
Dive deep into Ceilosca

TECHNICAL REPORT

KU-OSLAB-2018-02

FEB 08, 2018

Yoonjin Chung

Chung1685@naver.com

OPERATING SYSTEMS LABORATORY

KOREA UNIVERSITY

PAGE 1 / 9

ABSTRACT

본 문서는 오픈스택 컴포넌트인 Ceilosca 의 필요성과 세부 구조 및 동작방식을 기술하고 있는 문서이다. 오픈스택의 기존 성능 모니터링 컴포넌트인 Ceilometer 와 Monasca 와 그 두개의 컴포넌트를 연동한 새로운 컴포넌트인 Ceilosca 의 구조 분석을 목표로 한다.

CONTENTS

목차

1. Motivation.....	4
1.1. Needs.....	4
1.2. Ideas.....	4
2. Components.....	5
2.1. Publisher	5
2.2. Storage driver	5
3. Architecture.....	6
3.1. Overview	6
3.2. Specific structure.....	6

1. Motivation

1.1. Needs

- 기존의 Ceilometer 는 오픈스택 서버 내의 utilization 이나 capacity 등 현재 사용량에 대한 현재 Monasca 보다 다양한 metrics 를 수집할 수 있으나, 단순 checking 외에는 특별한 서비스가 제공되지 않았음. 무엇보다, Ceilometer Agent 와 API, Database 사이의 communication 이 모두 RabbitMQ 위에서 이루어졌기 때문에 오픈스택 서비스 전체의 성능 저하를 불러 일으키는 단점을 가지고 있었음. (물론 구조가 단순하고, 사용하고 있는 MongoDB 스토리지 역시 단순 명확하다는 점에서 장점은 있음.)
- 기존의 Monasca 는 그에 비해 현재 자원의 상태를 살펴보는 것뿐만 아니라 alarm/thresholding engine 등을 제공하면서 특정 자원의 상태가 설정된 threshold 를 뛰어 넘으면 alarm 을 발생 시키도록 설계되었음. 또한 multi-tenant monitoring 서비스 제공으로 인해 많은 플러그 인들이 함께 동작할 수 있기 때문에, Monasca 는 하이퍼바이저의 내부에서 일어나는 일들에 확실히 Ceilometer 보다 더 나은 모니터링을 제공하는 데 성공했다고 볼 수 있음.
- 뿐만 아니라, Ceilometer 가 RabbitMQ 를 그냥 사용했던 데에 비해, Monasca 는 자체 메시지 큐인 Kafka 를 사용하여 더 scalable 하고 빠른 처리 속도를 보장함. Database 역시 MongoDB 가 아닌 Influx DB 를 사용하여 저장 시간을 단축시키고 Batching (일괄처리)까지 가능하도록 했음. (Libvirt 가 Monasca 를 활용한 예시 중 하나임.)

1.2. Ideas

- Monasca 는 이러한 improvement 에도 불구하고, Ceilometer 에 비해 수집하는 metrics 의 종류가 다양하지 않고, 정확하지 않았음.
- 따라서 Ceilometer 의 metrics 수집력과 Monasca 의 scalability, performance and monitoring service 를 결합하자는 아이디어가 자연스레 등장하였고, CISCO 팀에 의해 Ceilosca 가 개발되었음.

2. Components

2.1. Publisher

- Publisher 는 Ceilometer Agent 가 수집한 resource data 를 Monasca API 에 publishing 할 수 있도록 Interface 를 제공하는 컴포넌트임.
- 이 publisher 를 통해 Ceilometer Agent 는 기존의 RabbitMQ 를 거치지 않고, Monasca API 에 바로 data 를 forwarding 하게 되므로, 기존 Ceilometer 의 문제점이었던 전체 서비스 성능 저하 문제를 보완함.
- 정리하자면, Publisher 의 핵심 목적은 Ceilometer Agent 가 RabbitMq 를 사용하지 대신 Monasca API 에 metrics data 를 직접 전달할 수 있도록 Interface 를 제공함에 있음. 이후, Monasca API 로 전달된 data 들은 Monasca 내부의 메세지 큐인 Kafka 위에서 다뤄지기 때문에 기존 Ceilometer 의 오버헤드가 약 2~3 배 줄어듦.

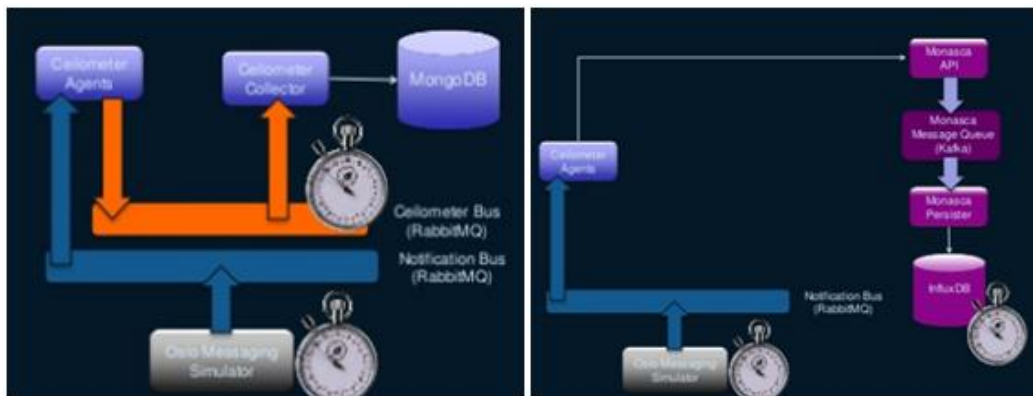


Figure 1. Flow of saving time series 비교. 왼쪽부터 Ceilometer, Ceilosca

2.2. Storage driver

- 기존의 Ceilometer storage driver 를 기반으로 구현한 드라이버로, write/read 요청을 Monasca-API 로 re-direct 해주는 역할을 함.
- 이때, Monasca API 는 REST based storage layer 로서 취급됨. 또한, Ceilometer-Monasca Driver 는 Monasca API 의 data structure 와 Ceilometer API 의 data structure 를 mapping 해주는 역할도 하여 두 개의 서로 다른 API 가 특별한 변화 없이 소통할 수 있도록 함.
- 이로 인해, 기존의 Ceilometer 를 사용하던 유저라면 굳이 Ceilometer 의 내용을 변경하지 않고도 Ceilosca 를 사용할 수 있음.

3. Architecture

3.1. Overview

- In a nutshell, Ceilosca is Ceilometer built on top of Monasca. Monasca 를 storage layer 로서 활용한 Ceilometer. 즉 기존의 Ceilometer 구조에서 MongoDB 의 위치에 Monasca API 를 끼워넣은 구조이며, RabbitMQ 를 거치지 않고 바로 data 를 전송함.

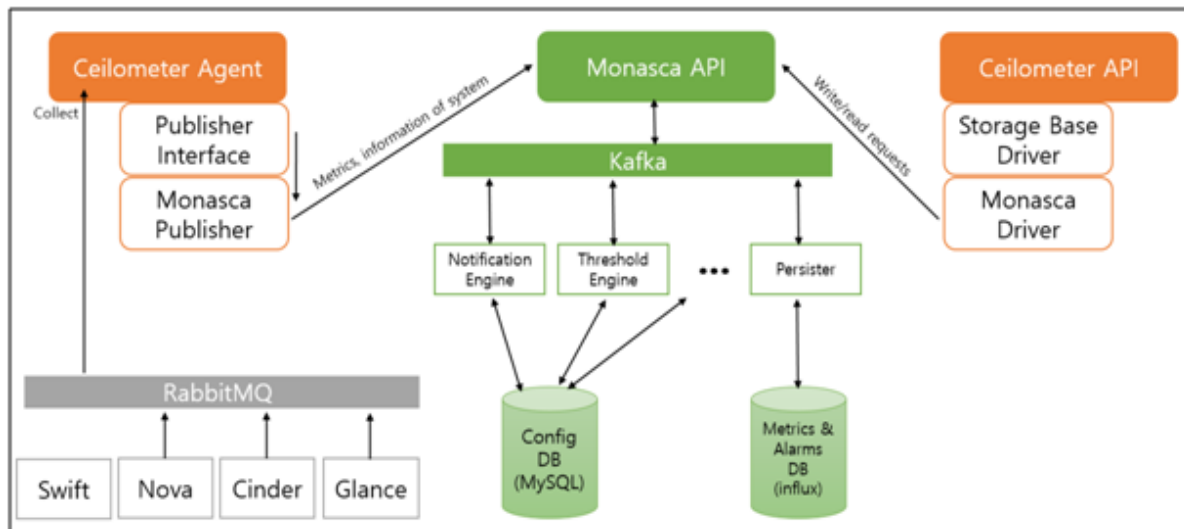


Figure 2. Ceilosca architecture

3.2. Specific structure

- Ceilometer

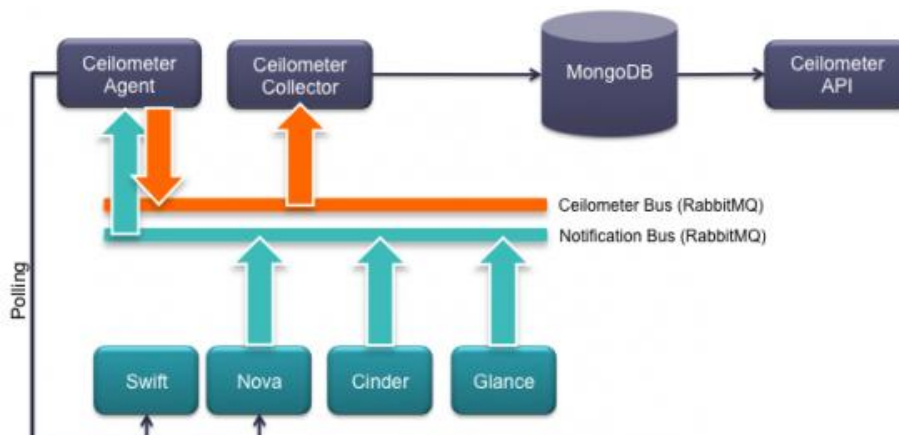


Figure 3. Ceilometer structure

- Main agent:
 - Notification, a.k.a Push model agent

- Central/Compute, a.k.a Pull model agent specification 에서 필수(Required)로 지정된 사항은 없음.
- 이 두 모델 모두 Monasca 와의 통합에 포함되어있음.
- Compute Agent 는 아마도 libvirt 또는 다른 가상화 계층에서 폴링 할 수있는 Monasca Agent 와 가장 중복되는 부분 일 것임

- Monasca

Monasca Architecture

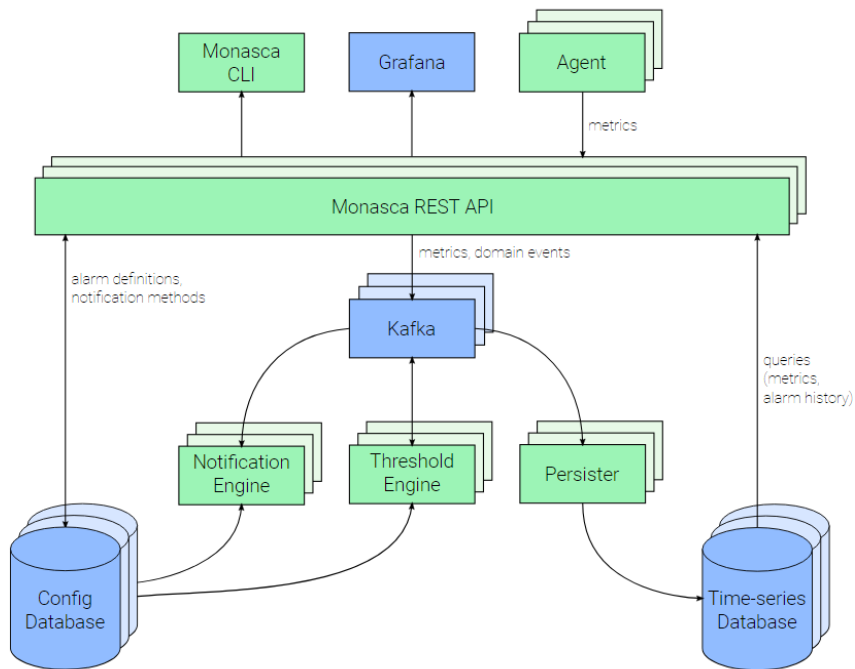


Figure 4. Monasca structure

- Monasca Agent: 모니터링 에이전트. 파이썬 베이스 모니터링 툴. 여러개의 sub-components 로 이루어져 있으며 cpu utilization, available memory, status 등등 여러 오픈스택 서비스들에 대한 metrics 를 POST 함.
 - Ceilosca 에서 이 컴포넌트를 Ceilometer 로 교체한 것이라고 볼 수 있음.
- Monasca API: 여러 서비스들, 요소들을 다루는 모니터링 API(RESTful API for monitoring tool).
 - Metrics: 방대한 양의 metrics 를 저장하고 쿼리를 전송.
 - Statistics: metrics 통계에 대한 query.
 - Alarm Definitions: Alarm 에 대한 정의를 create, update, delete and query
 - Alarms: Query and delete the alarm history.

- Simple expressive grammar for creating compound alarms composed of alarm subexpressions and logical operators.
- Alarm severities can be associated with alarms.
- The complete alarm state transition history is stored and queryable which allows for subsequent root cause analysis (RCA) or advanced analytics.
- Notification Methods: Create and delete notification methods and associate them with alarms, such as email. Supports the ability to notify users directly via email when an alarm state transitions occur.
- The Monasca API has both Java and Python implementations available.
- Persister: metrics 와 alarm state transitions 를 Kafka(Notification Bus)로부터 넘겨 받아 Time-Series Database 에 저장함.
- Notification/Threshold Engine: 각각 현재 서버에서 지정한 Notification 설정과 Threshold 설정을 Config Database 에 저장하기 전 Verifying 하는 엔진.
- Kafka: Openstack 의 RabbitMq 와 같은 역할로, Monasca 컴포넌트간의 데이터 송수신을 담당하는 Bus. Rabbitmq 와 분리되어 자체적으로 동작하므로 high performance, distributed, fault-tolerant, and scalable 함.
- Metrics and Alarms DB: Persister 가 metrics 와 alarms 을 저장하는 DB. 현재는 Vertica 와 influxDB 를 지원함.
- Config DB: 여러 configuration 이나 information of system 을 저장하는 DB. 현재는 MySQL 를 지원함.
- Monitoring Client(CLI): 파이썬 기반의 command-line communication 방식. 사용자가 직접 monitoring API 를 컨트롤할 수 있게 해줌.
- Monitoring UI: visualizing 을 위한 horizon 대시보드 지원.
- Ceilometer Publisher: Ceilometer 에 대한 multi-plugin 을 지원. 이 컴포넌트를 통해 Ceilometer-Monasca Publisher 가 Ceilometer 에 붙어 Ceilosca 를 구현된 것이고, 이 외에도 다양한 DB 에 Ceilometer API 를 연결하기 위해 반드시 필요한 컴포넌트임.
- Ceilosca: Ceilometer 와 Monasca 를 크게 변경하지 않고, 각각의 API 가 함께 동작할 수 있도록 Publisher 와 Storage Driver 라는 일종의 통역가와 같은 존재를 중간에 배치함으로써 구현되었음. 즉, Ceilometer 와 Monasca 각각의 metrics definition 을 mapping 하는 구조가 Ceilosca 의 핵심이라고 볼 수 있음.


```

meter_metric_map:
  - name: "network.outgoing.rate"
    monasca_metric_name: "vm.net.out_rate"
    resource_id: $.dimensions.resource_id
    project_id: $.dimensions.project_id
    user_id: $.dimensions.user_id
    region: "NA"
    type: "gauge"
    unit: "b/s"
    source: "NA"
    resource_metadata: $.measurement[0][2]

  - name: "network.incoming.rate"
    monasca_metric_name: "vm.net.in_rate"
    resource_id: $.dimensions.resource_id
    project_id: $.dimensions.project_id
    user_id: $.dimensions.user_id
    region: "NA"
    type: "gauge"
    unit: "b/s"
    source: "NA"
    resource_metadata: $.measurement[0][2]

```

Figure 5. Ceilometer-Monasca Mapping Configuration (Ceilosca_mapping.yaml)