Red Hat OpenStack Platform 17.1

오버클라우드 관찰 기능 관리

물리적 및 가상 리소스 추적 및 메트릭 수집

Last Updated: 2024-09-27

Red Hat OpenStack Platform 17.1 오버클라우드 관찰 기능 관리

물리적 및 가상 리소스 추적 및 메트릭 수집

OpenStack Team
rhos-docs@redhat.com

법적 공지

초록

운영 툴을 사용하여 Red Hat OpenStack Platform 환경을 측정 및 유지 관리하는 데 도움이 됩니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지 를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 더 나은지 알려주십시오.

직접 문서 피드백(DDF) 기능 사용

피드백 추가 DDF 기능을 사용하여 특정 문장, 단락 또는 코드 블록에 대한 직접 의견을 제출할 수 있습니다.

1. *다중 페이지 HTML* 형식으로 문서를 봅니다.
2. 문서의 오른쪽 상단에 **피드백** 버튼이 표시되는지 확인합니다.
3. 주석 처리하려는 텍스트 부분을 강조 표시합니다.
4. **피드백 추가를** 클릭합니다.
5. 코멘트를 사용하여 **피드백 추가** 필드를 완료합니다.
6. 선택 사항: 문서 팀이 문제에 대한 자세한 설명을 위해 연락을 드릴 수 있도록 이메일 주소를 추가합니다.
7. **Submit** 을 클릭합니다.

1장. 운영 측정 소개

ceilometer, collectd 및 로깅 서비스와 같은 관찰 기능 구성 요소를 사용하여 RHOSP(Red Hat OpenStack Platform) 환경에서 데이터를 수집할 수 있습니다. 자동 스케일링 사용 사례에 대해 Gnocchi에 수집한 데이터를 저장하거나 **metrics_qdr** 을 사용하여 데이터를 STF(Service Telemetry Framework)로 전달할 수 있습니다.

자동 스케일링에 대한 자세한 내용은 인스턴스 자동스케일링을 참조하십시오.

STF에 대한 자세한 내용은 Service Telemetry Framework 1.5를 참조하십시오.

1.1. 관찰 기능 아키텍처

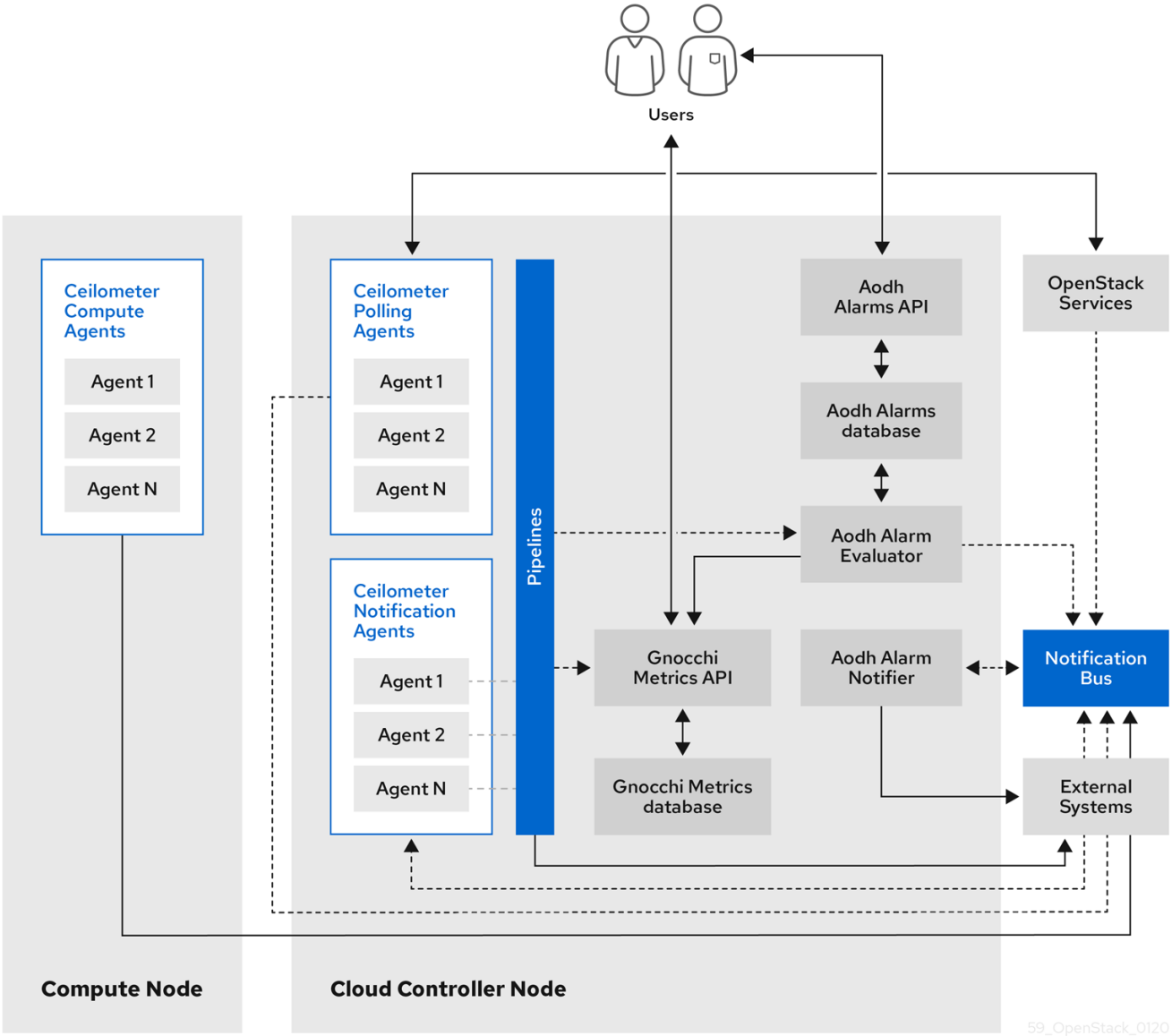
RHOSP(Red Hat OpenStack Platform) Observability는 OpenStack 기반 클라우드에 대한 사용자 수준 사용 데이터를 제공합니다. Compute usage events와 같은 기존 RHOSP 구성 요소에서 전송한 알림에서 데이터를 수집하거나 libvirt와 같은 RHOSP 인프라 리소스를 폴링하도록 관찰 구성 요소를 구성할 수 있습니다. Ceilometer에서는 수집된 데이터를 STF(Service Telemetry Framework)를 포함하여 데이터 저장소 및 메시지 큐와 같은 다양한 대상에 게시합니다.

관찰 기능은 다음 구성 요소로 구성됩니다.

- **데이터 수집**: Observability는 Ceilometer를 사용하여 메트릭 및 이벤트 데이터를 수집합니다. 자세한 내용은 1.2.1절. “ceilometer”의 내용을 참조하십시오.
- **스토리지**: Observability는 지표 데이터를 Gnocchi에 저장합니다. 자세한 내용은 1.3절. “Gnocchi를 사용한 스토리지”의 내용을 참조하십시오.
- **알람 서비스**: Observability는 Alarming 서비스(Aodh)를 사용하여 Ceilometer에서 수집한 메트릭 또는 이벤트 데이터에 대해 정의된 규칙을 기반으로 작업을 트리거합니다.

데이터를 수집한 후 타사 도구를 사용하여 메트릭 데이터를 표시 및 분석할 수 있으며, 알람 서비스를 사용하여 이벤트에 대한 알람을 구성할 수 있습니다.

그림 1.1. 관찰 기능 아키텍처



1.1.1. 모니터링 구성 요소의 지원 상태

이 표를 사용하여 RHOSP(Red Hat OpenStack Platform)에서 모니터링 구성 요소의 지원 상태를 확인합니다.

표 1.1. 지원 상태

구성 요소	이후 완전 지원	에서 더 이상 사용되지 않음	이후 삭제됨	참고
Aodh	RHOSP 9	RHOSP 15		자동 스케일링 사용 사례에 대해 지원됩니다.

구성 요소	이후 완전 지원	에서 더 이상 사용되지 않음	이후 삭제됨	참고
ceilometer	RHOSP 4			자동 스케일링 및 STF(Service Telemetry Framework) 사용 사례에서 RHOSP의 메트릭 및 이벤트 수집에 지원됩니다.
collectd	RHOSP 11			
Gnocchi	RHOSP 9	RHOSP 15		자동 스케일링 사용 사례에 대한 메트릭 스토리지에 지원됩니다.
Panko	RHOSP 11	RHOSP 12에서는 RHOSP 14 이후 기본적으로 설치되지 않음	RHOSP 17.0	
QDR	RHOSP 13	RHOSP 17.1		RHOSP에서 STF(Service Telemetry Framework)로 메트릭 및 이벤트 데이터 전송을 지원합니다.

1.2. RED HAT OPENSTACK PLATFORM의 데이터 수집

RHOSP(Red Hat OpenStack Platform)는 다음 두 가지 유형의 데이터 수집을 지원합니다.

- RHOSP 구성 요소 수준 모니터링용 Ceilometer입니다. 자세한 내용은 1.2.1절. "ceilometer"의 내용을 참조하십시오.
- 인프라 모니터링을 위한 collectd. 자세한 내용은 1.2.2절. "collectd"의 내용을 참조하십시오.

1.2.1. ceilometer

Ceilometer는 현재 모든 RHOSP 핵심 구성 요소에서 데이터를 정규화하고 변환하는 기능을 제공하는 RHOSP(Red Hat OpenStack Platform)의 기본 데이터 수집 구성 요소입니다. Ceilometer는 RHOSP 서비스와 관련된 미터링 및 이벤트 데이터를 수집합니다.

Ceilometer 서비스는 세 개의 에이전트를 사용하여 RHOSP(Red Hat OpenStack Platform) 구성 요소에서 데이터를 수집합니다.

- **컴퓨팅 에이전트(ceilometer-agent-compute):** 각 컴퓨팅 노드에서 실행되며 리소스 사용률 통계를 폴링합니다. 이 에이전트는 매개 변수 **--polling namespace-compute** 를 사용하여 실행되는 폴링 에이전트 **ceilometer-polling** 과 동일합니다.

- **중앙 에이전트(ceilometer-agent-central):** 중앙 관리 서버에서 실행되어 인스턴스 또는 컴퓨팅 노드에 연결되지 않은 리소스의 리소스 사용률 통계를 폴링합니다. 여러 에이전트를 시작하여 서비스를 수평으로 확장할 수 있습니다. 이는 **--polling namespace-central** 매개 변수로 작동하는 폴링 에이전트 **ceilometer-polling** 과 동일합니다.

- **알림 에이전트(ceilometer-agent-notification):** 중앙 관리 서버에서 실행되며 메시지 큐의 메시지를 사용하여 이벤트 및 미터링 데이터를 빌드합니다. 데이터는 정의된 대상에 게시합니다. Gnocchi가 기본 대상입니다. 이러한 서비스는 RHOSP 알림 버스를 사용하여 통신합니다.

Ceilometer 에이전트는 게시자를 사용하여 해당 엔드포인트(예: Gnocchi 또는 AMQP 버전 1(QDR))로 데이터를 전송합니다.

1.2.2. collectd

collectd는 인프라 메트릭을 제공하는 데 사용할 수 있는 또 다른 데이터 수집 에이전트입니다. 구성된 소스에서 데이터를 반복적으로 가져옵니다. 메트릭을 STF(Service Telemetry Framework)에 전달하여 데이터를 저장하고 시각화할 수 있습니다.

1.3. GNOCCHI를 사용한 스토리지

Gnocchi는 오픈 소스 시계열 데이터베이스입니다. gnocchi를 사용하여 Operator 및 사용자에게 메트릭 및 리소스에 대한 액세스를 저장하고 제공할 수 있습니다. Gnocchi는 보관 정책을 사용하여 계산할 집계와 보유할 집계 수를 정의합니다. 그리고 인덱서 드라이버를 사용하여 모든 리소스, 보관 정책 및 메트릭의 인덱스를 저장합니다.

자동 스케일링 사용 사례에 대해 RHOSP(Red Hat OpenStack Platform)에서 Gnocchi 사용이 지원됩니다. 자동 스케일링에 대한 자세한 내용은 인스턴스 자동스케일링을 참조하십시오.

1.3.1. 보관 정책: 시계열 데이터베이스에 단기 및 장기 데이터 저장

아카이브 정책은 계산할 집계와 유지할 집계 수를 정의합니다. Gnocchi는 최소, 최대값, 평균, Nth 백분위수 및 표준 편차와 같은 다양한 집계 방법을 지원합니다. 이러한 집계는 세분성이라는 기간 동안 계산되며 특정 기간 동안 유지됩니다.

보관 정책은 메트릭을 집계하는 방법과 저장 기간을 정의합니다. 각 아카이브 정책은 시간 초과의 포인트 수로 정의됩니다.

예를 들어 보관 정책이 1초 단위로 10개의 점으로 구성된 정책을 정의하는 경우 시계열 아카이브는 각각 1초 동안 집계를 나타내는 최대 10초 동안 유지됩니다. 즉, 시계열은 최대값에 더 최근 지점과 이전 지점 사이에 10초의 데이터를 유지합니다.

보관 정책은 사용되는 집계 메서드도 정의합니다. 기본값은 기본적으로 값이 mean, min, max. sum, std, count로 설정된 **default_aggregation_methods** 매개변수로 설정됩니다. 따라서 사용 사례에 따라 아카이브 정책과 세분성이 다릅니다.

추가 리소스

- 보관 정책에 대한 자세한 내용은 보관 정책 계획 및 관리를 참조하십시오.

1.3.2. 인덱서 드라이버

인덱서는 해당 정의, 유형 및 속성과 함께 모든 리소스, 보관 정책 및 메트릭의 인덱스를 저장합니다. 또한 리소스를 메트릭과 연결해야 합니다. Red Hat OpenStack Platform director는 기본적으로 인덱서 드라이버를 설치합니다. Gnocchi가 처리하는 모든 리소스와 메트릭을 인덱싱하려면 데이터베이스가 필요합니다. 지원되는 드라이버는 MySQL입니다.

1.3.3. Gnocchi 용어

이 표에는 Gnocchi 기능에 대해 일반적으로 사용되는 용어 정의가 포함되어 있습니다.

표 1.2. Gnocchi 용어

용어	정의
집계 방법	여러 측정값을 집계에 집계하는 데 사용되는 함수입니다. 예를 들어 min aggregation 메서드는 다양한 측정값의 값을 시간 범위의 모든 측정값의 최소 값으로 집계합니다.
집계	보관 정책에 따라 여러 조치에서 생성된 데이터 요소 튜플입니다. 집계는 타임스탬프 및 값으로 구성됩니다.
보관 정책	메트릭에 연결된 집계 스토리지 정책입니다. 보관 정책은 집계를 메트릭에 보관하는 기간과 집계 방법(통합 방법)을 결정합니다.
세분성	집계된 일련의 메트릭에 있는 두 집계 사이의 시간입니다.
측정	API에 의해 시계열 데이터베이스로 전송되는 들어오는 데이터 포인트 튜플입니다. 측정값은 타임스탬프와 값으로 구성됩니다.
메트릭	UUID로 식별된 집계를 저장하는 엔터티입니다. 지표를 이름을 사용하여 리소스에 연결할 수 있습니다. 지표가 집계를 저장하는 방법은 지표가 연결된 보관 정책에 의해 정의됩니다.
리소스	인프라에 있는 메트릭을 연결하는 모든 항목을 나타내는 엔터티입니다. 리소스는 고유 ID로 식별되며 속성을 포함할 수 있습니다.
시계열	시간별로 정렬된 집계 목록입니다.
Cryostat	지표가 집계를 유지하는 기간입니다. 보관 정책 컨텍스트에서 사용됩니다.

2장. 운영 측정 계획

Ceilometer 또는 collectd를 사용하여 자동 스케일링 또는 STF(Service Telemetry Framework)를 위한 Telemetry 데이터를 수집할 수 있습니다.

2.1. COLLECTD 측정

다음은 기본 collectd 측정값입니다.

- cpu
- 디스크 여유
- 디스크 사용량
- hugepages
- interface
- load
- memory
- unixsock
- uptime

2.2. 데이터 스토리지 계획

Gnocchi는 각 데이터 포인트가 집계되는 데이터 지점 컬렉션을 저장합니다. 스토리지 형식은 다른 기술을 사용하여 압축됩니다. 따라서 시계열 데이터베이스의 크기를 계산하려면 최악의 시나리오를 기반으로 크기를 추정해야 합니다.

주의

시계열 데이터베이스(Gnocchi) 스토리지에 RHOSP(Red Hat OpenStack Platform) Object Storage(swift) 사용은 소규모 및 비프로덕션 환경에서만 지원됩니다.

프로세스

1. 데이터 요소의 수를 계산합니다.
   포인트 수 = 시간 간격 / 단위

   예를 들어 1분 해상도로 1년의 데이터를 유지하려면 다음 공식을 사용합니다.

   데이터 지점 수 = (365 일 X 24 시간 X 60 분) / 1 분의 데이터 지점 = 525600

2. 시계열 데이터베이스의 크기를 계산합니다.
   size in bytes = 데이터 지점 X 8 바이트

이 공식을 예제에 적용하면 결과는 4.1MB입니다.

size in bytes = 525600 points X 8 bytes = 4204800 bytes = 4.1 MB

이 값은 집계된 단일 시계열 데이터베이스에 대한 예상 스토리지 요구 사항입니다. 아카이브 정책에서 여러 집계 방법(min, max, mean, sum, std, count)을 사용하는 경우 이 값을 사용하는 집계 방법 수로 곱합니다.

추가 리소스

- 1.3.1절. "보관 정책: 시계열 데이터베이스에 단기 및 장기 데이터 저장"
- 2.3절. "아카이브 정책 계획 및 관리"

2.3. 아카이브 정책 계획 및 관리

보관 정책을 사용하여 메트릭을 집계하는 방법과 시계열 데이터베이스에 지표를 저장하는 기간을 구성할 수 있습니다. 아카이브 정책은 시간 초과의 포인트 수로 정의됩니다.

보관 정책에서 1초 단위로 10포인트 정책을 정의하는 경우 시계열 아카이브는 각각 1초를 초과하는 집계를 나타내는 최대 10초 동안 유지됩니다. 즉, 시계열은 최신 지점과 이전 지점 사이에 최대 10초의 데이터를 유지합니다. 보관 정책은 사용할 집계 메서드도 정의합니다. 기본값은 **default_aggregation_methods** 매개변수로 설정됩니다. 여기서 기본값은 **mean**,**min**,**max** 입니다. **합계**,**std**,**count**. 따라서 사용 사례에 따라 아카이브 정책과 세분성이 다를 수 있습니다.

아카이브 정책을 계획하려면 다음 개념을 숙지해야 합니다.

- 지표. 자세한 내용은 2.3.1절. "메트릭"의 내용을 참조하십시오.
- 측정값. 자세한 내용은 2.3.2절. "사용자 정의 조치 생성"의 내용을 참조하십시오.

2.3.1. 메트릭

Gnocchi는 *metric* 이라는 오브젝트 유형을 제공합니다. 메트릭은 서버의 CPU 사용량, 방의 온도 또는 네트워크 인터페이스에서 보낸 바이트 수를 측정할 수 있는 모든 것입니다. 메트릭에는 다음과 같은 속성이 있습니다.

- 식별할 UUID입니다.
- 이름
- 측정값을 저장하고 집계하는 데 사용되는 보관 정책

추가 리소스

- 용어 정의는 Gnocchi Metric-as-a-Service 용어를 참조하십시오.

2.3.2. 사용자 정의 조치 생성

측정값은 API가 Gnocchi로 전송하는 들어오는 튜플입니다. 타임 스탬프와 값으로 구성됩니다. 고유한 사용자 지정 측정값을 만들 수 있습니다.

프로세스

- 사용자 지정 측정값을 생성합니다.

```
$ openstack metric measures add -m <MEASURE1> -m <MEASURE2> .. -r
<RESOURCE_NAME> <METRIC_NAME>
```

2.3.3. 메트릭 상태 확인

openstack metric 명령을 사용하여 성공적인 배포를 확인할 수 있습니다.

프로세스

- 배포를 확인합니다.

```
(overcloud) [stack@undercloud-0 ~]$ openstack metric status
+-----------------------------------------------------+-------+
| Field                                               | Value |
+-----------------------------------------------------+-------+
| storage/number of metric having measures to process | 0  |
| storage/total number of measures to process         | 0  |
+-----------------------------------------------------+-------+
```

오류 메시지가 없는 경우 배포에 성공합니다.

2.3.4. 아카이브 정책 생성

보관 정책을 생성하여 메트릭을 집계하는 방법과 시계열 데이터베이스에 지표를 저장하는 기간을 정의할 수 있습니다.

프로세스

- 아카이브 정책을 생성합니다. <archive-policy-name>을 정책 이름으로 바꾸고 <aggregation-method>를 집계 메서드로 바꿉니다.

```
$ openstack metric archive policy create <archive-policy-name> --definition <definition> \
--aggregation-method <aggregation-method>
```

참고

<definition>은 정책 정의입니다. 여러 속성을 쉼표(,)로 구분합니다. 아카이브 정책 정의의 이름과 값을 콜론(:)으로 구분합니다.

2.3.5. 아카이브 정책 보기

다음 단계를 사용하여 아카이브 정책을 검사합니다.

프로세스

1. 아카이브 정책을 나열합니다.

```
$ openstack metric archive policy list
```

2. 아카이브 정책의 세부 정보를 확인합니다.

```
# openstack metric archive-policy show <archive-policy-name>
```

2.3.6. 아카이브 정책 삭제

아카이브 정책을 삭제하려면 다음 단계를 사용하십시오.

프로세스

- 아카이브 정책을 삭제합니다. <archive-policy-name>을 삭제하려는 정책 이름으로 바꿉니다.

```
$ openstack metric archive policy delete <archive-policy-name>
```

검증

- 삭제한 보관 정책이 아카이브 정책 목록에서 없는지 확인합니다.

```
$ openstack metric archive policy list
```

2.3.7. 아카이브 정책 규칙 생성

아카이브 정책 규칙을 사용하여 메트릭과 아카이브 정책 간의 매핑을 구성할 수 있습니다.

프로세스

- 아카이브 정책 규칙을 생성합니다. <rule-name>을 규칙 이름으로 바꾸고 <archive-policy-name>을 아카이브 정책 이름으로 변경합니다.

```
$ openstack metric archive-policy-rule create <rule-name> /
--archive-policy-name  <archive-policy-name>
```

3장. LOGS 서비스 설치 및 구성

로그 메시지를 사용하여 시스템 이벤트 문제 해결 및 모니터링할 수 있습니다. 로그 수집 에이전트 Rsyslog는 클라이언트 측에서 로그를 수집하고 지원되는 RHOSP(Red Hat OpenStack Platform) 환경과 별도로 원격 Elasticsearch 스토리지 시스템으로 이러한 로그 레코드를 보냅니다.

3.1. 로그 시스템 아키텍처 및 구성 요소

모니터링 툴은 클라이언트가 RHOSP(Red Hat OpenStack Platform) 오버클라우드 노드에 배포된 클라이언트-서버 모델을 사용합니다. Rsyslog 서비스는 클라이언트 측 로깅을 제공합니다.

RHOSP의 로그 예는 다음과 같습니다.

- syslog 및 감사 로그 파일과 같은 운영 체제 로그입니다.
- RabbitMQ 및 MariaDB와 같은 인프라 구성 요소의 로그입니다.
- Identity(keystone) 및 Compute(nova)와 같은 RHOSP 서비스의 로그입니다.

로그 파일은 작업, 오류, 경고 및 기타 이벤트를 기록합니다. 분산 환경에서 한 위치에서 다양한 로그를 수집하면 디버깅 및 관리에 도움이 될 수 있습니다.

참고

RHOSP director는 로깅을 위해 서버 측 구성 요소를 배포하지 않습니다.

3.2. ELASTICSEARCH로 로깅 활성화

Elasticsearch는 로그를 저장하는 데 사용할 수 있는 서버 측 데이터베이스입니다. Elasticsearch에 대한 로그 서비스를 활성화하려면 Elasticsearch에 대한 로그 서비스를 인증해야 합니다.

참고

Rsyslog 서비스는 로깅을 위한 데이터 저장소로 Elasticsearch만 사용합니다.

사전 요구 사항

- Elasticsearch를 배포했습니다.
- 서버의 사용자 이름, 암호 및 URL이 있습니다.

프로세스

1. **tripleo-heat-templates/environments/logging-environment-rsyslog.yaml** 파일을 홈 디렉터리에 복사합니다.
2. 다음 예와 같이 환경에 대해 **RsyslogElasticsearchSetting** 매개변수를 구성합니다.

```
parameter_defaults:
  RsyslogElasticsearchSetting:
    uid: "elastic"
    pwd: "yourownpassword"
    skipverifyhost: "on"
    allowunsignedcerts: "on"
```

```
        server: "https://log-store-service-
telemetry.apps.stfcloudops1.lab.upshift.rdu2.redhat.com"
    serverport: 443
```

3. 로깅 환경 파일의 파일 경로를 다른 관련 환경 파일과 함께 **오버클라우드 배포** 명령에 추가합니다.

```
$ openstack overcloud deploy \
<overcloud_environment_files> \
-e /usr/share/openstack-tripleo-heat-templates/environments/logging-environment-
rsyslog.yaml
```

& **lt;overcloud_environment_files** >를 기존 배포의 환경 파일 목록으로 바꿉니다.

3.3. 구성 가능한 로깅 매개변수

이 표에는 RHOSP(Red Hat OpenStack Platform)에서 로깅 기능을 구성하는 데 사용하는 로깅 매개변수에 대한 설명이 포함되어 있습니다. 이러한 매개변수는 **/usr/share/openstack-tripleo-heat-templates/deployment/logging/rsyslog-container-puppet.yaml** 파일에서 찾을 수 있습니다.

표 3.1. 구성 가능한 로깅 매개변수

매개변수	설명
RsyslogElasticsearchSetting	**rsyslog-elasticsearch** 플러그인 구성.
RsyslogElasticsearchTlsCACert	Elasticsearch 서버 인증서를 발급한 CA의 CA 인증서 콘텐츠를 포함합니다.
RsyslogElasticsearchTlsClientCert	Elasticsearch에 대해 클라이언트 인증서 인증을 수행하기 위한 클라이언트 인증서의 내용이 포함되어 있습니다.
RsyslogElasticsearchTlsClientKey	cert **RsyslogElasticsearchTlsClientCert** 에 해당하는 개인 키의 내용을 포함합니다.

3.4. 로그 파일의 기본 경로 덮어쓰기

서비스 로그 파일의 경로를 포함하도록 기본 컨테이너를 수정하는 경우 기본 로그 파일 경로도 수정해야 합니다. 모든 구성 가능 서비스에는 < **service_name>LoggingSource** 매개변수가 있습니다. 예를 들어 **nova-compute** 서비스의 경우 매개 변수는 **NovaComputeLoggingSource** 입니다.

프로세스

- nova-compute 서비스의 기본 경로를 재정의하려면 구성 파일의 **NovaComputeLoggingSource** 매개변수에 경로를 추가합니다.

```
NovaComputeLoggingSource:
    tag: openstack.nova.compute
    file: /some/other/path/nova-compute.log
```

참고

서비스의 **tag** 및 **file** 매개변수 값을 정의합니다. 다른 매개변수에 기본값을 사용할 수 있습니다.

1. 특정 서비스의 형식을 수정할 수 있습니다. 형식은 Rsyslog 구성에 전달됩니다. 다음 예제에서는 기본 구문을 보여줍니다.

```
<service_name>LoggingSource:
   tag: <service_name>.tag
   path: <service_name>.path
   format: <service_name>.format
```

다음 예제에서는 더 복잡한 변환을 보여줍니다.

```
 ServiceLoggingSource:
   tag: openstack.Service
   path: /var/log/containers/service/service.log
   format: multiline
   format_firstline: '/^\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}.\d{3} \d+ \S+ \S+ \[(req-\S+ \S+ \S+ \S+ \S+ \S+|-)\]/'
   format1: '/^(?<Timestamp>\S+ \S+) (?<Pid>\d+) (?<log_level>\S+) (?<python_module>\S+) (\[(req-(?<request_id>\S+) (?<user_id>\S+) (?<tenant_id>\S+) (?<domain_id>\S+) (?<user_domain>\S+) (?<project_domain>\S+)|-)\])? (?<Payload>.*)?$/'
```

3.5. 로그 레코드 형식 수정

특정 서비스의 로그 레코드 시작 형식을 수정할 수 있습니다. 이렇게 하면 Rsyslog 구성에 직접 전달됩니다.

RHOSP(Red Hat OpenStack Platform) 로그 레코드의 기본 형식은 ('^[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}(.[0-9]++]입니다. (DEBUG|INFO|WARNING|ERROR) '.

절차

- 로그 레코드 시작을 구문 분석하기 위해 다른 정규식을 추가하려면 **startmsg.regex** 를 구성에 추가합니다.

```
NovaComputeLoggingSource:
      tag: openstack.nova.compute
      file: /some/other/path/nova-compute.log
      startmsg.regex: "^[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}(.[0-9]+ \\+[0-9]+)? [A-Z]+ \\([a-z]+\\)
```

3.6. RSYSLOG와 ELASTICSEARCH 간의 연결 확인

클라이언트 측에서 Rsyslog와 Elasticsearch 간의 통신을 테스트하고 확인할 수 있습니다.

프로세스

- Rsyslog 컨테이너의 **/var/log/rsyslog/omelasticsearch.log** 또는 호스트의 **/var/log/containers/rsyslog/omelasticsearch.log** 인 Elasticsearch 연결 로그 파일로 이동합니

다. 이 로그 파일이 없거나 로그 파일이 존재하지만 로그 파일이 없으면 연결 문제가 없습니다. 로그 파일이 있고 로그를 포함하는 경우 Rsyslog가 성공적으로 연결되지 않았습니다.

참고

서버 측에서 연결을 테스트하려면 Elasticsearch 로그에서 연결 문제가 있는지 확인합니다.

3.7. 역추적

문제를 해결하는 경우 역추적 로그를 사용하여 문제를 진단할 수 있습니다. 로그 파일에서 역추적에는 일반적으로 동일한 문제와 관련된 여러 정보 줄이 있습니다.

rsyslog는 로그 레코드 시작 방법을 정의하는 정규식을 제공합니다. 각 로그 레코드는 일반적으로 타임스탬프로 시작하며 역추적의 첫 번째 줄은 이 정보가 포함된 유일한 행입니다. rsyslog는 들여쓰기된 레코드를 첫 번째 행과 번들로 번들하고 하나의 로그 레코드로 보냅니다.

해당 동작 구성 옵션의 경우 <Service>LoggingSource에서 **startmsg.regex** 가 사용됩니다. 다음 정규식은 director의 모든 <service>LoggingSource 매개변수의 기본값입니다.

```
startmsg.regex='^[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}(.[0-9]+ [0-9]+)?
(DEBUG|INFO|WARNING|ERROR) '
```

이 기본값이 추가되거나 수정된 **LoggingSource** 의 로그 레코드와 일치하지 않는 경우 그에 따라 **startmsg.regex** 를 변경해야 합니다.

3.8. RED HAT OPENSTACK PLATFORM 서비스의 로그 파일 위치

각 RHOSP(Red Hat OpenStack Platform) 구성 요소에는 실행 중인 서비스와 관련된 파일이 포함된 별도의 로깅 디렉터리가 있습니다.

3.8.1. Bare Metal Provisioning (ironic) 로그 파일

Service	서비스 이름	로그 경로
OpenStack Ironic API	openstack-ironic-api.service	/var/log/containers/ironic/ironic-api.log
OpenStack Ironic Conductor	openstack-ironic-conductor.service	/var/log/containers/ironic/ironic-conductor.log

3.8.2. Block Storage(cinder) 로그 파일

Service	서비스 이름	로그 경로
블록 스토리지 API	openstack-cinder-api.service	/var/log/containers/cinder-api.log
블록 스토리지 백업	openstack-cinder-backup.service	/var/log/containers/cinder/backup.log

Service	서비스 이름	로그 경로
정보 메시지	cinder-manage 명령	/var/log/containers/cinder/cinder-manage.log
블록 스토리지 스케줄러	openstack-cinder-scheduler.service	/var/log/containers/cinder/scheduler.log
블록 스토리지 볼륨	openstack-cinder-volume.service	/var/log/containers/cinder/volume.log

3.8.3. 컴퓨팅(nova) 로그 파일

Service	서비스 이름	로그 경로
OpenStack Compute API 서비스	openstack-nova-api.service	/var/log/containers/nova/nova-api.log
OpenStack Compute 인증서 서버	openstack-nova-cert.service	/var/log/containers/nova/nova-cert.log
OpenStack Compute 서비스	openstack-nova-compute.service	/var/log/containers/nova/nova-compute.log
OpenStack Compute Conductor 서비스	openstack-nova-conductor.service	/var/log/containers/nova/nova-conductor.log
OpenStack Compute VNC 콘솔 인증 서버	openstack-nova-consoleauth.service	/var/log/containers/nova/nova-consoleauth.log
정보 메시지	nova-manage 명령	/var/log/containers/nova/nova-manage.log
OpenStack Compute NoVNC Proxy 서비스	openstack-nova-novncproxy.service	/var/log/containers/nova/nova-novncproxy.log
OpenStack Compute Scheduler 서비스	openstack-nova-scheduler.service	/var/log/containers/nova/nova-scheduler.log

3.8.4. 대시보드(horizon) 로그 파일

Service	서비스 이름	로그 경로
특정 사용자 상호 작용 로그	대시보드 인터페이스	/var/log/containers/horizon/horizon.log

Apache HTTP 서버는 대시보드 웹 인터페이스에 추가 로그 파일을 사용합니다. 이 파일은 웹 브라우저 또는 명령줄 클라이언트(예: keystone 및 nova)를 사용하여 액세스할 수 있습니다. 다음 로그 파일을 사용하여 대시보드 사용량을 추적하고 결함을 진단할 수 있습니다.

목적	로그 경로
처리된 모든 HTTP 요청	/var/log/containers/httpd/horizon_access.log
HTTP 오류	/var/log/containers/httpd/horizon_error.log
admin-role API 요청	/var/log/containers/httpd/keystone_wsgi_admin_access.log
admin-role API 오류	/var/log/containers/httpd/keystone_wsgi_admin_error.log
member-role API 요청	/var/log/containers/httpd/keystone_wsgi_main_access.log
member-role API 오류	/var/log/containers/httpd/keystone_wsgi_main_error.log

참고

동일한 호스트에서 실행 중인 다른 웹 서비스에서 보고한 오류를 저장하는 **/var/log/containers/httpd/default_error.log** 도 있습니다.

3.8.5. ID 서비스(keystone) 로그 파일

Service	서비스 이름	로그 경로
OpenStack ID 서비스	openstack-keystone.service	/var/log/containers/keystone/keystone.log

3.8.6. Image 서비스(glance) 로그 파일

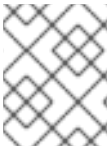
Service	서비스 이름	로그 경로
OpenStack Image Service API 서버	openstack-glance-api.service	/var/log/containers/glance/api.log
OpenStack 이미지 서비스 레지스트리 서버	openstack-glance-registry.service	/var/log/containers/glance/registry.log

3.8.7. networking(neutron) 로그 파일

Service	서비스 이름	로그 경로
OpenStack Neutron DHCP 에이전트	neutron-dhcp-agent.service	/var/log/containers/neutron/dhcp-agent.log
OpenStack Networking 계층 3 에이전트	neutron-l3-agent.service	/var/log/containers/neutron/l3-agent.log
메타데이터 에이전트 서비스	neutron-metadata-agent.service	/var/log/containers/neutron/metadata-agent.log
메타데이터 네임스페이스 프록시	해당 없음	/var/log/containers/neutron/neutron-ns-metadata-proxy-*UUID*.log
Open vSwitch 에이전트	neutron-openvswitch-agent.service	/var/log/containers/neutron/openvswitch-agent.log
OpenStack Networking 서비스	neutron-server.service	/var/log/containers/neutron/server.log

3.8.8. Object Storage(swift) 로그 파일

OpenStack Object Storage는 시스템 로깅 기능에만 로그를 보냅니다.



참고

기본적으로 모든 Object Storage 로그 파일은 local0, local1 및 local2 syslog 기능을 사용하여 **/var/log/containers/swift/swift.log** 로 이동합니다.

Object Storage 로그 메시지는 REST API 서비스 또는 백그라운드 데몬에서 수행됩니다.

- API 서비스 메시지에는 API 요청당 한 행이 포함됩니다. 프론트 엔드 및 백엔드 서비스 둘 다 메시지입니다.
- 데몬 메시지에는 데몬 작업에 대한 사람이 읽을 수 있는 정보가 포함됩니다. 소스 ID는 항상 행 시작 부분에 있습니다.

프록시 메시지의 예는 다음과 같습니다.

```
Apr 20 15:20:34 rhev-a24c-01 proxy-server: 127.0.0.1 127.0.0.1 20/Apr/2015/19/20/34 GET /v1/AUTH_zaitcev%3Fformat%3Djson%26marker%3Dtestcont HTTP/1.0 200 - python-swiftclient-2.1.0 AUTH_tk737d6... - 2 - txc454fa8ea4844d909820a-0055355182 - 0.0162 - - 1429557634.806570053 1429557634.822791100
```

다음은 데몬 메시지의 예입니다.

```
Apr 27 17:08:15 rhev-a24c-02 object-auditor: Object audit (ZBF). Since Mon Apr 27 21:08:15 2015: Locally: 1 passed, 0 quarantined, 0 errors files/sec: 4.34 , bytes/sec: 0.00, Total time: 0.23, Auditing time: 0.00, Rate: 0.00
Apr 27 17:08:16 rhev-a24c-02 object-auditor: Object audit (ZBF) "forever" mode completed: 0.56s.
```

```
Total quarantined: 0, Total errors: 0, Total files/sec: 14.31, Total bytes/sec: 0.00, Auditing time: 0.02,
Rate: 0.04
Apr 27 17:08:16 rhev-a24c-02 account-replicator: Beginning replication run
Apr 27 17:08:16 rhev-a24c-02 account-replicator: Replication run OVER
Apr 27 17:08:16 rhev-a24c-02 account-replicator: Attempted to replicate 5 dbs in 0.12589 seconds
(39.71876/s)
Apr 27 17:08:16 rhev-a24c-02 account-replicator: Removed 0 dbs
Apr 27 17:08:16 rhev-a24c-02 account-replicator: 10 successes, 0 failures
```

3.8.9. 오케스트레이션(heat) 로그 파일

Service	서비스 이름	로그 경로
OpenStack Heat API 서비스	openstack-heat-api.service	/var/log/containers/heat/heat-api.log
OpenStack Heat Engine Service	openstack-heat-engine.service	/var/log/containers/heat/heat-engine.log
오케스트레이션 서비스 이벤트	해당 없음	/var/log/containers/heat/heat-manage.log

3.8.10. 공유 파일 시스템 서비스(manila) 로그 파일

Service	서비스 이름	로그 경로
OpenStack Manila API 서버	openstack-manila-api.service	/var/log/containers/manila/api.log
OpenStack Manila Scheduler	openstack-manila-scheduler.service	/var/log/containers/manila/scheduler.log//
OpenStack Manila 공유 서비스	openstack-manila-share.service	/var/log/containers/manila/share.log

/var/log/containers/manila/manila-manage.log 의 Manila Python 라이브러리에서 정보를 기록할 수도 있습니다.

3.8.11. Telemetry(ceilometer) 로그 파일

Service	서비스 이름	로그 경로
OpenStack ceilometer 알림 에이전트	ceilometer_agent_notification	/var/log/containers/ceilometer/agent-notification.log
OpenStack ceilometer 중앙 에이전트	ceilometer_agent_central	/var/log/containers/ceilometer/central.log

Service	서비스 이름	로그 경로
OpenStack ceilometer 컬렉션	openstack-ceilometer-collector.service	/var/log/containers/ceilometer/collector.log
OpenStack ceilometer 컴퓨팅 에이전트	ceilometer_agent_compute	/var/log/containers/ceilometer/compute.log

3.8.12. 지원 서비스를 위한 로그 파일

다음 서비스는 코어 RHOSP 구성 요소에서 사용하며 자체 로그 디렉터리 및 파일이 있습니다.

Service	서비스 이름	로그 경로
메시지 브로커 (RabbitMQ)	rabbitmq-server.service	/var/log/rabbitmq/rabbit@*short_hostname*.log /var/log/rabbitmq/rabbit@*short_hostname*-sasl.log (Simple Authentication and Security Layer 관련 로그 메시지의 경우)
데이터베이스 서버 (MariaDB)	mariadb.service	/var/log/mariadb/mariadb.log
가상 네트워크 스위치 (Open vSwitch)	openvswitch-nonetwork.service	/var/log/openvswitch/ovsdb-server.log /var/log/openvswitch/ovs-vswitchd.log

3.8.13. Aodh (alarming service) 로그 파일

Service	컨테이너 이름	로그 경로
알람 API	aodh_api	/var/log/containers/httpd/aodh-api/aodh_wsgi_access.log
알람 평가기 로그	aodh_evaluator	/var/log/containers/aodh/aodh-evaluator.log
경보 리스너	aodh_listener	/var/log/containers/aodh/aodh-listener.log
경고 알림	aodh_notifier	/var/log/containers/aodh/aodh-notifier.log

3.8.14. gnocchi(metric storage) 로그 파일

Service	컨테이너 이름	로그 경로
Gnocchi API	gnocchi_api	/var/log/containers/httpd/gnocchi-api/gnocchi_wsgi_access.log

Service	컨테이너 이름	로그 경로
gnocchi metricd	gnocchi_metricd	/var/log/containers/gnocchi/gnocchi-metricd.log
gnocchi statsd	gnocchi_statsd	/var/log/containers/gnocchi/gnocchi-statsd.log

4장. COLLECTD 플러그인

RHOSP(Red Hat OpenStack Platform) 17.1-베타 환경에 따라 여러 collectd 플러그인을 구성할 수 있습니다.

다음 플러그인 목록은 기본값을 재정의하도록 설정할 수 있는 사용 가능한 heat 템플릿 **ExtraConfig** 매개변수를 보여줍니다. 각 섹션에는 **ExtraConfig** 옵션의 일반 구성 이름이 있습니다. 예를 들어 **example_plugin** 이라는 collectd 플러그인이 있는 경우 플러그인 제목의 형식은 **collectd::plugin::example_plugin** 입니다.

다음 예와 같이 특정 플러그인에 대해 사용 가능한 매개변수의 테이블을 참조합니다.

```
ExtraConfig:
  collectd::plugin::example_plugin::<parameter>: <value>
```

Prometheus 또는 Grafana 쿼리에 대한 특정 플러그인의 메트릭 테이블을 참조합니다.

4.1. COLLECTD::PLUGIN::AGGREGATION

집계 플러그인을 사용하여 여러 값을 하나로 **집계** 할 수 있습니다. **합계**,**평균**,**min**, **max** 와 같은 집계 함수를 사용하여 메트릭 (예: 평균 및 총 CPU 통계)을 계산합니다.

표 4.1. 집계 매개변수

매개변수	유형
host	문자열
plugin	문자열
plugininstance	정수
agg_type	문자열
typeinstance	문자열
sethost	문자열
setplugin	문자열
setplugininstance	정수
settypeinstance	문자열
groupby	문자열 배열
calculatesum	부울
calculatenum	부울

매개변수	유형
calculateaverage	부울
calculateminimum	부울
calculatemaximum	부울
calculatestddev	부울

설정 예:

다음 파일을 생성하려면 세 가지 집계 구성을 배포합니다.

1. **aggregation-calcCpuLoadAvg.conf**: 호스트 및 상태로 그룹화된 모든 CPU 코어의 평균 CPU 로드
2. **aggregation-calcCpuLoadMinMax.conf**: 호스트 및 상태별 최소 및 최대 CPU 로드 그룹
3. **aggregation-calcMemoryTotalMaxAvg.conf**: 유형별로 그룹화된 메모리의 최대, 평균 및 합계

집계 구성은 기본 **cpu** 및 **memory** 플러그인 구성을 사용합니다.

```
parameter_defaults:
  CollectdExtraPlugins:
    - aggregation

  ExtraConfig:
    collectd::plugin::aggregation::aggregators:
      calcCpuLoadAvg:
        plugin: "cpu"
        agg_type: "cpu"
        groupby:
          - "Host"
          - "TypeInstance"
        calculateaverage: True
      calcCpuLoadMinMax:
        plugin: "cpu"
        agg_type: "cpu"
        groupby:
          - "Host"
          - "TypeInstance"
        calculatemaximum: True
        calculateminimum: True
      calcMemoryTotalMaxAvg:
        plugin: "memory"
        agg_type: "memory"
        groupby:
          - "TypeInstance"
        calculatemaximum: True
        calculateaverage: True
        calculatesum: True
```

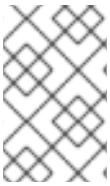
4.2. COLLECTD::PLUGIN::AMQP1

amqp1 플러그인을 사용하여 amqp1 메시지 버스에 값을 작성합니다(예: AMQ Interconnect).

표 4.2. amqp1 매개변수

매개변수	유형
manage_package	부울
전송	문자열
host	문자열
port	정수
user	문자열
암호	문자열
address	문자열
instances	hash
retry_delay	정수
send_queue_limit	정수
간격	정수

send_queue_limit 매개변수를 사용하여 발신 지표 대기열의 길이를 제한합니다.



참고

AMQP1 연결이 없는 경우 플러그인은 전송할 메시지를 계속 대기열에 추가하므로 바인딩되지 않은 메모리 사용량이 발생할 수 있습니다. 기본값은 0으로, 발신 메트릭 큐를 비활성화합니다.

메트릭이 누락된 경우 **send_queue_limit** 매개변수 값을 늘립니다.

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
    - amqp1

  ExtraConfig:
    collectd::plugin::amqp1::send_queue_limit: 5000
```

4.3. COLLECTD::PLUGIN::APACHE

apache 플러그인을 사용하여 Apache 웹 서버에서 제공하는 **mod_status** 플러그인에서 Apache 데이터를 수집합니다. 제공된 각 인스턴스에는 초 단위로**지정된 간격** 마다 값이 있습니다. 인스턴스에 **시간 초과** 간격 매개변수를 제공하는 경우 값은 밀리초 단위입니다.

표 4.3. Apache 매개변수

매개변수	유형
instances	hash
간격	정수
manage-package	부울
package_install_options	list

표 4.4. Apache 인스턴스 매개변수

매개변수	유형
url	HTTP URL
user	문자열
암호	문자열
verifypeer	부울
verifyhost	부울
cacert	AbsolutePath
sslciphers	문자열
timeout	정수

설정 예:

이 예에서 인스턴스 이름은 http://10.0.0.111/mod_status?auto 에서 Apache 웹 서버에 연결하는 **localhost** 입니다. 플러그인과 호환되지 않는 유형으로 상태 페이지가 반환되지 않도록 **?auto** 를 URL 끝에 추가해야 합니다.

```
parameter_defaults:
  CollectdExtraPlugins:
  - apache

  ExtraConfig:
```

```
collectd::plugin::apache::instances:
  localhost:
    url: "http://10.0.0.111/mod_status?auto"
```

추가 리소스

apache 플러그인 구성에 대한 자세한 내용은 apache 를 참조하십시오.

4.4. COLLECTD::PLUGIN::BATTERY

배터리 플러그인을 사용하여 랩탑 배터리의 나머지 용량, 전원 또는 마운전을 보고합니다.

표 4.5. 배터리 매개변수

매개변수	유형
values_percentage	부울
report_degraded	부울
query_state_fs	부울
간격	정수

추가 리소스

배터리 플러그인 구성에 대한 자세한 내용은 배터리를 참조하십시오.

4.5. COLLECTD::PLUGIN::BIND

bind 플러그인을 사용하여 DNS 서버에서 쿼리 및 응답에 대한 인코딩된 통계를 검색하고 해당 값을 collectd에 제출합니다.

표 4.6. 매개변수 바인딩

매개변수	유형
url	HTTP URL
memorystats	부울
opcodes	부울
parsetime	부울
qtypes	부울
resolverstats	부울
serverstats	부울

매개변수	유형
zonemaintstats	부울
뷰	array
간격	정수

표 4.7. 뷰 매개변수 바인딩

매개변수	유형
name	문자열
qtypes	부울
resolverstats	부울
cacherrsets	부울
영역	문자열 목록

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
  - bind

  ExtraConfig:
    collectd::plugins::bind:
      url: http://localhost:8053/
      memorystats: true
      opcodes: true
      parsetime: false
      qtypes: true
      resolverstats: true
      serverstats: true
      zonemaintstats: true
      views:
      - name: internal
        qtypes: true
        resolverstats: true
        cacherrsets: true
      - name: external
        qtypes: true
        resolverstats: true
        cacherrsets: true
        zones:
        - "example.com/IN"
```

4.6. COLLECTD::PLUGIN::CEPH

ceph 플러그인을 사용하여 ceph 데몬에서 데이터를 수집합니다.

표 4.8. Ceph 매개변수

매개변수	유형
데몬	array
longrunavglatency	부울
convertspecialmetrictypes	부울
package_name	문자열

설정 예:

```
parameter_defaults:
  ExtraConfig:
    collectd::plugin::ceph::daemons:
      - ceph-osd.0
      - ceph-osd.1
      - ceph-osd.2
      - ceph-osd.3
      - ceph-osd.4
```

참고

OSD(오브젝트 스토리지 데몬)가 모든 노드에 없는 경우 OSD를 나열해야 합니다.

collectd를 배포할 때 **ceph** 플러그인이 Ceph 노드에 추가됩니다. 배포가 실패하므로 **CollectdExtraPlugins** 에 Ceph 노드에 **ceph** 플러그인을 추가하지 마십시오.

추가 리소스

ceph 플러그인 구성에 대한 자세한 내용은 ceph 를 참조하십시오.

4.7. COLLECTD::PLUGINS::CGROUPS

cgroups 플러그인을 사용하여 cgroup의 프로세스 정보를 수집합니다.

표 4.9. cgroups 매개변수

매개변수	유형
ignore_selected	부울
간격	정수

매개변수	유형
cgroups	list

추가 리소스

cgroups 플러그인 구성에 대한 자세한 내용은 cgroup을 참조하십시오.
https://collectd.org/documentation/manpages/collectd.conf.5.shtml#plugin_cgroups

4.8. COLLECTD::PLUGIN::CONNECTIVITY

연결 플러그인을 사용하여 네트워크 인터페이스의 상태를 모니터링합니다.

참고

나열된 인터페이스가 없으면 기본적으로 모든 인터페이스가 모니터링됩니다.

표 4.10. 연결 매개변수

매개변수	유형
interfaces	array

설정 예:

```
parameter_defaults:
  ExtraConfig:
    collectd::plugin::connectivity::interfaces:
    - eth0
    - eth1
```

추가 리소스

연결 플러그인 구성에 대한 자세한 내용은 연결을 참조하십시오.

4.9. COLLECTD::PLUGIN::CONNTRACK

conntrack 플러그인을 사용하여 Linux connection-tracking 테이블의 항목 수를 추적합니다. 이 플러그인에 대한 매개변수가 없습니다.

4.10. COLLECTD::PLUGIN::CONTEXTSWITCH

ContextSwitch 플러그인을 사용하여 시스템에서 처리하는 컨텍스트 전환 수를 수집합니다. 사용 가능한 유일한 매개변수는 **간격** 이며, 이는 초 단위로 정의된 폴링 간격입니다.

추가 리소스

contextwitch 플러그인 구성에 대한 자세한 **내용은 context** witch 를 참조하십시오.

4.11. COLLECTD::PLUGIN::CPU

cpu 플러그인을 사용하여 CPU에서 다양한 상태(예: 유휴 상태, 실행 사용자 코드 실행, 시스템 코드 실행, IO 작업 대기 및 기타 상태)에서 사용하는 시간을 모니터링합니다.

cpu 플러그인은 백분율 값이 아닌 *jiffies* 를 수집합니다. jiffy의 값은 하드웨어 플랫폼의 클럭 빈도에 따라 달라지므로 절대 시간 간격 단위가 아닙니다.

백분율 값을 보고하려면 부울 매개변수 **reportbycpu** 및 **reportbystate** 를 **true** 로 설정한 다음 부울 매개변수 **값을** true로 설정합니다.

이 플러그인은 기본적으로 활성화되어 있습니다.

표 4.11. CPU 지표

이름	설명	쿼리
idle	유휴 시간 수	**collectd_cpu_total{...,type_instance='idle'}**
interrupt	인터럽트에 의해 차단된 CPU	**collectd_cpu_total{...,type_instance='interrupt'}**
nice	낮은 우선 순위 프로세스를 실행하는 시간	**collectd_cpu_total{...,type_instance='nice'}**
softirq	인터럽트 요청 서비스에 사용된 사이클의 양	**collectd_cpu_total{...,type_instance='waitirq'}**
훔치	하이퍼바이저가 다른 가상 프로세서를 제공하는 동안 가상 CPU가 실제 CPU를 기다리는 시간의 백분율	**collectd_cpu_total{...,type_instance='steal'}**
system	시스템 수준에서 소요된 시간(커널)	**collectd_cpu_total{...,type_instance='system'}**
user	사용자 프로세스에서 사용하는 jiffies	**collectd_cpu_total{...,type_instance='user'}**
wait	미해결 I/O 요청 시 대기 중인 CPU	**collectd_cpu_total{...,type_instance='wait'}**

표 4.12. CPU 매개변수

매개변수	유형	기본값
reportbystate	부울	true
valuespercentage	부울	true
reportbycpu	부울	true

매개변수	유형	기본값
reportnumcpu	부울	false
reportgueststate	부울	false
subtractgueststate	부울	true
간격	정수	120

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
    - cpu
  ExtraConfig:
    collectd::plugin::cpu::reportbystate: true
```

추가 리소스

cpu 플러그인 구성에 대한 자세한 내용은 cpu 를 참조하십시오.

4.12. COLLECTD::PLUGIN::CPUFREQ

cpufreq 플러그인을 사용하여 현재 CPU 빈도를 수집합니다. 이 플러그인에 대한 매개변수가 없습니다.

4.13. COLLECTD::PLUGIN::CSV

CSV **플러그인** 을 사용하여 CSV 형식의 로컬 파일에 값을 작성합니다.

표 4.13. CSV 매개변수

매개변수	유형
datadir	문자열
storerates	부울
간격	정수

4.14. COLLECTD::PLUGIN::DF

df 플러그인을 사용하여 파일 시스템의 디스크 공간 사용 정보를 수집합니다.

이 플러그인은 기본적으로 활성화되어 있습니다.

표 4.14. DF 메트릭

이름	설명	쿼리
무료	디스크 여유 공간의 양	**collectd_df_df_complex{...,type_instance="free"}**
예약됨	예약된 디스크 공간의 양	**collectd_df_df_complex{...,type_instance="reserved"}**
사용됨	사용된 디스크 공간의 양	**collectd_df_df_complex{...,type_instance="used"}**

표 4.15. DF 매개변수

매개변수	유형	기본값
devices	array	**[]**
fstypes	array	**['xfs']**
ignoreselected	부울	true
마운트 지점	array	**[]**
reportbydevice	부울	true
reportinodes	부울	true
reportreserved	부울	true
valuesabsolute	부울	true
valuespercentage	부울	false

설정 예:

```
parameter_defaults:
  ExtraConfig:
    collectd::plugin::df::fstypes: ['tmpfs','xfs']
```

추가 리소스

df 플러그인 구성에 대한 자세한 내용은 df 를 참조하십시오.

4.15. COLLECTD::PLUGIN::DISK

디스크 플러그인을 사용하여 하드 디스크 및 지원되는 경우 파티션의 성능 통계를 수집합니다.

참고

디스크 플러그인은 기본적으로 모든 디스크를 모니터링합니다. **ignoreselected** 매개변수를 사용하여 디스크 목록을 무시할 수 있습니다. 예제 구성은 *sda,sdb* 및 *sdc* 디스크를 무시하고 목록에 포함되지 않은 모든 디스크를 모니터링합니다.

이 플러그인은 기본적으로 활성화되어 있습니다.

표 4.16. 디스크 매개변수

매개변수	유형	기본값
디스크	array	[]
ignoreselected	부울	false
udevnameattr	문자열	*<undefined>*

표 4.17. 디스크 지표

이름	설명
병합	두 개 이상의 논리 작업을 제공하는 물리적 디스크 액세스와 같이 함께 병합할 수 있는 대기 중인 작업 수입니다.
time	I/O 작업을 완료하는 데 걸리는 평균 시간입니다. 값이 정확하지 않을 수 있습니다.
io_time	I/O(ms)를 수행하는 데 소요된 시간. 이 메트릭을 장치 로드 백분율로 사용할 수 있습니다. 1초의 값은 로드의 100%와 일치합니다.
weighted_io_time	I/O 완료 시간과 누적될 수 있는 백로그를 모두 측정합니다.
pending_operations	보류 중인 I/O 작업의 대기열 크기를 표시합니다.

설정 예:

```
parameter_defaults:
  ExtraConfig:
    collectd::plugin::disk::disks: ['sda', 'sdb', 'sdc']
    collectd::plugin::disk::ignoreselected: true
```

추가 리소스

디스크 플러그인 구성에 대한 자세한 내용은 디스크 를 참조하십시오.
https://collectd.org/documentation/manpages/collectd.conf.5.shtml#plugin_disk

4.16. COLLECTD::PLUGIN::HUGEPAGES

hugepages 플러그인을 사용하여 hugepages 정보를 수집합니다.

> This plugin is enabled by default.

표 4.18. hugepages 매개변수

매개변수	유형	기본값
report_per_node_hp	부울	true
report_root_hp	부울	true
values_pages	부울	true
values_bytes	부울	false
values_percentage	부울	false

설정 예:

```
parameter_defaults:
  ExtraConfig:
    collectd::plugin::hugepages::values_percentage: true
```

추가 리소스

- **hugepages** 플러그인 구성에 대한 자세한 내용은 hugepages 를 참조하십시오.

4.17. COLLECTD::PLUGIN::INTERFACE

인터페이스 플러그인을 사용하여 옥텟, 초당 패킷 및 오류 비율로 인터페이스 트래픽을 측정합니다.

> This plugin is enabled by default.

표 4.19. 인터페이스 매개변수

매개변수	유형	Default
interfaces	array	[]
ignoreselected	부울	false
reportinactive	부울	true

설정 예:

```
parameter_defaults:
  ExtraConfig:
    collectd::plugin::interface::interfaces:
```

```
    - lo
collectd::plugin::interface::ignoreselected: true
```

추가 리소스

- **인터페이스** 플러그인 구성에 대한 자세한 내용은 인터페이스를 참조하십시오.

4.18. COLLECTD::PLUGIN::LOAD

load 플러그인을 사용하여 시스템 로드와 시스템 사용 개요를 수집합니다.

This plugin is enabled by default.

표 4.20. 플러그인 매개변수

매개변수	유형	Default
report_relative	부울	true

설정 예:

```
parameter_defaults:
  ExtraConfig:
    collectd::plugin::load::report_relative: false
```

추가 리소스

- 로드 플러그인 구성에 대한 자세한 내용은 **load** 를 참조하십시오.

4.19. COLLECTD::PLUGIN::MCELOG

mcelog 플러그인을 사용하여 머신 검사 예외와 관련된 알림 및 통계를 보냅니다. 데몬 모드에서 실행되도록 **mcelog** 를 구성하고 로깅 기능을 활성화합니다.

표 4.21. mcelog 매개변수

매개변수	유형
Mcelogfile	문자열
메모리	hash **{ mcelogclientsocket[string], persistentnotification[boolean] }**

설정 예:

```
parameter_defaults:
    CollectdExtraPlugins: mcelog
    CollectdEnableMcelog: true
```

추가 리소스

- **mcelog** 플러그인 구성에 대한 자세한 내용은 mcelog 를 참조하십시오.

4.20. COLLECTD::PLUGIN::MEMCACHED

memcached 플러그인을 사용하여 memcached 캐시 사용, 메모리 및 기타 관련 정보에 대한 정보를 검색합니다.

표 4.22. Memcached 매개변수

매개변수	유형
instances	hash
간격	정수

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
  - memcached

  ExtraConfig:
    collectd::plugin::memcached::instances:
      local:
        host: "%{hiera('fqdn_canonical')}"
        port: 11211
```

추가 리소스

- **memcached** 플러그인을 구성하는 방법에 대한 자세한 내용은 memcached 를 참조하십시오.

4.21. COLLECTD::PLUGIN::MEMORY

메모리 플러그인을 사용하여 시스템의 메모리에 대한 정보를 검색합니다.

> This plugin is enabled by default.

표 4.23. 메모리 매개변수

매개변수	유형
기본값	valuesabsolute
부울	true
valuespercentage	부울

설정 예:

```
parameter_defaults:
  ExtraConfig:
    collectd::plugin::memory::valuesabsolute: true
    collectd::plugin::memory::valuespercentage: false
```

추가 리소스

- **메모리 플러그인 구성에 대한 자세한 내용은 메모리** 를 참조하십시오.
  https://collectd.org/documentation/manpages/collectd.conf.5.shtml#plugin_memory

4.22. COLLECTD::PLUGIN::NTPD

ntpd 플러그인을 사용하여 통계에 대한 액세스를 허용하고 구성된 매개 변수 및 시간 동기화 상태에 대한 정보를 검색하도록 구성된 로컬 NTP 서버를 쿼리합니다.

표 4.24. ntpd 매개변수

매개변수	유형
host	호스트 이름
port	포트 번호(정수)
reverselookups	부울
includeunitid	부울
간격	정수

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
  - ntpd

  ExtraConfig:
    collectd::plugin::ntpd::host: localhost
    collectd::plugin::ntpd::port: 123
    collectd::plugin::ntpd::reverselookups: false
    collectd::plugin::ntpd::includeunitid: false
```

추가 리소스

- **ntpd** 플러그인 구성에 대한 자세한 내용은 ntpd 를 참조하십시오.

4.23. COLLECTD::PLUGIN::OVS_STATS

ovs_stats 플러그인을 사용하여 OVS 연결 인터페이스의 통계를 수집합니다. **ovs_stats** 플러그인은 OVSDB 관리 프로토콜(RFC7047) 모니터 메커니즘을 사용하여 OVSDB에서 통계를 가져옵니다.

표 4.25. ovs_stats 매개변수

매개변수	유형
address	문자열
브리지	list
port	정수
소켓	문자열

설정 예:

다음 예제에서는 **ovs_stats** 플러그인을 활성화하는 방법을 보여줍니다. OVS를 사용하여 오버클라우드를 배포하는 경우 **ovs_stats** 플러그인을 활성화할 필요가 없습니다.

```
parameter_defaults:
  CollectdExtraPlugins:
    - ovs_stats
  ExtraConfig:
    collectd::plugin::ovs_stats::socket: '/run/openvswitch/db.sock'
```

추가 리소스

- **ovs_stats** 플러그인 구성에 대한 자세한 내용은 ovs_stats 를 참조하십시오.

4.24. COLLECTD::PLUGIN::PROCESSES

프로세스 플러그인은 시스템 프로세스에 대한 정보를 제공합니다. 사용자 정의 프로세스 일치를 지정하지 않으면 플러그인은 프로세스 수와 프로세스 포크 비율만 수집합니다.

특정 프로세스에 대한 자세한 정보를 수집하려면 **process** 매개변수를 사용하여 프로세스 이름 또는 **process_match** 옵션을 지정하여 정규식과 일치하는 프로세스 이름을 지정할 수 있습니다. **process_match** 출력의 통계는 프로세스 이름으로 그룹화됩니다.

> This plugin is enabled by default.

표 4.26. 플러그인 매개변수

매개변수	유형	기본값
프로세스	array	*<undefined>*
process_matches	array	*<undefined>*
collect_context_switch	부울	*<undefined>*
collect_file_descriptor	부울	*<undefined>*

매개변수	유형	기본값
collect_memory_maps	부울	*<undefined>*

추가 리소스

- **프로세스** 플러그인 구성에 대한 자세한 내용은 프로세스를 참조하십시오.

4.25. COLLECTD::PLUGIN::SMART

스마트 플러그인을 사용하여 노드의 물리적 디스크에서 SMART(자체 모니터링, 분석 및 보고 기술) 정보를 수집합니다. 또한 **스마트** 플러그인이 SMART Telemetry를 읽을 수 있도록 **CollectdContainerAdditionalCapAdd** 매개변수를 **CAP_SYS_RAWIO** 로 설정해야 합니다. **CollectdContainerAdditionalCapAdd** 매개변수를 설정하지 않으면 collectd 오류 로그에 다음 메시지가 기록됩니다.

스마트 플러그인: collectd를 root로 실행하지만 CAP_SYS_RAWIO 기능이 없습니다. 플러그인의 읽기 기능이 실패할 수 있습니다. init 시스템이 기능 삭제입니까?.

표 4.27. 스마트 매개변수

매개변수	유형
디스크	array
ignoreselected	부울
간격	정수

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
  - smart

  CollectdContainerAdditionalCapAdd: "CAP_SYS_RAWIO"
```

추가 정보

- 스마트 플러그인 구성에 대한 자세한 내용은 **smart** 를 참조하십시오. https://collectd.org/documentation/manpages/collectd.conf.5.shtml#plugin_smart

4.26. COLLECTD::PLUGIN::SWAP

스왑 플러그인을 사용하여 사용 가능한 스왑 공간에 대한 정보를 수집합니다.

표 4.28. swap 매개변수

매개변수	유형
reportbydevice	부울
reportbytes	부울
valuesabsolute	부울
valuespercentage	부울
reportio	부울

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
  - swap

  ExtraConfig:
    collectd::plugin::swap::reportbydevice: false
    collectd::plugin::swap::reportbytes: true
    collectd::plugin::swap::valuesabsolute: true
    collectd::plugin::swap::valuespercentage: false
    collectd::plugin::swap::reportio: true
```

4.27. COLLECTD::PLUGIN::TCPCONNS

tcpconns 플러그인을 사용하여 구성된 포트에서 인바운드 또는 아웃바운드로 연결된 TCP 연결 수에 대한 정보를 수집합니다. 로컬 포트 구성은 수신 연결을 나타냅니다. 원격 포트 구성은 송신 연결을 나타냅니다.

표 4.29. tcpconns 매개변수

매개변수	유형
localports	포트(Array)
remoteports	포트(Array)
수신 대기	부울
allportssummary	부울

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
  - tcpconns
```

```
ExtraConfig:
  collectd::plugin::tcpconns::listening: false
  collectd::plugin::tcpconns::localports:
  - 22
  collectd::plugin::tcpconns::remoteports:
  - 22
```

4.28. COLLECTD::PLUGIN::THERMAL

열 플러그인을 사용하여 ACPI 열 영역 정보를 검색합니다.

표 4.30. 열 매개변수

매개변수	유형
devices	array
ignoreselected	부울
간격	정수

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
  - thermal
```

4.29. COLLECTD::PLUGIN::UPTIME

가동 시간 플러그인을 사용하여 시스템 가동 시간에 대한 정보를 수집합니다.

This plugin is enabled by default.

표 4.31. uptime 매개변수

매개변수	유형
간격	정수

4.30. COLLECTD::PLUGIN::VIRT

virt 플러그인을 사용하여 호스트의 가상 머신의 **libvirt** API를 통해 CPU, 디스크, 네트워크 로드 및 기타 지표를 수집합니다.

이 플러그인은 컴퓨팅 호스트에서 기본적으로 활성화되어 있습니다.

표 4.32. virt 매개변수

매개변수	유형
연결	문자열
refresh_interval	hash
domain	문자열
block_device	문자열
interface_device	문자열
ignore_selected	부울
plugin_instance_format	문자열
hostname_format	문자열
interface_format	문자열
extra_stats	문자열

설정 예:

```
ExtraConfig:
  collectd::plugin::virt::hostname_format: "name uuid hostname"
  collectd::plugin::virt::plugin_instance_format: metadata
```

추가 리소스

virt 플러그인 구성에 대한 자세한 내용은 virt 을 참조하십시오.

4.31. COLLECTD::PLUGIN::VMEM

vmem 플러그인을 사용하여 커널 하위 시스템에서 가상 메모리에 대한 정보를 수집합니다.

표 4.33. vmem 매개변수

매개변수	유형
상세 정보	부울
간격	정수

설정 예:

```
parameter_defaults:
  CollectdExtraPlugins:
```

```
- vmem

ExtraConfig:
  collectd::plugin::vmem::verbose: true
```

4.32. COLLECTD::PLUGIN::WRITE_HTTP

write_http 출력 플러그인을 사용하여 POST 요청 및 JSON으로 메트릭을 사용하거나 **PUTVAL** 명령을 사용하여 HTTP 서버에 값을 제출합니다.

표 4.34. write_http 매개변수

매개변수	유형
확인	enum [*present*,*absent*]
노드	hash[String, Hash[String, Scalar]]
URL	hash[String, Hash[String, Scalar]]
manage_package	부울

설정 예:

```
parameter_defaults:
    CollectdExtraPlugins:
      - write_http
    ExtraConfig:
        collectd::plugin::write_http::nodes:
            collectd:
                url: “http://collectd.tld.org/collectd”
                metrics: true
                header: “X-Custom-Header: custom_value"
```

추가 리소스

- **write_http** 플러그인 구성에 대한 자세한 내용은 write_http 를 참조하십시오.

4.33. COLLECTD::PLUGIN::WRITE_KAFKA

write_kafka 플러그인을 사용하여 Kafka 주제로 값을 보냅니다. 하나 이상의 주제 블록을 사용하여 **write_kafka** 플러그인을 구성합니다. 각 topic 블록에 대해 고유한 이름과 Kafka 생산자를 지정해야 합니다. topic 블록 내에서 다음 주제별 매개변수를 사용할 수 있습니다.

표 4.35. write_kafka parameters

매개변수	유형
kafka_hosts	array[String]

매개변수	유형
주제	hash
속성	hash
meta	hash

설정 예:

```
parameter_defaults:
    CollectdExtraPlugins:
      - write_kafka
    ExtraConfig:
      collectd::plugin::write_kafka::kafka_hosts:
        - remote.tld:9092
      collectd::plugin::write_kafka::topics:
        mytopic:
          format: JSON
```

추가 리소스:

write_kafka 플러그인을 구성하는 방법에 대한 자세한 내용은 write_kafka 를 참조하십시오.