## Contents

About this Manual .......................................................................................................................... 5

1 Introduction ................................................................................................................................. 8
  1.1 Basic Usage Scenario ........................................................................................................... 9
  1.2 CMM Architecture and Components .................................................................................. 10
  1.3 User Management .............................................................................................................. 13
  1.4 Distribution Media ............................................................................................................. 13

2 Installation .................................................................................................................................. 14
  2.1 Prerequisites and Preparation ............................................................................................ 15
    2.1.1 Prerequisites .............................................................................................................. 15
    2.1.2 Control Machine ..................................................................................................... 16
    2.1.3 CMM Software and Setup Utilities ........................................................................... 17
    2.1.4 Security ................................................................................................................... 18
    2.1.5 HTTPS Support ....................................................................................................... 19
    2.1.6 Python Packages .................................................................................................... 19
    2.1.7 Bash Script Failures ................................................................................................. 19
  2.2 Configuring the Control Machine ...................................................................................... 20
  2.3 Installing the Offline Resources ......................................................................................... 21
  2.4 Installing the CMM Service ............................................................................................. 22
    2.4.1 Installation .............................................................................................................. 22
    2.4.2 Configuration .......................................................................................................... 27
  2.5 Installing the CMM Metrics Agent .................................................................................... 25
    2.5.1 Installation .............................................................................................................. 25
    2.5.2 Configuration .......................................................................................................... 27
  2.6 Installing the CMM Log Agent ........................................................................................... 29
    2.6.1 Installation .............................................................................................................. 29
    2.6.2 Configuration .......................................................................................................... 32
  2.7 Uninstallation ....................................................................................................................... 34
    2.7.1 Uninstalling the CMM Agents ................................................................................ 34
    2.7.2 Uninstalling the CMM Service and the OpenStack Extensions ................................ 35
    2.7.3 Cleanup .................................................................................................................. 35

3 Operation and Maintenance ...................................................................................................... 40
  3.1 Starting and Stopping Agents and Services ......................................................................... 40
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>Cleaning up Data</td>
<td>43</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Disabling Metrics for a CMM Metrics Agent</td>
<td>43</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Disabling Log Data for a CMM Log Agent</td>
<td>43</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Removing Metrics Data</td>
<td>44</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Removing Log Data</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Log File Handling</td>
<td>47</td>
</tr>
<tr>
<td>3.4</td>
<td>Backup and Recovery</td>
<td>48</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Databases</td>
<td>48</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Configuration Files</td>
<td>52</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Dashboards</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>Monitoring Services and Servers</td>
<td>54</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview of Services and Servers</td>
<td>54</td>
</tr>
<tr>
<td>4.2</td>
<td>Working with Data Visualizations</td>
<td>54</td>
</tr>
<tr>
<td>4.3</td>
<td>Defining Alarms</td>
<td>55</td>
</tr>
<tr>
<td>4.4</td>
<td>Defining Notifications</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>Managing Log Data</td>
<td>57</td>
</tr>
<tr>
<td>5.1</td>
<td>Configuring an Index Pattern</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>Working with the Log Management Dashboard</td>
<td>58</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Configuration Settings</td>
<td>59</td>
</tr>
<tr>
<td>A.1</td>
<td>Configuration Settings for CMM Metrics Agents</td>
<td>59</td>
</tr>
<tr>
<td>A.2</td>
<td>Configuration Settings for CMM Log Agent</td>
<td>60</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Supported Metrics</td>
<td>62</td>
</tr>
<tr>
<td>B.1</td>
<td>Standard Metrics</td>
<td>62</td>
</tr>
<tr>
<td>B.2</td>
<td>Additional Metrics</td>
<td>63</td>
</tr>
<tr>
<td>Glossary</td>
<td></td>
<td>68</td>
</tr>
</tbody>
</table>
About this Manual

This manual describes how CMM operators can install, operate, maintain, and monitor FUJITSU ServerView Cloud Monitoring Manager - hereafter referred to as Cloud Monitoring Manager (CMM).

The manual is structured as follows:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong> on page 8</td>
<td>Introduces CMM, its architecture, and users.</td>
</tr>
<tr>
<td><strong>Installation</strong> on page 14</td>
<td>Describes how to install and configure CMM.</td>
</tr>
<tr>
<td><strong>Operation and Maintenance</strong> on page 40</td>
<td>Describes the main operation and maintenance tasks for CMM.</td>
</tr>
<tr>
<td><strong>Monitoring Services and Servers</strong> on page 54</td>
<td>Describes the basic tasks involved in monitoring services and servers.</td>
</tr>
<tr>
<td><strong>Managing Log Data</strong> on page 57</td>
<td>Describes the basic tasks involved in managing the log data from the services and servers.</td>
</tr>
<tr>
<td><strong>Configuration Settings</strong> on page 59</td>
<td>Describes the configuration settings for the CMM Metrics Agent and the CMM Log Agent.</td>
</tr>
<tr>
<td><strong>Supported Metrics</strong> on page 62</td>
<td>Describes the metrics that are supported by CMM.</td>
</tr>
<tr>
<td><strong>Glossary</strong> on page 68</td>
<td>Defines the central terms relevant for CMM.</td>
</tr>
</tbody>
</table>

Readers of this Manual

This manual is written for operators who install, operate, and maintain CMM, and who monitor the host on which CMM is installed.

The manual assumes that you have profound knowledge of OpenStack and CMM, especially the individual services CMM is composed of. For installing the CMM components, you must be familiar with the administration and operation of LINUX systems.

Notational Conventions

This manual uses the following notational conventions:

<table>
<thead>
<tr>
<th>Add</th>
<th>The names of graphical user interface elements like fields are shown in boldface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>init</td>
<td>System names, for example command names and text that is entered from the keyboard, are shown in Courier font.</td>
</tr>
<tr>
<td>&lt;variable&gt;</td>
<td>Variables for which values must be entered are enclosed in angle brackets.</td>
</tr>
<tr>
<td>[option]</td>
<td>Optional items, for example optional command parameters, are enclosed in square brackets.</td>
</tr>
<tr>
<td>one</td>
<td>Alternative entries are separated by a vertical bar.</td>
</tr>
<tr>
<td>two</td>
<td></td>
</tr>
<tr>
<td>{one</td>
<td>two}</td>
</tr>
</tbody>
</table>
Abbreviations

This manual uses the following abbreviations:

- **CMM**: Cloud Monitoring Manager
- **IaaS**: Infrastructure as a Service
- **ICMP**: Internet Control Message Protocol
- **OS**: Operating System
- **OSS**: Open Source Software
- **PaaS**: Platform as a Service
- **SaaS**: Software as a Service

Available Documentation

The following documentation on CMM is available:

- **Overview**: A manual introducing CMM. It is written for everybody interested in CMM.
- **Tenant User's Guide**: A manual for tenant users describing how CMM supports them in monitoring their services and virtual machines in OpenStack.
- **OpenStack Operator's Guide**: A manual for OpenStack operators describing how CMM supports them in monitoring their OpenStack services as well as managing the log data available from the OpenStack services.
- **CMM Operator's Guide**: A manual for operators describing how to install, operate, and maintain CMM.

Related Web References

The following Web references provide information on open source offerings integrated with CMM:

- **OpenStack**: Documentation on OpenStack, the underlying platform technology.
- **OpenStack Horizon**: Documentation on the OpenStack Horizon dashboard.
- **Monasca Wiki**: Information on Monasca, the core of CMM.

More detailed Web references provided in this manual are subject to change without notice.

Trademarks

LINUX is a registered trademark of Linus Torvalds.

The OpenStack® Word Mark and OpenStack logo are registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation in the United States and other countries and are used with the OpenStack Foundation's permission. FUJITSU LIMITED is not endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

Red Hat is a trademark or a registered trademark of Red Hat Inc. in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

Python and PyCon are trademarks or registered trademarks of the Python Software Foundation.

ServerView and Systemwalker are registered trademarks of FUJITSU LIMITED.
Other company names and product names are trademarks or registered trademarks of their respective owners.

**Copyright**

Copyright FUJITSU LIMITED 2015 - 2016

All rights reserved, including those of translation into other languages. No part of this manual may be reproduced in any form whatsoever without the written permission of FUJITSU LIMITED.

**Export Restrictions**

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.
1 Introduction

As more and more applications are deployed on cloud systems and cloud systems are growing in complexity, managing the cloud infrastructure is becoming increasingly difficult. Cloud Monitoring Manager (CMM) helps mastering this challenge by providing a sophisticated Monitoring as a Service solution that is operated on top of OpenStack-based cloud computing platforms. The component architecture of OpenStack provides for high flexibility, yet it increases the burden of system operation because multiple services must be handled. CMM offers an integrated view of all services and assembles and presents related metrics and log data in one convenient access point. While being flexible and scalable to instantly reflect changes in the OpenStack platform, CMM provides the ways and means required to ensure multi-tenancy, high availability, and data security.

CMM covers all aspects of a Monitoring as a Service solution:

• Central management of monitoring data from medium and large-size OpenStack deployments.
• Storage of monitoring data in a resilient way.
• Horizontal and vertical scalability to support constantly evolving cloud infrastructures. When physical and virtual servers are scaled up or down to varying loads, the monitoring and log management solution can be adapted accordingly.
1.1 Basic Usage Scenario

The basic usage scenario of setting up and using the monitoring services of CMM looks as follows:

A tenant user acts as a service provider in the OpenStack environment. He books virtual machines to provide services to end users or to host services that he needs for his own development activities. CMM helps tenant users ensure that their services and the servers on which they are provided are configured and working as required.

The OpenStack operator is a special tenant user who is responsible for administrating and maintaining the underlying OpenStack platform. The monitoring and log management services of CMM enable him to ensure the availability and quality of the platform. He uses CMM for:

- Monitoring physical and virtual servers, hypervisors, and OpenStack services.
- Monitoring middleware components, for example, database services.
- Retrieving and analyzing the log data of the OpenStack services and servers, the middleware components, and the operating system.

As the CMM operator, you are responsible for providing the monitoring and log management services to the tenant users and the OpenStack operator. This enables the tenant users and the OpenStack operator to focus on operation and the quality of their services and servers without having to carry out the tedious tasks implied by setting up and administrating their own monitoring software. You use the monitoring services yourself for ensuring the quality of CMM.
The CMM Operator's Tasks
As the CMM operator, you have the following responsibilities:

- Installation and setup of the CMM Service, thus providing the monitoring services to the tenant users, and the monitoring and log management services to the OpenStack operator.
- Installation of the CMM Metrics Agent and Log Agent on the CMM host. The agents are required for monitoring the CMM Service.
- Regular maintenance of the components and services CMM consists of.
- Backup of the CMM databases, configuration files, and customized dashboards.
- Monitoring of the CMM Service to ensure the CMM quality. For monitoring, you use a graphical user interface that is seamlessly integrated into the cloud infrastructure. Based on OpenStack Horizon, the user interface visualizes the health and status of your cloud resources and enables user access to all monitoring and log management functionality.

1.2 CMM Architecture and Components
The following illustration provides an overview of the main components of CMM and their interaction:

OpenStack
CMM relies on OpenStack as technology for building cloud computing platforms for public and private clouds. OpenStack consists of a series of interrelated projects delivering various components for a cloud infrastructure solution and allowing for the deployment and management of Infrastructure as a Service (IaaS) platforms.
CMM Service

The CMM Service is the central CMM component. It is responsible for receiving, persisting, and processing monitoring and log data, as well as providing the data to the users.

The CMM Service relies on Monasca, an open source Monitoring as a Service solution. It uses Monasca for high-speed metrics querying and integrates the Threshold Engine (streaming alarm engine) and the Notification Engine of Monasca.

The CMM Service is installed on a single host. It consists of the following components:

- **Monitoring API**
  A RESTful API for monitoring. It is primarily focused on the following areas:
  - Metrics: Store and query massive amounts of metrics in real-time.
  - Statistics: Provide statistics for metrics.
  - Alarm Definitions: Create, update, query, and delete alarm definitions.
  - Alarms: Query and delete the alarm history.
  - Notification Methods: Create and delete notification methods and associate them with alarms. Users can be notified directly when alarms are triggered, for example, via email.

- **Message Queue**
  A component that primarily receives published metrics from the Monitoring API, alarm state transition messages from the Threshold Engine, and log data from the Log API. The data is consumed by other components, such as the Persister, the Notification Engine, and the Log Persister. The Message Queue is also used to publish and consume other events in the system. It is based on Kafka, a high-performance, distributed, fault-tolerant, and scalable message queue with durability built-in. For administrating the Message Queue, CMM uses Zookeeper, a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.

- **Persistor**
  A Monasca component that consumes metrics and alarm state transitions from the Message Queue and stores them in the Metrics and Alarms Database (InfluxDB).

- **Notification Engine**
  A Monasca component that consumes alarm state transition messages from the Message Queue and sends notifications for alarms, such as emails.

- **Threshold Engine**
  A Monasca component that computes thresholds on metrics and publishes alarms to the Message Queue when they are triggered. The Threshold Engine is based on Apache Storm, a free and open distributed real-time computation system.

- **Metrics and Alarms Database**
  An InfluxDB database used for storing metrics and the alarm history.

- **Config Database**
  A MariaDB database used for storing configuration information, alarm definitions, and notification methods.

- **Log API**
  A RESTful API for log management. The Log API gathers log data from the CMM Log Agents and forwards it to the Message Queue.
The CMM log management is based on Logstash, a tool for receiving, processing, and publishing all kinds of logs. It provides a powerful pipeline for querying and analyzing logs. Elasticsearch is used as the backend datastore, and Kibana as the front-end tool for retrieving and visualizing the log data.

- **Log Transformer**
  A Logstash component that consumes the log data from the Message Queue, performs transformation and aggregation operations on the data, and publishes the data that it creates back to the Message Queue.

- **Log Persister**
  A Logstash component that consumes the transformed and aggregated log data from the Message Queue and stores them in the Log Database (Elasticsearch).

- **Kibana Server**
  A Web browser-based analytics and search interface to Elasticsearch.

- **Log Database**
  An Elasticsearch database for storing the log data.

**Note:** The installation of the CMM Service includes the installation of all CMM third-party components that are required. From time to time, it may be necessary to install bug fixes or security patches for these third-party components. In order to guarantee for the interoperability and integrity of your entire CMM installation, you should obtain such fixes and patches solely from your CMM support organization.

**CMM Horizon Plugin**
CMM comes with a plugin for the OpenStack Horizon dashboard. The plugin extends the main dashboard in OpenStack with a view for monitoring. This enables CMM users to access the monitoring functions from a central Web-based graphical user interface. For details, refer to the OpenStack Horizon documentation.

Based on OpenStack Horizon, the monitoring data is visualized on a comfortable and easy-to-use dashboard which fully integrates with the following applications:
- Grafana (for metrics data). An open source application for visualizing large-scale measurement data.
- Kibana (for log data). An open source analytics and visualization platform designed to work with Elasticsearch.

**CMM Metrics Agent**
A CMM Metrics Agent is required for gathering metrics and sending them to the CMM Service. The agent supports metrics from a variety of sources as well as a number of built-in system and service checks. A CMM Metrics Agent can be installed on each virtual or physical server to be monitored.

The agent functionality is fully integrated into the source code base of the Monasca project. For details, refer to the Monasca Wiki.
CMM Log Agent
A CMM Log Agent is needed for collecting log data and forwarding it to the CMM Service for further processing. It can be installed on each virtual or physical server whose log data is to be retrieved.
The agent functionality is fully integrated into the source code base of the Monasca project. For details, refer to the Monasca Wiki.

1.3 User Management
CMM is fully integrated with Keystone, the identity service which serves as the common authentication and authorization system in OpenStack.
The CMM integration with Keystone requires any CMM user, including the CMM operator, to be registered as an OpenStack user. All authentication and authorization in CMM is done through Keystone. If a user requests monitoring data, for example, CMM verifies that the user is a valid user in OpenStack and allowed to access the requested metrics.
CMM users are created and administrated in OpenStack:
• Each user assumes a role in OpenStack to perform a specific set of operations. The OpenStack role specifies a set of rights and privileges.
• Each user is assigned to at least one project in OpenStack. A project is an organizational unit that defines a set of resources which can be accessed by the assigned users.
  Tenant users in CMM can monitor the set of resources that is defined for the projects to which they are assigned.
For details on user management, refer to the OpenStack documentation.

1.4 Distribution Media
CMM is distributed in the CMM installation package. It contains the CMM software and documentation for installing and configuring CMM.
The package includes:
• setup.sh script for setting up the CMM installer.
• Licenses subdirectory:
  License files of third-party software used by CMM.
• Service subdirectory:
  Installation package for CMM, CMM-Service-1.1.0.tar.gz. It provides the software and setup utilities required by the CMM Operator and the OpenStack Operator for installing CMM. For details, refer to Prerequisites and Preparation on page 15.
  FJSVsvcmm-1.1.0-1.noarch.rpm, an RPM required for checking version information.
• Manuals subdirectory:
  PDF manuals providing an overview of CMM, as well as information directed to the tenant user, the OpenStack operator, and the CMM operator.
2 Installation

The installation of CMM comprises several steps. They are partially executed by the CMM operator, and partially by the OpenStack operator.

The installation of CMM comprises the following steps:
1. Preparing the installation environment - done by the CMM operator and the OpenStack operator.
2. Configuring the Control Machine - done by the CMM operator and the OpenStack operator.
3. Installing the offline resources - done by the CMM operator only.
4. Installing the OpenStack extensions required for the CMM Service - done by the OpenStack operator only.
5. Installing the CMM Service - done by the CMM operator only.

As a CMM operator, you also install and configure the agents for monitoring CMM. The OpenStack operator installs and configures the agents for monitoring the OpenStack platform. He also prepares the monitoring environment for the tenant users who have booked a virtual machine in OpenStack.
CMM ships with an installer that is based on Ansible. You use it for installing the required software components.

**Installation Environment**

The following installation environment is the default installation environment:

- The CMM Operator installs the offline resources, the CMM components, and the required agents on one host (CMM host).
- The OpenStack Operator installs the OpenStack extensions and the required agents for the OpenStack Keystone service and the OpenStack Horizon service. Depending on the OpenStack environment, Keystone and Horizon can either be installed on the same or on two separate hosts. The OpenStack Operator can install additional agents for monitoring additional OpenStack services as required.

The installation of the CMM components is managed from a so-called Control Machine. You have to prepare a Control Machine for installing the CMM host. The OpenStack Operator prepares a Control Machine for installing the OpenStack extensions on the Keystone host and the Horizon host.

### 2.1 Prerequisites and Preparation

The following sections describe the prerequisites that must be fulfilled and the preparations you need to take before installing CMM.

#### 2.1.1 Prerequisites

CMM can be installed on a host machine with the following operating systems:
- Red Hat Enterprise Linux 7.2 (for Intel64)

As underlying platform technology, the following OpenStack platforms are supported:
- Red Hat Enterprise Linux OpenStack Platform 8
The following hardware resources are recommended:

<table>
<thead>
<tr>
<th>Hardware type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Fujitsu Server PRIMERGY RX300 S8</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel® Xeon® E5-2660 v2, 2.20 GHz or more</td>
</tr>
<tr>
<td>Memory</td>
<td>64 GB DIMM (DDR3) or more</td>
</tr>
</tbody>
</table>

CMM without any data requires about 2 GB of disk space. The disk capacity required for log data and metrics data varies considerably depending on the number of services and servers to be monitored.

Make sure that enough disk space is provided. If required, mount partitions for storing the log data and the metrics data. The default installation uses the following directories:
- The data directory of Elasticsearch is located in `/opt/elasticsearch/data`.
- The data directory of InfluxDB is located in `/var/opt/influxdb/data`.
- The data directory of MariaDB is located in `/var/lib/mysql`.
- The data directory of Kafka is located in `/var/kafka`.

CMM supports the following Web browsers:
- Google Chrome 50.0
- Microsoft Internet Explorer 11.0 (with Compatibility View disabled)
- Mozilla Firefox ESR 45.0

### 2.1.2 Control Machine

A so-called Control Machine is used for installing CMM. The Control Machine is a separate machine from which any number of remote machines can be managed on which software components are to be installed.

The remote machines are managed via SSH.

The Control Machine must fulfill the following system requirements:
- Ansible 1.9 must be installed and prepared as described below. Ansible is a simple IT automation platform that makes applications and systems easier to deploy. The automated installation of CMM is based on Ansible playbooks, and has been tested with Ansible 1.9.1.
- Python 2.7 must be installed.
For details on the operating system required for the Control Machine, refer to the *Ansible documentation*.

To install and prepare Ansible on the Control Machine, proceed as follows:

1. Log in to the Control Machine.

2. Enable the following repositories for Extra Packages for Enterprise Linux (EPEL):
   - *optional repository* (rhel-7-server-optional-rpms)
   - *extras repository* (rhel-7-server-extras-rpms)

   To enable the repositories, you have to activate the optional subchannel for Red Hat Network Classic. For details on certificate-based subscriptions, refer to the *Red Hat Subscription Management Guide*.

   To activate the subchannel, execute the following commands:

   ```
   subscription-manager repos --enable rhel-7-server-optional-rpms
   subscription-manager repos --enable rhel-7-server-extras-rpms
   ```

3. Install EPEL.

   To download EPEL for Red Hat Enterprise Linux 7.2, execute the following command:

   ```
   ```

   To install EPEL, execute the following command:

   ```
   sudo rpm -i epel-release-latest-7.noarch.rpm
   ```

4. To install Ansible, execute the following command:

   ```
   sudo yum install ansible1.9
   ```

### 2.1.3 CMM Software and Setup Utilities

The software and setup utilities are distributed in the CMM installation package on the CMM DVD. To set up the installation package, you have to mount the DVD and execute the `setup.sh` script that is provided. It extracts the utilities required for installing CMM.

Proceed as follows:

1. Log in to the Control Machine.

2. Mount the DVD to a directory of your choice. It is recommended that you explicitly specify the ISO9660 file system.

   Execute the following command:

   ```
   # mount -t iso9660 -r /dev/<file_name> <mount_dir>
   ```

   Replace `<file_name>` by the file name of the device and `<mount_dir>` by the directory to which you want to mount the DVD.

3. To run the `setup.sh` script, execute the following command:

   ```
   # <mount_dir>/setup.sh
   ```

   Replace `<mount_dir>` by the directory to which you have mounted the DVD.
The content of the installation package is extracted to the `/opt/FJSVsvcmm` directory on the Control Machine. After extraction, the following files and directories are available:

- **group_vars**
  Directory with the Ansible group variables required for configuring the Control Machine from which CMM is installed.

- **host_vars**
  Directory with the Ansible host variables required for configuring the Control Machine from which CMM is installed.

- **roles and tasks**
  Directories with artifacts required by the CMM installer.

- **templates**
  Directory with sample Ansible configuration files. They can be used for configuring the Control Machine from which CMM is installed.

- **uninstaller**
  Directory with artifacts required by the CMM uninstaller.

- ***.yml files**
  Ansible playbooks required for installing and uninstalling the CMM software components.

- **services.sh**
  Script for starting, stopping, and viewing the status of the CMM agents and services.

- ***.txt files**
  CMM11_PythonRequirements.txt lists the Python libraries required for the CMM Service, OSP8_PythonRequirements.txt lists the Python libraries required for the OpenStack extensions. These libraries are installed to the `/usr/lib/python2.7/site-packages` directory.

**Note:** As a prerequisite for the CMM Service and the OpenStack extensions, specific versions of the Python libraries must be installed. Make sure that the installed versions are not overwritten in the `/usr/lib/python2.7/site-packages` directory.

CMM11_RPMPackages.txt lists the RPM (RedHat Repository Manager) packages required for installing the CMM components, OSP8_RPMPackages.txt lists the RPM packages required for installing the OpenStack extensions.

**Note:** The CMM installer automatically installs any missing RPM packages from your YUM repository server. You have to ensure that the required packages are available in your YUM repository.

### 2.1.4 Security

In a default Red Hat Enterprise Linux installation, the following security precautions are taken:

- A firewall exists to prevent unauthorized user access.
- Security-Enhanced Linux (SELinux), a security module that adds mandatory access control mechanisms to the Linux kernel, is enabled.
Before installing the CMM components, make sure that the firewall is enabled and SELinux is running. Make sure that neither the firewall nor SELinux block the communication between the Control Machine and the remote machines.

The Ansible installer you use to install the CMM components opens the following ports:

- Port 8888 for the offline resources required for the installation.
- Port 5607 for the Log API.
- Port 8070 for the Monitoring API.
- Port 5601 for the Kibana Server.

Port 8081 is required internally by the Monitoring API, for example, for healthchecks or threads. The DropWizard framework that is used by the Ansible installer enables external access to this port.

CMM uses the following ports to integrate with the required OpenStack services:

- Port 80 for the OpenStack Horizon service.
- Port 5000 and 35357 for the OpenStack Keystone service.

If it is necessary to manually enable access to the ports, you can execute the following commands. Repeat them for each port to which access must be enabled.

```
firewall-cmd --add-port=<port_number>/tcp
firewall-cmd --permanent --add-port=<port_number>/tcp
```

Replace `<port_number>` with the port.

### 2.1.5 HTTPS Support

By default, the CMM components use the HTTP protocol for communication. For a more secure connection, CMM can be configured to use the HTTPS protocol. If you want to use the HTTPS protocol, contact your FUJITSU support organization for information.

### 2.1.6 Python Packages

CMM uses pip for installing Python packages. When installing CMM software components, pip V7.0.3 is automatically installed.

**Note:** Check whether pip is already installed on the host machine on which the CMM components are to be installed. The installation fails if a pip version is installed that is different from V7.0.3.

### 2.1.7 Bash Script Failures

When executing the `sudo` command in bash scripts during installation, services may fail to start properly. This may be caused by the `tty` specification in the `/etc/sudoers` file.

To prevent failures, proceed as follows:

1. Open the `sudoers` file with your favorite editor.
   
   **Example:**
   
   ```
   sudo visudo
   ```
2. Make sure that the following line is commented by adding a # at the beginning of the line:

```
#Defaults requiretty
```

### 2.2 Configuring the Control Machine

The Control Machine must be connected to all remote machines on which CMM components are to be installed. To configure the Control Machine, proceed as follows:

1. Log in to the Control Machine.
2. Open the Ansible configuration file with your favorite editor.
   
   Example:

   ```
   sudo vim /etc/ansible/ansible.cfg
   ```

3. Specify the following SSH settings in the file:

   ```
   [defaults]
   transport = ssh
   
   [ssh_connection]
   ssh_args = -o ControlMaster=auto -o BatchMode=yes -o ForwardAgent=yes
   ```

4. Open the configuration file of the Control Machine with your favorite editor.
   
   Example:

   ```
   sudo vim /etc/ansible/hosts
   ```

   For detailed information on the configuration of the Control Machine, refer to the Ansible Inventory documentation.

5. Configure the required hosts in the configuration file. CMM ships with a sample configuration file that you can use as a template:

   ```
   hosts-single-cmm.j2
   ```

   The sample is located in the `/opt/FJSVsvcmm/templates` directory.

   Copy the complete content of the sample file to your configuration file.

6. Replace the variables in every section that is defined:

   - **Offline Host** for installing the offline resources required for the installation.
   - **Monasca Host** for installing the CMM Service.
   - `<cmm_component>` for installing the individual CMM components.
   - **Monasca Agent Hosts** for installing a CMM Metrics Agent on the CMM host.
   - **Monasca Log Agent Hosts** for installing a CMM Log Agent on the CMM host.

   Replace `{{ monasca_host }}` with the host name or IP address of the remote machine on which you want to install the corresponding CMM component. Replace `{{ ssh_user_monasca }}` with the user name of the user who is to install it.

   **Note:** The user who is to install the CMM component must have the privilege to execute sudo commands without password. Check your sudo configuration for this purpose.
7. To establish the connection between the Control Machine and the remote machine, generate the SSH key for Ansible.
   Execute the following command:
   
   ```bash
   ssh-keygen -t rsa
   ```

   **Note:** If you have specified a passphrase for the SSH key, execute any install or uninstall command with the `ssh-agent` tool.
   Make sure to stop the `ssh-agent` tool as soon as the installation or uninstallation is successful.

8. To copy the SSH key to the CMM host, execute the following command:

   ```bash
   ssh-copy-id <user_name>@<cmm_host_ip>
   ```

   Replace `user_name` by the user name to be used for accessing the host, and replace `cmm_host_ip` by the host name or IP address of the host.

   Example response for a successful connection to the CMM host:
   
   ```
   /usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
   to filter out any that are already installed
   /usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you
   are prompted now it is to install the new keys
   <user_name>@<cmm_host_ip>''s password:
   Number of key(s) added: 1
   Now try logging into the machine, with:   "ssh
   '
   '<user_name>@<cmm_host_ip>''
   and check to make sure that only the key(s) you wanted were added.
   ```

### 2.3 Installing the Offline Resources

To install the offline resources that are required for installing the OpenStack extensions and the CMM Service, proceed as follows:

1. Log in to the Control Machine.

2. Configure the remote machines on which to install the offline resources, the CMM Service, and the OpenStack Keystone service. The installer uses Ansible group variables for configuration purposes. The group variables are located in the `/opt/FJSVsvcmm/group_vars` directory.

   Open the `all_group` file with your favorite editor. It defines the main configuration settings.

   Example:
   
   ```bash
   sudo vim /opt/FJSVsvcmm/group_vars/all_group
   ```

3. Specify the IP addresses of your environment in the `hosts` section:
   - `offline_host` for the offline resources
   - `monasca_host` for the CMM Service
   - `keystone_host` for the OpenStack Keystone service
4. To install the offline resources, specific configuration settings are required. They are specified in the `group_vars/offline_group` file.
The file contains the default configuration. No changes are required.

**Note:** Make changes only if you have to deviate from the default.

To view the default configuration, you can open the file with your favorite editor.
Example:

```
sudo vim /opt/FJSVsvcmm/group_vars/offline_group
```

5. To install the offline resources, execute the following command:

```
ansible-playbook /opt/FJSVsvcmm/offline.yml
```

The following response is displayed if the installation is successful.

Example:

```
PLAY RECAP ****************************************************
offline-resources-host : ok=40  changed=23  unreachable=0  failed=0
```

**Note:** There are checks performed during installation that may result in failures or warnings.
These failures and warnings are reported but they do not block the installation. It is only this final message that is relevant for a successful installation.

In case the installation fails, retry it in verbose mode. To collect debug information, you can execute the following command:

```
ansible-playbook -vvvv /opt/FJSVsvcmm/offline.yml
```

As soon as the offline resources are successfully installed, the OpenStack operator must prepare the OpenStack platform for CMM. Refer to the `OpenStack Operator's Guide` for details on the tasks of the OpenStack operator. He has to install the required OpenStack extensions and has to create the users and roles in OpenStack required for installing and working with CMM.
Before proceeding with the installation of the CMM Service, contact your OpenStack operator and make the corresponding arrangements.

### 2.4 Installing the CMM Service

As a prerequisite for installing the CMM Service, the OpenStack operator must have installed the OpenStack extensions for CMM.

To install the CMM Service, proceed as follows:

1. Log in to the Control Machine.
2. Configure the remote machine on which to install the CMM Service. The installer uses Ansible group variables for configuration purposes. The group variables are located in the `/opt/FJSVsvcmm/group_vars` directory.

Open the `all_group` file with your favorite editor. It defines the main configuration settings.

Example:

```
sudo vim /opt/FJSVsvcmm/group_vars/all_group
```

3. Check the IP addresses of your environment in the `hosts` section. They have already been specified for the installation of the offline resources:

- `offline_host` for the offline resources
- `monasca_host` for the CMM Service
- `keystone_host` for the OpenStack Keystone service

Example:

```
# hosts
offline_host: 192.168.10.4
monasca_host: 192.168.10.4
keystone_host: 192.168.10.5
```

4. Enter the user credentials of a valid OpenStack user in the `Monasca Api` section. Contact your OpenStack Operator for this purpose. He is responsible for creating the required users and roles.

The user you enter must have full access to the CMM Service. He must be assigned a role that authorizes him to read and write data. The role must be specified for `defaultAuthorizedRoles` (see Step 8 below).

Example:

```
# Monasca Api CLI - Credentials
monasca_user:
  user: cmm-operator
  password: "{{ keystone_cmm_operator_user_password }}"
  project: cmm
```

5. Provide the passwords required for installing the CMM Service. They are defined in the `credentials.yml` file that is located in the `/opt/FJSVsvcmm` directory.

Open the `credentials.yml` file with your favorite editor.

Example:

```
sudo vim /opt/FJSVsvcmm/credentials.yml
```

6. Specify the following passwords in the file:

- `influxdb_mon_api_password`: for the Monitoring API user accessing the InfluxDB.
- `influxdb_mon_persister_password`: for the Persister user accessing the InfluxDB.
- `database_notification_password`: for the Notification Engine user accessing the MariaDB database.
- `database_monapi_password`: for the Monitoring API user accessing the MariaDB database.
2: Installation

• database_thresh_password: for the Threshold Engine user accessing the MariaDB database.
• database_logapi_password: for the Log API user accessing the MariaDB database.
• keystone_cmm_operator_user_password: for Keystone access of the CMM operator (see Step 4 above). The user account for the CMM operator is automatically created with the installation.
• keystone_admin_password of an OpenStack user account that already exists, for example, admin. This user account is used for creating new user accounts with the installation. It is the user account of the OpenStack operator.

7. To install the CMM components, specific configuration settings are required. They are specified in the group_vars/monasca_group file.
   The file contains the default configuration. No changes are required for the individual components. You have to specify only the authorized OpenStack roles (see Step 8 below) and a file system for backing up the Elasticsearch database (see Step 9 below).

**Note:** Make changes for the individual components only if you have to deviate from the default.

To view the default configuration, you can open the file with your favorite editor.
Example:

```
 sudo vim /opt/FJSVsvcmm/group_vars/monasca_group
```

8. Specify the OpenStack roles that have been authorized by the OpenStack operator in the group_vars/monasca_group file:
   • defaultAuthorizedRoles specifies the roles which grant the users full access.
   • agentAuthorizedRoles specifies the roles which grant agent access only. The users' access is restricted to sending monitoring data.

Example:

```
defaultAuthorizedRoles: "user,domainuser,domainadmin,cmm-user"
agentAuthorizedRoles: "cmm-agent"
```

9. Specify a file system for backing up the Elasticsearch database in the elasticsearch section of the group_vars/monasca_group file. The file system you specify must later be used for backing up your Elasticsearch database.

Example:

```
elasticsearch_repo_dir: [/mount/backup/elasticsearch]
```

10. To install the CMM components, execute the following command:

    `ansible-playbook /opt/FJSVsvcmm/monasca.yml`

The following response is displayed if the installation is successful.

Example:

```
PLAY RECAP ********************************************************************
cmm-node : ok=262 changed=189 unreachable=0 failed=0
```
2: Installation

Note: There are checks performed during installation that may result in failures or warnings. These failures and warnings are reported but they do not block the installation. It is only this final message that is relevant for a successful installation.

In case the installation fails, check your configuration settings and passwords and retry the installation in verbose mode. To collect debug information, you can execute the following command:

```
ansible-playbook -vvvv /opt/FJSVsvcmm/monasca.yml
```

As soon as the CMM Service is successfully installed, you can proceed with installing and configuring the agents for monitoring CMM.

2.5 Installing the CMM Metrics Agent

For monitoring CMM, a CMM Metrics Agent must be installed on your CMM host.

As a prerequisite for installing the agent on your CMM host, the offline resources, the OpenStack extensions, and the CMM Service must have been installed.

The installation consists of the following steps:

1. Installing the agent.
2. Configuring the agent.

2.5.1 Installation

To install a CMM Metrics Agent, proceed as follows:

1. Log in to the Control Machine.
2. Open the configuration file of the Control Machine with your favorite editor.
   
   Example:

   ```
sudo vim /etc/ansible/hosts
   ```

3. Configure the host on which you want to install the agent. For this purpose, check the following section in the configuration file:

   ```
   # Monasca Agent Hosts
   [monasca_agent_group]
cmm-monasca-agent ansible_ssh_host=<monasca_host>
   ansible_ssh_user=<ssh_user_monasca>
   ```

   If required, you can add a line for installing an additional agent on a different host. The installer allows you to install one or multiple instances of an agent at a time. When adding a line for an additional agent, you have to specify a unique alias for the host machine on which to install the agent.

4. Replace `<monasca_host>` with the host name or IP address of the remote machine on which to install the agent. Replace `<ssh_user_monasca>` with the user name of the user who is to install the agent.

Note: The user who installs the agent must have access to the machine on which the agent is to be installed. Access as a user with root privileges is required.
5. To establish the connection between the Control Machine and the remote machines, generate the SSH key for Ansible.
   Execute the following command:

   ```
   ssh-keygen -t rsa
   ```

6. To copy the SSH key to a remote machine, execute the following command. Repeat the command for each remote machine on which to install an agent.

   ```
   ssh-copy-id <user_name>@<host_ip_address>
   ```

   Replace `<user_name>` by the user name to be used for accessing the remote machine, and replace `<host_ip_address>` by the host name or IP address of the remote machine.

   Example response for a successful connection to a remote machine:

   ```
   /usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you
are prompted now it is to install the new keys
<user_name>@<host_ip_address>''s password:
Number of key(s) added: 1
Now try logging into the machine, with:   "ssh
''<user_name>@<host_ip_address>''"
and check to make sure that only the key(s) you wanted were added.
```

7. Create a configuration file for the agent. It must be stored in the installation directory. An individual configuration file is required for each agent instance that is to be installed.

   Execute the following command to create a configuration file. Repeat the command for all host machines on which to install an agent.

   ```
   sudo vim /opt/FJSVsvcmm/host_vars/<file_name>
   ```

   Replace `<file_name>` with the alias for the host machine. `<file_name>` must be identical to the alias specified in the configuration file of the Control Machine (see Step 3 above).

8. Configure the agents. The following example shows the minimum set of parameters that must be configured to start an agent. The minimum set is restricted to the credentials of an OpenStack user used for the communication between the CMM Service and the agent.

   You can specify additional configuration settings, if required. For a list of the configuration settings that are supported, refer to [Configuration Settings for CMM Metrics Agents](#) on page 59.

   To define the minimum set of parameters, insert the following content into each configuration file. Specify the user credentials of a valid OpenStack user who is authorized to send monitoring data to the CMM Service. Contact your OpenStack Operator. He is responsible for creating the required users and roles.

   Example:

   ```
   # file: host_vars/cmm-monasca-agent
   
   # Monasca Agent user credentials
   monasca_agent_user: cmm-agent
   monasca_agent_password: "{{ cmm_monasca_agent_keystone_password }}"
   ```
9. Provide the passwords required for installing an agent. They are defined in the credentials.yml file that is located in the /opt/FJSVsvcmm directory.
Open the credentials.yml file with your favorite editor.
Example:

```
sudo vim /opt/FJSVsvcmm/credentials.yml
```

10. Specify the following passwords in the file:
- `cmm_monasca_agent_keystone_password` for Keystone access of the user used for configuration purposes of the CMM Metrics Agent on the CMM host. The user account is automatically created with the installation.
- `cmm_monasca_agent_database_password` for agent access to the Maria DB database.
- `openstack_monasca_agent_keystone_password` for Keystone access of the user used for configuration purposes of the CMM Metrics Agent on the OpenStack host. The user account is automatically created with the installation.

11. To run the installer, execute the following command:

```
ansible-playbook /opt/FJSVsvcmm/monasca-agent.yml
```

The following response is displayed if the installation is successful.
Example:

```
PLAY RECAP ********************************************************
monasca-agent : ok=40    changed=23    unreachable=0    failed=0
```

**Note:** There are checks performed during installation that may result in failures or warnings. These failures and warnings are reported but they do not block the installation. It is only this final message that is relevant for a successful installation.

In case the installation fails, check your configuration settings and passwords and retry the installation in verbose mode. To collect debug information, you can execute the following command:

```
ansible-playbook -vvvv /opt/FJSVsvcmm/monasca-agent.yml
```

The agents are installed and automatically started after a successful installation. The installer creates a startup script that automatically starts the agent when the machine is booted.
A CMM Metrics Agent is installed in a virtualenv environment. By default, the virtualenv environment of the CMM Metrics Agent is located in the `/opt/monasca-agent` directory.
An agent is provided as a LINUX service.

### 2.5.2 Configuration

The agent installation creates the configuration files required for monitoring the services on the server where the agent is installed. The configuration files are in .yaml format. The following configuration files are created:
- `agent.yaml` located in the `/etc/monasca/agent` directory.
It defines the basic agent configuration. The installer auto-detects applications and processes that are running on your machine and saves the corresponding settings to this file.

- *.yaml files located in the `/etc/monasca/agent/conf.d` directory. They provide the standard metrics the agent uses for monitoring your services and servers. Standard metrics for system checks are automatically configured by the installer.

- *.yaml files located in the `/opt/monasca-agent/share/monasca/agent/conf.d` directory. They provide additional metrics that you can activate as enhancement to the standard metrics. These files provide template configurations that you can adapt to your environment.

The installation of an agent includes its initial configuration. You have the following options for reconfiguring an agent:

- You can update the `agent.yaml` file.
- You can activate metrics in addition to the standard metrics.

Refer to the following sections for details.

**Updating the Configuration File**

To edit the `agent.yaml` file, proceed as follows:

1. Log in to the server where the agent is installed.
2. To stop the agent, execute the following command:

   ```bash
   sudo systemctl stop monasca-agent
   ```

3. Open the file with your favorite editor. Example:

   ```bash
   sudo vim /etc/monasca/agent/agent.yaml
   ```

4. Adapt the configuration settings as required. For a list of the parameters that are supported by CMM, refer to `Configuration Settings for CMM Metrics Agents` on page 59.

5. To start the agent again, execute the following command:

   ```bash
   sudo systemctl start monasca-agent
   ```

   The agent is instantly available with the updated configuration settings.

**Activating Additional Metrics**

The installer automatically activates standard metrics for monitoring your services and servers. They include system checks, for example, on CPU usage, disk space, or the average system load. No manual configuration is required for these system checks. The installer automatically configures them to retrieve monitoring data from the server where your agent is installed. For details on these metrics, refer to `Standard Metrics` on page 62.

As enhancement to the standard metrics, CMM allows you to activate additional metrics. For a list of the metrics that are supported by CMM, refer to `Additional Metrics` on page 62.

The agent ships with metrics templates that you can adapt to your environment and use for monitoring your services and servers.
To activate additional metrics, proceed as follows:
1. Log in to the server where the agent is installed.
2. To stop the agent, execute the following command:

   ```bash
   sudo systemctl stop monasca-agent
   ```

3. Copy the required template file.
   Example:

   ```bash
   sudo cp -p
   /opt/monasca-agent/share/monasca/agent/conf.d/rabbitmq.yaml.example
   /etc/monasca/agent/conf.d/rabbitmq.yaml
   ```

4. Open the template file with your favorite editor.
   Example:

   ```bash
   sudo vim /etc/monasca/agent/conf.d/rabbitmq.yaml
   ```

5. Adapt the configuration to your environment. For configuration examples, refer to Additional Metrics on page 62.

6. To start the agent again, execute the following command:

   ```bash
   sudo systemctl start monasca-agent
   ```

The activated metrics can instantly be used by the agent for retrieving monitoring data.

**Note:** The installer auto-detects applications and processes that are running on your system. It saves the detected settings to the corresponding configuration files (e.g. `host_alive.yaml`, `http_check.yaml`, or `process.yaml`). These files are automatically provided in the `/etc/monasca/agent/conf.d/` directory. You must complete their configuration.

### 2.6 Installing the CMM Log Agent

For monitoring CMM, a CMM Log Agent must be installed on your CMM host.

As a prerequisite for installing the agent, the offline resources, the OpenStack extensions, and the CMM Service must have been installed.

The installation consists of the following steps:

1. Installing the agent.
2. Configuring the agent.

#### 2.6.1 Installation

To install a CMM Log Agent, proceed as follows:

1. Log in to the Control Machine.
2. Open the configuration file of the Control Machine with your favorite editor.
Example:

```
sudo vim /etc/ansible/hosts
```

3. Configure the host on which you want to install the agent. For this purpose, check the following section in the configuration file:

```
# Monasca Log Agent Hosts
[monasca_log_agent_group]
cmm-monasca-log-agent ansible_ssh_host=<monasca_host>
anime_ssh_user=<ssh_user_monasca>
```

If required, you can add a line for installing an additional agent on a different host. The installer allows you to install one or multiple instances of an agent at a time. When adding a line for an additional agent, you have to specify a unique alias for the host machine on which to install the agent.

4. Replace `<monasca_host>` with the host name or IP address of the remote machine on which to install the agent. Replace `<ssh_user_monasca>` with the user name of the user who is to install the agent.

**Note:** The user who installs the agent must have access to the machine on which the agent is to be installed. Access as a user with root privileges is required.

5. To establish the connection between the Control Machine and the remote machine, generate the SSH key for Ansible.
   Execute the following command:

```
ssh-keygen -t rsa
```

6. To copy the SSH key to a remote machine, execute the following command. Repeat the command for each remote machine on which to install an agent.

```
ssh-copy-id <user_name>@<host_ip_address>
```

Replace `<user_name>` by the user name to be used for accessing the remote machine, and replace `<host_ip_address>` by the host name or IP address of the remote machine.

**Example response for a successful connection to a remote machine:**

```
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
<user_name>@<host_ip_address>'s password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh '
'<user_name>@<host_ip_address>'" and check to make sure that only the key(s) you wanted were added.
```

7. Create a configuration file for the agent. It must be stored in the installation directory. An individual configuration file is required for each agent instance that is to be installed.
Execute the following command to create a configuration file. Repeat the command for all host machines on which to install an agent.

```
sudo vim /opt/FJSVssvcmm/host_vars/<file_name>
```

Replace `<file_name>` with the alias for the host machine. `<file_name>` must be identical to the alias specified in the configuration file of the Control Machine (see Step 3 above).

8. Configure the agents. The following example shows the minimum set of parameters that must be configured to start an agent. The minimum set is restricted to the credentials of an OpenStack user used for the communication between the CMM Service and the agent.

You can specify additional configuration settings, if required. For details on the configuration settings, refer to Configuration Settings for CMM Log Agent on page 60.

To define the minimum set of parameters, insert the following content into each configuration file. Replace the user credentials with the credentials of a valid OpenStack user who is authorized to send monitoring data to the CMM Service. Contact your OpenStack Operator. He is responsible for creating the required users and roles.

Example:

```
# file: host_vars/cmm-monasca-log-agent
# logstash monasca user credentials
project_name: cmm
username: cmm-operator
password: "{{ cmm_monasca_log_agent_keystone_password }}"
domain_id: default
```

**Note:** The installation automatically configures the agent so that a minimum set of log data can be retrieved. For information on how to update this set of log data, refer to Configuration on page 32.

9. Provide the passwords required for installing an agent. They are defined in the credentials.yml file that is located in the /opt/FJSVssvcmm directory.

Open the credentials.yml file with your favorite editor.

Example:

```
sudo vim /opt/FJSVssvcmm/credentials.yml
```

10. Specify the following passwords in the file:

   • `cmm_monasca_log_agent_keystone_password` for Keystone access of the user used for configuration purposes of the CMM Log Agent on the CMM host. The user account is automatically created with the installation.

   • `openstack_monasca_log_agent_keystone_password` for Keystone access of the user used for configuration purposes of the CMM Log Agent on the OpenStack host. The user account is automatically created with the installation.

11. To run the installer, execute the following command:

```
ansible-playbook /opt/FJSVssvcmm/monasca-log-agent.yml
```

The following response is displayed if the installation is successful.
Example:

```
PLAY RECAP ********************************************************
monasca-log-agent : ok=40   changed=23   unreachable=0   failed=0
```

**Note:** There are checks performed during installation that may result in failures or warnings. These failures and warnings are reported but they do not block the installation. It is only this final message that is relevant for a successful installation.

In case the installation fails, check your configuration settings and passwords and retry the installation in verbose mode. To collect debug information, you can execute the following command:

```
ansible-playbook -vvvv /opt/FJSVsvcmm/monasca-log-agent.yml
```

The agents are installed and automatically started after a successful installation. The installer creates a startup script that automatically starts the agent when the machine is booted.

An agent is provided as a LINUX service.

### 2.6.2 Configuration

The agent installation automatically configures the agent so that a minimum set of log data can instantly be retrieved. You can enhance the agent configuration or update the initial configuration, if required.

The following sections provide information on the file that stores the agent configuration and describe how to update and enhance the configuration.

#### Configuration File

The installer stores all configuration settings of the CMM Log Agent in the following file:

```
/etc/monasca/monasca-log-agent/agent.conf
```

The file is composed of an input and an output section:

- The input section specifies which log data is to be retrieved.
  
  The CMM Log Agent is based on the so-called ELK stack, a solution for searching and analyzing log data that combines the open source projects Elasticsearch, Logstash, and Kibana. For details on the ELK stack, refer to the documentation on [Elasticsearch, Logstash, and Kibana](#).

  CMM supports the file plugin of Logstash as input mechanism. The file plugin enables Logstash to read log data from any log file on your file system. Logstash supports additional plugins. For details, refer to [Logstash Input Plugins](#). Contact Fujitsu if you want to integrate a different plugin with CMM.

- The output section specifies all parameters required for retrieving the log data and sending it to the CMM Server for further processing.

#### Updating the Configuration File

To edit the `agent.conf` file, proceed as follows:

1. Log in to the server where the agent is installed.
2. To stop the agent, execute the following command:

```
sudo systemctl stop monasca-log-agent
```

3. Open the file with your favorite editor.
   Example:

```
sudo vim /etc/monasca/monasca-log-agent/agent.conf
```

4. Adapt the input section, if required.
   If you want to add files to be monitored, add a corresponding file block.
   If you want to define dimensions for the log files of a file block, define them with `add_field`.
   Dimensions allow you to collect meta information with the log data that is retrieved by the agent. The meta information is attached to each log entry. It is represented as a field in the CMM log management dashboard. For the user who is working with the log data, dimensions provide additional filtering options.
   Example configuration:

```yaml
input {
  file {
    path => "/var/log/keystone/*.log"
  }
  file {
    path => "/var/log/monasca/agent/*.log"
  }
  file {
    path => "/var/log/monasca/monasca-log-agent/*.log"
  }
  file {
    add_field => { "dimensions" => { "service" => "monasca-api" }}
    add_field => { "dimensions" => { "language" => "java" }}
    add_field => { "dimensions" => { "log_level" => "error" }}
    path => "/var/log/monasca/api/error.log"
  }
}
```

5. Adapt the output section, if required. Update the corresponding parameter values. Each value must be enclosed in double quotes (").
   For details on the configuration settings that can be defined, refer to *Configuration Settings for CMM Log Agent* on page 60.
   Example configuration:

```yaml
output {
  monasca_log_api {
    monasca_log_api_url => "http://192.168.10.4:5607/v3.0"
    keystone_api_url => "http://192.168.10.5:35357/v3"
    project_name => "cmm"
    username => "cmm-operator"
    password => "password"
    domain_id => "default"
    dimensions => ['app_type:kafka', 'priority:high']
    num_of_logs => 100
    delay => 1
    elapsed_time_sec => 600
    max_data_size_kb => 5120
  }
}
```
6. To start the agent again, execute the following command:

```
sudo systemctl start monasca-log-agent
```

The agent is instantly available with the updated configuration settings.

## 2.7 Uninstallation

CMM ships with uninstallers for automatically uninstalling the CMM software components.

Before uninstalling, it is recommended that you make a backup of the configuration files and the data created during operation. For details on backups, refer to Backup and Recovery on page 48.

When uninstalling the CMM software components, stick to the following sequence:

1. Uninstall the CMM agents.
2. Uninstall the CMM Service and the OpenStack extensions.
3. Manually remove remaining CMM components if they are no longer required.

### 2.7.1 Uninstalling the CMM Agents

To uninstall a CMM Metrics Agent or a CMM Log Agent, proceed as follows:

1. Log in to the Control Machine.
2. Open the configuration file of the Control Machine with your favorite editor.
   
   Example:
   ```
sudo vim /etc/ansible/hosts
```

3. Check that the hosts from which to uninstall an agent are configured correctly. Multiple instances of an agent are uninstalled if multiple hosts are specified in the file.

   The host name or IP address of the remote machines as well as the user name of the user who is to uninstall the agent must be specified.

   **Note:** The user who uninstalls the agents must have access to the corresponding hosts. Access as a user with root privileges is required.

4. To run the uninstaller for one or multiple instances of a CMM Metrics Agent, execute the following command:

   ```
   ansible-playbook /opt/FJSVsvcmm/monasca-agent-uninstaller.yml
   ```

   To run the uninstaller for one or multiple instances of a CMM Log Agent, execute the following command:

   ```
   ansible-playbook /opt/FJSVsvcmm/monasca-log-agent-uninstaller.yml
   ```

The agents and their dependencies are uninstalled:

- The agents are stopped.
- The corresponding services are removed (monasca-agent or monasca-log-agent).
The agents' configuration files and the directories where they are located are removed.

2.7.2 Uninstalling the CMM Service and the OpenStack Extensions

To uninstall the CMM Service and the OpenStack extensions, proceed as follows:

1. Log in to the Control Machine.
2. Open the configuration file of the Control Machine with your favorite editor.
   
   Example:
   
   ```
   sudo vim /etc/ansible/hosts
   ```
3. Check that the CMM host, the OpenStack Keystone host, and the OpenStack Horizon host from which you want to uninstall the CMM components are configured correctly.

   The host name or IP address of the remote machines from which to uninstall the software components as well as the user name of the user who is to uninstall the software components must be specified.

   **Note:** The user who uninstalls the software components must have access to the corresponding host. Access as a user with root privileges is required.
4. To run the uninstaller, execute the following command:
   
   ```
   ansible-playbook /opt/FJSVsvcmm/uninstaller.yml
   ```
5. To remove the technical users that were used by CMM, execute the following commands:

   ```
   userdel -r mon-schema
   userdel -r mon-api
   userdel -r mon-log-api
   userdel -r mon-persister
   userdel -r mon-notification
   userdel -r mon-thresh
   userdel -r mon-transformer
   userdel -r mon-agent
   userdel -r mon-log-agent
   userdel -r elastic
   userdel -r kafka
   userdel -r memcached
   ```

   As soon as the uninstallation is finished, you can delete the OpenStack users, roles, and projects that are no longer relevant.

   **Note:** Before installing CMM again, it is required to reboot the operating system.

2.7.3 Cleanup

The automatic uninstallation of the CMM Service and the OpenStack extensions does not include the removal of databases and message queues. In this way, possible dependencies to other systems are not broken deliberately. The following components are affected:

- MariaDB database
- InfluxDB database
- Elasticsearch database
• Kafka Message Queue
• Apache Storm Threshold Engine

If you no longer need one of the above components, and want to manually remove them, proceed as follows:
1. Log in to the host where the above components are installed as a user with root privileges.
2. For every component, proceed as described in the subsequent sections.

**MariaDB Database**
1. Stop the MariaDB service:

   ```bash
   systemctl stop mariadb
   ```

2. Disable the automatic start of the MariaDB service during system booting:

   ```bash
   systemctl disable mariadb
   ```

3. Remove the MariaDB packages from your YUM repository, provided that they are not used by other services:

   ```bash
   yum remove mariadb \
   mariadb-devel \
   mariadb-server \
   perl-Compress-Raw-Bzip2 \
   perl-Compress-Raw-Zlib \
   perl-DBD-MySQL \
   perl-DBI \
   perl-Data-Dumper \
   perl-IO-Compress \
   perl-Net-Daemon \
   perl-PlRPCrm
   ```

4. Remove the Python module required for the MariaDB database, provided that it is not used by other services:

   ```bash
   pip uninstall PyMySQL
   ```

5. Remove the MariaDB data directory:

   ```bash
   rm -rf /var/lib/mysql
   ```

6. Remove the MariaDB configuration file:

   ```bash
   rm -rf /etc/my.cnf
   ```

7. Remove the daemon directory:

   ```bash
   rm -rf /etc/my.cnf.d
   ```

8. Remove the log error directory:

   ```bash
   rm -rf /var/log/mariadb
   ```
9. Remove the MariaDB user and group:

   userdel -r mysql

10. Optionally, restart the systemctl daemon:

   systemctl daemon-reload

---

### InfluxDB Database

1. Stop the InfluxDB service:

   systemctl stop influxdb

2. Disable the automatic start of the InfluxDB service during system booting:

   /sbin/chkconfig influxdb off

3. Remove the InfluxDB package from your YUM repository:

   yum remove influxdb

4. Remove the remaining folders and files:

   rm -rf /var/log/influxdb
   rm -rf /var/opt/influxdb
   rm -rf /etc/opt/influxdb
   rm /etc/default/influxdb
   rm /etc/logrotate.d/influxdb

5. Remove the InfluxDB user and group:

   userdel -r influxdb

6. Optionally, restart the systemctl daemon:

   systemctl daemon-reload

---

### Elasticsearch Database

1. Stop the Elasticsearch service:

   systemctl stop elasticsearch

2. Disable the automatic start of the Elasticsearch service during system booting:

   /sbin/chkconfig elasticsearch off

3. Remove the remaining folders and files:

   rm -rf /var/data
   rm -rf /var/log/elasticsearch
   rm -rf /tmp/hsperfdata_elastic
Kafka Message Queue

1. Stop the Kafka service:

   `systemctl stop kafka`

2. Disable the automatic start of the Kafka service during system booting:

   `systemctl disable kafka`

3. Remove the remaining folders and files:

   `rm -rf /tmp/hsperfdata_kafka`
   `rm -rf /var/log/kafka`

4. Optionally, restart the `systemctl` daemon:

   `systemctl daemon-reload`

Apache Storm Threshold Engine

1. Stop the Storm services:

   `systemctl stop storm-nimbus storm-supervisor`

2. Disable the automatic start of the Storm services during system booting:

   `systemctl disable storm-nimbus storm-supervisor`

3. Remove the Storm service scripts:

   `rm /etc/systemd/system/storm-nimbus.service`
   `rm /etc/systemd/system/storm-supervisor.service`

4. Remove the Storm folders:

   `rm -rf /opt/storm`
   `rm -rf /tmp/hsperfdata_storm`
   `rm -rf /var/log/storm`
   `rm -rf /var/storm`

5. Remove the Storm user and group:

   `userdel -r storm`
6. Optionally, restart the `systemctl` daemon:

```
systemctl daemon-reload
```
3  Operation and Maintenance

Regular operation and maintenance includes:
• Starting and stopping agents and services.
• Disabling the monitoring of specific metrics in the configuration of a CMM Metrics Agent.
• Disabling the collection of specific log data in the configuration of a CMM Log Agent.
• Removing metrics data from the InfluxDB database.
• Removing log data from the Elasticsearch database.
• Handling log files of agents and services.
• Backup and recovery of databases, configuration files, and dashboards.

3.1 Starting and Stopping Agents and Services

The CMM installation package contains a script that you can use to start, stop, and view the status of the CMM agents and services. CMM consists of:

<table>
<thead>
<tr>
<th>#</th>
<th>Service Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>zookeeper</td>
<td>Centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.</td>
</tr>
<tr>
<td>2</td>
<td>storm-nimbus</td>
<td>Storm is a distributed real-time computation system for processing large volumes of high-velocity data. The Storm Nimbus daemon is responsible for distributing code around a cluster, assigning tasks to machines, and monitoring for failures.</td>
</tr>
<tr>
<td>3</td>
<td>storm-supervisor</td>
<td>The Storm supervisor listens for work assigned to its machine and starts and stops worker processes as necessary based on what Nimbus has assigned to it.</td>
</tr>
<tr>
<td>4</td>
<td>mariadb</td>
<td>MariaDB database service. CMM stores configuration information in this database.</td>
</tr>
<tr>
<td>5</td>
<td>kafka</td>
<td>Message Queue service.</td>
</tr>
<tr>
<td>6</td>
<td>influxdb</td>
<td>InfluxDB database service. CMM stores metrics and alarms in this database.</td>
</tr>
<tr>
<td>7</td>
<td>elasticsearch</td>
<td>Elasticsearch database service. CMM stores the log data in this database.</td>
</tr>
<tr>
<td>8</td>
<td>memcached</td>
<td>Memcached service. CMM uses it for caching authentication and authorization information required for the communication between the Log API and OpenStack Keystone.</td>
</tr>
<tr>
<td>9</td>
<td>monasca-notification</td>
<td>Notification Engine.</td>
</tr>
<tr>
<td>10</td>
<td>monasca-thresh</td>
<td>Threshold Engine.</td>
</tr>
<tr>
<td>11</td>
<td>monasca-log-transformer</td>
<td>Log Transformer.</td>
</tr>
<tr>
<td>12</td>
<td>monasca-log-api</td>
<td>Log API.</td>
</tr>
<tr>
<td>#</td>
<td>Service Name</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>-------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>13</td>
<td>monasca-persister</td>
<td>Persister.</td>
</tr>
<tr>
<td>14</td>
<td>monasca-api</td>
<td>Monitoring API.</td>
</tr>
<tr>
<td>15</td>
<td>monasca-agent</td>
<td>CMM Metrics Agent.</td>
</tr>
<tr>
<td>16</td>
<td>kibana</td>
<td>Kibana server.</td>
</tr>
<tr>
<td>17</td>
<td>monasca-log-persister</td>
<td>Log Persister.</td>
</tr>
<tr>
<td>18</td>
<td>monasca-log-agent</td>
<td>CMM Log Agent.</td>
</tr>
</tbody>
</table>

### Configuration
For starting and stopping agents and services, the script must be configured in the `services.yml` file. This file is contained in the installation package for CMM, `CMM-Service-1.1.0.tar.gz`, which you have extracted to `/opt/FJSVsvcmm` on the Control Machine.

In the configuration file, specify the CMM host where the CMM services and agents are installed. To do this, proceed, for example, as follows:

```
cd /opt/FJSVsvcmm
vim services.yml
```

- `hosts: cmm-node` defines the host where CMM is installed. The host name must match the host name specified for the CMM host in the `/etc/ansible/hosts` file on the Ansible Master, for example:

```
# Monasca Hosts
[monasca_group]
cmm-node ansible_ssh_host=192.168.10.4
    ansible_ssh_user=administrator
```

**Note:** Do not change the sequence of the services defined in the `services.yml` file.

### Usage
Proceed as follows to run the script:

1. Log in to the CMM host.
2. Connect to the Control Machine via SSH.
3. Go to the directory where the CMM installation package is located.

   Example:
   ```
cd /opt/FJSVsvcmm
   ```

4. Execute the script with the appropriate options:
   - To display help information:
     ```
     ./services.sh --help
     ```
• To start all CMM agents and services:

```
./services.sh start -i /etc/ansible/hosts
```

• To stop all CMM agents and services:

```
./services.sh stop -i /etc/ansible/hosts
```

• To view the status of all CMM agents and services:

```
./services.sh status -i /etc/ansible/hosts
```

Starting and Stopping Individual Agents and Services

You can start or stop the CMM agents and services individually. When doing so, you need to stick to the following sequence:

1. zookeeper
2. storm-nimbus
3. storm-supervisor
4. mariadb
5. kafka
6. influxdb
7. elasticsearch
8. memcached
9. monasca-notification
10. monasca-thresh
11. monasca-log-transformer
12. monasca-log-api
13. monasca-persister
14. monasca-api
15. monasca-agent
16. kibana
17. monasca-log-persister
18. monasca-log-agent

When stopping individual services, proceed in the reverse order.

Use the following command to start an individual service:

```
sudo systemctl start <service>
```

Example:

```
sudo systemctl start monasca-api
```

Use the following command to stop an individual service:

```
sudo systemctl stop <service>
```
Example:

```
sudo systemctl stop monasca-api
```

Use the following command to view the status of an individual service:

```
sudo systemctl status <service>
```

Example:

```
sudo systemctl status monasca-api
```

### 3.2 Cleaning up Data

At regular intervals, you should check for the amount and size of the collected metrics and log data, and delete any unnecessary data to free disk space. This section describes how to:

- Disable the monitoring of specific metrics in the configuration of a CMM Metrics Agent.
- Disable the collection of specific log data in the configuration of a CMM Log Agent.
- Remove metrics data from the InfluxDB database.
- Remove log data from the Elasticsearch database.

#### 3.2.1 Disabling Metrics for a CMM Metrics Agent

To disable the monitoring of a specific metrics that is no longer needed in the configuration of a CMM Metrics Agent, you have to delete the corresponding .yaml file. Proceed as follows:

1. Log in to the server where the agent is installed.
2. To stop the agent, execute the following command:

```
sudo systemctl stop monasca-agent
```

3. Change to the directory that stores the metrics used for monitoring.
   Example:

   ```
cd /etc/monasca/agent/conf.d
```

4. Delete the .yaml file that is no longer needed.
   Example:

   ```
sudo rm -i process.yaml
```

5. To start the agent again, execute the following command:

```
sudo systemctl start monasca-agent
```

#### 3.2.2 Disabling Log Data for a CMM Log Agent

To disable the collection of specific log data that is no longer needed in the configuration of a CMM Log Agent, you have to delete the corresponding entries in the agent.conf configuration file. Proceed as follows:

1. Log in to the server where the agent is installed.
2. To stop the agent, execute the following command:

    sudo systemctl stop monasca-log-agent

3. Open the agent configuration file with your favorite editor.

   Example:

    sudo vim /etc/monasca/monasca-log-agent/agent.conf

4. In the input section, delete the file block for the log data you no longer want to monitor.

   If you do not want to monitor log data on Keystone any longer, for example, delete the following file block:

   ```
   file {
       path => "*/var/log/keystone/*".log"
   }
   ```

5. To start the agent again, execute the following command:

    sudo systemctl start monasca-log-agent

### 3.2.3 Removing Metrics Data

Metrics data is stored in the Metrics and Alarms InfluxDB Database. InfluxDB features an SQL-like query language for querying data and performing aggregations on that data.

The CMM Metrics Agent configuration defines the metrics and types of measurement for which data is stored. For each measurement, a so-called series is written to the InfluxDB database. A series consists of a timestamp, the metric, and the value measured.

Example:

<table>
<thead>
<tr>
<th>Measurement: cpu.user_perc</th>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-08-11T11:55:03Z</td>
<td>cpu.user_perc</td>
<td>80</td>
</tr>
<tr>
<td>2015-08-11T12:55:03Z</td>
<td>cpu.user_perc</td>
<td>70</td>
</tr>
<tr>
<td>2015-08-11T13:55:03Z</td>
<td>cpu.user_perc</td>
<td>100</td>
</tr>
</tbody>
</table>

Every series can be assigned key tags. In the case of CMM, this is the _tenant_id tag. This tag identifies the OpenStack project for which the metrics data has been collected.

From time to time, you may want to delete outdated or unnecessary metrics data from the Metrics and Alarms Database, for example, to save space or remove data for metrics you are no longer interested in.

Proceed as follows to delete metrics data from the database:

1. Create a backup of the database. For details, refer to Backup and Recovery on page 48.
2. Determine the ID of the OpenStack project for the data to be deleted:

   Log in to the OpenStack dashboard and go to Identity > Projects.
The following OpenStack projects are initially relevant for metrics and alarms management:

- **admin**: All metrics data related to OpenStack.
- **cmm**: All metrics data related to CMM.

In the course of the productive operation of CMM, additional projects may be created, for example, for tenant users. The **Project ID** field shows the relevant tenant ID.

3. Log in to the host where CMM is installed.

4. Go to the directory where InfluxDB is installed:

   ```
   cd /opt/influxdb
   ```

5. Connect to InfluxDB using the InfluxDB command-line interface as follows:

   ```
   ./influx -host <cmm_host_ip>
   ```

   Replace **cmm_host_ip** with the host name or IP address of the machine on which CMM is installed.

   The output of this command is, for example, as follows:

   ```
   Connected to http://localhost:8086 version 0.9.1
   InfluxDB shell 0.9.1
   ```

6. Connect to the InfluxDB database of CMM (**mon**):

   ```
   > show databases
   name: databases
   ----------
   name
   mon
   > use mon
   Using database mon
   ```

7. Delete the desired data.

   - When a project is no longer relevant or a specific tenant is no longer used, delete all series for the project as follows:

     ```
     DROP SERIES WHERE _tenant_id = '<project ID>'
     ```

     Example:

     ```
     DROP SERIES WHERE _tenant_id = '27620d7ee6e948e29172f1d0950bd6f4'
     ```

   - When a metric is no longer relevant for a project, delete all series for the specific project and metric as follows:

     ```
     DROP SERIES FROM "<metric>" WHERE _tenant_id = '<project ID>'
     ```

     Example:

     ```
     DROP SERIES FROM "cpu.user_perc" WHERE _tenant_id = '27620d7e'
     ```
• In order to avoid that the entire database becomes too large, you can, for example, create a backup, then drop the database and create a new, empty one:

```sql
DROP DATABASE mon
CREATE DATABASE mon
```

8. Restart the `influxdb` service, for example, as follows:

```bash
sudo systemctl restart influxdb
```

You can view all measurements for a specific project as follows:

```sql
SHOW MEASUREMENTS WHERE _tenant_id = '<project ID>'
```

You can view the series for a specific metric and project, for example, as follows:

```sql
SHOW SERIES FROM "cpu.user_perc" WHERE _tenant_id = '<project ID>'
```

### 3.2.4 Removing Log Data

Log data is stored in the Elasticsearch database. Elasticsearch stores the data in indices. One index per day is created for every OpenStack project.

By default, the indices are stored in the following directory on the host where the CMM Service is installed:

```
/var/data/elasticsearch/<cluster-name>/nodes/<node-name>
```

Example:

```
/var/data/elasticsearch/elasticsearch/nodes/0
```

**Note:** If your system is configured in a different way, look up the directory in the Elasticsearch configuration file, `/opt/elasticsearch/config/elasticsearch.yml`.

If you want to delete outdated or unnecessary log data from the Elasticsearch database, proceed as follows:

1. Make sure that `curl` is installed. If this is not the case, download and install it, for example, from the following Web site: [http://curl.haxx.se/download.html](http://curl.haxx.se/download.html).
2. Create a backup of the Elasticsearch database. For details, refer to *Backup and Recovery* on page 48.
3. Determine the ID of the OpenStack project for the data to be deleted:
   
   Log in to the OpenStack dashboard and go to **Identity > Projects**. 
   
   The following OpenStack projects are relevant for log management:
   
   • **admin**: All log information related to OpenStack.
   
   • **cmm**: All log information related to CMM.
   
   In the course of the productive operation of CMM, additional projects may be created.
   
   The **Project ID** field shows the relevant ID.
4. Log in to the host where CMM is installed.
5. Make sure that the data you want to delete exists by executing the following command:

```
curl -XHEAD -i 'http://localhost:<port>/<projectID-date>'
```

For example, if Elasticsearch is listening at port 9200 (default), the ID of the OpenStack project is abc123, and you want to check the index of 2015, July 1st, the command is as follows:

```
```

If the HTTP response is 200, the index exists; if the response is 404, it does not exist.

6. Delete the index as follows:

```
curl -XDELETE -i 'http://localhost:<port>/<projectID-date>'
```

Example:

```
```

This command either returns an error, such as IndexMissingException, or acknowledges the successful deletion of the index.

**Note:** Be aware that the -XDELETE command immediately deletes the index file!

Both, for -XHEAD and -XDELETE, you can use wildcards for processing several indices. For example, you can delete all indices of a specific project for the whole month of July, 2015:

```
```

**Note:** Take extreme care when using wildcards for the deletion of indices. You could delete all existing indices with one single command!

### 3.3 Log File Handling

In case of trouble with the CMM services, you can study their log files to find the reason. The log files are also useful if you need to contact your support organization. The table below lists the location of the log files for all agents and services. The files are written to the directories on the hosts where the agents or services are installed.

<table>
<thead>
<tr>
<th>#</th>
<th>Service Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Apache Storm</td>
<td>/var/log/storm</td>
</tr>
<tr>
<td>2</td>
<td>Elasticsearch</td>
<td>/var/log/elasticsearch</td>
</tr>
<tr>
<td>3</td>
<td>InfluxDB</td>
<td>/var/log/influxdb</td>
</tr>
<tr>
<td>4</td>
<td>Kafka</td>
<td>/var/log/kafka</td>
</tr>
<tr>
<td>5</td>
<td>Kibana server</td>
<td>/var/log/kibana</td>
</tr>
<tr>
<td>6</td>
<td>Log Agent</td>
<td>/var/log/monasca/monasca-log-agent</td>
</tr>
<tr>
<td>7</td>
<td>Log API</td>
<td>/var/log/monasca/log-api</td>
</tr>
<tr>
<td>8</td>
<td>Log Persister</td>
<td>/var/log/monasca/monasca-log-persister</td>
</tr>
</tbody>
</table>
### 3.4 Backup and Recovery

Typical tasks of the CMM operator are to make regular backups, particularly of the data created during operation.

At regular intervals, you should make a backup of all:
- Databases.
- Configuration files of the individual agents and services.
- Monitoring and log dashboards you have created and saved.

CMM does not offer integrated backup and recovery mechanisms. Instead, you use the mechanisms and procedures of the individual components.

#### 3.4.1 Databases

You need to create regular backups of the following databases on the host where the CMM Service is installed:
- Elasticsearch database
- InfluxDB database
- MariaDB database

It is recommended that backup and restore operations for databases are carried out by experienced operators only.
Elasticsearch Database

For backing up and restoring your Elasticsearch database, you can use the Snapshot and Restore module of Elasticsearch.

To make a backup of the database, proceed as follows:

1. Make sure that `curl` is installed. If this is not the case, download and install it, for example, from the following Web site: [http://curl.haxx.se/download.html](http://curl.haxx.se/download.html).

2. Log in to the host where the CMM Service is installed.

3. Create a snapshot repository.

   Example:

   ```
   curl -XPUT http://localhost:9200/_snapshot/my_backup -d '{
   "type": "fs",
   "settings": {
     "location": "/mount/backup/elasticsearch/my_backup",
     "compress": true
   }
   }'
   ```

   The example registers a shared file system repository ("type": "fs") that uses the `/mount/backup/elasticsearch/my_backup` directory for storing snapshots.

   **Note:** The directory for storing snapshots must be specified as configured during the installation of the CMM Service in the `group_vars/monasca_group` file.

   `compress` is turned on to compress the metadata files.

4. Check whether the repository was created successfully.

   Example:

   ```
   curl -XGET http://localhost:9200/_snapshot/my_backup
   ```

   Example response for a successfully created repository:

   ```
   {
   "my_backup": {
     "Type": "fs",
     "settings": {
       "compress": "true",
       "location": "/mount/backup/elasticsearch/my_backup"
     }
   }
   }
   ```

5. Create a snapshot of your database that contains all indices. A repository can contain multiple snapshots of the same database. The name of a snapshot must be unique within the snapshots created for your database.

   Example:

   ```
   curl -XPUT http://localhost:9200/_snapshot/my_backup/snapshot_1?wait_for_completion=true
   ```

   The example creates a snapshot named `snapshot_1` for all indices in the `my_backup` repository.
To restore the database instance, proceed as follows:

1. Close all indices of your database.
   Example:
   ```bash
curl -XPOST http://localhost:9200/_all/_close
   ``

2. Restore all indices from the snapshot you have created.
   Example:
   ```bash
curl -XPOST http://localhost:9200/_snapshot/my_backup/snapshot_1/_restore
   ``

   The example restores all indices from snapshot_1 that is stored in the my_backup repository.

For additional information backing up and restoring an Elasticsearch database, refer to the *Elasticsearch documentation*.

**InfluxDB Database**

For backing up and restoring your InfluxDB database, you can use the InfluxDB shell. The shell is part of your InfluxDB distribution. If you installed InfluxDB via a package manager, the shell is, by default, installed in the `/opt/influxdb` directory.

To make a backup of the database, proceed as follows:

1. Log in to the InfluxDB database as a user who is allowed to run the `influxdb` service.
   Example:
   ```bash
   $ su influxdb
   ```

2. Back up the database.
   Example:
   ```bash
   $ ./influxd backup -host "<host>:<port>" /mount/backup/mysnapshot
   ```

   Replace `<host>` and `<port>` with the host name or IP address and port number that are defined in the meta section in your InfluxDB configuration file. By default, the influxdb.conf file is located in the `/etc/opt/influxdb/` directory. The example creates the backup for the database in `/mount/backup/mysnapshot`.

Before restoring the database, make sure that all database processes are shut down. To restore the database, you can then proceed as follows:

1. Stop the InfluxDB database service. Use the following command:
   ```bash
   $ systemctl stop influxdb
   ```

2. Log in to the InfluxDB database as a user who is allowed to run the `influxdb` service.
   Example:
   ```bash
   $ su influxdb
   ```

3. Restore the database.
Example:

$ ./influxd restore -config /etc/opt/influxdb/influxdb.conf /mount/backup/mysnapshot

The example restores the backup from /mount/backup/mysnapshot to /etc/opt/influxdb/influxdb.conf.

4. Start the InfluxDB database service. Use the following command:

$ systemctl start influxdb

For additional information on backing up and restoring an InfluxDB database, refer to the InfluxDB documentation.

**MariaDB Database**

For backing up and restoring your MariaDB database, you can use the `mysqldump` utility program. `mysqldump` performs a logical backup that produces a set of SQL statements. These statements can later be executed to restore the database.

To back up your MariaDB database, you must be the owner of the database or a user with superuser privileges. You can use the following command.

Example:

`mysqldump -u root -p mon > dumpfile.sql`

In addition to the name of the database, you have to specify the name and the location where `mysqldump` stores its output.

To restore your MariaDB database, proceed as follows:

1. Log in to the host where the CMM Service is installed as a user with root privileges.

2. Go to the directory where the CMM installation package is located.

Example:

`cd /opt/FJSVsvcmm`

3. Stop all CMM agents and services:

`./services.sh stop`

4. Start the `mariadb` service:

`systemctl start mariadb`

5. Log in to the database you have backed up as a user with root privileges.

Example:

`mysql -u root -p mon`

6. Remove and create the database:

>` DROP DATABASE mon;`
> CREATE DATABASE mon;

7. Exit mariadb:

> \q

8. Restore the database.

Example:

```
mysql -u root -p mon < dumpfile.sql
```

9. Stop the mariadb service:

```
systemctl stop mariadb
```

10. Start all CMM agents and services:

```
./services.sh start
```

For additional information on backing up and restoring a MariaDB database with mysqldump, refer to the MariaDB documentation.

### 3.4.2 Configuration Files

Below you find a list of the configuration files of the agents and the individual services included in the CMM Service. Back up these files at least after you have installed and configured CMM and after each change in the configuration.

```
/etc/default/influxdb
/etc/firewalld/firewalld.conf
/etc/firewalld/lockdown-whitelist.xml
/etc/firewalld/zones/public.xml
/etc/kafka/server.properties
/etc/my.cnf
/etc/my.cnf.d/client.cnf
/etc/my.cnf.d/mysql-clients.cnf
/etc/my.cnf.d/server.cnf
/etc/monasca/agent/agent.yaml
/etc/monasca/agent/conf.d/*
/etc/monasca/agent/supervisor.conf
/etc/monasca/api-config.yml
/etc/monasca/log-api-config.yml
/etc/monasca/log/persister.conf
/etc/monasca/monasca-log-agent/agent.conf
/etc/monasca/notification.yaml
/etc/monasca/persister-config.yml
/etc/monasca/thresh-config.yml
/etc/opt/influxdb/influxdb.conf
/etc/rc.d/rc0.d/K50influxdb
/etc/rc.d/rc1.d/K50influxdb
/etc/rc.d/rc2.d/S50influxdb
/etc/rc.d/rc3.d/S50influxdb
/etc/rc.d/rc4.d/S50influxdb
/etc/rc.d/rc5.d/S50influxdb
/etc/rc.d/rc6.d/K50influxdb
/etc/systemd/system/elasticsearch.service
```
3.4.3 Dashboards

Kibana can persist customized log dashboard designs to the Elasticsearch database, and allows you to recall them. For details on saving, loading, and sharing log management dashboards, refer to the Kibana documentation.

Grafana allows you to export customized monitoring dashboards to a location of your choice and re-import them when necessary. For backing up and restoring the exported dashboards, use the standard mechanisms for the file system. For details on exporting monitoring dashboards, refer to the Getting Started tutorial of Grafana.

Recovery

If you need to recover the configuration of one or more agents or services, the recommended procedure is as follows:

1. If necessary, uninstall the agents or services, and install them again.
2. Stop the agents or services.
3. Copy the backup of your configuration files to the correct location according to the table above.
4. Start the agents or services again.
4 Monitoring Services and Servers

CMM offers various features for monitoring your services and the servers on which they are provisioned. They include:

- A monitoring overview which allows you to keep track of the services and servers and quickly check their status.
- Metrics dashboards for visualizing your monitoring data.
- Alerting features for monitoring.

In the following sections, you will find information on the monitoring overview and the integrated metrics dashboards as well as details on how to define and handle alarms and notifications.

Accessing CMM

To perform monitoring tasks, you must have access to the OpenStack platform as a user with the cmm-user role or any other role that is authorized to use the CMM monitoring functions. Additional roles are optional. In addition, you must be assigned to the OpenStack project you want to monitor.

Log in to the OpenStack platform using the credentials of a corresponding user. This allows you to access the OpenStack Horizon dashboard. The CMM functionality is available on the Monitoring tab. You can access all monitoring data of the project to which you are assigned.

4.1 Overview of Services and Servers

CMM provides a comfortable status overview of your services and servers. Use Monitoring > Overview to view their status at a glance.

The Overview page enables access to all monitoring data retrieved by the agent. Use the options at the top border of the Overview page to access preconfigured CMM metrics dashboards. You can create your own dashboard that visualizes your monitoring data, as required. For details, refer to Working with Data Visualizations on page 54.

As soon as you have defined an alarm for a service or server, there is status information displayed for it on the Overview page:

- A service or server in a green box indicates that it is up and running. There are alarms defined for it, but the defined thresholds have not yet been reached or exceeded.
- A service or server in a red box indicates that there is a severe problem that needs to be checked. One or multiple alarms defined for the service or server have been triggered.
- A service or server in a yellow box indicates a problem. One or multiple alarms have already been triggered, yet, the severity of these alarms is low.
- A service or server in a gray box indicates that alarms have been defined, yet, the monitoring data has not yet been collected and sent.

For details on defining alarms, refer to Defining Alarms on page 55.

4.2 Working with Data Visualizations

The user interface for monitoring your services and servers integrates with Grafana, an open source application for visualizing large-scale monitoring data on metrics dashboards. CMM ships with preconfigured metrics dashboards for operators. You can also configure your own dashboards, if required.
Each metrics dashboard is composed of individual panels that are arranged in a number of rows. You can define which content is displayed in which panel and which time range is visualized. You can drag and drop panels within and between rows. Additional display options enable you to customize how your dashboard presents the defined content.

In addition to creating your own dashboards, CMM allows you to export and import dashboards. You can also create and save dashboard templates that facilitate customizations. For details on working with the panel editing user interface of the metrics dashboard, refer to the Getting Started tutorial of Grafana.

Use Monitoring > Overview in the OpenStack Horizon dashboard to access the CMM metrics dashboard. The options at the top border of the Overview page allow you to access the preconfigured dashboards for operators. You can customize them to visualize your monitoring data, if required.

### 4.3 Defining Alarms

CMM allows you to define alarms for monitoring your cloud resources. An alarm definition specifies the metrics to be collected and the threshold at which an alarm is to be triggered. If the specified threshold is reached or exceeded for a cloud resource, notifications can be sent to inform the CMM users involved.

You can create simple or complex alarm definitions to handle a large variety of monitoring requirements. The syntax for defining alarms is based on a simple expressive grammar.

For an alarm definition, you specify the following elements:

- **Name.** Mandatory identifier of the alarm. The name must be unique within the project for which you define the alarm.

- **Expression.** Sub-expressions and logical operators that define the alarm. As soon as this expression evaluates to true, the alarm is triggered.

  To define the expression, proceed as follows:

  1. Select the metrics you want to monitor. The select list allows you to access all metrics provided by your OpenStack operator.

  2. Select a statistical function for the metrics: min to monitor the minimum values, max to monitor the maximum values, sum to monitor the sum of the values, count for the monitored number, or avg for the arithmetic average.

  3. Enter one or multiple dimensions in the **Add dimension** field to further qualify the metrics. Dimensions filter the data to be monitored. Each dimension consists of a key/value pair that allows for a flexible and concise description of the data to be monitored, for example, region, availability zone, service tier, or resource ID.

    The dimensions available for the selected metrics are displayed in the Matching Metrics section. Type the name of the key you want to associate with the metrics in the **Add dimension** field. You are offered a select list for adding the required key/value pair.

  4. Enter the threshold value at which an alarm is to be triggered, and combine it with a relational operator: <, >, <=, or >=.

    The unit of the threshold value is related to the metrics for which you define the threshold, for example, the unit is percentage for cpu.system_perc or MB for disk.total_used_space_mb.
4: Monitoring Services and Servers

Note: If you need to define alarm definitions that are more complex, create the alarm definition first and update it afterwards. The Edit Alarm Definition page allows you to directly edit the expression syntax.

- **Match by.** Enter the dimensions that should be taken into account for triggering an alarm. If you want them to be ignored in the expression that is to be evaluated, leave the field empty.

- **Description.** Optional. A short description that depicts the purpose of the alarm.

- **Severity.** The following severities for an alarm are supported: **Low**, **Medium**, **High**, or **Critical**.

  The severity affects the status information on the **Overview** page. If an alarm that is defined as **Critical** is triggered, the corresponding resource is displayed in a red box. If an alarm that is defined as **Low**, **Medium**, or **High** is triggered, the corresponding resource is displayed in a yellow box only.

  The severity level is subjective. Choose a level that is appropriate for prioritizing your alarms.

- **Notifications.** Optional. Alerts to be used for the alarm. As soon as the alarm is triggered, the notifications you select are sent.

  The notifications must have been predefined. For details, refer to the **Defining Notifications** on page 56.

For additional details on alarm definitions, refer to the **Monasca API documentation**.

To create, edit, and delete alarms, use **Monitoring > Alarm Definitions**.

### 4.4 Defining Notifications

Notifications define how CMM users are informed when a threshold value defined for an alarm is reached or exceeded. In the alarm definition, you can assign one or multiple notifications.

For a notification, you specify the following elements:

- **Name.** A unique identifier of the notification. The name is offered for selection when defining an alarm.

- **Type.** Email is the notification method supported by CMM. If you want to use **WebHook** or **PagerDuty**, contact your FUJITSU support organization for information.

- **Address.** The email address to be notified when an alarm is triggered.

Note: Generic top-level domains such as business domain names are not supported in email addresses (for example, user@xyz.fujitsu.com).

To create, edit, and delete notifications, use **Monitoring > Notifications**.
5 Managing Log Data

For managing the log data of your services and the virtual and physical servers on which they are provisioned, CMM provides a log management dashboard. The dashboard offers options for visualizing and analyzing your log data.

You have to configure an index pattern before you have access to the log data that is retrieved by your CMM Log Agent. Based on index patterns, you can view and analyze specific log data in the dashboard.

Accessing CMM

To perform log management tasks, you must have access to the OpenStack platform as a user with the admin and the cmm-user role, or any other role that is authorized to use the CMM log management functions. Additional roles are optional.

Log in to the OpenStack platform using the credentials of a corresponding user. This allows you to access the OpenStack Horizon dashboard. The CMM log management functions are available on the Monitoring tab.

Note: CMM does not support multi-tenant log management. This means that any user with the admin role has access to all logs, independent of the project that is specified in the agent configuration.

For accessing the log management dashboard from the Monitoring tab, use Monitoring > Overview. The Log Management option is provided at the top border of the Overview page.

As soon as you access the dashboard, the complete log management functionality is available. To view log data, configure an index pattern first.

5.1 Configuring an Index Pattern

In order to view and analyze log data in the log management dashboard, you must configure at least one index pattern. Index patterns are used to identify the search index to run search and analytics against. They are also used to automatically configure the fields displayed in the dashboard.

CMM enables dynamic mapping of fields. After configuring an index pattern, the indices that match the pattern are automatically scanned to display a list of the index fields. This guarantees that the fields are correctly visualized in the log management dashboard.

The first index pattern you configure is automatically set as the default. You can create one or multiple index patterns per project. You can also create index patterns for different projects. When you create more than one index pattern, you have to select the default pattern. The default pattern defines which log data is visualized when the CMM log management dashboard is accessed.

When you access the log management dashboard for the first time, you are automatically prompted to configure an index pattern. For an index pattern, you specify the following elements:

- **Index contains time-based events.** It is recommended that this option is selected. This improves search performance by enabling searches only on those indices that contain data on time-based events.

- **Use event times to create index names.** It is recommended that this option is selected. This improves search performance by enabling searches only on those indices that contain data in the time range you specify.
5: Managing Log Data

• **Index pattern interval.** Select Daily as index pattern interval. Daily intervals are supported by the CMM Service.

• **Index name or pattern.** The pattern allows you to define dynamic index names. Static text in a pattern is denoted using brackets. Replace the predefined pattern ([logstash-]* or [logstash-]YYYY.MM.DD) as follows:

  Replace logstash- by the project ID of the OpenStack project whose log data is to be visualized in the dashboard.

  Replace * or YYYY.MM.DD by YYYY-MM-DD as naming pattern. This naming pattern is supported by the CMM Service.

  Example: [557aff4bf007473d84069aca202a1633-]YYYY-MM-DD

• **Time-field name.** Select @timestamp as time-field name. @timestamp matches the YYYY-MM-DD naming pattern.

5.2 Working with the Log Management Dashboard

The user interface for managing your log data is based on Kibana, an open source analytics and visualization platform that uses Elasticsearch as the underlying database technology. Kibana allows you to search, view, and interact with your log data, thus supporting you in understanding large data volumes. You can easily perform advanced data analysis and visualize your data in a variety of charts, tables, and maps. Changes to the underlying log data are displayed in real time.

The log management dashboard provides the following options:

• Submitting search queries, filtering the search results, and viewing and examining the data in the log entries that are returned. You can see the number of log entries that match the search query, and you get field value statistics. If a time field is configured for the index pattern, the distribution of log entries over time is displayed in a histogram at the top of the log management dashboard.

• Constructing a visualization of search results. You can save visualizations, use them individually, or combine them in a dashboard. Different visualization types are supported, for example, data tables, line charts, pie charts, or vertical bar charts.

• Combining visualizations in dashboards for correlating related information. A dashboard can be saved and reloaded later. A saved dashboard can be accessed and used by any OpenStack or CMM operator.

• Sharing dashboards with other users by providing a direct link to the dashboard or embedding it in a Web page. A user must have Kibana access in order to use an embedded dashboard.

For details on working with the log management dashboard, refer to the Kibana documentation.
Appendix A: Configuration Settings

A.1 Configuration Settings for CMM Metrics Agents

The configuration settings for the CMM Metrics Agent are provided in the following file:
/etc/monasca/agent/agent.yaml

Find below a list of the parameters that are supported by CMM.
For a description of the available parameters, you can execute the following command:

```
monasca-setup --help
```

- amplifier
- ca_file
- check_frequency
- config_dir
- detection_args
- detection_plugins
- dry_run
- insecure
- keystone_url
- log_dir
- log_level
- monasca_url
- overwrite
- password
- project_domain_id
- project_domain_name
- project_id
- project_name
- remove
- service
- skip_enable
- system_only
- template_dir
- user
- username
- user_domain_id
- user_domain_name
- verbose
A.2 Configuration Settings for CMM Log Agent

The configuration settings for the CMM Log Agent are provided in the following file:
/etc/monasca/monasca-log-agent/agent.conf

Below you find a detailed description of the individual settings.

**monasca_log_api_url**
Mandatory. The URL used to access the machine where the CMM Service is installed.
Example: http://192.168.10.4:5607/v3.0

**keystone_api_url**
Mandatory. The URL used to access the server where the OpenStack Keystone service is installed. The service is used for authenticating the user specified in **username**.
Example: http://192.168.10.5:35357/v3

**project_name**
Mandatory. The name of the OpenStack project for which log data is to be retrieved by the agent.
Example: cmm

**username**
Mandatory. The user to be used for authenticating the agent against Keystone. The user specified here must have the **cmm-agent** role in OpenStack and be assigned to the OpenStack project that is to be monitored by the agent. The project is specified in **project_name**. It is recommended that this user is used only for configuration purposes and not for actually monitoring services and servers.
Example: admin-agent

**password**
Mandatory. The password of the user specified in **username**.

**domain_id**
Mandatory. The ID of the OpenStack Keystone domain to which the user specified in **username** is assigned.
Example: default

**dimensions**
Optional. Meta information to be collected with the log data that is retrieved by the agent. The information can be defined as an array.
The meta information defined by a dimension is attached to each log entry. It is represented as one or more fields in the CMM log management dashboard. For the user who is working with the log data, dimensions provide additional filtering options.

**Example:** 
['app_type:kafka', 'priority:high']

### num_of_logs
Optional. The maximum number of logs per request that is sent to the Log API.

It is recommended that you check the number of logs you have to manage. The lower the maximum number of logs you specify, the higher the log management performance.

Allowed values: Any value above 0  
Default: 125

### max_data_size_kb
Optional. The maximum data load in kilobyte for sending a request to the Log API.

Allowed values: Any value above 0  
Default: 5120

### elapsed_time_sec
Optional. The time interval in seconds for sending logs to the Log API.

Logs are sent in the interval you specify provided that the maximum number of logs specified in num_of_logs or the maximum data load specified in max_data_size_kb is not reached earlier.

Allowed values: Any value above 0  
Default: 30

### delay
Optional. The delay time in seconds until it is checked whether the time interval specified in elapsed_time_sec is reached.

Allowed values: Any value above 0  
Default: 10
Appendix B: Supported Metrics

The sections below describe the metrics supported by CMM:

- Standard metrics for general monitoring of servers and networks.
- Additional metrics for monitoring specific servers and services.

For you as the CMM operator, the following metrics are of relevance for monitoring the services CMM consists of and the server on which they are installed:

- All the standard metrics.
- The following additional metrics: elastic, host_alive, http_check, kafka_consumer, postgres, process, zk.

The CMM Metrics Agent can also run Nagios plugins and send status codes returned by the plugins as metrics to the Monitoring API. For information on how to use Nagios checks in CMM, contact your FUJITSU support organization.

| Note:  | Adding dimensions for metrics is not supported by CMM. The installer auto-detects applications and processes that are running on your machine and saves the corresponding settings to the agent's configuration file. Additional dimensions cannot be specified. |

B.1 Standard Metrics

CMM supports the following standard metrics for monitoring servers and networks. These metrics usually do not require specific settings. The metrics are grouped by metrics types. Each metrics type references a set of related metrics.

- **cpu.yaml**
  Metrics on CPU usage, e.g. the percentage of time the CPU is idle when no I/O requests are in progress, or the percentage of time the CPU is used at system level or user level.

- **disk.yaml**
  Metrics on disk space, e.g. the percentage of disk space that is used on a device, or the total amount of disk space aggregated across all the disks on a particular node.

- **load.yaml**
  Metrics on the average system load over different periods (e.g. 1 minute, 5 minutes, or 15 minutes).

- **memory.yaml**
  Metrics on memory usage, e.g. the number of megabytes of total memory or free memory, or the percentage of free swap memory.

- **network.yaml**
  Metrics on the network, e.g. the number of network bytes received or sent per second, or the number of network errors on incoming or outgoing network traffic per second.
B.2 Additional Metrics

CMM supports the additional metrics described below for monitoring specific servers and services. The metrics are grouped by metrics types. Each metrics type references a set of related metrics. Depending on the services running on the host where you install a CMM Monitoring Agent, some or all of these metrics are added to the agent configuration. You should check the individual YAML files and change or correct the settings as required, or remove individual YAML files from the agent configuration if you do not want to monitor the metrics they include.

Note that in addition to the metrics below, many more metrics are provided by the Monasca project. These are not installed and supported by CMM. For additional information on the supported metrics, you can also refer to the Monasca documentation.

apache.yaml
Apache Web Server checks gather metrics from an Apache Web Server. The configuration file must contain the URL of the server, as well as the user name and password for accessing it.

Example configuration:

```yaml
init_config:
instances:
  - apache_status_url: http://localhost/server-status?auto
    apache_user: Root
    apache_password: password
```

elastic.yaml
Elastic checks gather metrics for Elasticsearch databases, such as the Log Database of CMM. The configuration file must specify the URL for HTTP requests. If basic authentication is used, for example, elasticsearch-http-basic, the configuration file must also specify the user name and password for every instance that requires authentication.

Example configuration:

```yaml
init_config:
instances:
  - url: http://localhost:9200
    username: username
    password: password
```

host_alive.yaml
Host alive checks perform active checks on a remote host to determine whether it is alive. The checks use ping (ICMP) or SSH.

SSH checks provide extensive tests on the availability of remote host machines. They check the banner that is returned. A remote host machine may still respond to a ping request but may not return an SSH banner. Therefore it is recommended that you use SSH checks instead of ping checks if possible.

Example configuration:

```yaml
init_config:
  ssh_port: 22
  # ssh_timeout is a floating point number (seconds)
```
ssh_timeout: 0.5

# ping_timeout is an integer number (seconds)
ping_timeout: 1

instances:
# alive_test can be either "ssh" for an SSH banner test (port 22)
# or "ping" for ICMP ping test instances:
- name: ssh to somehost
  host_name: somehost.somedomain.net
  alive_test: ssh

- name: ping gateway
  host_name: gateway.somedomain.net
  alive_test: ping

- name: ssh to 192.168.0.221
  host_name: 192.168.0.221
  alive_test: ssh

http_check.yaml

HTTP endpoint checks perform up/down checks on HTTP endpoints. Based on a list of URLs, the agent sends an HTTP request and reports success or failure to the CMM Service.

Example configuration:

```
init_config:

instances:
  url: http://192.168.0.254/healthcheck
  timeout: 1
  include_content: true
  collect_response_time: true
  match_pattern: '.*OK.*OK.*OK.*OK.*OK'
```

kafka Consumer.yaml

Kafka consumer checks gather metrics related to services consuming Kafka topics, such as the Persisters or Notification Engine of CMM.

For Kafka consumer checks, the Kafka consumer module (kafka-python) must be installed in the virtualenv environment of the CMM Metrics Agent. To install it in the default directory, execute the following command:

```bash
# source /opt/monasca-agent/bin/activate
# pip install kafka-python
# deactivate
```

Example configuration:

```
init_config:

instances:
  - consumer_groups:
      '1_alarm-state-transitions':
        'alarm-state-transitions': ['3', '2', '1', '0']
      '1_metrics':
        'metrics': &id001 ['3', '2', '1', '0']
        'test':
          'healthcheck': ['1', '0']
```
Appendix B: Supported Metrics

libvirt.yaml
Libvirt checks provide metrics for virtual machines that run on a hypervisor. The checks provide a set of metrics for the owner of the virtual machine as well as for the owner of the hypervisor.

Example configuration:

```yaml
init_config:
  admin_password: pass
  admin_tenant_name: services
  admin_user: nova
  identity_uri: 'http://192.168.10.5:35357/v2.0'
  region_name: 'region1'
  cache_dir: /dev/shm
  nova_refresh: 14400
  vm_probation: 300
  ping_check: /usr/bin/fping -n -c1 -t250 -q
  ping_only: false

instances:
  - {}
```

mysql.yaml
MySQL checks gather metrics from a MySQL database server. The metrics are related to the server status variables of MySQL.

For MySQL checks, the MySQL module (pymysql) must be installed in the virtualenv environment of the CMM Metrics Agent. To install it in the default directory, execute the following command:

```
# source /opt/monasca-agent/bin/activate
# pip install pymysql
# deactivate
```

Example configuration:

```yaml
init_config:

instances:
  defaults_file: /root/.my.cnf
  server: localhost
  user: root
```

ntp.yaml
Network Time Protocol checks monitor the time offset between the NTP server and the host machine. The configuration file must specify the host name, the port number, version information, and the timeout.

Example configuration:

```yaml
init_config:

instances:
```
Appendix B: Supported Metrics

- host: pool.ntp.org
  port: ntp
  version: 3
  timeout: 5

**postfix.yaml**
Postfix checks monitor a Postfix mail server. The configuration file must specify the name, directory, and queue of the server to be monitored.

Example configuration:

```yaml
init_config:
instances:
  - name: /var/spool/postfix
    directory: /var/spool/postfix
    queues:
      - incoming
      - active
      - deferred
```

**postgres.yaml**
Postgres checks gather metrics from a PostgreSQL database.

Example configuration:

```yaml
init_config:
instances:
  - host: localhost
    port: 5432
    username: my_username
    password: my_password
    dbname: db_name
    relations:
      - my_table
      - my_other_table
```

**process.yaml**
Process checks verify that a defined set of processes is up and running. The processes can be identified by specifying the process name or a pattern match.

Example configuration:

```yaml
init_config:
instances:
  - name: ssh
    search_string: ['ssh', 'sshd']
  - name: mysql
    search_string: ['mysql']
    exact_match: True
```

**rabbitmq.yaml**
RabbitMQ checks gather metrics on nodes, exchanges, and queues from a RabbitMQ server. The configuration file must specify the names of the exchanges and queues to be monitored.
For RabbitMQ checks, the RabbitMQ Management plugin must be installed. It is included in the RabbitMQ distribution. To enable the plugin, execute the following command:

```
rabbitmq-plugins enable rabbitmq_management
```

**Example configuration:**

```
init_config:
instances:
  - exchanges: [nova, cinder, ceilometer, glance, keystone, neutron, heat]
    nodes: [rabbit@devstack]
    queues: [conductor]
    rabbitmq_api_url: http://localhost:15672/api
    rabbitmq_user: guest
    rabbitmq_pass: guest
```

**zk.yaml**

ZooKeeper checks gather metrics on nodes and connections covered by ZooKeeper, a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. The check parses the result of the ZooKeeper `stat admin` command.

**Example configuration:**

```
init_config:
instances:
  - host: localhost
    port: 2181
    timeout: 3
```
Glossary

CMM Operator
A person responsible for maintaining and administrating CMM.

Dimension
A key/value pair that allows for a flexible and concise description of the data to be monitored, for example, region, availability zone, service tier, or resource ID. Each dimension describes a specific characteristic of the metrics to be monitored.
In CMM, metrics are uniquely identified by a name and a set of dimensions. Dimensions can serve as a filter for the monitoring data.

Elasticsearch
An open source application that provides a highly scalable full-text search and analytics engine. CMM uses Elasticsearch as the underlying technology for storing, searching, and analyzing large volumes of log data.

Grafana
An open source application for visualizing large-scale measurement data. CMM integrates with Grafana for visualizing the CMM monitoring data.

Infrastructure as a Service (IaaS)
The delivery of computer infrastructure (typically a platform virtualization environment) as a service.

Kibana
An open source analytics and visualization platform designed to work with Elasticsearch. CMM integrates with Kibana for visualizing the CMM log data.

Logstash
An open source application that provides a data collection engine with pipelining capabilities. CMM integrates with Logstash for collecting, processing, and outputting logs.

Metrics
Self-describing data structures that allow for a flexible and concise description of the data to be monitored. Metrics values represent the actual monitoring data that is collected and presented in CMM.

Monasca
An open source Monitoring as a Service solution that integrates with OpenStack. It forms the core of CMM.

OpenStack Operator
A person responsible for maintaining and administrating OpenStack, the underlying platform technology of CMM.
**Platform as a Service (PaaS)**
The delivery of a computing platform and solution stack as a service.

**Software as a Service (SaaS)**
A model of software deployment where a provider licenses an application to customers for use as a service on demand.

**Tenant User**
A person with limited access to cloud resources in OpenStack. A tenant user provides services to end users or hosts services for his own development activities.