

# **FUJITSU Software ServerView Cloud Monitoring Manager V1.1**

A horizontal band featuring a red abstract graphic with flowing, curved lines and a bright light source, creating a sense of motion and energy.

## **OpenStack Operator's Guide**

J2UL-2075-02ENZ0(00)  
July 2016



# Contents

	<b>About this Manual.....</b>	<b>5</b>
<b>1</b>	<b>Introduction.....</b>	<b>8</b>
1.1	Basic Usage Scenario.....	9
1.2	The OpenStack Operator's Tasks.....	10
1.3	Components.....	11
1.4	User Management.....	12
<b>2</b>	<b>Installation.....</b>	<b>13</b>
2.1	Prerequisites and Preparation.....	14
2.1.1	Prerequisites.....	14
2.1.2	Control Machine.....	15
2.1.3	CMM Software and Setup Utilities.....	16
2.1.4	Security.....	17
2.1.5	HTTPS Support.....	18
2.1.6	Python Packages.....	18
2.1.7	Bash Script Failures.....	18
2.2	Configuring the Control Machine.....	18
2.3	Installing the OpenStack Extensions.....	20
2.4	Installing the CMM Metrics Agent.....	22
2.4.1	Installation.....	23
2.4.2	Configuration.....	25
2.5	Installing the CMM Log Agent.....	27
2.5.1	Installation.....	27
2.5.2	Configuration.....	30
2.6	Operation and Maintenance.....	31
2.6.1	Starting and Stopping an Agent.....	32
2.6.2	Disabling Metrics for a CMM Metrics Agent.....	32
2.6.3	Disabling Log Data for a CMM Log Agent.....	33
2.6.4	Collecting the Agent Log Files.....	33
2.6.5	Backup and Recovery.....	33
2.7	Uninstallation.....	34
<b>3</b>	<b>Preparation for Tenant Users.....</b>	<b>35</b>

---

3.1	Creating a Role and a User Account in OpenStack.....	35
3.2	Installing the CMM Metrics Agent.....	35
3.3	Installing Additional Libraries.....	35
3.4	Configuring the CMM Metrics Agent.....	36
4	Monitoring Services and Servers.....	37
4.1	Overview of Services and Servers.....	37
4.2	Working with Data Visualizations.....	37
4.3	Defining Alarms.....	38
4.4	Defining Notifications.....	39
5	Managing Log Data.....	40
5.1	Configuring an Index Pattern.....	40
5.2	Working with the Log Management Dashboard.....	41
Appendix A Configuration Settings.....		42
A.1	Configuration Settings for CMM Metrics Agents.....	42
A.2	Configuration Settings for CMM Log Agent.....	43
Appendix B Supported Metrics.....		45
B.1	Standard Metrics.....	45
B.2	Additional Metrics.....	46
Glossary	.....	51

## About this Manual

This manual describes how OpenStack operators can monitor their OpenStack platforms with FUJITSU ServerView Cloud Monitoring Manager - hereafter referred to as Cloud Monitoring Manager (CMM).

The manual is structured as follows:

Chapter	Description
<i>Introduction</i> on page 8	Introduces CMM and the basic usage scenario of monitoring an OpenStack platform.
<i>Installation</i> on page 13	Describes how to install the CMM components required for monitoring an OpenStack platform.
<i>Preparation for Tenant Users</i> on page 35	Describes how to prepare the monitoring environment for tenant users.
<i>Monitoring Services and Servers</i> on page 37	Describes the basic tasks involved in monitoring the OpenStack services and servers.
<i>Managing Log Data</i> on page 40	Describes the basic tasks involved in managing the log data from the OpenStack services and servers.
<i>Configuration Settings</i> on page 42	Describes the configuration settings for the CMM Metrics Agent and the CMM Log Agent.
<i>Supported Metrics</i> on page 45	Describes the metrics that are supported by CMM.
<i>Glossary</i> on page 51	Defines the central terms relevant for CMM.

## Readers of this Manual

This manual is directed to OpenStack operators who use CMM for monitoring their OpenStack platform.

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

## Notational Conventions

This manual uses the following notational conventions:

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

{one   two}	Mandatory entries with alternatives are enclosed in curly brackets.
-------------	---

## Abbreviations

This manual uses the following abbreviations:

<b>CMM</b>	Cloud Monitoring Manager
<b>IaaS</b>	Infrastructure as a Service
<b>ICMP</b>	Internet Control Message Protocol
<b>OS</b>	Operating System
<b>OSS</b>	Open Source Software
<b>PaaS</b>	Platform as a Service
<b>SaaS</b>	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.

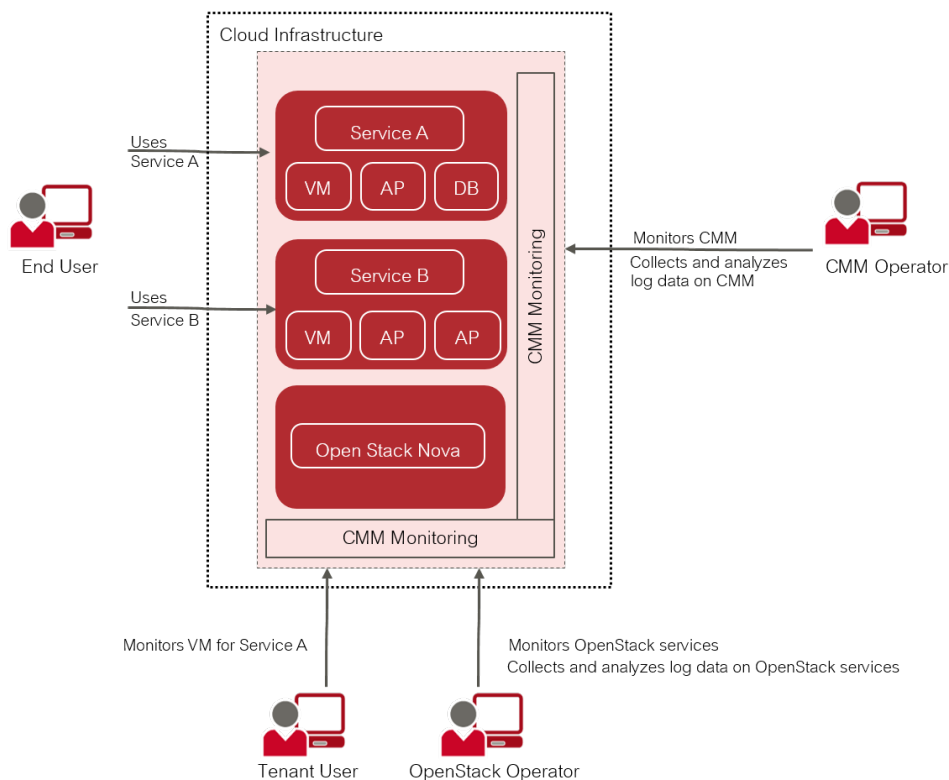
CMM covers all aspects of a Monitoring as a Service solution:

- 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.

As an **OpenStack operator**, you are a special tenant user who is responsible for administrating and maintaining the underlying OpenStack platform. The monitoring and log management services of CMM enable you to ensure the availability and quality of your platform. You use 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.

The **CMM operator** is responsible for providing the monitoring and log management services to the tenant users and the OpenStack operator. This enables them 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. The CMM operator uses the monitoring services himself for ensuring the quality of CMM.

## 1.2 The OpenStack Operator's Tasks

In order to use CMM for monitoring your OpenStack services and servers, you must install and configure so-called agents:

- A CMM Metrics Agent is required for monitoring your services and servers.
- A CMM Log Agent is required for collecting the log data that is generated for your services and servers.

### Monitoring

A CMM Metrics Agent can be installed on each virtual or physical server where services to be monitored are provisioned. The agent is responsible for querying metrics and sending them to the CMM Service for further processing.

Metrics are self-describing data structures that are uniquely identified by a name and a set of dimensions. 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 CMM Metrics Agent supports various types of metrics including the following:

- System metrics, for example, CPU usage, consumed disk space, or network traffic.
- Host alive checks. The agent can perform active checks on a host to determine whether it is alive using ping (ICMP) or SSH.
- Process checks. The agent can check and monitor a process, for example, the number of instances, memory size, or number of threads.
- HTTP endpoint checks. The agent can perform up/down checks on HTTP endpoints by sending an HTTP request and reporting success or failure to the CMM Service.
- Service checks. The agent can check middleware services, for example, MySQL, Kafka, or RabbitMQ.
- OpenStack services. The agent can perform specific checks on each process that is part of an OpenStack service.

Your individual agent configuration determines which metrics are available for monitoring your services and servers. For details on installing and configuring a CMM Metrics Agent, refer to *Installing the CMM Metrics Agent* on page 22.

As soon as an agent is available, you have access to the CMM monitoring features. You work with a graphical user interface that is seamlessly integrated into your cloud infrastructure. Based on OpenStack Horizon, the user interface enables access to all monitoring functionality and the resulting large-scale monitoring data. A comfortable dashboard visualizes the health and status of your cloud resources.

CMM provides functions for alarm and notification management. Template-based alarm definitions allow for monitoring a dynamically changing set of resources without the need for reconfiguration. While the number of underlying virtual machines is changing, for example, this ensures the efficient monitoring of scalable cloud services. Notifications can be configured in order to inform CMM users when an alarm is triggered.

For details on the monitoring functions, refer to *Monitoring Services and Servers* on page 37.

### Log Management

A CMM Log Agent can be installed and configured on each physical and virtual server where log data is to be retrieved. The agent collects the log data from the services and servers and sends

them to the CMM Service for further processing. For details on installing and configuring a CMM Log Agent, refer to *Installing the CMM Log Agent* on page 27.

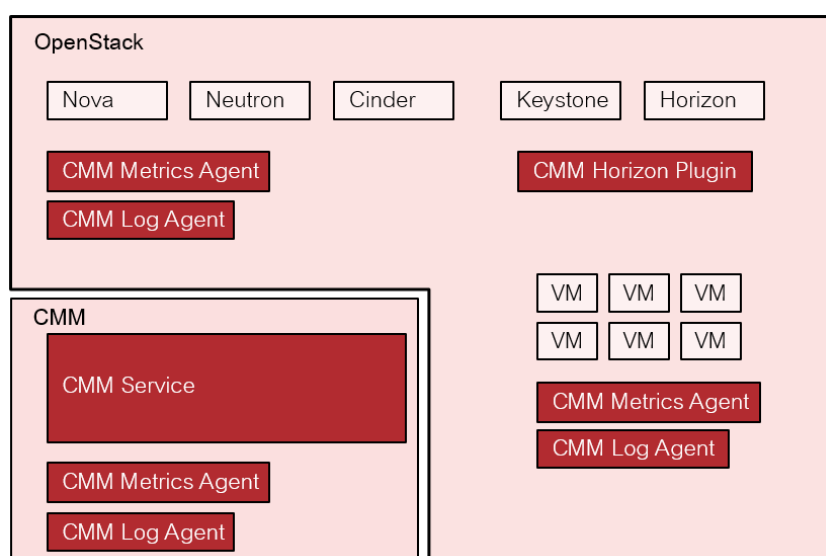
CMM stores the log data in a central database. This forms the basis for visualizing the log data for the CMM users. Advanced data analysis and visualization of the log data is supported in a variety of charts, tables, and maps. Visualizations can easily be combined in dynamic dashboards that display changes to search queries in real time.

Based on OpenStack Horizon, the customizable dashboards are seamlessly integrated into your cloud infrastructure. They enable user access to all log management functionality.

For details on the log management functions, refer to *Managing Log Data* on page 40.

## 1.3 Components

The following illustration provides an overview of the main components of CMM:



### 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 metrics and log data, as well as providing the data to the users.

The CMM Service relies on Monasca. It uses Monasca for high-speed metrics querying and integrates the streaming alarm engine and the notification engine of Monasca. For details, refer to the [Monasca Wiki](#).

## 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](#).

# 1.4 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 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.

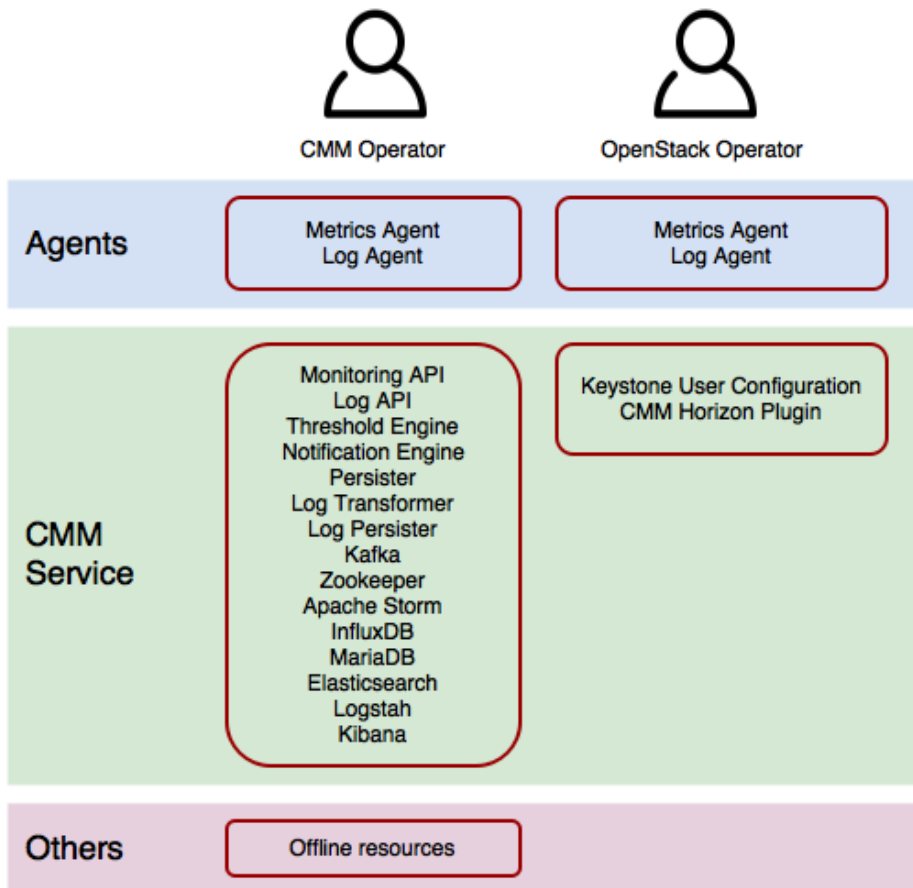
CMM users are created and administrated in OpenStack:

- 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](#).

## 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:

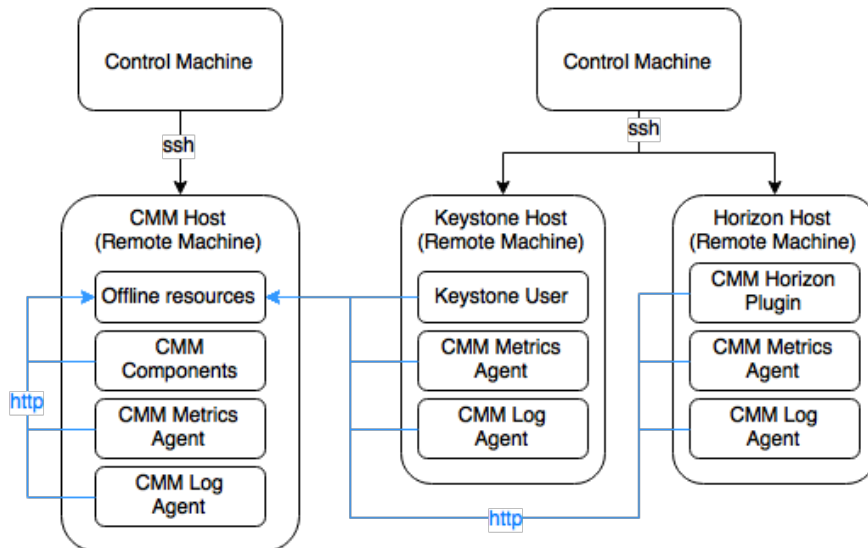
1. Preparing the installation environment - done by the OpenStack operator and the CMM operator.
2. Configuring the Control Machine - done by the OpenStack operator and the CMM operator.
3. Installing the offline resources - done by the CMM operator only.
4. Installing the OpenStack extensions required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

As an OpenStack operator, you also install and configure the agents for monitoring your OpenStack platform.

CMM ships with an installer that is based on Ansible. You use it for installing the OpenStack extensions and the required agents.

## Recommended Installation Environment

The following installation environment is recommended:



- By default, the CMM Operator installs the offline resources, the CMM components, and the required agents on one host (CMM host).
- As an OpenStack Operator, you install the OpenStack extensions and the required agents for the OpenStack Keystone service and the OpenStack Horizon service. Depending on your OpenStack environment, Keystone and Horizon can either be installed on the same or on two separate hosts.

You 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.

## 2.1 Prerequisites and Preparation

The following sections describe the prerequisites that must be fulfilled and the preparations you need to take before installing the OpenStack extensions and the required agents.

### 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

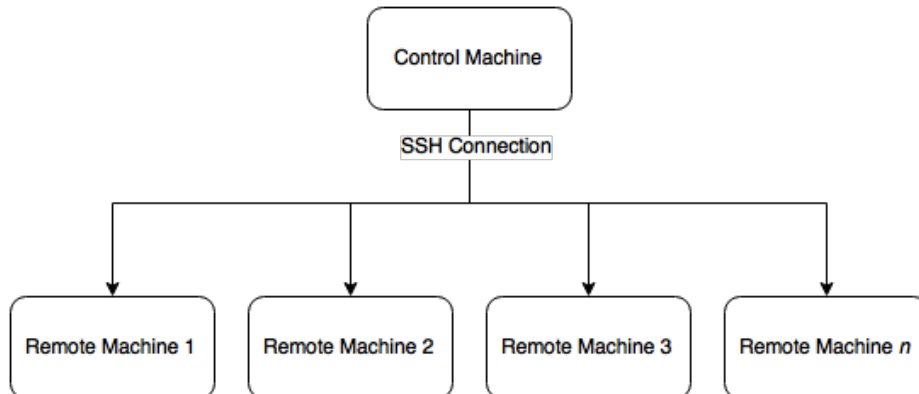
CMM supports the following Web browsers:

- 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:

```
wget \
https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

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/FJSVsvcomm` 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.
- `*.yaml` 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-os.j2
```

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

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

6. Replace the variables in every section that is defined:
  - `OpenStack Hosts` for installing the OpenStack extensions on the Horizon host and the Keystone host.
  - `Monasca Agent Hosts` for installing a CMM Metrics Agent on the Horizon host and the Keystone host.
  - `Monasca Log Agent Hosts` for installing a CMM Log Agent on the Horizon host and the Keystone host.

Replace `{{ openstack_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_openstack }}` 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 machines, generate the SSH key for Ansible.

Execute the following command:

```
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 remote machines, execute the following command. Repeat the command for each remote machine on which to install a software component.

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

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

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>'s password:
```

```
Number of key(s) added: 1

Now try logging into the machine, with:  "ssh
'<user_name>@<host_ip>'
and check to make sure that only the key(s) you wanted were added.
```

## 2.3 Installing the OpenStack Extensions

As a prerequisite for installing the OpenStack Extensions, the CMM operator must have installed the offline resources.

To install the OpenStack extensions for CMM, proceed as follows:

1. Log in to the Control Machine.
2. Configure the remote machines required for installing the OpenStack extensions. 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:

- `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. The installer creates OpenStack user accounts for working with CMM. For this purpose, specify your access credentials for the OpenStack Keystone service in the `Keystone Admin credentials` section:

Example:

```
# Keystone Admin credentials
keystone_admin_token: ADMIN
keystone_admin: 'admin'
keystone_admin_project: 'admin'
```

**Note:** If `ADMIN` authentication is disabled for the OpenStack Keystone service, `keystone_admin_token: ADMIN` must be removed.

5. Provide the passwords required for installing the OpenStack extensions. 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:

- `keystone_cmm_operator_user_password` for Keystone access of the CMM operator. The user account for the CMM operator is automatically created with the installation.
- `keystone_cmm_agent_password` for Keystone access of the user used for configuration purposes of the CMM Metrics Agent. The user account is automatically created with the installation.
- `keystone_admin_agent_password` for Keystone access of the user used for configuration purposes of the CMM Log Agent. The user account 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 the new user accounts with the installation. It is the user account of the OpenStack operator.

7. Specify your region name for the OpenStack Keystone service. The region name is defined by the `keystone_region` parameter in the `/opt/FJSVsvcmm/roles/monasca-keystone/defaults/main.yml` file.

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

Example:

```
sudo vim /opt/FJSVsvcmm/roles/monasca-keystone/defaults/main.yml
```

8. Adapt the `keystone_region` parameter in the file to your environment.

Example:

```
keystone_region: RegionOne
```

9. To install the OpenStack extensions, specific configuration settings are required. They are specified in the `group_vars/openstack_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/openstack_group
```

By default, the installer creates the following users, roles, and projects in OpenStack:

- A CMM operator who is to monitor CMM.  
User name: `cmm-operator`. Assigned user role: `cmm-user` and `admin`. Assigned OpenStack project: `cmm`.
- A user for configuration purposes of the CMM Metrics Agent. It is recommended that this user account is used only for the agent configuration and not for actually monitoring services and servers.  
User name: `cmm-agent`. Assigned user role: `cmm-agent`. Assigned OpenStack project: `cmm`.

- A user for configuration purposes of the CMM Log Agent. It is recommended that this user account is used only for the agent configuration and not for actually accessing the log management functions in CMM.

User name: `admin-agent`. Assigned user role: `cmm-agent`. Assigned OpenStack project: `admin`.

- The OpenStack operator who is to monitor the OpenStack platform uses the existing `admin` user account.

Assigned user role: `cmm-user`. Assigned OpenStack project: `admin`.

**Note:** The roles that are created by the installer must be authorized for accessing the CMM Service. If you change the default configuration of the user accounts, contact your CMM operator. He is responsible for configuring the CMM Service.

10. To install the OpenStack extensions, execute the following command:

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

The following response is displayed if the installation is successful.

Example:

```
PLAY RECAP *****
horizon-node      : ok=34    changed=21    unreachable=0    failed=0
keystone-node     : ok=21    changed=3     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/openstack.yml
```

As soon as the OpenStack extensions are successfully installed, the CMM operator can install the CMM Service.

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

## 2.4 Installing the CMM Metrics Agent

For monitoring the OpenStack platform, a CMM Metrics Agent must be installed on each OpenStack node you want to monitor.

As a prerequisite for installing an 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.

## 2.4.1 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 hosts on which you want to install an agent. For this purpose, check the following section in the configuration file:

```
# Monasca Agent Hosts
[monasca_agent_group]
horizon-monasca-agent          ansible_ssh_host=<horizon_host>
ansible_ssh_user=<ssh_user_monasca>
keystone-monasca-agent        ansible_ssh_host=<keystone_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 the host variables with the host name or IP address of the remote machines on which to install an 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 each agent. The configuration files 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 42.

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 the send monitoring data to the CMM Service.

Example:

```
# file: host_vars/horizon-monasca-agent

# Monasca Agent user credentials
monasca_agent_user: cmm-agent
monasca_agent_password:
  "{{ openstack_monasca_agent_keystone_password }}"
monasca_agent_project: cmm
```

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:

- `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.
- `cmm_monasca_agent_database_password` for agent access to the Maria DB database.
- `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.



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

```
ansible-playbook /opt/FJSVsvcomm/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/FJSVsvcomm/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.

## 2.4.2 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:

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

3. Open the file with your favorite editor.

Example:

```
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 42.

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

```
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 45.

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 45.

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:

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

3. Copy the required template file.

Example:

```
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:

```
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 45.
6. To start the agent again, execute the following command:

```
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.

## 2.5 Installing the CMM Log Agent

For monitoring the OpenStack platform, a CMM Log Agent must be installed on each OpenStack node you want to monitor.

As a prerequisite for installing an 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.

### 2.5.1 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 hosts on which you want to install an agent. For this purpose, check the following section in the configuration file:

```
# Monasca Log Agent Hosts
[monasca_log_agent_group]
horizon-monasca-log-agent
  ansible_ssh_host=<horizon_host> ansible_ssh_user=<ssh_user_monasca>
keystone-monasca-log-agent
  ansible_ssh_host=<keystone_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 the host variables with the host name or IP address of the remote machines on which to install an 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 each 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/FJSVsvcomm/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 43.

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.

Example:

```
# file: host_vars/cmm-monasca-log-agent

# logstash monasca user credentials
project_name: cmm
username: cmm-operator
password: "{{ openstack_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 30.

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:

- `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.
- `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.

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

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

## 2.5.2 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:

```
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 43.

Example configuration:

```
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.

## 2.6 Operation and Maintenance

Regular operation and maintenance includes:

- Starting and stopping the agents.
- 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.
- Collecting the agent log files.

- Backup and recovery.

## 2.6.1 Starting and Stopping an Agent

System operation and maintenance may require that an agent is manually stopped and restarted.

To start an agent, proceed as follows:

1. Log in to the server where the agent is installed.
2. To start a CMM Metrics Agent, execute the following command:

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

To start a CMM Log Agent, execute the following command:

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

To stop an agent, proceed as follows:

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

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

To stop a CMM Log Agent, execute the following command:

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

## 2.6.2 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
```



### 2.6.3 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
```

### 2.6.4 Collecting the Agent Log Files

The agent log files record the events that occur in operating the agent. CMM provides an individual log file for each agent component. The log files provide important information, for example, for trouble-shooting.

The log files of the CMM Metrics Agent are written to the following directory:

```
/var/log/monasca/agent
```

The log files of the CMM Log Agent are written to the following directory:

```
/var/log/monasca/monasca-log-agent
```

### 2.6.5 Backup and Recovery

CMM does not offer integrated backup and recovery mechanisms. Use the standard file system mechanisms instead.

It is recommended that you create a regular backup of the configuration files:

- The configuration files of the CMM Metrics Agent located in the `/etc/monasca/agent` directory.
- The configuration file of the CMM Log Agent located in the `/etc/monasca/monasca-log-agent` directory.

The frequency of backups depends on the amount of configuration changes.

If you need to recover your agent installation, the recommended procedure is as follows:

1. Uninstall the agent. For details, refer to *Uninstallation* on page 34.
2. Reinstall the agent. For details on the CMM Metrics Agent, refer to *Installation* on page 22. For details on the CMM Log Agent, refer to *Installation* on page 27.
3. Stop the agent.
4. Copy the backup of your configuration files to the `/etc/monasca/agent` or the `/etc/monasca/monasca-log-agent` directory, respectively.
5. Start the agent again.

## 2.7 Uninstallation

As an OpenStack operator you are responsible for uninstalling CMM agents, if required. The OpenStack extensions you have installed for your platform are uninstalled by the CMM operator who is responsible for uninstalling the CMM Service.

Before uninstalling a CMM agent, it is recommended that you make a backup of the configuration files created for operation. For details on backups, refer to *Backup and Recovery* on page 33.

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. Note that 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 agent 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.

## 3 Preparation for Tenant Users

Tenant users who have booked a virtual machine in OpenStack can monitor their machine with libvirt. Libvirt provides a toolkit for managing virtual machines.

As an OpenStack operator, you have to prepare the monitoring environment for the tenant users. The following steps are required:

1. Create a role and a user account for libvirt monitoring in OpenStack.
2. Install a CMM Metrics Agent on each hypervisor where the virtual machines are provisioned for the tenant users.
3. Install additional libraries for libvirt monitoring.
4. Configure the agent.

### 3.1 Creating a Role and a User Account in OpenStack

As a prerequisite for installing a CMM Metrics Agent, you need to take the following preparations:

- Create the `monitoring-delegate` role in OpenStack Keystone.  
This role is required for cross-tenant metrics submission. A tenant user who has booked a virtual machine must receive only the monitoring data related to his virtual machine. This role enables the agent to submit metrics on behalf of an individual tenant user.
- Create a user account for configuring the agent. The user must be assigned the `monitoring-delegate` role.  
The user name and password of this user will be specified in the agent configuration when the agent is installed. It is recommended that this user account is used only for configuration purposes and not for actually monitoring the services.

### 3.2 Installing the CMM Metrics Agent

To enable monitoring for tenant users, a CMM Metrics Agent must be installed on the hypervisor that hosts the virtual machines of the tenant users.

Enter the credentials of the OpenStack user you have created for libvirt monitoring in the agent configuration. This user is used for the communication between the CMM Service and the agent.

For details on installing a CMM Metrics Agent, refer to *Installation* on page 22.

### 3.3 Installing Additional Libraries

Additional RPM packages and Python libraries are required for libvirt monitoring.

Install the following RPM package from your YUM repository server:

- `libvirt-devel`

Install the following Python libraries in the virtualenv environment of the CMM Metrics Agent:

- `libvirt-python`
- `lxml`
- `python-neutronclient`
- `python-novaclient`. Version 2.26 is required.

By default, the virtualenv environment of the CMM Metrics Agent is located in the `/opt/monasca-agent` directory.

## 3.4 Configuring the CMM Metrics Agent

The installation of the CMM Metrics Agent includes its initial configuration. To prepare the environment for tenant users, you have to reconfigure the agent. It is necessary to explicitly activate the libvirt metrics that the tenant users use for monitoring. 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.

To activate the metrics, refer to *Configuration* on page 25. For an example configuration, refer to *Additional Metrics* on page 45.

As soon as the agent is started again, the tenant users can use the libvirt metrics for monitoring.

## 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 37.

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 38.

### 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`.

---

**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 *Defining Notifications* on page 39.

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.



- **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`

**Note:** The `dimensions` parameter 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.

## 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 OpenStack operator, the following metrics are of relevance for monitoring the servers and services of your OpenStack platform:

- All the standard metrics.
- The following additional metrics: `apache`, `host_alive`, `http_check`, `libvirt`, `mysql`, `ntp`, `postfix`, `process`, `rabbitmq`.

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:

```
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:

```
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:

```
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:

```

# 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']

```

```
'thresh-event':
  'events': ['3', '2', '1', '0']
'tresh-metric':
  'metrics': *id001
kafka_connect_str: localhost:9092
zk_connect_str: localhost:2181
```

### 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:

```
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:

```
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:

```
init_config:

instances:
```



```
- 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:

```
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:

```
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:

```
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.