plug-in to the z/OS Management Facility (z/OS SMF).

Runtime environment

When running IBM Common Data Provider for z Systems these components play an important role:

- Data Gatherers collect data

  IBM Common Data Provider for z Systems Data Gatherers collect IT operational data in near real-time streams or in batch mode. The batch mode option is used to collect detailed data for analysis or troubleshooting and retrieve archive information to investigate recurring problems. The IT operational data gathered includes:

  - SMF records such as SMF 30 and SMF 80 records
    - SYSLOG The IBM z/OS System Log and unformatted system services (USS) SyslogD
    - JOBLOGs output written to a data definition (DD) by a running job
    - Application logs including IBM CICS Transaction Server logs and IBM WebSphere Application Server logs
    - Generic files such as operational data generated by your business applications

  The Data Gatherers pass the data gathered to the Data Streamer.

- Data Streamer streams data to analytic solutions

  Data Streamer transforms the structured and unstructured data to the format(s) required by the target analytics solutions. Any transformation and filtering is done before transmitting the data to the analytics solutions.

Analytic solutions

IBM Common Data Provider for z Systems (https://www.ibm.com/us-en/marketplace/common-data-provider-for-z-systems) Data Streamer controls data formats and streams the data to subscribers. The subscribers are analytic solutions which are configured as subscribers to the IT operational data. On-platform analytic solutions are currently supplied by running the System Data Engine in batch mode, bypassing the use of the Data Streamer.

- **On-platform analytic solutions**

  IBM Db2 Analytics Accelerator for z/OS (https://www.ibm.com/us-en/marketplace/db2-analytics-accelerator) is a high performance appliance, which is tightly integrated with Db2 for z/OS. You can use it to do simple to complex Db2 queries with the operational data and produce reports and gain insights from the operational data supporting time sensitive decisions.

  IBM Tivoli Decision Support for z/OS (https://www.ibm.com/us-en/marketplace/decision-support-for-zos) collects log data and provides easy access to the historical enterprise-wide IT utilization information for use in performance reporting, service level management, and usage accounting. IBM Tivoli Decision Support for z/OS enables you to effectively manage the performance of your system by collecting performance data in a Db2 database and presenting the data in a variety of formats for use in systems management.

  IBM Common Data Provider for z Systems can take mainframe operational data, such as System Management Facility (SMF) records, and send them directly to the IBM Db2 Analytics Accelerator for storage, analytics, and reporting. The data is stored using a schema supplied by analytics components provided by IBM Tivoli Decision Support for z/OS. This approach removes the need to store data on Db2 for z/OS and allows for more detailed timestamp level records to be stored. It also moves more of the CPU work from z/OS to the Db2 Analytics Accelerator appliance and allows for reporting to make use of the high query speeds of the Db2 Analytics Accelerator.

- **Off-platform subscribers**

  Included in the off-platform subscribers are IBM and third-party analytic solutions such as:

  - IBM Operations Analytics for z Systems (also on platform) (https://www.ibm.com/us-en/marketplace/operations-analytics-for-z-systems)
    Operations Analytics for z Systems is a tool that enables you to search, visualize, and analyze large amounts of structured and unstructured operational data across IBM Z® environments, including log, event, and service request data and peformance metrics. Identify issues in your workloads, locate hidden problems, and perform root cause analysis faster.
  - IBM Security zSecure™ (https://www.ibm.com/us-en/marketplace/zsecure-administration)
    Security zSecure provides mainframe security solutions that can help you protect your enterprise, detect threats, comply with policy and regulations, and reduce costs.
  - IBM Qradar (https://www.ibm.com/us-en/marketplace/ibm-qradar-siem)
    Qradar detects anomalies, uncovers advanced threats, and removes false positives. It consolidates log events and network flow data from thousands of devices, endpoints and applications distributed throughout a network. It then uses an advanced Sense Analytics engine to normalize and correlate this data and identifies security offenses requiring investigation.
  - Elasticsearch (https://www.elastic.co/products/elasticsearch)
    Elasticsearch combines the power of a full text search engine with the indexing strengths of a JSON document database to create a powerful tool for rich data analysis on large volumes of data. With Elasticsearch your searching can be scored for exactness letting you dig through your data set for those close matches and near misses which you could be missing. IBM Compose for Elasticsearch makes Elasticsearch even better by managing it for you.
  - Hadoop (https://hadoop.apache.org/)
    Apache Hadoop is a highly scalable storage platform designed to process very large data sets across hundreds to thousands of computing nodes that operate in parallel. It provides a cost-effective storage solution for large data volumes with no format requirements.
  - Splunk software (https://www.splunk.com/)
    Splunk provides software products that enable search, analysis, and visualization of machine-generated data gathered from the websites, applications, sensors, devices, and so on that comprise an organization's IT infrastructure or business. IT operations teams can quickly access this data; identify trends and patterns to gain insights; and act based on those insights.

Subscribers flows

This section discusses the way IT operational data flows to the subscribers.

Off-platform subscribers flows

Figure 8 focuses on the off-platform subscribers flows.

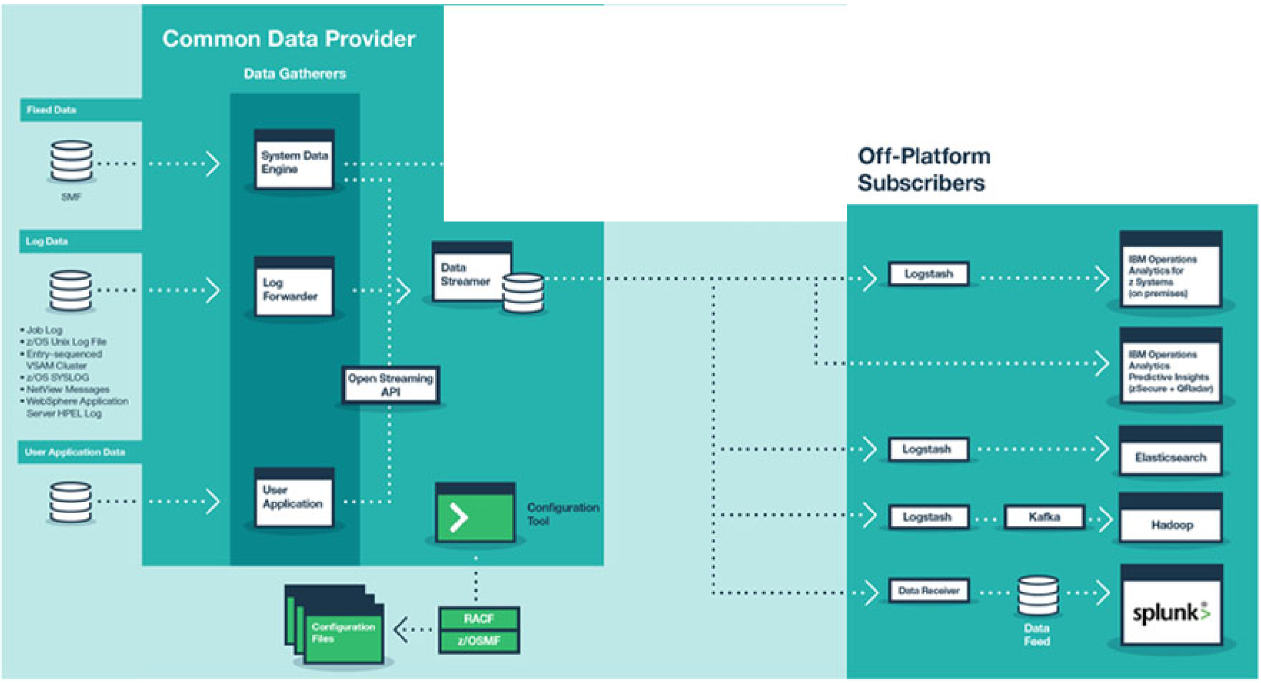


Figure 8 Off-platform subscribers flow

IBM Operations Analytics for z Systems

The IBM Common Data Provider for z Systems data gatherers collect the IT operational data specified by the policy and sends them to the Data Streamer for IBM Common Data Provider for z Systems. The Data Streamer transforms the data into UTF-8 format and forwards it to the Logstash server that is provided as part of IBM Operations Analytics for z Systems. IBM Operations Analytics for z Systems expects operational data to be sent unsplit.

Elasticsearch

Near real-time streaming to Elasticsearch is supported for SMF, IMS and SYSLOG data. The ingestion of this data into Elasticsearch requires additional data processing in Logstash. The ELK ingestion toolkit provided with IBM Common Data Provider for z Systems contains the Logstash configurations needed to perform this processing. The data streams need to be split and converted to UTF-8 in IBM Common Data Provider for z Systems before being sent to Logstash.

Splunk Enterprise

IBM Common Data Provider for z Systems can stream SMF, IMS and SYSLOG data in near real-time to Splunk Enterprise. The Data Streamer sends the collected data to a Data Receiver installed on a Splunk node (either Splunk Enterprise or a Splunk Heavy Forwarder). The Data Receiver writes the data out to files which are then read by Splunk Enterprise or the Heavy Forwarder. Additionally, a Splunk Ingestion App provided by the Common Data Provider needs to be installed into Splunk Enterprise to assist in the processing of ingested mainframe data.

On-platform analytic platforms flows

Figure 9 shows the on-platform analytic platform flow.

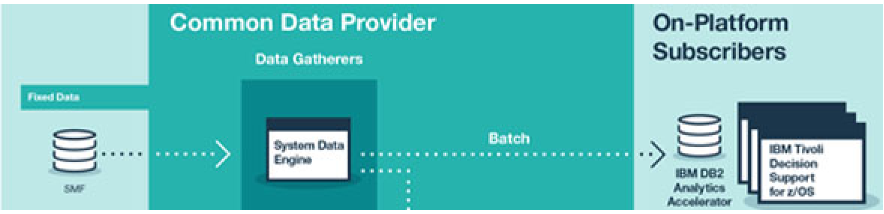


Figure 9 On-platform subscribers flow

IBM Db2 Analytics Accelerator

The IBM Common Data Provider for z Systems System Data Engine is used to convert SMF log data into z/OS UNIX files that conform to the Tivoli Decision Support for z/OS analytics components tables in Db2 internal format. The Db2 Analytics Accelerator Loader for z/OS is then used to load the Db2 internal format data sets directly into the Db2 Analytics Accelerator.

IBM Db2 Analytics Accelerator

The IBM Common Data Provider for z Systems System Data Engine is used to convert SMF log data into z/OS UNIX files that conform to the Tivoli Decision Support for z/OS analytics components tables in Db2 internal format. The Db2 Analytics Accelerator Loader for z/OS is then used to load the Db2 internal format data sets directly into the Db2 Analytics Accelerator.

IBM Common Data Provider for z Systems best practices and tuning

A document providing clear and concise tuning guidelines and best practices for IBM Common Data Pro-vider on z Systems product is available at http://ibm.biz/CDPzBestPractices.

This document is a "living" document. It will be updated as new and relevant information is gathered, reviewed, and deemed suitable for placement in this document.

System requirements

Verify that your z/OS system meets the requirements for running IBM Common Data Provider for z Systems. You must run the IBM Common Data Provider for z Systems in each z/OS logical partition (LPAR) from which you want to gather z/OS operational data.

IBM Common Data Provider for z Systems must be run with IBM z/OS V2.1, V2.2 or V2.3, IBM z/OSMF V2.1, V2.2 or V2.3 and IBM SDK for z/OS Java Technology Edition V7.0.1 or V8. Consult the z/OS system requirements topic in the IBM Common Data Provider for z Systems documentation for more details.

Security and authorization prerequisites

Different authorizations are required for installing and configuring IBM Common Data Provider for z Systems components and for accessing component-related libraries and configuration files during run time. Consult the IBM Common Data Provider for z Systems document and the Program Directories for more details.

- Required authorizations for Common Data Provider for z Systems Configuration Tool

  To run the setup script (savingpolicy.sh) to set up a working directory you must be logged into the z/OS system with a user ID that is in z/OSMF administrator group 1 and is a TSO

ID that has the UID 0 attribute. The working directory must be readable and writable by the user ID that runs the Configuration Tool.

To install, uninstall or run the Configuration Tool, you must be logged in to z/OSMF with a TSO user ID that is in z/OSMF administrator group 1.

- Required authorizations for Data Streamer

  The user ID that is associated with the Data Streamer started task must have:

  appropriate authority to access the IBM Common Data Provider for z Systems program files, which include the installation files and the policy file

  read/execute permissions to the Java libraries in the UNIX System Services file system

- Required authorizations for System Data Engine operations
- If you are collecting SMF data from an in-memory resource or log stream, the user ID that is associated with the System Data Engine started task must have authority to read the SMF in-memory resource or log stream. Also, if you are collecting SMF data from a log stream, the user ID must have update access to the RACF profile `MVS™.SWITCH.SMF` in the OPERCMDS RACF class.
- If you are collecting SMF data from the SMF user exit, there are no other requirements for the user ID.
- The following information further describes the required authorities:
  - Authority to read the SMF log stream or in-memory resource:
    If you are using the Resource Access Control Facility (RACF) as your System Authorization Facility (SAF) product, you must give the System Data Engine user ID read authority to the profile that you set up to secure your SMF in-memory resource or log stream. In the following examples, `IFASMF.resource` represents the name of the SMF in-memory resource or log stream that is being used to gather SMF records, and userid represents the System Data Engine user ID.
    - For an in-memory resource:
      `PERMIT IFA.IFASMF.resource CLASS(FACILITY) ACCESS(READ) ID(userid)`
    - For a log stream resource:
      `PERMIT IFASMF.resource CLASS(LOGSTRM) ACCESS(READ) ID(userid)`
  - Update access to the RACF profile `MVS.SWITCH.SMF` in the OPERCMDS RACF class (only if you are collecting SMF data from a log stream):
    Update access to the RACF profile `MVS.SWITCH.SMF` in the OPERCMDS RACF class is required when collecting SMF data from a log stream, so that the user ID can issue the `MVS SWITCH SMF` command. The System Data Engine periodically issues the `MVS SWITCH SMF` command to ensure that it is accessing the most up-to-date data from the log stream. To grant the user ID update access to this RACF profile, issue the following commands:

    ```
    PERMIT MVS.SWITCH.SMF CLASS(OPERCMDS) ACCESS(UPDATE) ID(userid)
    SETROPTS RACLIST(OPERCMDS) REFRESH
    ```

    This authority is not required to process data from an SMF in-memory resource.

- Required authorities for Log Forwarder operations

  The user ID that is associated with the Log Forwarder start procedure must have the required authorities for file access and for issuing console messages.

  - File access authority:
    The Log Forwarder user ID must have the appropriate authority to access the Log Forwarder program files, which include the installation files, the configuration files, and the files in the working directory.

        - Installation file access: The Log Forwarder user ID must have read and execute permissions to the Log Forwarder installation files in the UNIX System Services file system.
        - Configuration file access: The Log Forwarder user ID must have read permission to the Log Forwarder configuration files in the UNIX System Services file system.
        - Working directory access: The Log Forwarder user ID must have read and write permissions to the Log Forwarder working directory.
    - Authority to issue console messages:
      The Log Forwarder user ID must have the authority to issue z/OS console messages.

  If you are using RACF as your System Authorization Facility (SAF) product, either use the GLARACF procedure in the SGLASAMP library to create a user ID for the Log Forwarder started task (GLAPROC procedure) and associate that user ID with the started task, or complete the following steps to assign this authority:

  a. In RACF, add the BPX.CONSOLE resource to the class FACILITY by using the General Resource Profiles option in the RACF Services Option Menu.
  b. In the BPX.CONSOLE profile that was created (or updated) in the preceding step, add the user ID that the Log Forwarder started task is associated with and assign read access to the user ID.
  c. Issue the following command to activate your changes: **`SETROPTS RACLIST(FACILITY) REFRESH`**

- Authorization the System Data Engine with APF:
  For the System Data Engine to gather System Management Facilities (SMF) data, the **SHBOLOAD** library must be authorized with the authorized program facility (APF).

  To authorize the SHBOLOAD library, a library name and volume ID must be in the list of authorized libraries in the PROG*xx* member of the SYS1.PARMLIB library.

  Use one of the following methods to authorize the SHBOLOAD library:

  - To include the SHBOLOAD library in APF at system IPL, add the following statement to a PROG*xx* member:
    ```
    APF ADD DSNAME(hlq.SHBOLOAD) VOLUME(volname)
    ```
  - To dynamically add the SHBOLOAD library to APF after system IPL, issue the following MVS command:
    ```
    SETPROG APF,ADD,DSNAME=hlq.SHBOLOAD,VOLUME=volname
    ```

Part 3. Practical integration scenarios

This part includes the following topics:

Integration with IBM Tivoli Decision Support for z/OS

Tivoli Decision Support for z/OS customers can leverage their existing reporting systems by loading Decision Support for z/OS data direct into IBM Db2 Analytics Accelerator (IDAA). Db2 Analytics Accelerator is a high-performance appliance that integrates business insights into operational processes.

By including IBM Common Data Provider for z Systems in this environment the System Data Engine (a data gatherer) can in batch provide Tivoli Decision Support for z/OS SMF data to Db2 Analytics Accelerator for z/OS for analysis and queries. This approach saves money and time in the following ways:

- Reduced Db2 for z/OS footprint because no data lands in Db2 tables
- Saves CPU consumption (no copy and no aggregation on a z System)
- Allows you to keep more data over a longer period
- Provides the ability to store and query timestamp and interval data for deep analysis
- Enables the retention of existing reporting systems

Integration with Splunk

Scenario

An IBM Z customer has a mobile application (MyMobileApp) that contains components on open systems, but also leverages CICS in the customer’s z/OS environment. Splunk is already in use by the MyMobileApp development team to analyze MyMobileApp open systems application logs. The development team needs to load CICS operational data related to MyMobileApp into Splunk in order to perform end-to-end problem determination. Testing of MyMobileApp new features is performed in specific time windows.

By incorporating IBM Common Data Provider for z Systems into the IT environment, the development team can:

- Gather CICS operational data in specific time windows.
- Filter CICS operational data based on the content (like a custom field referring to MyMobileApp).
- Send only selected data to the Splunk server accessible to the development team.
- With the analysis results the development team can pinpoint performance issues and modify the application as needed to improve its performance and to minimize conflicts for key resources.

Integration steps

Integrating Common Data Provider for z Systems with Splunk entails the following key steps:

Install and configure the Data Receiver on the target Splunk platform.

Install the CDPz Splunk Ingestion Kit app into the Splunk server

Configure the Common Data Provider for z Systems components to extract, format, and forward data to the Data Receiver

Installing and configuring the Data Receiver

The Data Receiver is a .jar file (DataReceiver.jar) that is downloaded to the Splunk platform. A Java runtime environment must be installed; Java 1.8 was used in this example.

The DataReceiver.jar file can be placed in any location; in this example `/opt/cdpdatareceiver` is used.

Two environment variables must be defined before the Data Receiver is started:

CDPDR_HOME – the location of the DataReceiver.jar file (in this example `/opt/cdpdatareceiver`).

CDPDR_PATH – the location where the Data Receiver will write the data received from CDPz into files that will be read by Splunk (in this example `/opt/cdpdatareceiver/data`).

> **Note:** The Data Receiver environment variables must also be available to Splunk. Once the environment variables are set, you will need to restart Splunk to ensure that they are picked up in the Splunk runtime environment.

A **cdpdr.properties** file should be created in the CDPRDR_HOME directory to define the Data Receiver parameters. Here is an example of the properties file:

```
port = 19080
cycle = 3
trace = y
ssl = n
```

The **port** parameter is the IP port the Data Receiver will listen on to receive data from CDPz. In this example we use port 19080. Whatever port is chosen should be opened through any local firewall.

The **cycle** parameter is the number if cycles used for the files storing the data records received from CDPz. This number determines how often the Data Receiver will cycle through files as it stores the records. For example, with a value of 3, it will initially create 3 files suffixed with -0, -1, and -2. At the top of each hour it will switch to the next file in the sequence to store incoming data. When it reaches the end it will begin again with the first file, thus limiting the amount of disk space used. It is best to start with this default, observe the pattern of received data and subsequent file sizes, and then adjust as needed for optimal disk space usage.

The **trace** parameter indicates if tracing is to be turned on or off. This is normally used for debugging purposes and should not be active for normal operations.

The **ssl** parameter indicates if Secure Sockets Layer (SSL) will be used between the Data Receiver and the CDPz Data Streamer. While this example does not use SSL, it is a good practice to use it. To use SSL some additional steps must be performed. Details are in the Common Data Provider documentation in the IBM Knowledge Center (https://www.ibm.com/support/knowledgecenter/SSGE3R_1.1.0/welcome.html).

- Download the `setupDataReceiverSSL.sh` script from the Common Data Provider installation directory on z/OS to the Data Receiver platform.

Note: If you use Microsoft Windows, download `setupDataReceiverSSL.bat.`

- (NOTE: If using Microsoft Windows, download `setupDataReceiverSSL.bat`)
- Make the file executable.
- Ensure that the CDPDR_HOME and CDPDR_PATH environment variables are defined.
- Ensure that the JAVA_HOME variable points to the Java home directory.
- Execute the `setupDataReceiverSSL.sh` with the input parameters documented in the script. The script will generate a local keystore and a public certificate.
- Upload the public certificate to the z/OS system in which the Common Data Provider is installed.
- Execute the `importCertificate.sh` script (provided within the Common Data Provider installation directory tree) to import the certificate into the Data Streamer's keystore.

Note: The parameters can be overridden when the Data Receiver is started using command line parameters; consult the Common Data Provider documentation for details.

Using the cdpdr.properties file, the Data Receiver can be started with the command

java -Dfile.encoding=UTF-8 -jar DataReceiver.jar

The Java parameter `–Dfile.encoding=UTF-8` is required, as that is the encoding CDPz uses to send data to the Data Receiver.

The command can be placed in a script to start the Data Receiver as a background process (and also to perform any other setup actions such as setting the location of the java executable). Such a script can be used as a basis to define the Data Streamer as a system service.

The Data Receiver will emit a startup message similar to the following:

<timestamp> HBO6064I CDP Data Receiver - Version: V1R1M0, Build ID: <ID>, APAR: <APAR ID>

Its process listening on the input port – in this example port 19080 – will verify that it is active and waiting to process data.

Installing the CDPz Splunk Ingestion Kit

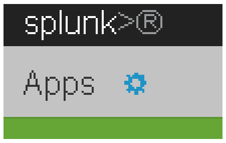
CDPz provides a Splunk Ingestion Kit consisting of a Splunk app file (a compressed gzipped TAR file with a `.spl` file extension) which is imported into Splunk. It contains the various definitions required for Splunk to process the incoming data. Versions are provided for both UNIX/Linux and Microsoft Windows platforms.

For this example, the UNIX/Linux `.spl` file (`ibm_cdpz_buffer_nix.spl`) is installed into Splunk with the following steps:

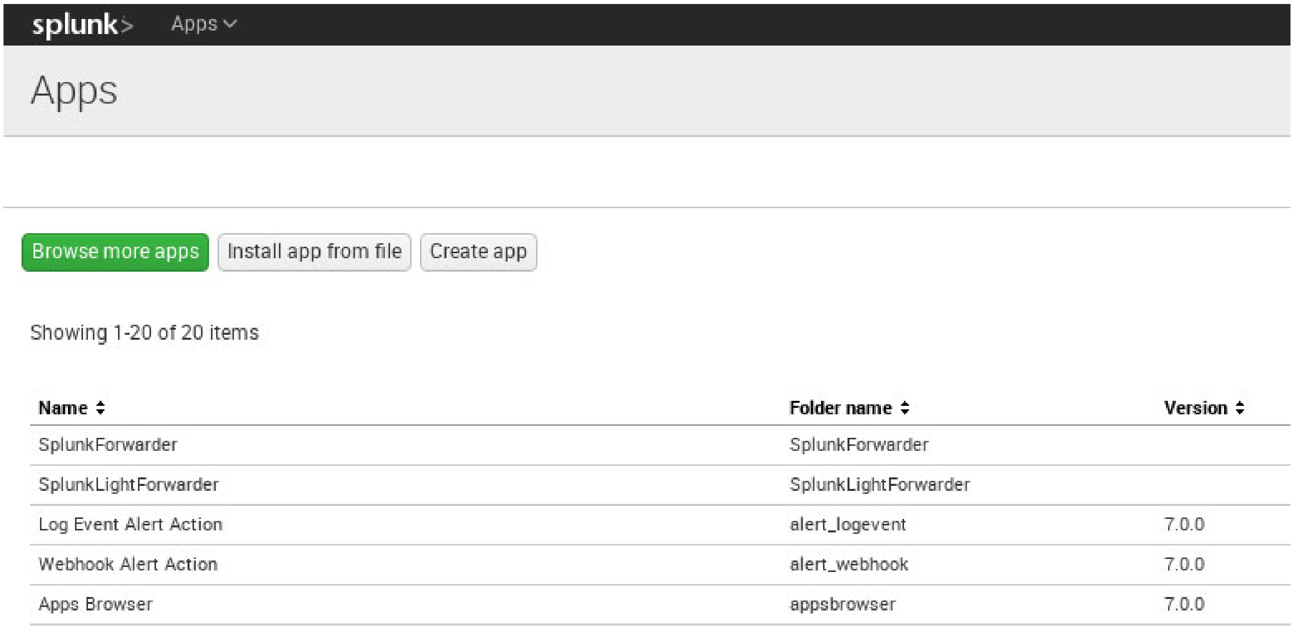
Download the `.spl` file to a local workstation (a workstation which can provide a browser session to log into Splunk).

Log in to the Splunk user interface on the same platform where the Data Receiver is running, using a Splunk administrator user ID.

On the left side of the menu, click the blue gear next to the word Apps:



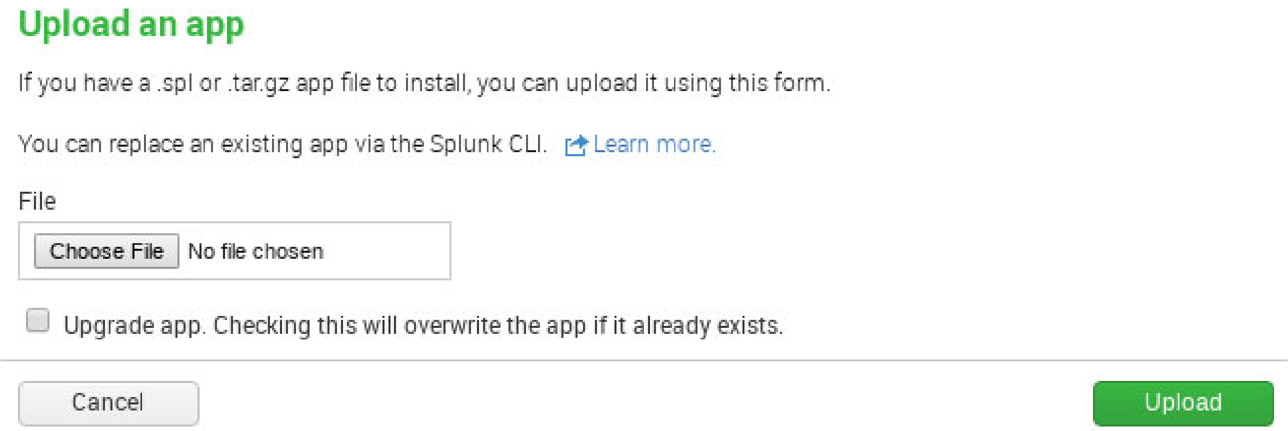
This will bring up the list of installed apps:



Showing 1-20 of 20 items

Name ⇕	Folder name ⇕	Version ⇕
SplunkForwarder	SplunkForwarder	
SplunkLightForwarder	SplunkLightForwarder	
Log Event Alert Action	alert_logevent	7.0.0
Webhook Alert Action	alert_webhook	7.0.0
Apps Browser	appsbrowser	7.0.0

Click the **Install app from file** button. This will display an upload window:



Upload an app

If you have a .spl or .tar.gz app file to install, you can upload it using this form.

You can replace an existing app via the Splunk CLI. Learn more.

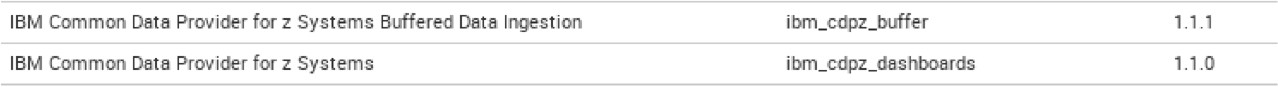
File

Choose File No file chosen

□ Upgrade app. Checking this will overwrite the app if it already exists.

Browse to where the `.spl` file was saved and select it. When prompted, select **Enable Now**.

When the upload and processing is complete, the integration app will be shown as IBM CDPz Buffered Ingestion:



IBM Common Data Provider for z Systems Buffered Data Ingestion	ibm_cdpz_buffer	1.1.1
IBM Common Data Provider for z Systems	ibm_cdpz_dashboards	1.1.0

Configuring the Common Data Provider

A flow must be defined in IBM Common Data Provider for z Systems to identify:

The data sources that will be used;

Any required transformations;

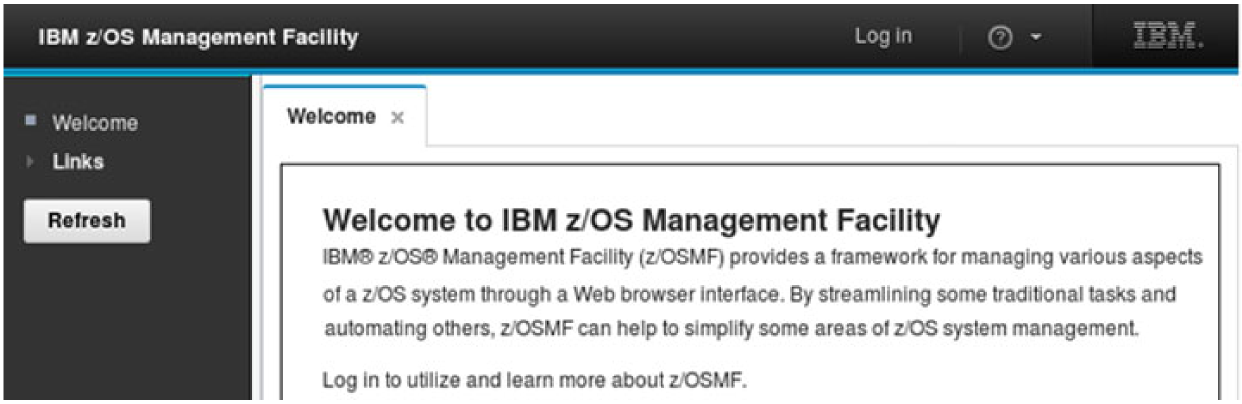
The subscriber to which the data is sent.

These actions are performed in the CDPz configuration tool, which is installed as a plugin to the z/OS Systems Management Facility (z/OSMF). The tool allows a graphical representation of the flow to be built, which then generates the files that the CDPz runtime components will read at their startup and use to collect, transform, and send data.

For the purposes of this example, we will create a policy that collects z/OS SYSLOG data and SMF Type 110 monitoring exception records.

Access the configuration tool as follows.

Enter the following URL in a browser: https://<*mainframe-url*>/zosmf/
The following browser page is displayed, including the **Log in** text in the upper right corner of the screen.

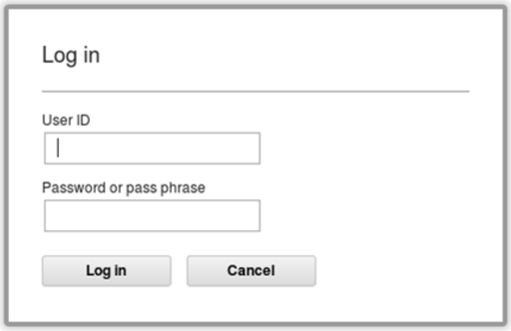


Welcome to IBM z/OS Management Facility

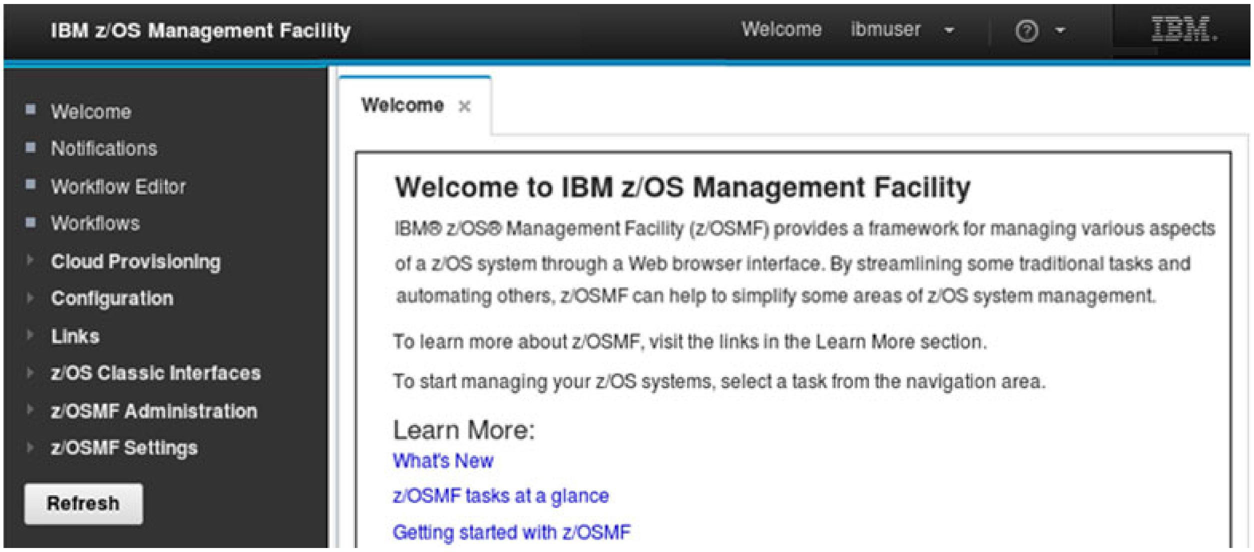
IBM® z/OS® Management Facility (z/OSMF) provides a framework for managing various aspects of a z/OS system through a Web browser interface. By streamlining some traditional tasks and automating others, z/OSMF can help to simplify some areas of z/OS system management.

Log in to utilize and learn more about z/OSMF.

Click **Log in**. The following dialog box is displayed.



Specify a TSO user ID and password, and click **Log in**. The following welcome screen is displayed.



Welcome to IBM z/OS Management Facility

IBM® z/OS® Management Facility (z/OSMF) provides a framework for managing various aspects of a z/OS system through a Web browser interface. By streamlining some traditional tasks and automating others, z/OSMF can help to simplify some areas of z/OS system management.

To learn more about z/OSMF, visit the links in the Learn More section.

To start managing your z/OS systems, select a task from the navigation area.

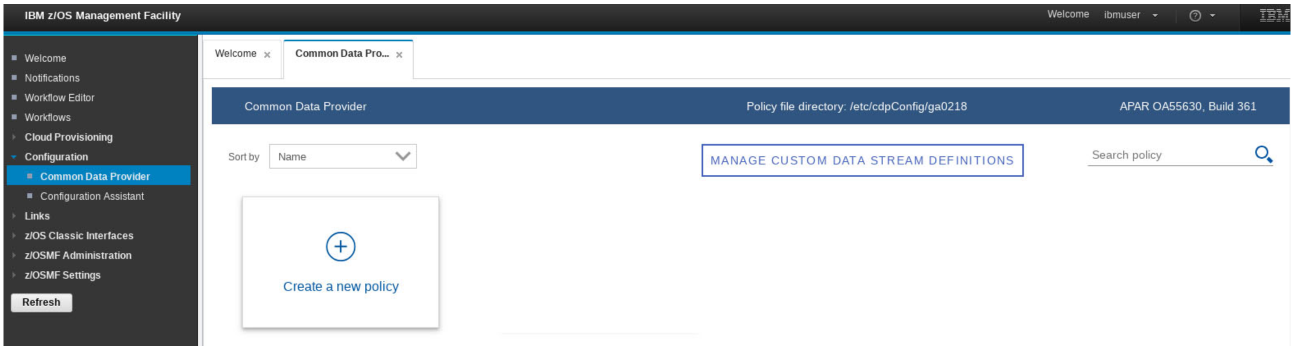
Learn More:

What's New

z/OSMF tasks at a glance

Getting started with z/OSMF

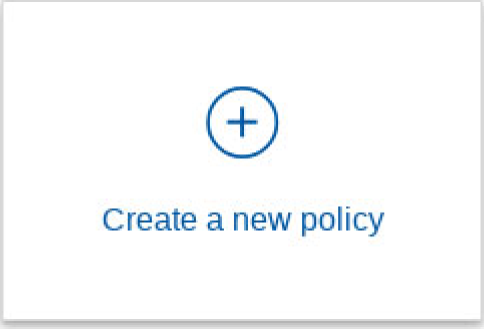
Click **Configuration > Common Data Provider** to access the configuration tool.
If no policies exist, your screen will look as follows:



Policy file directory: /etc/cdpConfig/ga0218

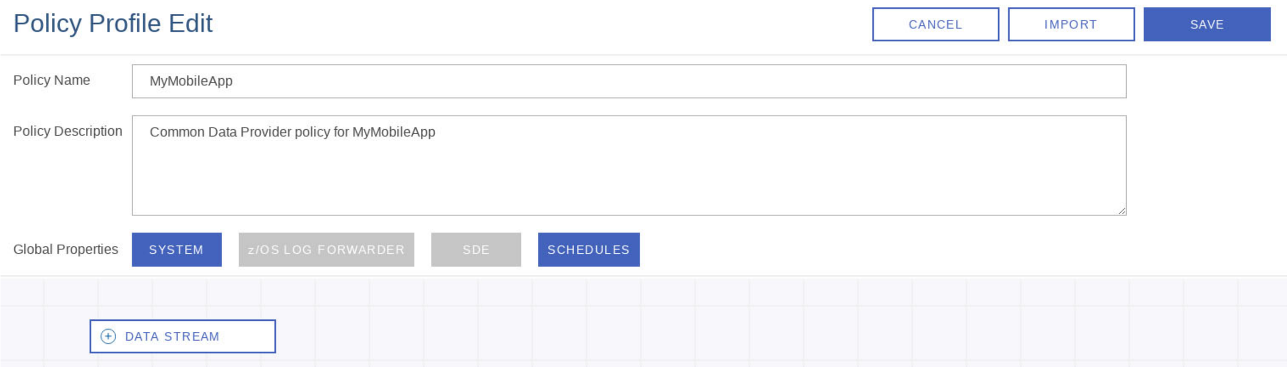
APAR OA55630, Build 361

After initial logon, click on **+** to create a new policy:



Note: Each policy must have a unique name. On the editor screen for the new policy, provide a new name: "MyMobileApp"

Click the **+ DATA STREAM** icon. This will display a list of defined data streams, grouped into **IMS Data, Log Data, SMF Data** and **KPI Data**.

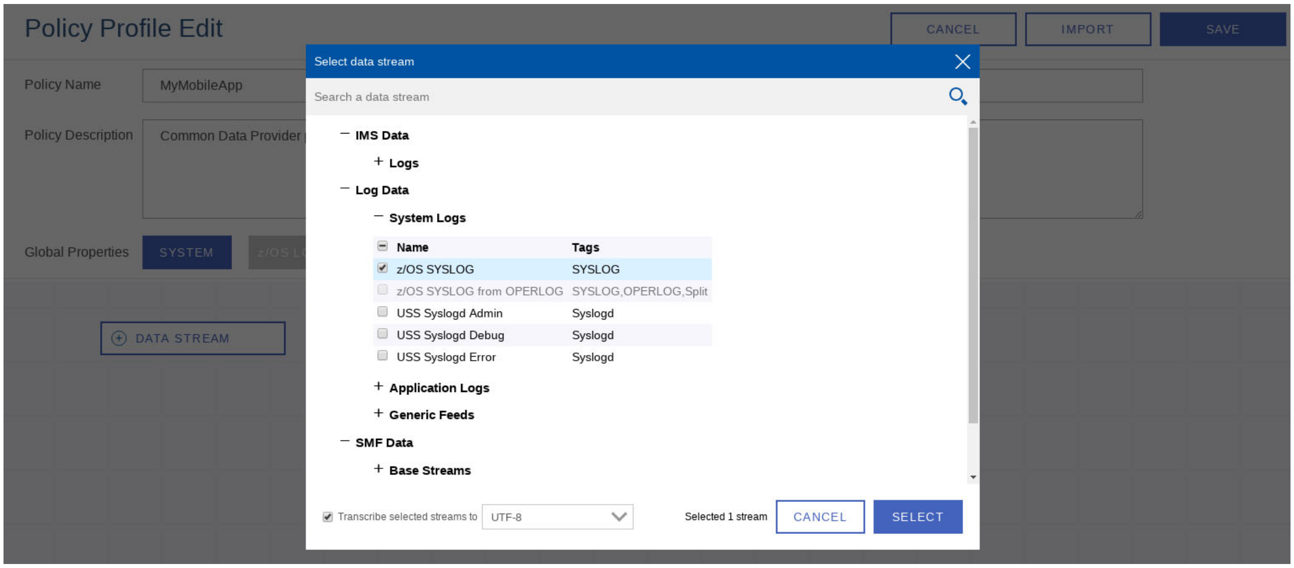


Policy Name: MyMobileApp

Policy Description: Common Data Provider policy for MyMobileApp

Global Properties

For a list of data streams under the **Log Data** click the **(+)** icon and select **z/OS SYSLOG**; then click on **SELECT**.



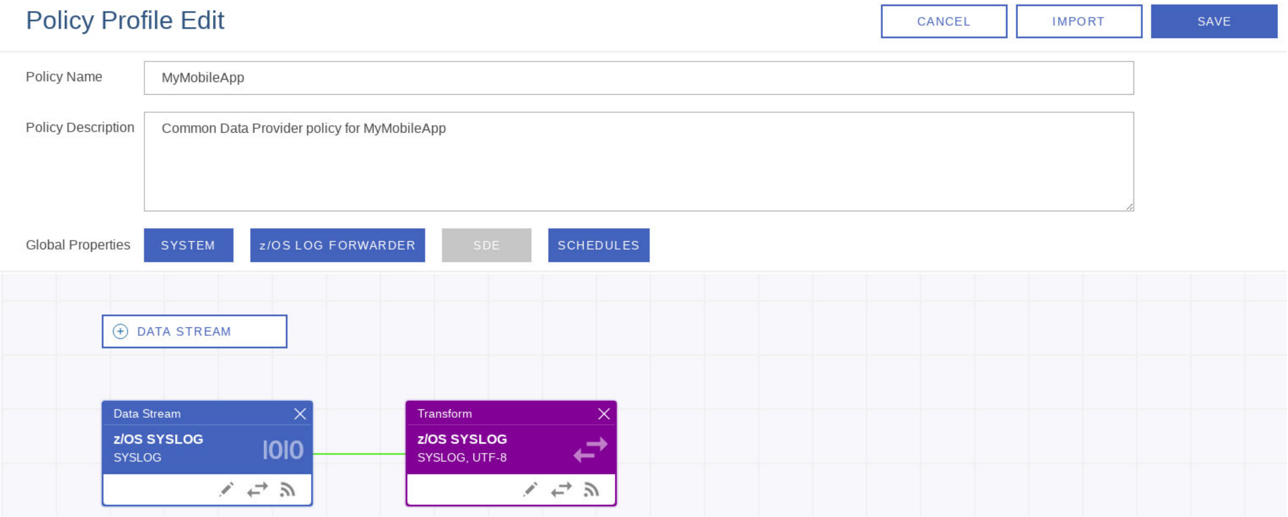
	Name	Tags
☑	z/OS SYSLOG	SYSLOG
□	z/OS SYSLOG from OPERLOG	SYSLOG,OPERLOG,Split
□	USS Syslogd Admin	Syslogd
□	USS Syslogd Debug	Syslogd
□	USS Syslogd Error	Syslogd

☑ Transcribe selected streams to UTF-8

Selected 1 stream

Note: A **Transcribe** transform translates data into an alternate encoding expected by the subscriber. By default, all streams are converted from EBCDIC to UTF-8.

This will create two new icons on the policy canvas: **Data Stream: z/OS SYSLOG** and **Transform: z/OS SYSLOG**.



Policy Profile Edit

Policy Name: MyMobileApp

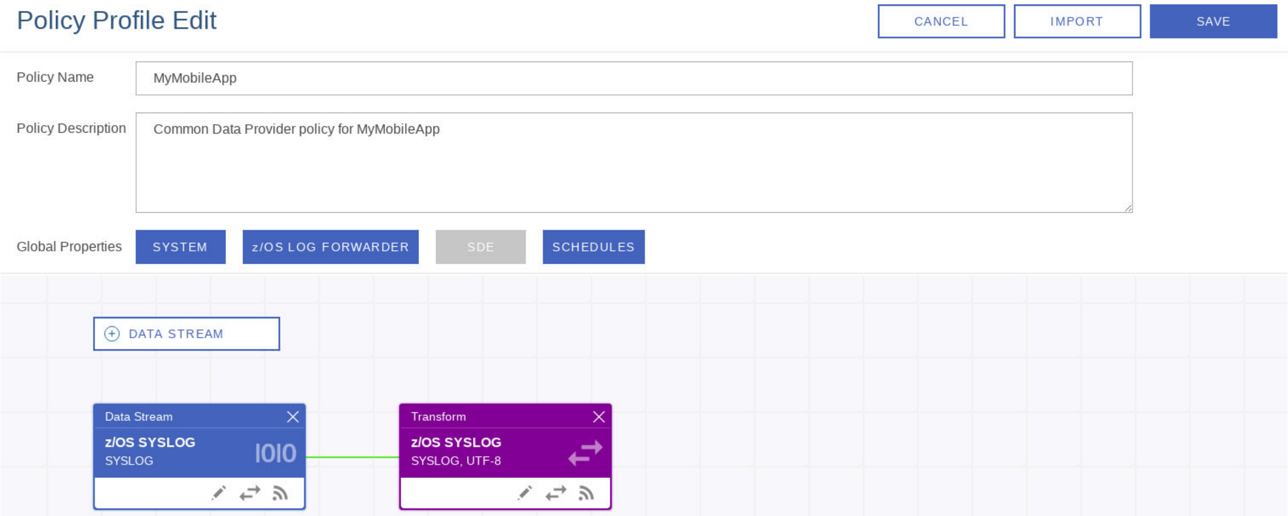
Policy Description: Common Data Provider policy for MyMobileApp

Splunk expects z/OS operational data to be sent in SPLIT format. To accomplish this, we will need to add in an extra transform step.

> **Note:** A **Splitter** transform splits a string of multiple messages into individual message records. Different Splitter transforms are available for different data stream types. Be aware that SMF records are always split by default and do not require a Splitter transform. z/OS SYSLOG data, on the other hand, or not split by default.

Click the two arrow button:

⇄

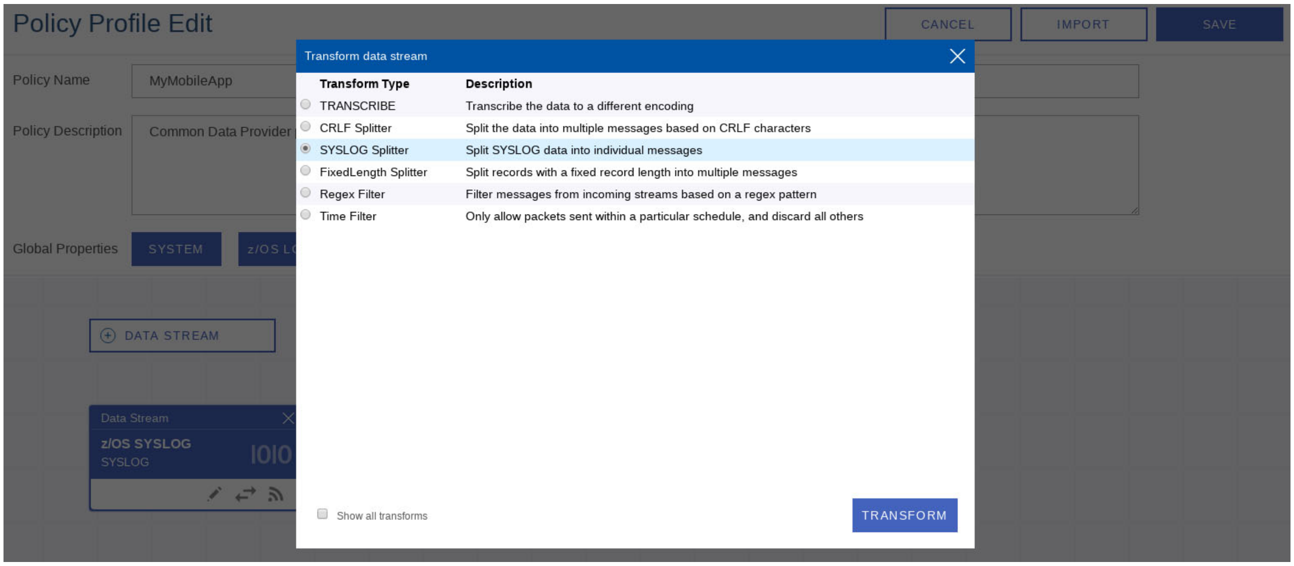


Policy Profile Edit

Policy Name: MyMobileApp

Policy Description: Common Data Provider policy for MyMobileApp

On the popup, select **SYSLOG Splitter** and click on **TRANSFORM**.



Policy Profile Edit

CANCEL IMPORT SAVE

Policy Name: MyMobileApp

Policy Description: Common Data Provider

Global Properties: SYSTEM z/OS L

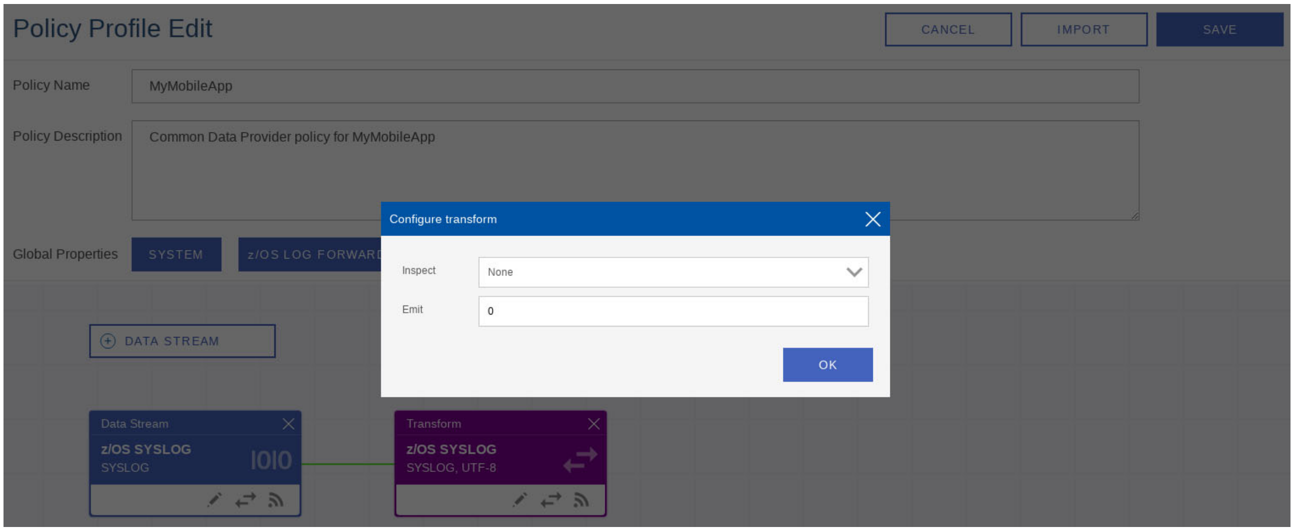
Transform data stream

	Transform Type	Description
○	TRANSCRIBE	Transcribe the data to a different encoding
○	CRLF Splitter	Split the data into multiple messages based on CRLF characters
●	SYSLOG Splitter	Split SYSLOG data into individual messages
○	FixedLength Splitter	Split records with a fixed record length into multiple messages
○	Regex Filter	Filter messages from incoming streams based on a regex pattern
○	Time Filter	Only allow packets sent within a particular schedule, and discard all others

□ Show all transforms

TRANSFORM

Click **OK** in the **Configure transform** dialog.



Policy Profile Edit

CANCEL IMPORT SAVE

Policy Name: MyMobileApp

Policy Description: Common Data Provider policy for MyMobileApp

Global Properties: SYSTEM z/OS LOG FORWARD

Configure transform

Inspect: None

Emit: 0

OK

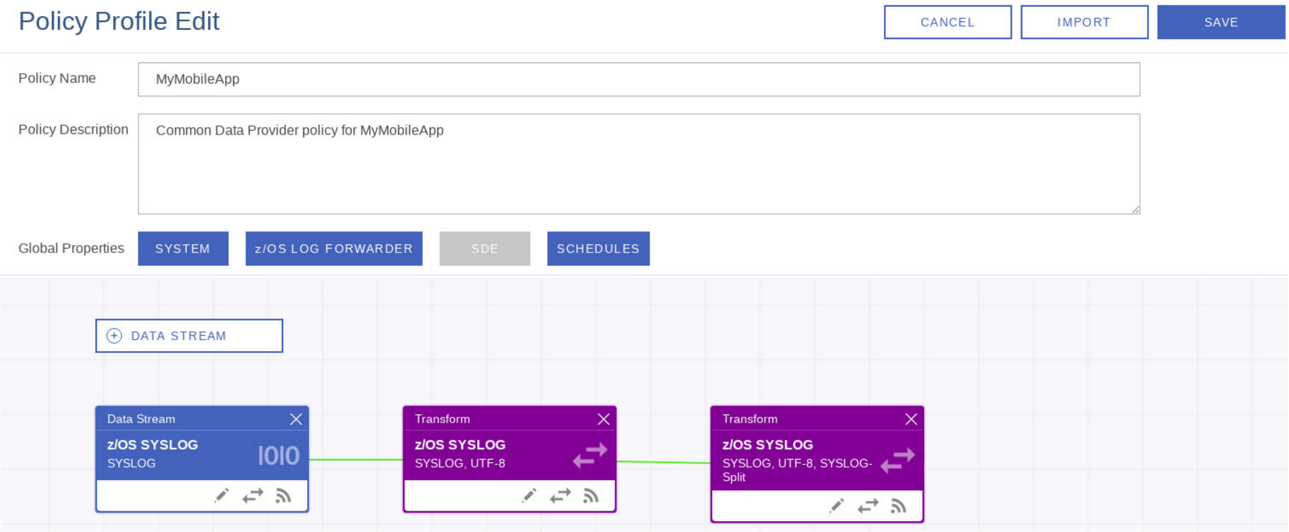
A new Transform icon is added to the policy canvas.

If you wish to filter the z/OS SYSLOG data stream, you can add a Regex or Time Filter transform to the policy.

> **Note:** A **Regex Filter** transform filters messages in the data stream according to a regular expression (regex) pattern. A **Time Filter** transform filters messages in the data stream according to a specified schedule. You can define both regex patterns and schedules.

Click the two-arrow button on the most recently created Transform icon:

⇄


Policy Profile Edit

Policy Name: MyMobileApp

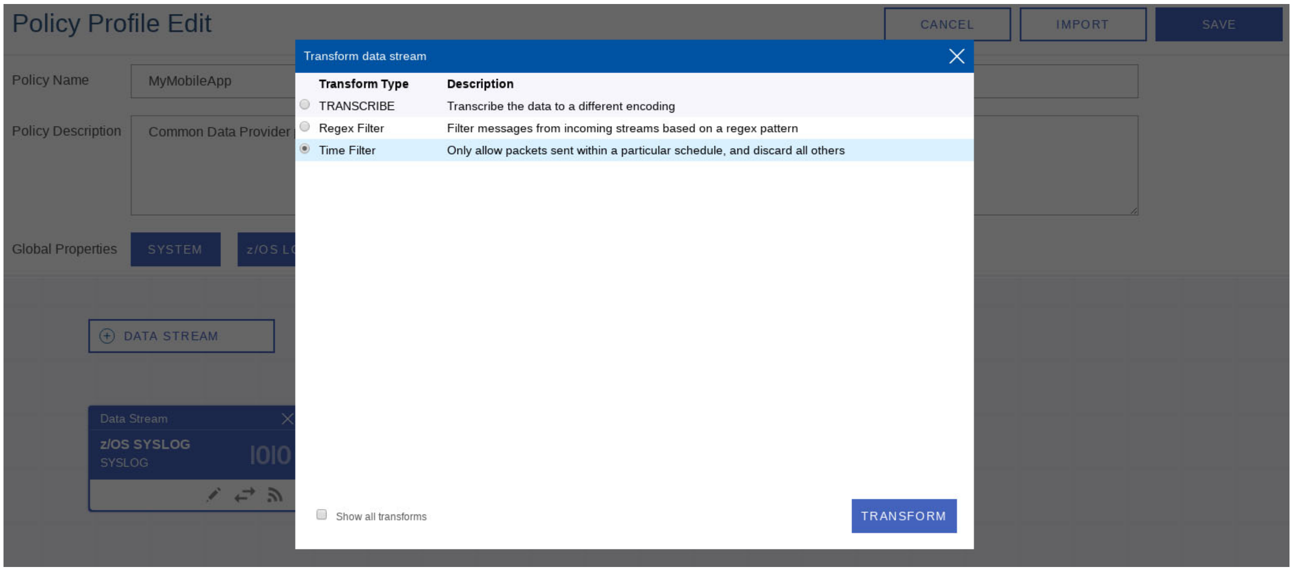
Policy Description: Common Data Provider policy for MyMobileApp

Data Stream — z/OS SYSLOG — SYSLOG

Transform — z/OS SYSLOG — SYSLOG, UTF-8

Transform — z/OS SYSLOG — SYSLOG, UTF-8, SYSLOG-Split

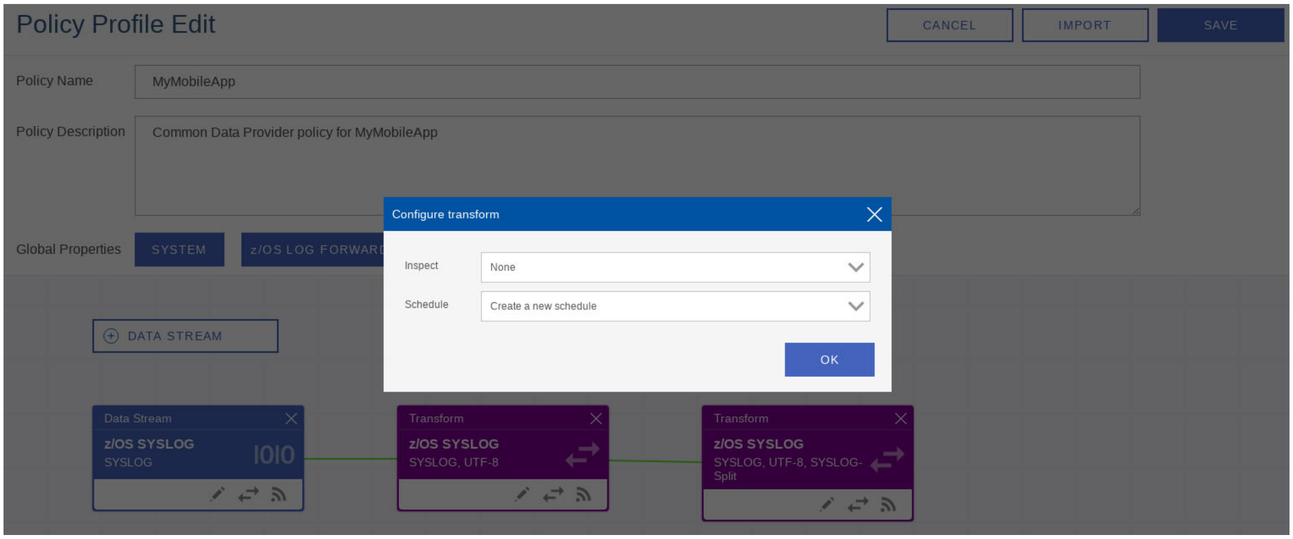
On the popup, select **Time Filter** and click on **TRANSFORM**.


Transform data stream

	Transform Type	Description
○	TRANSCRIBE	Transcribe the data to a different encoding
○	Regex Filter	Filter messages from incoming streams based on a regex pattern
●	Time Filter	Only allow packets sent within a particular schedule, and discard all others

□ Show all transforms

Click **OK** in the **Configure transform** dialog.



Policy Name: MyMobileApp

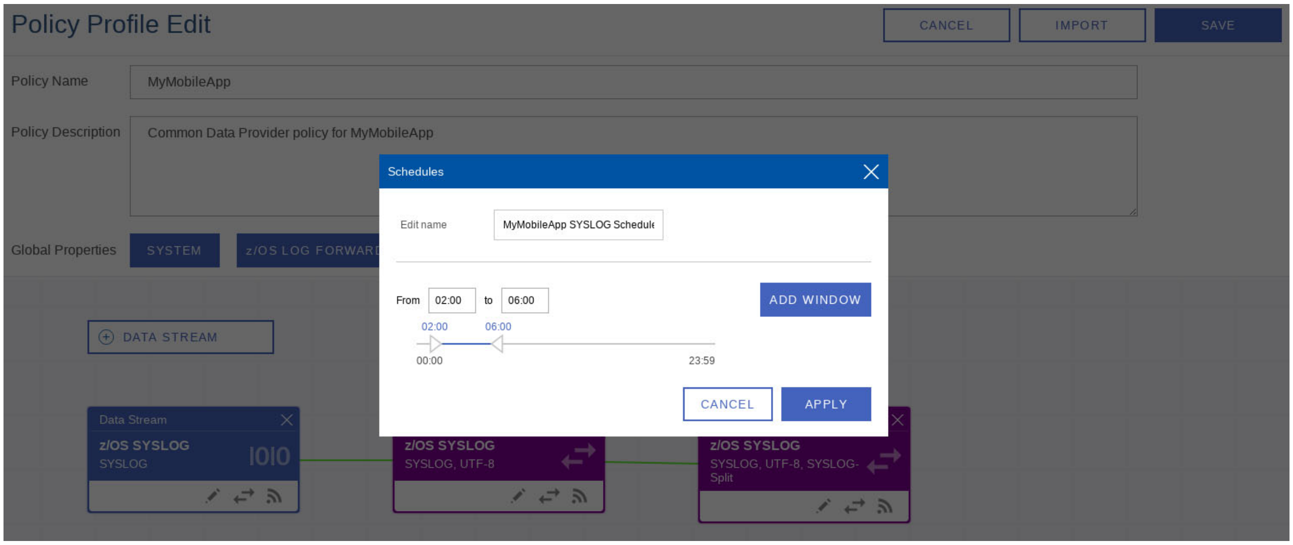
Policy Description: Common Data Provider policy for MyMobileApp

Configure transform

Inspect: None

Schedule: Create a new schedule

In the **Schedules** dialog, provide a name for the new schedule and fill in the time window information. Then, click on **Apply**.



Policy Name: MyMobileApp

Policy Description: Common Data Provider policy for MyMobileApp

Schedules

Edit name: MyMobileApp SYSLOG Schedule

From 02:00 to 06:00

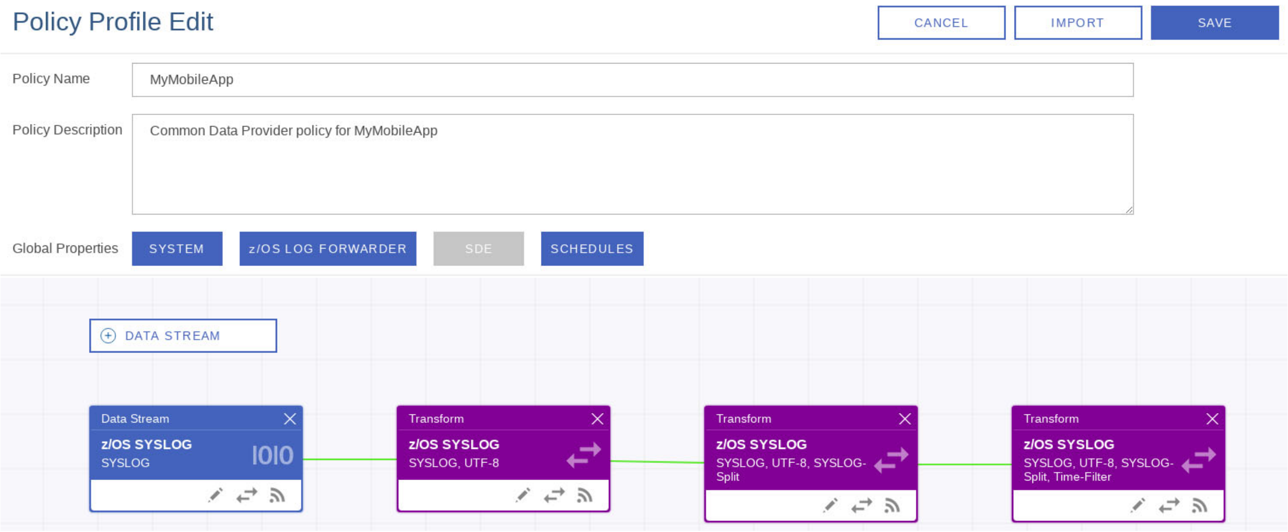
A third transform icon is added to the policy canvas.

We will now need to add a subscriber.

Note: A **Subscriber** defines a target destination for the operational data collected by CDPz. Since each CDPz instance can send operational data to multiple different analytics platforms, a policy may contain one or more subscribers.

Click the Subscribe button:

This, in turn, will open a popup.



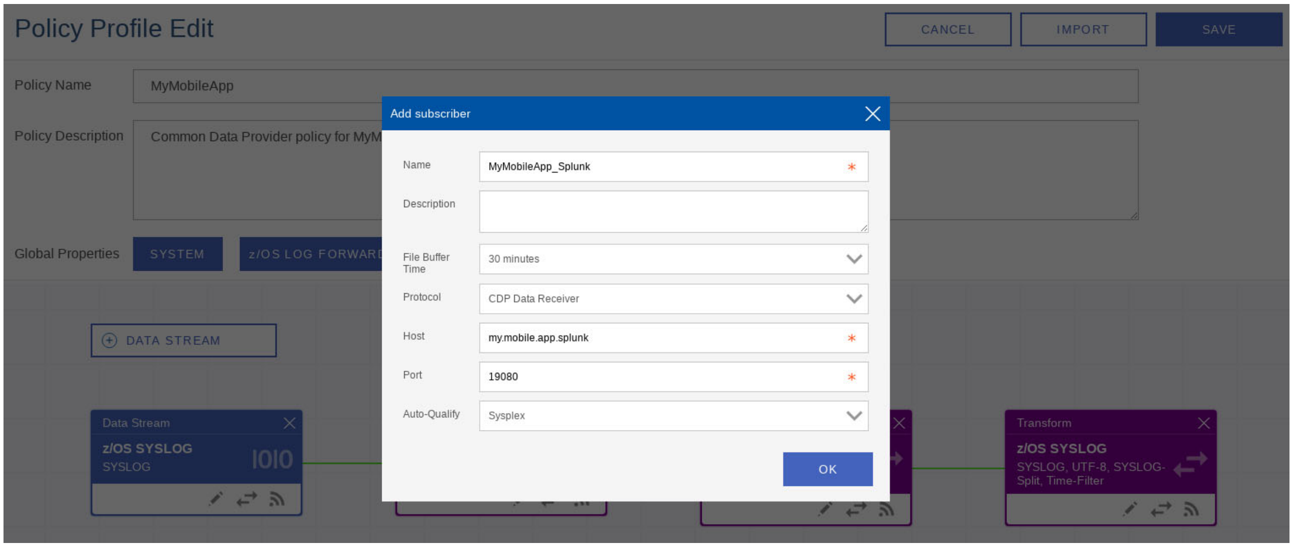
Policy Profile Edit

Policy Name: MyMobileApp

Policy Description: Common Data Provider policy for MyMobileApp

As no subscribers have been added yet, the list of subscribers is empty. Click **ADD SUBSCRIBER**.

Provide a name for your subscriber, as well as the following settings:
Protocol: CDP Data Receiver (CDP Data Receiver SSL if you with to set up secure communication)
Host: <your_splunk_host>
Port: 19080 (this value must match the port you defined for the Data Receiver on the Splunk node)



Add subscriber

Name: MyMobileApp_Splunk

Description:

File Buffer Time: 30 minutes

Protocol: CDP Data Receiver

Host: my.mobile.app.splunk

Port: 19080

Auto-Qualify: Sysplex

Click **OK**. Then, click on **UPDATE SUBSCRIPTIONS**.

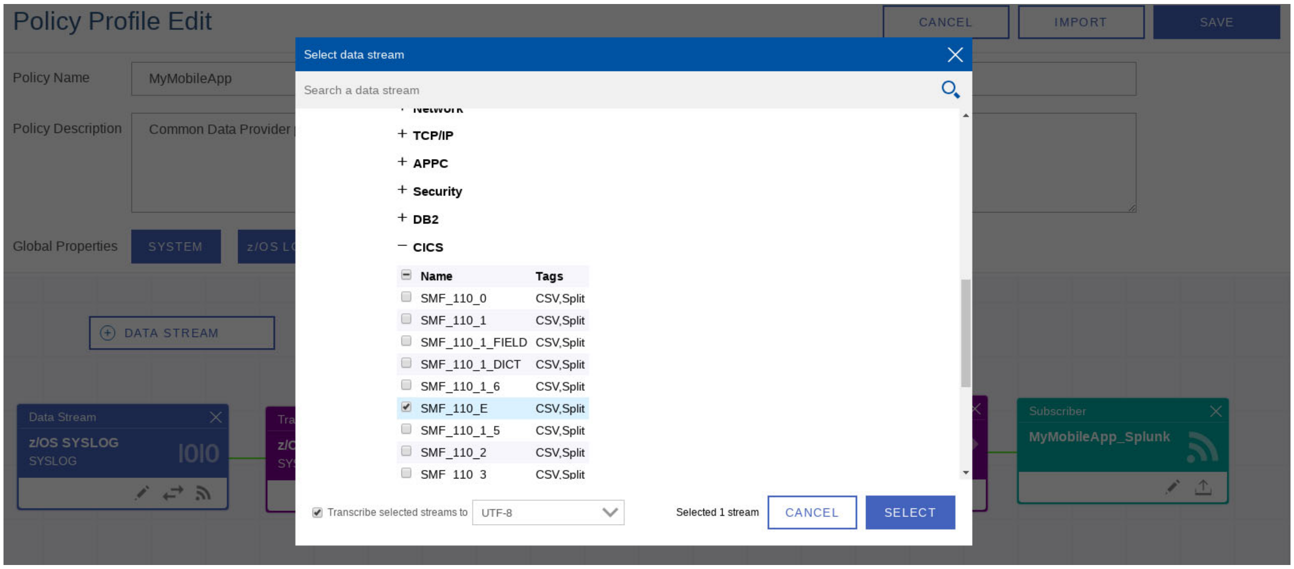
We now have a data stream that is transcribed to UTF-8 encoding, uses a SYSLOG splitter and a Time filter, and will send data to the Data Receiver located on your Splunk host.

Next, we will add support for SMF Type 110 monitoring exception records to the policy, following the same procedure as for z/OS SYSLOG data.

As the SMF data stream reference in the Common Data Provider documentation shows (`https://www.ibm.com/support/knowledgecenter/SSGE3R_1.1.0/smf_reference.html`), SMF

Type 110 monitoring exception records are represented by a data stream named SMF_110_E in the Common Data Provider. Consequently, we will add that data stream to the policy:

From the **Select data stream** dialog, select **SMF Data > Base Streams > CICS > SMF_110_E**:



Policy Profile Edit

CANCEL IMPORT SAVE

Policy Name: MyMobileApp

Policy Description: Common Data Provider

Global Properties: SYSTEM z/OS L

Select data stream

Search a data stream

Network
+ TCP/IP
+ APPC
+ Security
+ DB2
− CICS

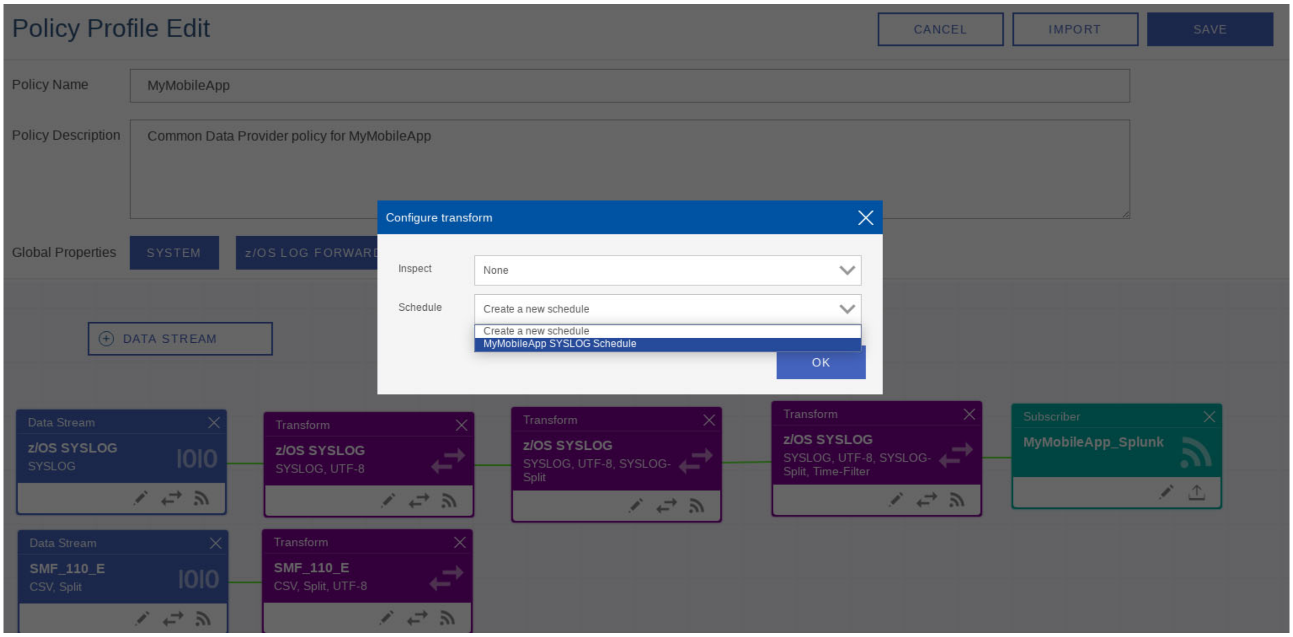
Name	Tags
□ SMF_110_0	CSV,Split
□ SMF_110_1	CSV,Split
□ SMF_110_1_FIELD	CSV,Split
□ SMF_110_1_DICT	CSV,Split
□ SMF_110_1_6	CSV,Split
☑ SMF_110_E	CSV,Split
□ SMF_110_1_5	CSV,Split
□ SMF_110_2	CSV,Split
□ SMF_110_3	CSV,Split

☑ Transcribe selected streams to UTF-8

Selected 1 stream CANCEL SELECT

Notice that this data stream is tagged with the word *Split*, indicating that data for this stream is always collected in split (single-message) form. A Splitter transform is therefore not needed for this data stream.

If desired, add a Time Filter transform. You can either re-use the schedule created for the z/OS SYSLOG stream, or set up a new schedule:



Policy Profile Edit

CANCEL IMPORT SAVE

Policy Name: MyMobileApp

Policy Description: Common Data Provider policy for MyMobileApp

Global Properties: SYSTEM z/OS LOG FORWARD

Configure transform

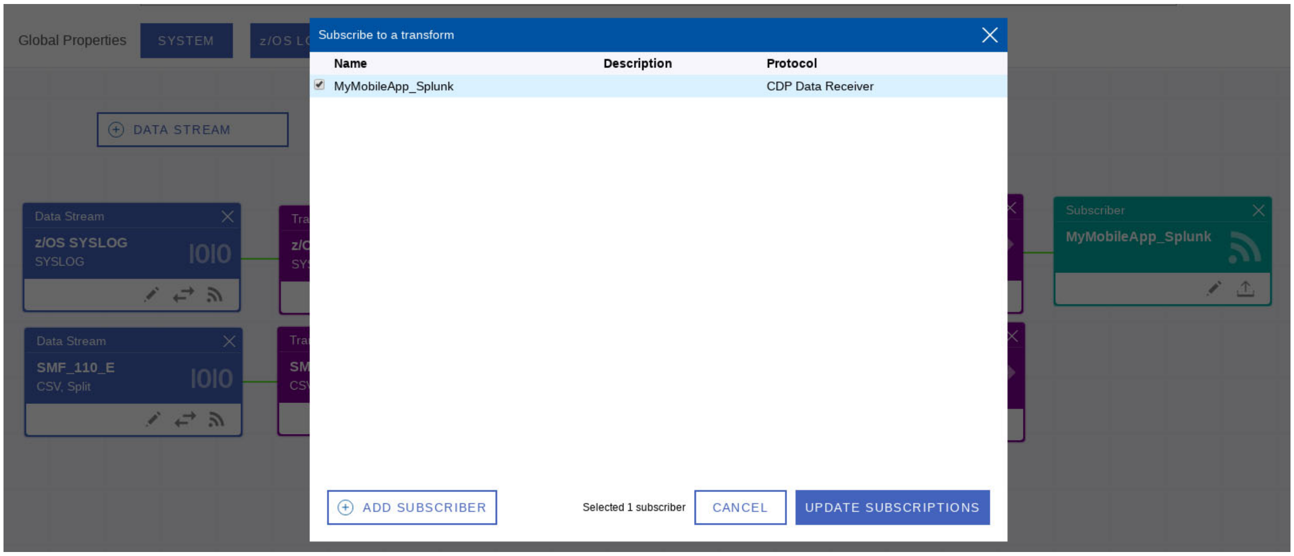
Inspect: None

Schedule: Create a new schedule

Create a new schedule
MyMobileApp SYSLOG Schedule

OK

Finally, add the stream to the existing Splunk subscriber:

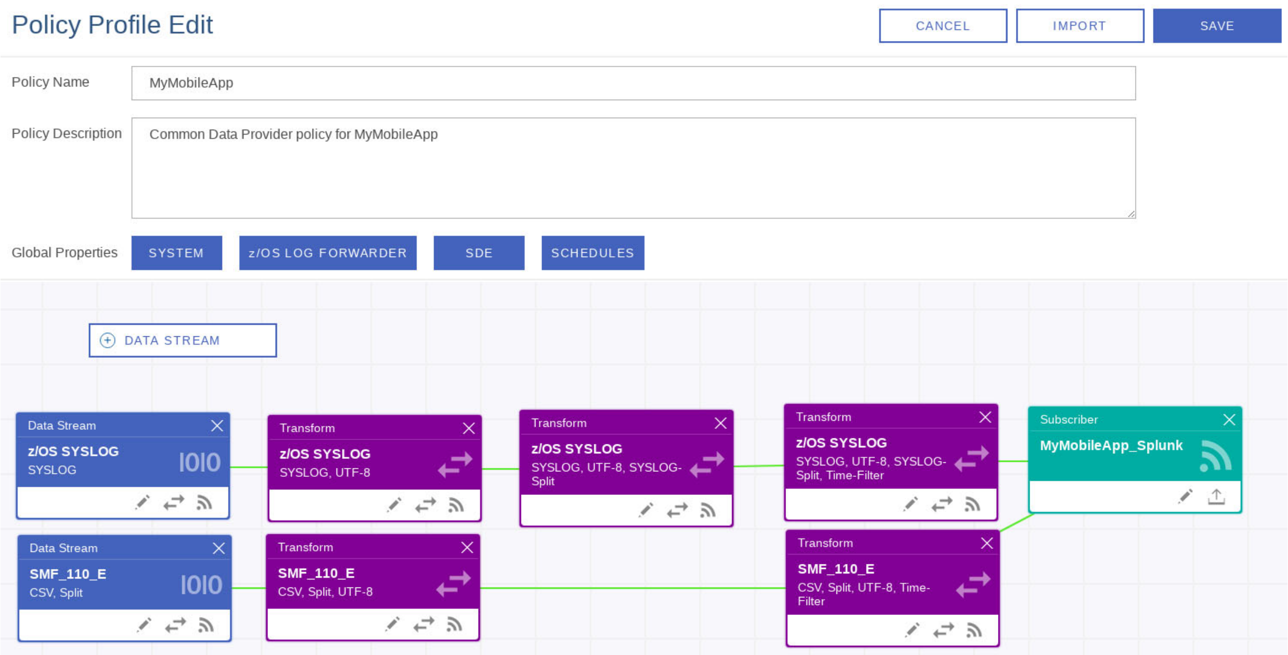


Subscribe to a transform

	Name	Description	Protocol
☑	MyMobileApp_Splunk		CDP Data Receiver

Selected 1 subscriber

The policy canvas now looks like this:



Policy Profile Edit

Policy Name: MyMobileApp

Policy Description: Common Data Provider policy for MyMobileApp

Before closing the policy, click on the **SDE** button and on the **z/OS LOG FORWARDER** button and make sure that all settings in the resulting dialogs are appropriate for your environment.

Then, click on the **SAVE** button to save and close the policy.
This will cause the creation of a set of configuration files in the policy directory. You will need to make these configuration files available to the Common Data Provider run-time components (System Data Engine, z/OS Log Forwarder and Data Streamer) as documented:

- Data Streamer
(`https://www.ibm.com/support/knowledgecenter/SSGE3R_1.1.0/ds_config.html`)

- z/OS Log Forwarder (https://www.ibm.com/support/knowledgecenter/SSGE3R_1.1.0/lf_config_files.html)
- System Data Engine (https://www.ibm.com/support/knowledgecenter/SSGE3R_1.1.0/sde_st.html)

The run-time components will need to be restarted for changes to the configuration files to take effect.

When messages arrive at the Data Receiver, they are written to files in the directory defined in the CDPDR_PATH variable. Each data stream will be represented by its own file set. Each record is in Comma Separated Value (CSV) format. The first seven fields of each record identify:

- Sysplex Name
- System Name
- Hostname
- Path
- Source type
- Source name
- Time zone offset from GMT time

The fields after that are parsed from the message contents.

Integration with the Elastic Stack

Scenario

The MyMobileApp scenario used to illustrate the integration with Splunk (Appendix , “Integration with Splunk” on page 19) can also be applied to the Common Data Provider’s integration with the Elastic Stack.

Integration steps

Integrating Common Data Provider for z Systems with the Elastic Stack entails the following key steps:

Install and configure the target Logstash instance.

Install the Common Data Provider Elasticsearch Ingestion Kit files onto the same platform as the Logstash instance and customize Logstash to use the files.

Configure the Common Data Provider for z Systems components to extract, format, and forward data to Logstash.

Installing and configuring Logstash

Logstash is relativity easy to installed, from binary distributions or package repositories. The Elastic Company provides the Logstash code installation steps at https://www.elastic.co/guide/en/logstash/current/installing-logstash.html

After the code is installed, the main configuration steps are to define a) the input sources, b) the processing to be applied against the received data, and c) the output sources to send the processed data. This is defined in a set of Logstash configuration files that define one or more of the following sections, in order:

- `input` – the source of the data (e.g. a file or a TCP/IP port)
- `filter` – processing to be done against the data
- `output` – the destination for the data that has passed through the filter stage.

For the Common Data Provider, this information will be provided in the next step.

Installing and configuring the Elasticsearch Ingestion Kit

The Common Data Provider Elasticsearch Ingestion Kit consists of a set of Logstash configuration files to process most of the data the Common Data Provider might send:

A Logstash input definition file for receiving data from CDPz.

A pair of Logstash files for each log and SMF record type supported by CDPz, to map and filter the input for Elasticsearch usage. Only the pairs for the actual data to be sent from CDPz should be used; providing Logstash with additional, unnecessary configuration files can have a significant negative impact on its performance.

A Logstash output definition file to send the data to an Elasticsearch index server, for use in the Kibana user interface or other processes that can access data stored in Elasticsearch.

The ingestion kit is packaged as a gzipped TAR file and must be download from the Common Data Provider installation directory on z/OS to the server running the target Logstash instance.

Once downloaded, the TAR file can be unpacked, and the appropriate configuration files can be copied to the defined directory from which Logstash reads its configuration files. The file names have the following naming convention:

File name prefix	**Processing stage**
B_	Input stage
E_	Preparation stage
H_	Field name annotation stage
N_	Timestamp resolution stage
Q_	Output stage

The naming allows the stages to be processed in the proper sequences, as Logstash reads the configuration files in lexical order. For example, to process SYSLOG data, these files are used:

`B_CDPz_Input.lsh` – this defines the input port Logstash will be listening on, as well as the format of the incoming data, which must be json. For example:

```
input {
    tcp {
        port => 8081
        codec => "json"
    }
}
```

`E_CDPz_Index.lsh` – provides a filter section that adds an index field to the data (required by Elasticsearch) that will be used as part of the overall record index (which is set in the output section).

`H_SYSLOG_OR_OPERLOG.lsh` – provides a filter section that maps the CSV data within the last JSON field in the incoming data to column names.

`N_SYSLOG_OR_OPERLOG.lsh` – provides a filter section that adds a timestamp field for when the record was received by Logstash.

`Q_CDPz_Elastic.lsh` – provides an output section that sends the processed data to Elasticsearch. For example:

```
output {
    elasticsearch {
        hosts => [ "10.1.1.132:9200" ]
        user => "elastic"
        password => "changemeorelse"
        action => "index"
        index => "cdp-%{[@metadata][indexname]}-%{+yyyyMMdd}"
    }
}
```

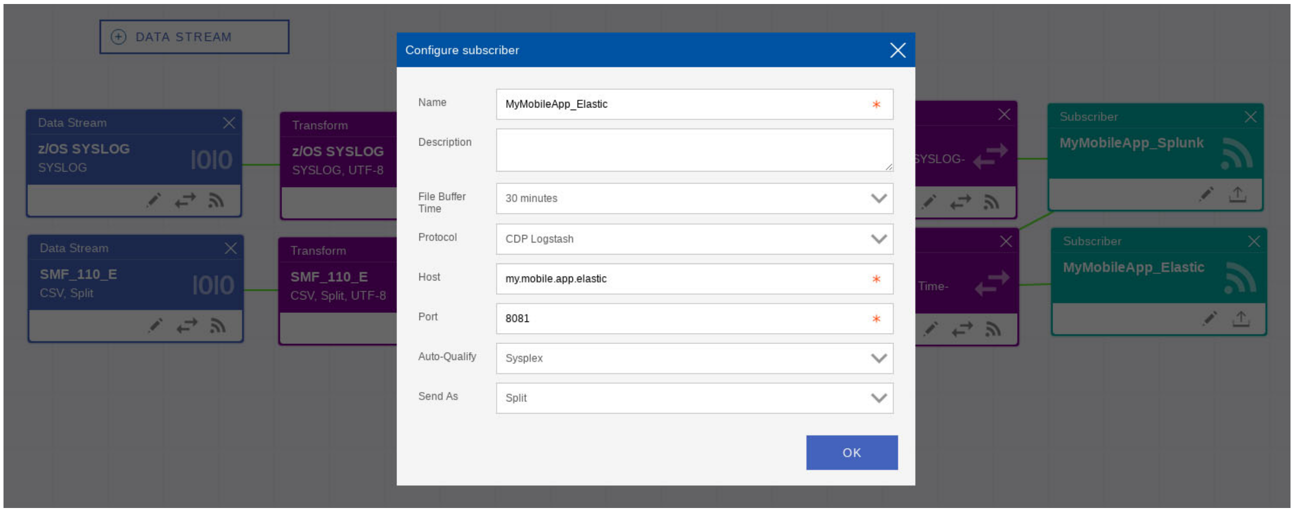
The `B_CDPz_Input.lsh`, `E_CDPz_Index.lsh` and `Q_CDPz_Elastic.lsh` files will always be used, as they are not associated with any specific data source. Which of the `H_` and `N_` files will be used depends on the types of data being sent. For example, if SMF Type 30 usage records are being sent from the Common Data Provider in addition to SYSLOG data, the `H_` and `N_` files for both SYSLOG and the SMF Type 30 records will need to be placed into the Logstash configuration file directory.

Configuring the IBM Common Data Provider for z Systems

A policy for integrating the Common Data Provider with the Elastic Stack will be nearly identical to an equivalent policy for integration with Splunk. The only difference is in the subscriber definition. As a matter of fact, it is easy to define both a Splunk subscriber and an Elastic Stack subscriber in the same policy should your use case require this.

To create a subscriber for the Elastic Stack, specify the following parameters:
Protocol: CDP Logstash (CDP Logstash SSL if you with to set up secure communication)
Host: <your_logstash_host>
Port: 8081 (this value must match the port you defined for Logstash in the `B_CDPz_Input.lsh` file)
Send As: Split

The **Configure subscriber** dialog will look similar to the following:



Configure subscriber

Name: MyMobileApp_Elastic *
Description:
File Buffer Time: 30 minutes
Protocol: CDP Logstash
Host: my.mobile.app.elastic *
Port: 8081 *
Auto-Qualify: Sysplex
Send As: Split

OK

As in the case of integration with Splunk, the new or updated configuration files need to be made available to the Common Data Provider run-time components, and the run-time components will need to be restarted to pick up the changes.

When messages arrive at Logstash, they will – by default – not be written out to the file system. Instead, they will be processed by Logstash in memory and then passed on to Elasticsearch. If you wish to write out the data to the file system in addition to sending them to Elasticsearch, you can easily extend the Logstash configuration with an additional output stage using the Logstash file or `csv` output plugins. It is a good practice to set up Logstash to pick up configuration changes automatically (without the need for a restart) to simplify the modification of the processing pipeline configuration; this can be accomplished through one of the following resources:

- The `–config.reload.automatic` command line parameter (`https://www.elastic.co/guide/en/logstash/current/running-logstash-command-line.html#command-line-flags`) or
- The `config.reload.automatic` property in the logstash.yaml file (`https://www.elastic.co/guide/en/logstash/current/logstash-settings-file.html`)

Processing the z/OS operational data in Logstash makes them available to the entire palette of Logstash filter and output plugins, enabling further integration with tools such as Apache Kafka, Apache Hadoop, and many others.

Additional resources

For more information on IBM Common Data Provider for z Systems, review the following information sources:

- IBM Common Data Provider for z Systems product page:

  https://www.ibm.com/ms-en/marketplace/common-data-provider-for-z-systems

- IBM Common Data Provider for z Systems Knowledge Center:

  https://www.ibm.com/support/knowledgecenter/SSGE3R

- IBM developerWorks wiki for IBM Common Data Provider for z Systems:

  https://ibm.co/2h1Tojk

- IBM Whitepaper: IBM Common Data Provider for z Systems: Splunk Integration Example:

  https://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102713

Authors

This guide was produced by a team of specialists from around the world working with the International Technical Support Organization (ITSO).

Michael Bonett is a Certified Executive IT Specialist (retired) in the United States of America. He has 39 years of IT experience with a focus on IBM Z Service Management and Cloud computing solutions and the integration capabilities of IBM Z. He holds a degree in Applied Mathematics and Computer Science from Yale University.

Domenico D'Alterio is an Offering Manager in Italy. He has 19 years of experience in the IT Service Management field. He holds a degree in Electronic Engineering from "La Sapienza" University. Rome. His areas of expertise include Workload Scheduling, Analytics and System Automation. He has written extensively on business value of IBM Common Data Provider for z Systems.

Keith Miller is a Senior Software Engineer in the United States of America. He has 30 years of experience in the software engineering field. He holds a degree in Computer Science from Pennsylvania State University.

Volkmar Burke Siegemund is a Software Services Delivery Manager in the United States of America. He has 18 years of experience in software development. He holds a degree in Ethnology from the Eberhard Karls Universität Tübingen, Germany.

Other authors include:

Eric Goodson

Matt Hunter

Fabio Riva

John Strymecki

Thanks to the following people for their contributions to this project:

Bill White and LindaMay Patterson
International Technical Support Organization

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, 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 http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

CICS®
Db2®
developerWorks®
IBM®
IBM Z®
IMS™
MVS™
NetView®
RACF®
Redbooks®
Redbooks (logo) ®
Tivoli®
WebSphere®
z Systems®
z/OS®
zSecure™

The following terms are trademarks of other companies:

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

REDP-5465-00

ISBN 073845706x

Printed in U.S.A.