

2.0

*IBM RMF Operational Analytics (as part
of Integration Monitor)
Installation and User Guide*



Note:

Before using this information and the product it supports, read the "Notices" topic at the end of this information.

April 2021 edition

This edition applies to Version 1 Release 1 of IBM RMF Operational Analytics and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2021; Copyright Rocket Software Inc., 2021.**

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

© **Copyright International Business Machines Corporation .**

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

Contents

About this information.....	V
Chapter 1. IBM RMF Operational Analytics: a part of IBM Z OMEGAMON	
Integration Monitor.....	1
Chapter 2. Installing RMF OA.....	3
IBM RMF Operational Analytics basic architecture.....	3
Before you begin.....	3
Installation procedure.....	4
Install Java.....	5
Install Elasticsearch.....	5
Configure Elasticsearch.....	5
Install and configure Kafka.....	6
Set up Zookeeper.....	7
Set up Kafka Broker.....	7
Set up Kafka Distributed Worker.....	8
Check Kafka Distributed Worker.....	9
Review potential security issues.....	9
Preparing for RMF OA.....	9
Install Ansible.....	10
Review and set parameter files prior to installation.....	10
Managing static fields.....	12
Execute the RMF OA installation file.....	12
Set up password encryption.....	13
Perform installation checks.....	14
View your Kibana environment space.....	14
Chapter 3. How to use RMF OA.....	15
IBM RMF Operational Analytics Spaces.....	15
RMF Reports.....	15
RMF OA Indexes.....	15
Naming standards.....	16
Index Patterns.....	16
RMF OA dashboards and visualizations.....	17
Analytics Space.....	17
Monitoring Space.....	21
Notices.....	25
Index.....	29

About this information

IBM Z OMEGAMON Integration Monitor displays performance information from a variety of sources, including OMEGAMON monitors and third-party software, in a single location. IBM RMF Operational Analytics (also known as RMF OA) is a new offering that helps you monitor and troubleshoot z/OS performance issues whilst leveraging historical data for trend analysis and analytics. RMF OA is available as part of IBM Z OMEGAMON Integration Monitor.

These topics provide instructions for installing, configuring, and using IBM Z OMEGAMON Integration Monitor.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform these tasks:

- Plan for the installation of RMF OA
- Install and operate RMF OA
- Use RMF OA

Chapter 1. IBM RMF Operational Analytics: a part of IBM Z OMEGAMON Integration Monitor

IBM RMF Operational Analytics, also known as RMF OA, allows you to monitor the activity of your z/OS® environment from a workstation through a modern and intuitive interface.

RMF OA takes its input data from the Distributed Data Server (DDS). One DDS per sysplex is needed to have a complete overview of your z/OS environment.

RMF OA provides sample dashboards and visualizations on the most used and expected Key Performance Indicators (KPIs). The information displayed is a subset of all the data collected from RMF reports. The contents of each collected RMF report is stored in a corresponding Elasticsearch index. These indexes allow the creation of archived data that is available indefinitely for data analytics purposes.

From the sample dashboards and visualizations provided, you can define your own customized KPIs fitted to your environment. The Elasticsearch data structure and visualizations may change over time to provide a migration path to IBM Z® Common Data Provider.

This document describes IBM RMF Operational Analytics, how to install it, and how to use it.

Chapter 2. Installing RMF OA

How to install and run RMF OA on z/Linux® Red Hat.

IBM RMF Operational Analytics basic architecture

An overview of how RMF OA is built.

- Each **IBM RMF Operational Analytics source connector** requests RMF data from an individual GPM server and stores it as XML messages into a Kafka topic
- **RMF OA Streams** transform XML message topics of GPM server data format into Kafka topics
- **Sink connectors** stream Kafka topics into Elasticsearch indexes

RMF OA collects data from RMF Distributed Data Servers (*GPMSEVERE*) over a period of time, or interval. Activate these intervals according to your analysis needs and the records that you wish to keep. Possible interval values are 60 or 100 seconds.

In our installation example, we define the following defaults:

- 1 source connector per GPMSEVERE
- 1 stream application
- 1 sink connector

Naming conventions used in our example:

Host name of the server	SERVERNAME
RMF OA Username	USERNAME
RMF OA User home directory	USERHOME
RMF OA Install directory	RMFOAHOME
Elastic Install directory	ELASTICHOME
Kafka Install directory	KAFKAHOME
GPM SERVER URL	GPMSEVEREURL
GPM SERVER NAME	GPMSEVERENAME

Before you begin

A checklist of tasks to perform before starting the installation process.

- **Supported platforms and prerequisites:**
 - Operating system: Red Hat Enterprise Linux 8.2.x (x386 or z)
 - z/OS versions: 2.2, 2.3, 2.4, 2.5
 - Other prerequisites: Java™ 11, Elasticsearch 7.12.1, Kibana 7.12.1, Kafka 2.4.1 (scala 2.11), Confluent Kafka Connect Elasticsearch 5.4.1
 - Recommended tools: Ansible 2.9.10, Python 3.8
- The RMF OA installation process requires a user account with root privileges (directly or via sudo).
- Ensure that the following TCP or UDP ports are open in both directions:
 - **9200** for the Elasticsearch server
 - **8083** for the Kafka server
 - Ensure that the Kafka server can access all configured DDS (port **8803**)

- **5601** for Kibana access
- Ensure that the RMF APAR number OA60119 for z/OS v2.3, v2.4 and v2.5, is installed on the DDS LPARs.
- LPAR z/OS versions 2.2, 2.3, 2.4, and 2.5 are supported.
- If all LPARs of sysplexes are running on different z/OS levels, ensure that the DDS LPAR is at the highest level for complete data collection.
- Set up memory parameters for the following components:
 - Elasticsearch: **jvm.options** in elastic/config directory
 - Kafka broker: **kafka-server-start.sh** in kafka/bin directory
 - Kafka worker: **connect-standalone.sh** in kafka/bin directory
 - RMF OA Streams: **rmfoa.yml** in the RMF OA Ansible installation
- Install the Elastic Stack Kibana component on a non-z/Linux server (as not compatible with z/Linux OS).
- Ensure the following components are installed:
 - OS
 - RHEL 8.2 running on z/Linux platform
 - selinux security should be disabled (root account needed)
 - Java
 - S/390® specific Java release used for all RMF OA components recommended is:

```
OpenJDK11U-jdk_s390x_linux_hotspot_11.0.7_10.tar.gz
```

It can be found on <https://adoptopenjdk.net/>

- Python 3.8
- Ansible 2.9.10
- Download the following RMF OA packages locally, using the USERNAME account:
 - int-rmfoa-app.connector.jar (used to connect to the gpm-servers)
 - int-rmfoa-app.stream.jar (used to administrate the Kafka in/out topics)
 - int-rmfoa-lib.sink-transform.jar (used to create specific index names for Elasticsearch)
 - int-rmfoa-tools.delivery.zip
- Download Elasticsearch version 7.12.1
- Download Apache Kafka 2.4.1 Scala 2.11
- Download the Confluent Elasticsearch Sink connector: <https://www.confluent.io/hub/confluentinc/kafka-connect-elasticsearch> (download previous version confluentinc-kafka-connect-elasticsearch-5.4.1.zip)

Note on the resources needed to collect 10 LPARs with a single RMFOA node:

Standard VM sizing	0-3 months	4-12 months
Memory GB	64	128
Disk TB	0.5	1
vCPU	16	32

Installation procedure

Provides an overview of the steps to install, configure, and deploy a new installation of RMF OA.

Install Java

The Java programming language is needed to run RMF OA.

1. As root, extract S/390 java archive to /usr/lib/jvm
2. Change default Java to this release, using symbolic links:

```
[roa]/home/roa>java -version
openjdk version "11.0.7" 2020-04-14
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.7+10)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.7+10, mixed mode)
```

Install Elasticsearch

Elasticsearch is an open source search and analytics solution.

1. Unzip the Elasticsearch distribution in USERHOME directory.
2. For ease of use, you may create and use a symlink to Elasticsearch directory.

```
-s USERHOME/ elasticsearch-7.12.1 USERHOME/elastic
```

3. If necessary, set the JAVA_HOME variable:

```
export JAVA_HOME=/usr/lib/jvm/jdk-11.0.7+10
```

4. With the root account, change the default for maximum number of file descriptors opened. To do so, edit /etc/security/limits.conf and add the following lines:

```
* soft nofile 65536
* hard nofile 65536
* root soft nofile 65536
* root hard nofile 65536
```

5. Restart your session.
6. Check the new limit with the command `ulimit -n`, it should be now 65365.
7. Increase vm max map count:

```
$ sudo sysctl -w vm.max_map_count=262144
```

8. To set it permanently, in /etc/sysctl.conf add:

```
vm.max_map_count=262144
```

9. All Elasticsearch indexes are stored under a directory called data. We recommend creating a symbolic link to a larger and faster disk to improve Elasticsearch performance:

```
-s /home/roa/elasticsearch-7.12.1/data /disk1/roa/data
```

Configure Elasticsearch

Elasticsearch configuration is performed using two configuration files located in ELASTICHOME/config directory, they are called **elasticsearch.yml** and **jvm.options**.

1. For Elasticsearch indexes, we recommend using a larger and faster disk to improve Kafka engine performance:

```
mkdir -p /disk1/roa/data
```

2. Adapt the following elasticsearch.yml example to your needs:

```
cluster.name: rmfoacluster
node.name: node-1
network.host: [_site_, _local_]
http.port: 9200
discovery.seed_hosts: ["127.0.0.1", "[:,1]"]
cluster.initial_master_nodes: ["node-1"]
```

```
bootstrap.system_call_filter: false
xpack.ml.enabled: false
search.max_buckets: 100000
```

3. In the `jvm.options` file, adjust the JVM memory according to the ingestion load and the number of end users.
4. Set the memory settings in the `jvm.options` file as the following (leave all other default memory parameters untouched, `Xms` is the initial size of total heap space, `Xmx` represents the maximum size of total heap space):

```
-Xms4g
-Xmx4g
```

5. Check the configuration, by running Elasticsearch using the script called `elasticsearch` located in the `ELASTICHOME/bin` directory. If the daemon stays up, this means the configuration is correct.
6. Cancel the process and start Elasticsearch using the following script to be set in the `USERHOME` directory:

```
startelastic.sh:
export JAVA_HOME=/usr/lib/jvm/jdk-11.0.7+10
cd elastic
rm -f nohup.out
rm -rf ./logs/*.log
nohup ./bin/elasticsearch -d
```

7. Run the script and after a few seconds, look in the `ELASTICHOME/logs` directory to see if the cluster is healthy.
8. You can also check that the Elasticsearch server is running correctly and visible from outside using the following URL in a browser: `http://SERVERNAME:9200/`

Install and configure Kafka

Apache Kafka is a distributed message queueing system. It logs events into channels, and Kafka holds on to those messages while consumers process them. Unlike a traditional “dumb” message queue, Kafka lets consumers keep track of which messages have been read. Its main assets are reliance, scalability and performance.

There are four different sections in Kafka:

- **Broker:** receives messages from producers, and stores them on disk.
- **Worker:** JVM Service, based on Kafka Connect, which is a framework service around connectors to move data into and out of Kafka. The service is a worker. Several connectors can be handled with one worker. Management is done through REST APIs and the connect name is often used to describe the REST API.
- **Topics:** A Topic is a category, or feed name to which records are stored and published. All Kafka records are organized into topics. Producer applications write data to topics through the Broker, and consumer applications read from topics. Records published stay until a configurable retention period has passed by. Kafka retains records in the log, making the consumers responsible for tracking the position in the log, known as the “offset”. Typically, a consumer advances the offset in a linear manner as messages are read. However, the position is actually controlled by the consumer, which can consume messages in any order.
- **Messages:** These are records of a topic.

1. Unpack the archive `kafka_2.11-2.4.1.tgz` to the `USERHOME` directory.
2. For convenience, symlink the kafka root directory `kafka_2.11-2.4.1`:

```
-s USERHOME/ kafka_2.11-2.4.1 USERHOME/kafka <= KAFKAHOME
```

Set up Zookeeper

The Kafka engine depends on ZooKeeper to store cluster configuration. By default, a Zookeeper server is embedded in the Kafka distribution, and the Zookeeper data directory resides in a directory by default set to /tmp. We recommend changing this device to another location for safety purposes.

Change KAFKAHOME/config/zookeeper.properties configuration file as follows:

```
dataDir=/disk1/roa/zookeeper
clientPort=2181
maxClientCnxns=0
admin.enableServer=false
```

Set up Kafka Broker

A Kafka broker handles all query requests and is managed using this configuration file: KAFKAHOME/config/server.properties.

1. We recommend using a larger and faster disk to improve Kafka engine performance:

```
mkdir -p /disk1/roa/kafka-logs
```

2. Change the logs-dirs variable of KAFKAHOME/config/server.properties:

```
mkdir -p /disk1/roa/kafka-logs
```

3. Check the following important parameters:

- **kafka-logs:** kafka data directory.
- **auto.create.topics.enable:** flag to leave kafka creating the topic needed, should be set to true.
- **log.retention.hours:** retention period for all data in kafka (in log.dirs), normally more than 48.

Here is an example of a complete server.properties file we have tested:

```
broker.id=0
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
log.dirs=/disk1/roa/kafka-logs
num.partitions=1
auto.create.topics.enable=true
num.recovery.threads.per.data.dir=1
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
log.retention.hours=48
log.segment.bytes=1073741824
log.retention.check.interval.ms=300000
zookeeper.connect=localhost:2181
zookeeper.connection.timeout.ms=6000
group.initial.rebalance.delay.ms=0
message.max.bytes=7000000
```

4. If required, put in place **stop/start** scripts to facilitate the handing of Zookeeper processes.
5. As root, create a file called kafka-zookeeper.service in /etc/systemd/system directory. Edit this file as follows (where User is the user that will execute the process: USERNAME and ExecStart is the starting command: KAFKAHOME/bin/zookeeper-server-start.sh KAFKAHOME/config/zookeeper.properties):

```
[Service]
User=roa
ExecStart=/home/roa/kafka/bin/zookeeper-server-start.sh /home/roa/kafka/config/
zookeeper.properties
ExecStop=/home/roa/kafka/bin/zookeeper-server-stop.sh /home/roa/kafka/config/
zookeeper.properties
```

6. Create and start the service:

```
systemctl enable kafka-zookeeper.service
systemctl start kafka-zookeeper.service
systemctl status kafka-zookeeper.service
```

7. If required, put in place **stop/start** scripts to facilitate the handing of the broker processes.
8. As root, create a file called `kafka-broker.service` in the `/etc/systemd/system` directory. Edit this file as follows (where `User` is the user that will execute the process `USERNAME` and `ExecStart` is the starting command: `KAFKAHOME/bin/kafka-server-start.sh KAFKAHOME/config/server.properties`):

```
[Service]
User=roa
ExecStart=/home/roa/kafka/bin/kafka-server-start.sh /home/roa/kafka/config/server.properties
ExecStop=/home/roa/kafka/bin/kafka-server-stop.sh /home/roa/kafka/config/server.properties
```

9. Create and start the service:

```
systemctl enable kafka-broker.service
systemctl start kafka-broker.service
systemctl status kafka-broker.service
```

Set up Kafka Distributed Worker

Kafka workers manage connectors: producers and consumers that interact with topics and work on messages. Distributed worker hosts REST Api to check, create and control connectors. It is used by the RMF OA Ansible installation to add and configure all the required connectors depending on the GPM servers, reports and collection interval configuration. The service managing the distributed worker is mandatory.

1. As root, create a file called `kafka-connect.service` in `/etc/system.d/system` directory. Edit this file accordingly (where `User` is the user that will execute the process: `USERNAME` and `ExecStart` is the Starting command: `KAFKAHOME/bin/connect-distributed.sh KAFKAHOME/config/connect-distributed.properties`):

```
[Service]
User=roa
ExecStart=/home/roa/kafka/bin/connect-distributed.sh
/home/roa/kafka/config/connect-distributed.properties
ExecStop= kill <pid>
```

2. `confluentinc-kafka-connect-elasticsearch-5.4.1` is the standard connector Kafka uses to connect to ElasticSearch. Put the whole directory in the plugins directory:

```
[roa]/home/roa/plugins>ls
confluentinc-kafka-connect-elasticsearch-5.4.1
int-rmfoa-app.connector.jar
```

3. Check that the file is executable, then still as root, create and start the broker service:

```
systemctl enable kafka-connect.service
systemctl start kafka-connect.service
systemctl status kafka-connect.service
systemctl status kafka-broker.service
kafka-broker.service
Loaded: loaded (/etc/systemd/system/kafka-broker.service; static;
vendor preset: disabled)
Active: active (running) since Thu 2020-07-16 09:33:18 EDT; 17h ago
Main PID: 29654 (java)
CGroup: /system.slice/kafka-broker.service
└─29654 java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMill...

Jul 17 01:33:20 *****.rocketsoftware.com kafka-server-start.sh[29654]: [20...
Jul 17 01:43:20 *****.rocketsoftware.com kafka-server-start.sh[29654]: [20...
```

4. Here is a broker configuration `KAFKA_HOME/config/ connect-distributed.properties` example we have tested:

```
bootstrap.servers=localhost:9092
group.id=connect-cluster
```

```
key.converter=org.apache.kafka.connect.json.JsonConverter
value.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=false
value.converter.schemas.enable=false
offset.storage.topic=connect-offsets
offset.storage.replication.factor=1
config.storage.topic=connect-configs
config.storage.replication.factor=1
status.storage.topic=connect-status
status.storage.replication.factor=1
offset.flush.interval.ms=10000
rest.host.name=*****
rest.port=8083
plugin.path=/home/roa/plugins
connector.client.config.override.policy=All
producer.max.request.size=7000000
```

Check Kafka Distributed Worker

Use a web browser to perform a check on the distributed worker.

1. From a browser access `http://SERVERNAME:8083/`. This shows information on the worker process.
2. Access `http://SERVERNAME:8083/connectors` to see a list of the installed connectors. It should be empty at this stage.

Review potential security issues

On some sites security is strict and firewall rules cannot be avoided.

If necessary, open ports using the root account:

```
[roa]/etc/systemd/system> sudo systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2020-07-08 05:46:24 EDT; 1 weeks 1 days ago
    Docs: man:firewalld(1)
   Main PID: 2153 (firewalld)
  CGroup: /system.slice/firewalld.service
          └─2153 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

=>

sudo firewall-cmd --zone=public --add-port=9200/tcp
sudo firewall-cmd --zone=public --add-port=8083/tcp
sudo firewall-cmd --zone=public --add-port=9601/tcp
```

Preparing for RMF OA

Some useful definitions to prepare you for the next stages of the installation process.

Control node: Any machine with Ansible installed. You can run commands and playbooks from any control node.

Managed nodes: The network devices you manage with Ansible. Ansible is not installed on managed nodes.

Inventory: A list of managed nodes. Your inventory can specify information like IP address for each managed node.

Modules: The units of code Ansible executes. Each module has a particular use. You can invoke a single module with a task, or invoke several different modules in a playbook.

Tasks: The units of action in Ansible. You can execute a single task once with an ad-hoc command.

Playbooks: Ordered lists of tasks, saved so you can run those tasks in that order repeatedly. Playbooks are written in YAML and are easy to read, write, share and understand.

Install Ansible

Ansible is a tool to automate installation and deployment of applications through multiple servers. Ensure to use a root account for these steps.

1. Install PIP (python installation process) by changing dir to /tmp, and get the PIP module

```
:curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

2. Install PIP:

```
python get-pip.py
```

3. Search for Python development module:

```
yum search python |grep devel
```

4. Install an extra Python module:

```
yum install python-devel.s390x
```

5. Install additional modules as required:

```
yum install libffi-devel.s390x
yum install openssl-devel
```

6. Run ansible : pip install ansible. Ansible should then be available:

```
$ansible --version
ansible 2.10.3
  config file = None
  configured module search path = ['/home/roa/.ansible/plugins/modules', '/usr/share/ansible/
plugins/modules']
  ansible python module location = /usr/lib/python3.7/site-packages/ansible
  executable location = /usr/local/bin/ansible
  python version = 3.7.4 (default, Nov 13 2020, 07:57:27) [GCC 4.8.5 20150623 (Red Hat
4.8.5-44)]
```

Review and set parameter files prior to installation

RMF OA installation files consist of binaries for connectors and streaming, as well as an Ansible script and config files for the installation process. We describe here the main parameters used for a single instance installation. All the scripts and configuration files are located in the RMFOAHOME/delivery directory.

1. **servername.ini** is the root configuration file specified in the installation command line. In the following sample, comments are displayed in **bold** after the parameter:

```
[elasticsearch] List of Elasticsearch servers
SERVERNAME

[rhel] List of the RMF OA Servers nodes (kafka)
SERVERNAME

[all:vars] Variables for all nodes
ansible_connection=ssh
ansible_python_interpreter=/usr/bin/python Binary or link of python executable
ansible_remote_path=/usr/bin/ansible Binary or link of ansible executable
ansible_ssh_pass=password Password of the RMF OA User
ansible_sudo_pass=rootpassword Password of the root user
ansible_user=USERNAME Username of RMF OA
tls_ansible_path="/ca-certificates"
tls_host_path="/etc/ssl/server"
tls_host_certificate_pfx_path="/etc/ssl/server/{{ ansible_hostname }}.pfx"
tls_docker_certificate_pfx_path="/etc/ssl/server/{{ ansible_hostname }}.pfx"

rs_pass= Password of the RMF OA User
rs_user= Username of RMF OA

[rmfoa] Main RMF OA Server group, see rmfoa.yml for details
SERVERNAME

[rs] Elastic server group, see rs.yml for details
```

```
SERVERNAME
```

2. **rs.yml** contains all the Elasticsearch index structure configuration

```
---
# file: group_vars/rs

elastic_template_cfg:
  - { number_of_replicas: '1', number_of_shards: '1', refresh_interval: '50s', interval:
100, index_prefix: 'report', report_name: 'cpc-table' }
  topic_reports: ['cpc_table', 'cfact_table'] by topic_reports: [ 'cpc_table', 'cfact_table' ]
  - { number_of_replicas: '1', number_of_shards: '1', refresh_interval: '50s', interval:
100, index_prefix: 'report', report_name: 'cfact-table' }

elastic_template_cluster_cfg:
  - { number_of_replicas: '3', number_of_shards: '1', refresh_interval: '150s', interval:
100, index_prefix: 'report', report_name: 'cpc-header' }
  topic_reports: ['cpc_table', 'cfact_table'] by topic_reports: [ 'cpc_table', 'cfact_table' ]
  - { number_of_replicas: '3', number_of_shards: '1', refresh_interval: '150s', interval:
100, index_prefix: 'report', report_name: 'cfact-table' }

topic_replication_factor: 1
topic_partitions: 1
topic_reports: ['cpc_table', 'cfact_table']
topic_intervals: [100]

rmfoa_certificate_authorities_for_es: "/etc/ssl/server/rs-ca-v1.pem"
```

3. **rmfoa.yml** contains the main installation parameters for the servers. Comments in **bold** after the parameter.

```
---
---
# file: group_vars/rmfoa

# --> COMMON PARAMETERS
jolokia_jvm_agent_name: "jolokia-jvm-1.6.2-agent.jar"

docker_enable: false Install on docker machines
download_enable: false Download Apps from Nexus

rmfoa_connector_remote_path: "/usr/local/ansible/rmfoa/java" Local source dir for connectors
rmfoa_stream_remote_path: "/usr/local/ansible/rmfoa/java" Local source dir for streams
rmfoa_kafka_remote_path: "/usr/local/ansible/rmfoa/kafka"
rmfoa_remote_path: "/usr/local/ansible/rmfoa"
rmfoa_log_path: "/var/log/rmfoa"
rmfoa_elastic_remote_path: "/etc/elasticsearch" Temporary directory for Elastic templates,
subdir templates will be created
rmfoa_jvm_agent_path: "/usr/share/java"
rmfoa_cdp_templates:
  - { src: "cdp-elastic-template-commons.json.j2", target: "rmfoa-indices-template-v1-cdp-
commons.json", template_name: "rmfoa-indices-template-v1-cdp-commons" }

# --> ELK PARAMETERS
elastic_templates: false
elastic_policies: false

# --> CONNECTOR PARAMETERS
connector_sink_name_table: "rmfoa_sink_table"
connector_source_dds_buffer_size_s: 7200 Recovery period in seconds
connector_source_name: "rmfoa_source"
connector_sleep: 11
connector_producer_override_compression_type: "lz4"
connector_sink_backup_tasks_max: 2
#connector_tasks_max: 10
#connector_tasks_length: "{{ connector_tasks_max | int - 1 }}"
connector_topic_prefix: "rmfoa_raw_xml"
connector_sink_tasks_max: 25
connector_enable: true
connector_sink_security: false
connector_threads_check_enable: false

# --> STREAM PARAMETERS
# ----> TZ
stream_jvm_parameters: "-Xms1G -Xmx5G" Should be simplified for z/Linux
stream_jvm_monitor_parameters: ""
stream_gc_enable: false
stream_enable: true
```

```
# --> KAFKA PARAMETERS
kafka_clusters_container_names: ['control-center','connect','broker1','broker2','broker3']
kafka_plugin_path: "/usr/share/java"
kafka_connect_service: "confluent-kafka-connect" Should be the same as Kakfa service

# --> KIBANA
# ----> TZ
kibana_base_path: "/usr/share/kibana"
kibana_css_path: "${kibana_base_path}/optimize/bundles/" Kibana path for style sheets
kibana_spaces_enable: false
kibana_settings_dateFormat_tz: "UTC" Timezone used by Kibana display
kibana_files_local_path: "../../../commons/kibana"
kibana_files_remote_path: "${rmfoa_remote_path}/kibana"

kibana_spaces_CDP:
- { id: "rmf-cdp-analysis", space_filename: "RMFOA_CDP_SPACE-0A.json",
  saveobjects_filename: ["RMFOA_CDP_KPI_OA-v0.0.1.ndjson"] }
- { id: "rmf-cdp-monitoring", space_filename: "RMFOA_CDP_SPACE-0M.json",
  saveobjects_filename: ["RMFOA_CDP_KPI_OM-v0.0.1.ndjson"] }
```

4. **servername.yml** contains the collect configuration. `connector_cfg` section should contain one line per couple Gpm Server / interval and includes report selection. Use the optional `system_id` parameter to filter the request on a specific system id on the sysplex. The interval parameter must be set to 100.

```
---
interval: [100]
connector_cfg:
- { id: 'gpm', interval: '100', protocol: 'http', url: 'server:8803', index: 1, report:
  'CPC,CFACT', system_id: 'XXXX', tasks_max: 5, delay: 25 }
stream_cfg:
- { interval: '100', tomcat_port: '9501', actuator_port: '9601', profiles:
  'ansible-100' }
```

5. To connect to GPMServe with the HTTPS protocol, use the following configuration:

```
connector_cfg:
- { id: 'gpm', interval: '100', protocol: 'https', username: 'user', password: 'pass',
  secretpath: 'path_to_store', url: 'server:8803', index: 1, report: 'CPC,CFACT', system_id:
  'XXXX', tasks_max: 5, delay: 25 }
```

For tighter security, see section [Set up password encryption](#), and encrypt the password.

Managing static fields

A few fields, present in the CDP Indexes are not populated through GPMServe data.

- hostname
- path
- sourcetype
- sourcename

To customize the definition of these fields:

1. Create a properties file **ibm.properties** in the SOURCE directory.
2. Populate this file with the static string values that are going to be added in the indexes for each record:

```
ibm.rmfoa.hostname=myhost
ibm.rmfoa.path=mypath
ibm.rmfoa.sourcetype=mysourcetype
ibm.rmfoa.sourcename=mysourcename
```

Execute the RMF OA installation file

Once the parameter files have been reviewed, you are now set to execute the installation process.

Run the installation command:

```
[roa]/home/roa>cat install.sh
ansible-playbook -c local
./delivery/scripts/ansible/commons/playbook/rs_rmfoa.yml -i
/home/roa/delivery/scripts/ansible/inventories/SERVERNAME.ini
-e "is_cdp_version=true docker_enable=false elastic_templates=true
ansible_connection=local elk_reset=false kibana_spaces_enable=true" -l SERVERNAME
```

The installation process will:

- Stop the existing stream or connectors if needed.
- Install Elasticsearch template if needed.
- Install the new binaries.
- Restart the platform with the latest configuration.

Set up password encryption

We strongly recommend you use this password encryption procedure. The connection to the DDS Server requires a userID and password authentication. The connector module has to provide this password in clear form to the DDS server.

1. To execute Kafka connect, create a service account on the machine used to deploy the connector module.
2. Using this account, encrypt your password with the dedicated library:

```
int-rmf-lib.utils-{version}-fat.jar
```

3. Once executed, the encrypted password is displayed. The secret key is stored in the `rmfoa.sec` file in the home directory of the service account. Restrict access to this file to the service account.
4. Copy the encrypted password into the Connector configuration file of the DDS Source Connector. This is also possible using an Ansible script.
5. Launch Kafka connect using the service account.
6. If encrypting several passwords, use the same encryption key by omitting the secret parameter when executing. It is possible, but more complicated, to use different store paths for different password.
7. The command to encrypt a password:

```
java -jar int-rmfoa-lib.utils.jar -s secret -p mypassword
```

Available parameters :

Parameter	Description	Example
<code>--passToEncrypt</code> or <code>-p</code>	The password you want to encrypt. Required: yes	mypassword
<code>--secret</code> or <code>-s</code>	A string you must keep secret, used to generate the secret key stored in <code>rmfoa.sec</code> . Required: no	secret123
<code>--storePath</code> or <code>-sp</code>	The path to the <code>rmfoa.sec</code> file. If not provided the user home directory is used. Required: no	/home/rmf

Example on Windows:

```
C:\temp>java -jar int-rmfoa-lib.utils.jar -s secret -p
mypassword
{CRYPT}B3CB473AF1AE2B27E4C5B608DC86F19FC88543D816EFA2D04584A01CA467998B83BFD78993E6
```

Example for multiple passwords:

First password:

```
java -jar int-rmfoa-lib.utils.jar -s secret -p mypassword1
```

Second password:

```
java -jar int-rmfoa-lib.utils.jar -p mypassword2
```

Perform installation checks

Use these commands to check the RMF OA running processes.

1. To show the installed connectors, in a browser, type the URL: `http://SERVERNAME:8083/connectors`
 - "rmfoa_source_rs28_100": source connector for GPM Server RS28 and 100s interval
 - "rmfoa_source_rbs1_100": source connector for GPM Server RBS1 and 100s interval
 - "rmfoa_sink_100": sink connector for 100s interval
2. To list all the topics currently used in Kafka, run `listtopic.sh`:

```
[roa]/home/roa>./listtopic.sh
__consumer_offsets
__transaction_state
connect-configs
connect-offsets
connect-status
rmfoa_raw_xml_100
rmfoa_v1_100_cfact_table_json
rmfoa_v1_100_cpc_table_json
...
```

3. To show the number of records per topic (only for RMF OA topics), use the command `numrecordtopic.sh`:

```
[roa]/home/roa>numrecordtopic.sh
rmfoa_raw_xml_100:0:29602
rmfoa_v1_100_cfact_table_json:0:2316836
rmfoa_v1_100_cpc_table_json:0:6192
...
```

4. To show all the Elasticsearch indexes, run `listindexes.sh`

```
[roa]/home/roa>listindexes.sh
-----INDEXES-----
yellow open cdp-zos-rmf_cpc-rsplex01-20211006          vdAH2giPTomTeuPWBbPKQw 1
1      0    0      283b      283b
...
```

View your Kibana environment space

Kibana is the data visualization tool for the Elasticsearch stack. RMF OA provides sample dashboards and KPIs to visualize RMF data. These samples are created automatically during the previous installation process.

View these sample spaces for RMF OA directly in Kibana.

Chapter 3. How to use RMF OA

RMF OA comes with sample embedded dashboards and visualizations to monitor the activity of your z/OS environment. Customize these dashboards and visualizations according to your needs.

RMF OA uses the Elastic stack (Kibana and Elasticsearch) as user interface. Use Kibana to perform the following actions:

- create dashboards
- define visualizations or KPIs
- manage indexes
- visualize your raw data

When on Kibana's introduction panel, select a Space, then select a dashboard corresponding to the topic you want to monitor or analyze. Then define a time range.

The Coupling Facility and CPC Monitoring Dashboards have visualizations which have been defined with an interval set to 100 seconds. If your shortest interval is not 100 but 60 seconds, feel free to change the index and the interval of all the visualizations of these two dashboards.

On each dashboard, you can set filters in order to have a more narrow view on your analysis and also pin them to propagate them to other dashboards.

IBM RMF Operational Analytics Spaces

On deployment, the following Spaces are defined to host the embedded dashboards and visualizations.

- **RMF OA Analytics** is dedicated to the analysis of data collected from RMF Reports. From this space you can visualize and understand the activities on your z/OS environments during different intervals.
- **RMF OA Monitoring** is dedicated to the near real-time monitoring of KPIs on activity patterns, Coupling Facility, CP consumption.

RMF Reports

RMF Reports are requested through GPMSERVE by using RMF Monitor III data collection.

Report	Report Full Name	Report Description
CFACT	Coupling Facility Activity	Provides data on activities on all structure
CPC	Central Processor Complex	Provide data to monitor values that are relevant for software pricing as well as partition related processor activities

RMF OA Indexes

Data collected from RMF Reports are stored in indexes managed by Elasticsearch. These indexes are accessible through Index Patterns defined in Kibana and selected during the creation of a visualization.

Manage an index from the interface by selecting **Management > Index Management** (Elasticsearch section). In addition, it is also possible to display the settings, structure, mapping and stats of an index.

Index Name	RMF Report/ Metric ID	Key of unicity
------------	--------------------------	----------------

cdp-zos-rmf_cfact- sysplexName-date	CFACT	RMF Report, sysplex, cpuSerialNo, CFAISTRU, CFAINAM, CFAPSYS, CFAICASI, timestamp
cdp-zos-rmf_cpc- sysplexName-date	CPC	RMF Report, sysplex, CPCHCPU,CPCHPNAM, timestamp

The CFACT and CPC indexes are populated with data collected from RMF Report which have respectively the same name.

Interval (data range)

The intervals 60 and 100, in seconds, are available for RMF data collections through GPMSEVERE for short-term data analysis.

You can only use either 60 or 100 seconds interval at a time.

Naming standards

Each index follows these naming standards:

indexname - cdp-sourcetype_*metricid*-*sysplexName-date*, where:

- *sourcetype* - zos-rmf
- *metricid* - the requested RMF Report. For example: CPC, CFACT.
- *sysplexName* - Name of the sysplex.
- *Date* - Date of the data into the index. For example: 20210602.

Index Patterns

Index patterns tell Kibana which Elasticsearch indexes contain the data that you want to work with. They are definitions used in a visualization to define which selection of indexes should be queried.

- Define index patterns in the interface by selecting **Management > Index Patterns** in the Kibana section.
- Define index patterns directly in TSVB or Timelion visualizations

The following is a list of the predefined RMF OA Analytics index patterns.

Index Pattern	Metric ID	Indexes
cdp-zos-rmf*	ALL	ALL
cdp-zos-rmf_cfact*	CFACT	cdp-zos-rmf_cfact-sysplexName-date
cdp-zos-rmf_cpc*	CPC	cdp-zos-rmf_cpc-sysplexName-date

The following is a list of the predefined RMF OA Monitoring index patterns.

Index Pattern	Metric ID	Indexes
cdp-zos-rmf*	ALL	ALL
cdp-zos-rmf_cpc*	CPC	cdp-zos-rmf_cpc-sysplexName-date

RMF OA dashboards and visualizations

RMF OA comes with a set of predefined visualizations. Over time, these will change and additions will be made.

Analytics Space

This section gives an overview of the monitoring dashboards, KPIs and visualizations supplied with RMF OA.

Note on accessibility issues: Kibana uses high contrast, color-blind safe palettes that have been tested with most assistive technology. Details can be found on the Elastic UI website.

For each report, a Search query has been defined and is reachable from the Discover panel or from the link defined in the **Representation of RMF Data Portal's Report** visualization of each Analytics Dashboard.

[Analytics] Dashboard - Report CFACT

Displays information on Coupling Facility Activities through defined structures.

In the Analytics Dashboard, select an interval before performing any analysis.

Used Index patterns: see section [Index Patterns](#) for some insights.

Visualizations	Description
[Analytics] CFACT - Links	List of different links available in RMF Analytics Space: - Analytics Dashboards (CFACT, CPC Sub-Capacity/TFP) - Spaces (RMF Analytics, RMF Monitoring)
[Analytics] CFACT - Top Utilization % by CF	List of the 10 CF with the highest total of processor utilization % for all the available data ranges. Limit: CF Name=10
[Analytics] CFACT - CF Utilization % by CF Structure	Visualization displaying the top 5 CF utilization % trend by CF Structure (CACHE, LIST & LOCK) for all the available data ranges. Limits: Structure=5, Structure Type=3, CF Name=10
[Analytics] CFACT - CF Utilization % by CACHE Structure	Visualization displaying the top 5 CF utilization % trend by CACHE Structure for all the available data ranges. Limits: Structure=5, Structure Type=1 (include CACHE), CF Name=10
[Analytics] CFACT - CF Utilization % by LOCK Structure	Visualization displaying the top 5 CF utilization % trend by LOCK Structure for all the available data ranges. Limits: Structure=5, Structure Type=1 (include LOCK), CF Name=10
[Analytics] CFACT - CF Utilization % by LIST Structure	Visualization displaying the top 5 CF utilization % trend by LIST Structure for all the available data ranges. Limits: Structure=5, Structure Type=1 (include LIST), CF Name=10
[Analytics] CFACT - Async & Sync Request Rate by CF Structure	Visualization displaying the CF asynchronous & synchronous request rate trend by structure for all the available data ranges. Limits: LPAR=*ALL, Structure=100

Visualizations	Description
[Analytics] CFACT - Async Request Count & Rate by CF & LPAR	Visualization displaying the CF asynchronous request count & request rate trends by CF & LPAR for all the available data ranges. Limits: CF Name=10, LPAR=50
[Analytics] CFACT - Sync Request Count & Rate by CF & LPAR	Visualization displaying the CF synchronous request count & request rate trends by CF & LPAR for all the available data ranges. Limits: CF Name=10, LPAR=50
[Analytics] CFACT - Representation of RMF Data Portal's Report	Links to a representation of the data as displayed in RMF Data Portal's CFACT Report

[Analytics] Dashboards - Report CPC - Subcapacity

Provides information on Software Pricing as well as partition related processor activity.

Visualizations	Description
[Analytics] CPC SubCapacity - Analytics - Links	List of different links available in RMF Analytics Space: - Analytics Dashboards (CPC Sub-Capacity/TFP) - Spaces (Analytics, Monitoring)
[Analytics] CPC SubCapacity - Top CPC by Highest MSU	Sub-Capacity Visualization displaying the list of the 5 CPC with the highest MSU consumption for all the available data ranges
[Analytics] CPC SubCapacity - Highest MSU Consumption by Lpar	Sub-Capacity Visualization displaying the synthesis of the 10 LPARs with the highest MSU consumption and their metric for all the available data ranges
[Analytics] CPC SubCapacity - Top Group Capacity by Highest MSU	Sub-Capacity Visualization displaying the list of the 10 Group Capacity with the highest MSU consumption for all the available data ranges
[Analytics] CPC SubCapacity - Top Sysplex by Highest MSU	Sub-Capacity Visualization displaying the list of the 10 Sysplex with the highest MSU consumption for all the available data ranges
[Analytics] CPC SubCapacity - Top Lpar by Highest MSU	Sub-Capacity Visualization displaying the list of the 10 Lpar with the highest MSU consumption for all the available data ranges
[Analytics] CPC SubCapacity - MSU Trend by Lpar	Sub-Capacity Visualization displaying the consumed MSU trend by LPAR for all the available data ranges. Limits: CPC=5 & LPAR=50
[Analytics] CPC - Mainframe Hardware Configuration	Most recent synthesis of the CPC hardware configuration. Limits: CPC=5
[Analytics] CPC - Processor allocation	List of Shared & Dedicated processors allocated for each System ID. Limits: CPC=5, Sysplex=10, System ID=50, LPAR=50, Processor Type=5
[Analytics] CPC - Physical CP Utilization % Trend	Visualization displaying the physical CP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *CP)

Visualizations	Description
[Analytics] CPC - Logical CP Utilization % Trend	Visualization displaying the logical CP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50 & Processor Type=1 (include set to *CP)
[Analytics] CPC - Physical zIIP Utilization % Trend	Visualization displaying the physical zIIP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *IIP)
[Analytics] CPC - Logical zIIP Utilization % Trend	Visualization displaying the logical zIIP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *IIP)
[Analytics] CPC - Physical ICF Utilization % Trend	Visualization displaying the physical ICF utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *ICF)
[Analytics] CPC - Logical ICF Utilization % Trend	Visualization displaying the logical ICF utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to ICF)
[Analytics] CPC - Physical IFL Utilization % Trend	Visualization displaying the physical IFL utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *IFL)
[Analytics] CPC - Logical IFL Utilization % Trend	Visualization displaying the logical IFL utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to IFL)
[Analytics] CPC - Physical zAAP Utilization % Trend	Visualization displaying the physical zAAP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *IFL)
[Analytics] CPC - Logical zAAP Utilization % Trend	Visualization displaying the logical zAAP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to IFL)
[Analytics] CPC - Representation of RMF Data Portal's Report	Links to a representation of the data as displayed in RMF Data Portal's CPC Report

[Analytics] Dashboards - Report CPC - TFP

Provides information on Software Pricing as well as partition related processor activity.

Visualizations	Description
[Analytics] CPC TFP - Analytics - Links	List of different links available in RMF Analytics Space: - Analytics Dashboards (CFACT, CPC Sub-Capacity/TFP) - Spaces (RMF Analytics, RMF Monitoring)
[Analytics] CPC TFP - Total MSU vs CPC Capacity Trend	Tailored Fit Pricing Visualization displaying the consumed MSU & CPC Capacity trends by CPC for all the available data ranges. Limit: CPC=5
[Analytics] CPC TFP - Total MSU	Tailored Fit Pricing Visualization displaying the sum of consumed MSU by CPC during the selected time range for all the available data ranges. Limit: CPC=5
[Analytics] CPC TFP - Total MSU Consumption by Lpar	Tailored Fit Pricing Visualization displaying the sum of consumed MSU by Lpar during the selected time range for all the available data ranges. Limit: Lpar=50.
[Analytics] CPC - Physical CP Utilization % Trend	Visualization displaying the physical CP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *CP)
[Analytics] CPC - Logical CP Utilization % Trend	Visualization displaying the logical CP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50 & Processor Type=1 (include set to *CP).
[Analytics] CPC - Physical zIIP Utilization % Trend	Visualization displaying the physical zIIP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *IIP).
[Analytics] CPC - Logical zIIP Utilization % Trend	Visualization displaying the logical zIIP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *IIP).
[Analytics] CPC - Physical ICF Utilization % Trend	Visualization displaying the physical ICF utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *ICF)
[Analytics] CPC - Logical ICF Utilization % Trend	Visualization displaying the logical ICF utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to ICF).
[Analytics] CPC - Logical IFL Utilization % Trend	Visualization displaying the logical IFL utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to IFL).

Visualizations	Description
[Analytics] CPC - Physical IFL Utilization % Trend	Visualization displaying the physical IFL utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *IFL)
[Analytics] CPC - Logical zAAP Utilization % Trend	Visualization displaying the logical zAAP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to IFL).
[Analytics] CPC - Physical zAAP Utilization % Trend	Visualization displaying the physical zAAP utilization % trend for all the available data ranges. Limits: CPC=5, Lpar=50, Processor Type=1 (include set to *IFL)
[Analytics] CPC - Representation of RMF Data Portal's Report	Links to a representation of the data as displayed in RMF Data Portal's CPC Report.

[Analytics] Dashboard - Reports Overview

Displays information collected through GPMSEVE in tables.

Only access this dashboard from the *Representation of RMF Data Portal's Report* visualization as the interval and metric ID filters will already be set.

Reports Overview	Description
[Analytics] CFACT - Overview	All the information collected from the RMF Report CFACT. For each data range and a view defined by the filter (SYSPLEX), it is possible to see information on the Coupling Facility Activity by Structure.
[Analytics] CPC - Overview	The information collected from the RMF Report CPC. For each data range and a view defined by the filter (CPC, CPU Serial #, SYSPLEX, LPAR, processor type), it is possible to see each metric time range by time range.

Monitoring Space

This section gives an overview of the Monitoring dashboards, KPIs and visualizations supplied with RMF OA.

[Monitoring] Dashboard - CPC SubCapacity

Provides information on Software Pricing as well as partition related processor activity.

Visualizations	Description
[Monitoring] CPC SubCapacity - Links	List of different links available in RMF Monitoring Space: - Monitoring Dashboards (CPC Sub-Capacity/TFP) - Spaces (Analytics, Monitoring)
[Monitoring] CPC SubCapacity - Top CPC by Highest MSU	Sub-Capacity Visualization displaying the list of the 5 CPC with the highest MSU consumption for a data range set to 100s.

Visualizations	Description
[Monitoring] CPC SubCapacity - Top Sysplex by Highest MSU	Sub-Capacity Visualization displaying the list of the 10 Sysplex with the highest MSU consumption for a data range set to 100s.
[Monitoring] CPC SubCapacity - Top Group Capacity by Highest MSU	Sub-Capacity Visualization displaying the list of the 10 Group Capacity with the highest MSU consumption for a data range set to 100s.
[Monitoring] CPC SubCapacity - Top LPAR by Highest MSU	Monitoring Sub-Capacity Visualization displaying the list of the 10 LPAR with the highest MSU consumption for a data range set to 100s.
[Monitoring] CPC SubCapacity - Highest MSU Consumption by LPAR	Monitoring Sub-Capacity Visualization displaying the synthesis of the 10 LPARs with the highest MSU consumption and their metric for a data range set to 100s.
[Monitoring] CPC SubCapacity - MSU Trend by LPAR	Monitoring Sub-Capacity Visualization displaying the consumed MSU by LPAR for a data range set to 100s. Limit: LPAR=50
[Monitoring] CPC - Mainframe Hardware Configuration	Most recent synthesis of the CPC hardware configuration. Limit: CPC=5
[Monitoring] CPC - Processor allocation	List of Shared & Dedicated processors allocated for each LPAR. Limits: CPC=5, Sysplex=10, System ID=50, LPAR=50, Processor Type=5
[Monitoring] CPC - Physical CP utilization %	Visualization displaying the physical CP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=*CP
[Monitoring] CPC - Logical CP Utilization %	Visualization displaying the logical CP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=*CP
[Monitoring] CPC - Physical IFL utilization %	Visualization displaying the physical IFL utilization % trend by LPAR. Limits: LPAR=50, Processor Type=IFL
[Monitoring] CPC - Logical IFL Utilization %	Visualization displaying the logical IFL utilization % trend by LPAR. Limits: LPAR=50, Processor Type=IFL
[Monitoring] CPC - Physical zAAP utilization %	Visualization displaying the physical zAAP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=zAAP
[Monitoring] CPC - Logical zAAP Utilization %	Visualization displaying the logical zAAP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=zAAP
[Monitoring] CPC - Physical zIIP utilization %	Visualization displaying the physical zIIP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=*IIP
[Monitoring] CPC - Logical ICF Utilization %	Visualization displaying the logical ICF utilization % trend by LPAR. Limits: LPAR=50, Processor Type=ICF

Visualizations	Description
[Monitoring] CPC - Physical ICF Utilization %	Visualization displaying the physical ICF utilization % trend by LPAR. Limits: LPAR=50, Processor Type=ICF
[Monitoring] CPC - Logical zIIP Utilization %	Visualization displaying the logical zIIP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=*IIP

[Monitoring] Dashboard - Report CPC - TFP

Provides information on Software Pricing as well as partition related processor activity.

Visualizations	Description
[Monitoring] CPC TFP - Links	List of different links available in RMF Monitoring Space: - Monitoring Dashboards (CPC Sub-Capacity/TFP) - Spaces (Analytics, Monitoring)
[Monitoring] CPC TFP - Top CPC by Total MSU	Tailored Fit Pricing Visualization displaying the list of the top CPC with the highest total of MSU during the selected time range for a data range set to 3600s.
[Monitoring] CPC TFP - Total MSU	Tailored Fit Pricing Visualization displaying the sum of consumed MSU by CPC during the selected time range for a data range set to 3600s. Limit: CPC=5
[Monitoring] CPC TFP - Total MSU by LPAR	Tailored Fit Pricing Visualization displaying the sum of consumed MSU by LPAR during the selected time range for a data range set to 3600s. Limit: LPAR=50
[Monitoring] CPC - Mainframe Hardware Configuration	Most recent synthesis of the CPC hardware configuration. Limit: CPC=5
[Monitoring] CPC - Processor allocation	List of Shared & Dedicated processors allocated for each LPAR. Limits: CPC=5, Sysplex=10, System ID=50, LPAR=50, Processor Type=5
[Monitoring] CPC - Physical CP utilization %	Visualization displaying the physical CP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=*CP
[Monitoring] CPC - Physical zIIP utilization %	Visualization displaying the physical zIIP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=*IIP
[Monitoring] CPC - Logical CP Utilization %	Visualization displaying the logical CP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=*CP
[Monitoring] CPC - Logical zIIP Utilization %	Visualization displaying the logical zIIP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=*IIP
[Monitoring] CPC - Logical ICF Utilization %	Visualization displaying the logical ICF utilization % trend by LPAR. Limits: LPAR=50, Processor Type=ICF

Visualizations	Description
[Monitoring] CPC - Physical ICF utilization %	Visualization displaying the physical ICF utilization % trend by LPAR. Limits: LPAR=50, Processor Type=ICF
[Monitoring] CPC - Logical IFL Utilization %	Visualization displaying the logical IFL utilization % trend by LPAR. Limits: LPAR=50, Processor Type=IFL
[Monitoring] CPC - Physical IFL utilization %	Visualization displaying the physical IFL utilization % trend by LPAR. Limits: LPAR=50, Processor Type=IFL
[Monitoring] CPC - Logical zAAP Utilization %	Visualization displaying the logical zAAP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=zAAP
[Monitoring] CPC - Physical zAAP utilization %	Visualization displaying the physical zAAP utilization % trend by LPAR. Limits: LPAR=50, Processor Type=zAAP

Notices

This information was developed for products and services offered in the U.S.A.

This material may 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

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 the names and addresses used by an actual business enterprise 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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.html>.

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

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

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

Index

C

cookie policy [25](#), [27](#)

L

legal notices

 cookie policy [25](#), [27](#)

 notices [25](#)

 programming interface information [25](#), [26](#)

 trademarks [25](#), [26](#)

N

notices [25](#), [26](#)

P

programming interface information [25](#), [26](#)

T

trademarks [25](#), [26](#)

