FUJITSU

FUJITSU Software Cloud Monitoring Manager V2.0.14

Application Operator`s Guide

July 2021

Contents

About this Manual **2**

1 Introduction **5**

1.1 Basic Usage Scenario 6

1.2 The Application Operator's Tasks 7

1.3 Components 9

1.4 User Management 10

2 Accessing CMM **11**

3 Monitoring Services and Virtual Machines **12**

3.1 Overview 12

3.2 Viewing Metrics Data 12

3.3 Defining Alarms 13

3.4 Defining Notifications 16

3.5 Status of Services and Virtual Machines 17

Glossary **18**

About this Manual

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

The manual is structured as follows:

Chapter	Description
Introduction to CMM	Introduces CMM, its architecture and users.
Installation	Describes how to install and configure CMM.
Preparations for Application Operators	Describes how to prepare the monitoring environment for application operators.
Operation and Maintenance	Describes the main operation and maintenance tasks for CMM.
Monitoring	Describes the basic tasks involved in monitoring services and servers.
Log Management	Describes the basic tasks involved in managing the log data from the services and servers.
Glossary	Defines the central terms relevant for CMM.

Readers of this Manual

This manual is written for operators who install, operate, and maintain CMM. It also describes how the operators use CMM for monitoring and log management. The manual assumes that you have profound knowledge of OpenStack and CMM, especially the individual services CMM is composed of. For installing the CMM components, you must be familiar with the administration and operation of LINUX systems.

Notational Conventions

This manual uses the following notational conventions:

Notation	Description
`Add`	Names of graphical user interface elements.
`init`	System names, for example command names and text that is entered from the keyboard.
`<variable>`	Variables for which values must be entered.
`[option]`	Optional items, for example optional command parameters.
`one | two`	Alternative entries.
`{one | two}`	Mandatory entries with alternatives.

Abbreviations

This manual uses the following abbreviations:

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

Available Documentation

The following documentation on CMM is available:

- *Overview:* A manual introducing CMM. It is written for everybody interested in CMM.
- *System Operator's Guide:* A manual for system operators describing how to install, operate, and maintain CMM. The manual also describes how to prepare the OpenStack platform for CMM and how to use the CMM monitoring functions.
- *Application Operator's Guide:* A manual for application operators describing how CMM supports them in monitoring their services and virtual machines in OpenStack.

Related Information

The following links 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* : Information on Monasca, the core of CMM.
- *Grafana* : Documentation on Grafana, the open-source application used for visualizing metrics data.
- *Kibana* : Documentation on Kibana, the open-source application used for visualizing log data.

Links to more detailed information provided in this manual are subject to change without notice.

Trademarks

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

Other company names and product names are trademarks or registered trademarks of their respective owners.

High Risk Activity

The Customer acknowledges and agrees that the Product is designed, developed and manufactured as contemplated for general use, including without limitation, general office use, personal use, household use, and ordinary industrial use, but is not designed, developed and manufactured as contemplated for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could lead directly to death, personal injury, severe physical damage or other loss (hereinafter “High Safety Required Use”), including without limitation, nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system. The Customer shall not use the Product without securing the sufficient safety required for the High Safety Required Use. In addition, FUJITSU (or other affiliate’s name) shall not be liable against the Customer and/or any third party for any claims or damages arising in connection with the High Safety Required Use of the Product.

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

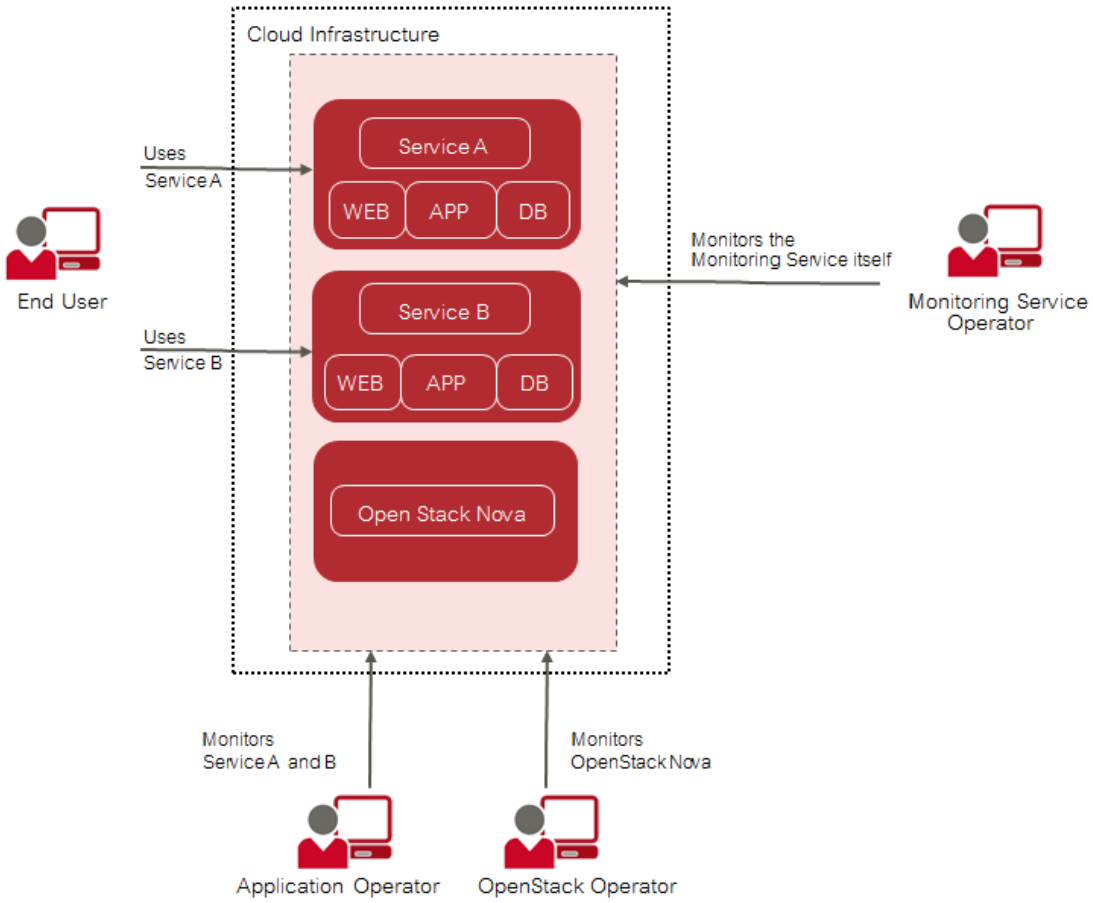
As more and more applications are deployed on cloud systems and cloud systems are growing in complexity, managing the cloud infrastructure is becoming increasingly difficult. Cloud Monitoring Manager (CMM) helps mastering this challenge by providing a sophisticated Monitoring as a Service solution that is operated on top of OpenStack-based cloud computing platforms. CMM addresses the requirements of large-scale public and private clouds where high numbers of physical and virtual servers need to be monitored and huge amounts of monitoring data need to be managed. It consolidates metrics, alarms, and notifications, as well as health and status information from multiple systems, thus reducing the complexity and allowing for a higher-level analysis of the monitoring data.

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

- Storage of monitoring data in a resilient way.
- Multi-tenancy architecture for submitting and streaming metrics. The architecture ensures the secure isolation of metrics data.
- 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 solution can be adapted accordingly.

1.1 Basic Usage Scenario

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



Basic Usage Scenario

As an **application operator** , you have booked virtual machines in OpenStack to provide services to end users or to host services that you need for your own development activities. CMM helps you ensure that the virtual machines on which these services are provided are working as required.

The **OpenStack operator** uses CMM to monitor physical and virtual servers, hypervisors, and services of the underlying platform. In addition, an OpenStack operator is responsible for the middleware components, for example the database services. The OpenStack operator also prepares the monitoring environment including available metrics for application operators.

The **Monitoring Service operator** is responsible for providing the cloud monitoring features to the application operators and the OpenStack operator. This enables the application operators and the OpenStack operator to focus on the operation and quality of their services without having to carry out the tedious tasks implied by setting up and administrating their own monitoring software. The Monitoring Service operator uses the monitoring features for monitoring the operation of CMM.

1.2 The Application Operator's Tasks

The OpenStack operator is responsible for preparing your monitoring environment and activates the metrics that you can use for monitoring the virtual machines you have booked.

Metrics

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 standard metrics include:

- Metrics on CPU usage, for example 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.
- Metrics on disk space, for example 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.
- Metrics on the average system load over different periods, for example 1 minute, 5 minutes, or 15 minutes.
- Metrics on memory usage, for example the number of megabytes of total memory or free memory, or the percentage of free swap memory.
- Metrics on the network, for example the number of network bytes received or sent per second, or the number of network errors on incoming or outgoing network traffic per second.

Monitoring Functions

CMM provides a graphical user interface that is seamlessly integrated into your cloud infrastructure. Based on OpenStack Horizon, it visualizes the health and status of your virtual machines and enables access to all monitoring functionality and the resulting large-scale monitoring data.

A convenient dashboard visualizes the health and status of your virtual machines. It allows you to experiment with many ways of analyzing the performance of your resources in real-time. You cannot only view but also share and explore visualizations of your monitoring data. For monitoring your virtual machines, 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. This ensures the efficient monitoring of scalable cloud services.

Alarm definitions allow you to specify expressions that are evaluated based on the metrics data that is received. Notifications can be configured in order to inform CMM users when an alarm is triggered.

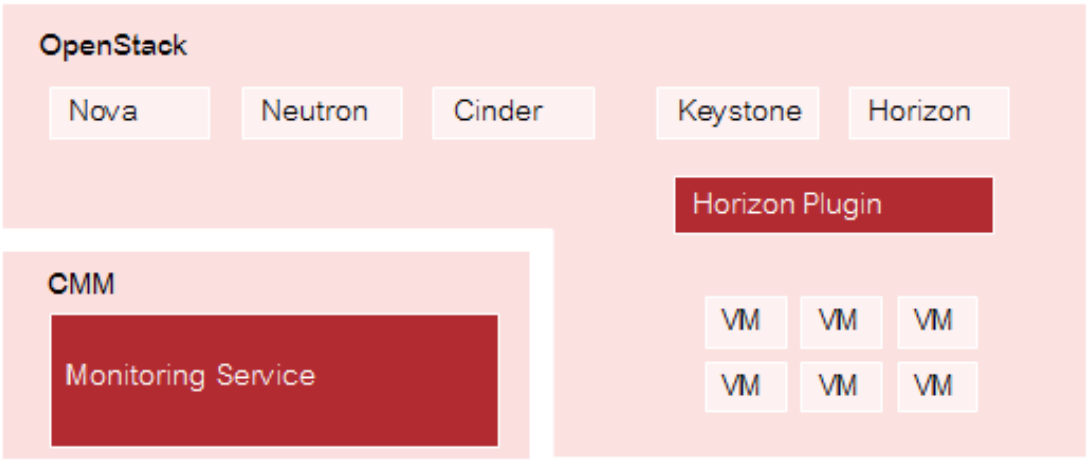
As an application operator, you:

- Build, explore, and share visualizations of your monitoring data.
- Create, update, and delete alarms.
- Create, update, and delete notifications.

For details on the monitoring functions, refer to *Monitoring Services and Virtual Machines* on page 12.

1.3 Components

The following illustration provides an overview of the CMM components:



Components

OpenStack

CMM relies on OpenStack, a 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.

Monitoring Service

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

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. Monitoring data is displayed on a convenient and easy-to-use dashboard which fully integrates with Grafana, an open-source application for visualizing large-scale measurement data.

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 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. Application operators 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 Accessing CMM

As an application operator, you must fulfill the following prerequisites:

- You must have access to the OpenStack platform as a user with the `monasca-user` role or any other role that is authorized to use the CMM monitoring functions. Additional roles are optional.
- You must be assigned to the OpenStack project whose resources you want to monitor.

Log in to OpenStack Horizon with your user name and password provided as login information by OpenStack. For monitoring your services and virtual machines, you use the CMM monitoring user interface which is integrated into OpenStack Horizon. The CMM functionality is available on the **Monitoring** tab. It provides access to the monitoring data of all projects to which you are assigned. Before you start, select the project you want to work on.

Web Browsers

CMM has been tested with the following Web browsers:

- Google Chrome 90.
- Microsoft Edge 91.
- Mozilla Firefox 58.

Notes:

- Mozilla Firefox 58:

  When accessing Grafana, a message is displayed: '`Your browser is not fully supported. A newer Browser version is recommended`'. This browser version has been extensively tested. There is no known restriction when using Firefox 58 to access CMM metric information via Grafana.

- All Browsers:

  When accessing Kibana, a message is displayed: '`Your browser doesn't meet the security requirements for Kibana`'. However, this message is not related to browser versions. Security in CMM is ensured: Only users with access to Horizon can successfully use Kibana to access CMM log information. Thus, this message can be safely ignored.

3 Monitoring Services and Virtual Machines

CMM offers various features for monitoring your services and the virtual machines on which they are provisioned. The available metrics in combination with early warnings about problems and outages assist you in analyzing and troubleshooting any issue you encounter in your environment. The monitoring features of CMM include:

- A monitoring overview which allows you to access all monitoring information.
- Metrics dashboards for visualizing your monitoring data.
- Alerting features.

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

3.1 Overview

CMM provides one convenient access point to your monitoring data. Use **Monitoring > Overview** to keep track of your services and virtual machines and quickly check their status. On the **Overview** page, you can:

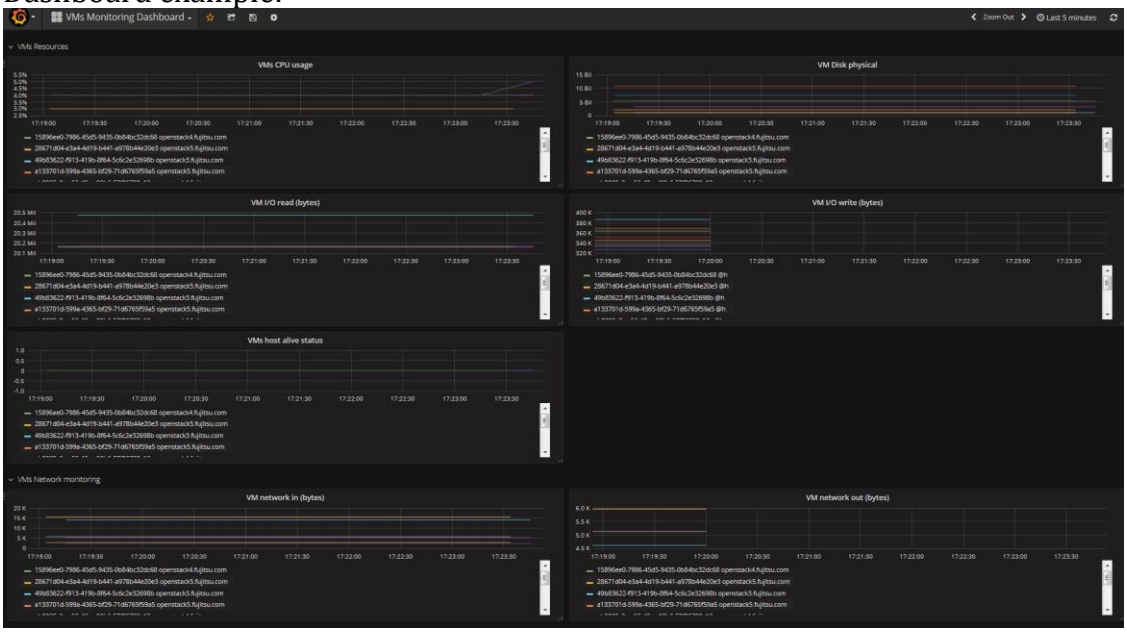
- View the status of your resources at a glance. As soon as you have defined an alarm for a service or virtual machine and metrics data has been received, relevant status information is displayed on the **Overview** page. For details on the status information, refer to *Status of Services and Virtual Machines* on page 17. For details on defining alarms, refer to *Defining Alarms* on page 13.
- Access a metrics dashboard. For details, refer to next section *Viewing Metrics Data*.

3.2 Viewing Metrics Data

The user interface for monitoring your services and virtual machines integrates with Grafana, an open-source application for visualizing large-scale metrics data. CMM ships with a preconfigured dashboard that visualizes the monitoring data collected from your services and virtual machines.

To access the dashboard, use the **Grafana Home** option located along the top of the **Overview** page.

Dashboard example:



3.3 Defining Alarms

You have to define alarms to monitor your resources. An alarm definition specifies the metrics to be collected and the threshold at which an alarm is to be triggered for a resource. By default, an alarm definition is evaluated over a succession of one-minute periods. If the specified threshold is reached or exceeded within one such period, the alarm is triggered and notifications can be sent to inform users. The alarm definition is re-evaluated in each subsequent time period.

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

The elements that define an alarm are grouped into **Details**, **Expression** and **Notifications**. They are described in the following sections.

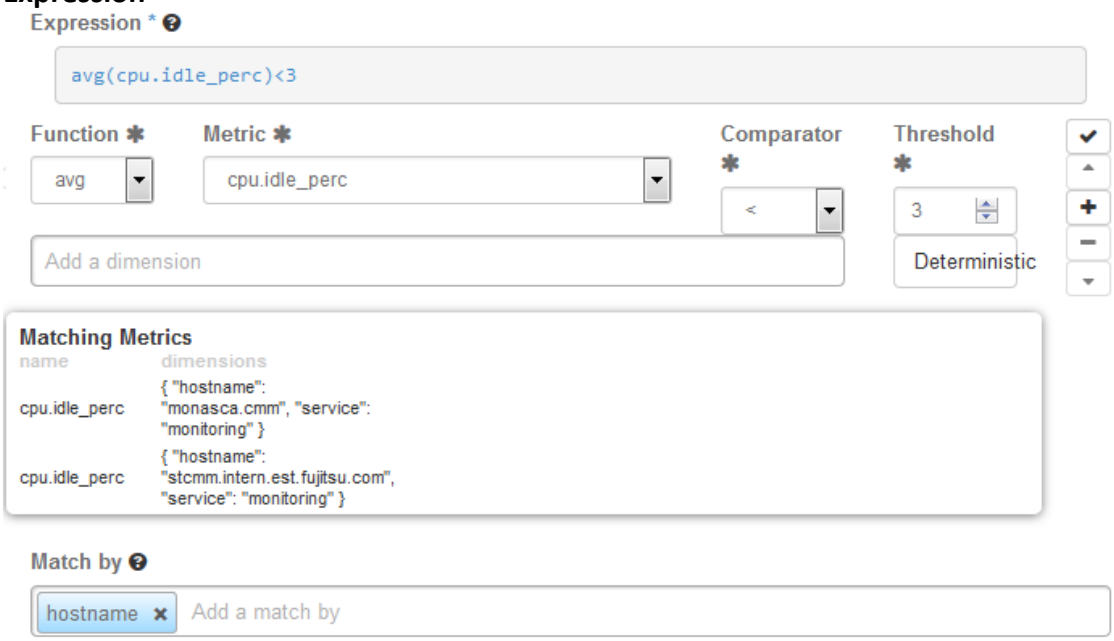
Details

Details

For an alarm definition, you specify the following details:

- **Name**. Mandatory identifier of the alarm. The name must be unique within the project for which you define the alarm.
- **Description**. Optional. A short description that outlines the purpose of the alarm.
- **Severity**. The following severities for an alarm are supported: **Low** (default), **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. The severity level is subjective. Choose a level that is appropriate for prioritizing the alarms in your environment.

Expression



Expression

The expression defines how to evaluate the metrics. The expression syntax is based on a simple expressive grammar. For details, refer to the *Monasca API documentation*.

To handle a large variety of monitoring requirements, you can create either simple alarm definitions that refer to one metrics only, or compound alarm definitions that combine multiple metrics in one expression.

Example for a simple alarm definition that checks whether the system-level load of the CPU exceeds a threshold of 90 percent:

```
cpu.system_perc{hostname=monasca} > 90
```

Example for a simple alarm definition that checks the average time of the system-level load of the CPU over a period of 480 seconds. The alarm is triggered only if this average is greater than 95 percent:

```
avg(cpu.system_perc{hostname=monasca}, 120) > 95 times 4
```

Example for a compound alarm definition that evaluates two metrics. The alarm is triggered if either the system-level load of the CPU exceeds a threshold of 90 percent, or if the disk space that is used by the specified service exceeds a threshold of 90 percent:

```
avg(cpu.system_perc{hostname=monasca}) > 90 OR
max(disk.space_used_perc{service=monitoring}) > 90
```

To define an alarm expression, proceed as follows:

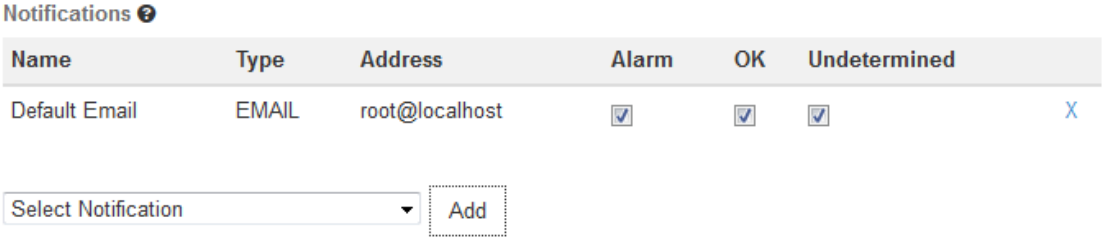
1. Select the metrics to be evaluated.
2. Select a statistical function for the metrics: `min` to monitor the minimum values, `max` to monitor the maximum values, `sum` to monitor the sum of the values, `count` for the monitored number, `avg` for the arithmetic average, or `last` for the most recent value.
3. Enter one or multiple dimensions in the **Add a dimension** field to further qualify the metrics. Dimensions filter the data to be monitored. They narrow down the evaluation to specific entities. 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 a dimension** field. You are offered a selection 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.idle_perc` or MB for `disk.total_used_space_mb`.
5. Switch on the **Deterministic** option if you evaluate a metrics for which data is received only sporadically. The option should be switched on, for example, for all log metrics. This ensures that the alarm status is `OK` and displayed as a green box on the **Overview** page although metrics data has not yet been received. Do not switch on the option if you evaluate a metrics for which data is received regularly. This ensures that you instantly notice, for example, that a host machine is offline and that there is no metrics data for the agent to collect. On the **Overview** page, the alarm status therefore changes from OK to UNDETERMINED and is displayed as a white box.
6. Enter one or multiple dimensions in the **Match by** field if you want these dimensions to be taken into account for triggering alarms. Example: If you enter `hostname` as a dimension, individual alarms will be created for each host machine on which metrics data is collected. The expression you have defined is not evaluated as a whole but individually for each host machine in your environment. If **Match by** is set to a dimension, the number of alarms depends on the number of dimension values on which metrics data is received. An empty **Match by** field results in exactly one alarm. To enter a dimension, you can simply type the name of the dimension in the **Match by** field. The dimensions you enter cannot be changed once the alarm definition is saved.
7. Build a compound alarm definition to combine multiple metrics in one expression. Using the logical operators `AND` or `OR`, any number of expressions can be combined. Use the **Add** button to append an expression, and choose either AND or OR as the **Operator** to connect it to the one you have already defined. Proceed with the second expression as described in

Step 1 to Step 6 above. The following options are provided for creating and organizing compound alarm definitions:

- Append an additional expression using the **Add** button.
- Finish editing an appended expression using the **Submit** button.
- Delete an appended expression using the **Remove** button.
- Change the position of an appended expression using the **Up** or **Down** button.

Notes: * `Function`, `Metric`, `Comparator` and `Threshold` are required fields and must not be empty. * You can also edit the expression syntax directly. For this purpose, save your alarm definition and update it using the **Edit Alarm Definition option**. When updating the alarm definition, you can also change the default time period for each alarm definition evaluation. For syntax details, refer to the Monasca API documentation on *Alarm Definition Expressions*. * Restrictions apply to usage of time/times and compound alarm definitions. Please refer to *Release Notes* for details.

Notifications



Notifications

You can enable notifications for an alarm definition. As soon as an alarm is triggered, the enabled notifications will be sent.

The **Notifications** tab allows you to select the notifications from the ones that are defined in your environment. For a selected notification, you specify whether you want to send it for a status transition to **Alarm** , **OK** , and/or **Undetermined**.

For details on defining notifications, refer to *Defining Notifications*. For details on alarm statuses, refer to *Status of Services and Virtual Machines*.

3.4 Defining Notifications

Notifications define how users are informed when a threshold value defined for an alarm is reached or exceeded. In an 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**. The notification method to be used. `Email`, `Pagerduty`, or `WebHook` can be selected provided that the methods were enabled when installing the Monitoring Service.
- **Address**. For `Email`, 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.company`).

- **Address**. For `WebHook`, the WebHook URL to be loaded when an alarm is triggered.
- **Period**. For `WebHook` only. The integer value indicates how often a notification is to be resent. To create, edit, and delete notifications, use **Monitoring > Notifications**.

3.5 Status of Services and Virtual Machines

The following alarm statuses are distinguished when an alarm definition has been evaluated:

- `Alarm`. The alarm expression has evaluated to true. An alarm has been triggered for the cloud resource.
- `OK`. The alarm expression has evaluated to false. There is no need to trigger an alarm.
- `Undetermined`. No metrics data has been received within the defined time period.

As soon as you have defined an alarm for a resource, status information is displayed for it on the **Overview** page:

The color of the boxes in the three sections indicates the status:

- A green box for a service or virtual machine indicates that it is up and running. There are alarms defined for it, but no alarms have been triggered.
- A red box for a service or virtual machine indicates that there is a severe problem that needs to be checked. One or multiple alarms defined for it have been triggered.
- A yellow box indicates a problem. One or multiple alarms have already been triggered, however, the severity of these alarms is low.
- A white box indicates that though alarms have indeed been defined, metrics data has not yet been received.

The status information on the **Overview** page results from one or multiple alarms that have been defined for the corresponding resource. If multiple alarms are defined, the severity of the individual alarms controls the status color.

You can click a resource on the **Overview** page to display details on the related alarms. The details include the status of each alarm and the expression that is evaluated. For each alarm, you can drill down into the alarm history. To narrow down the problem, the history presents detailed information on the status transitions.

Glossary

Application Operator

A person responsible for providing services to end users or hosting services for development activities. An application operator has limited access to cloud resources in OpenStack.

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 monitoring data.

InfluxDB

An open-source time-series database that supports high write loads and large data set storage. CMM uses InfluxDB as the underlying technology for storing metrics and the alarm history.

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 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.

Monitoring Service Operator

A person responsible for maintaining and administrating CMM.

MySQL

An open-source relational database that provides an SQL-compliant interface for accessing data. CMM uses MySQL as the underlying technology for storing configuration information, alarm definitions, and notification methods.

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.