Engineering
Simplicity

# Contrail Insights™

# Contrail Insights User Guide

Published
2023-10-03

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

*Contrail Insights™ Contrail Insights User Guide*

The information in this document is current as of the date on the title page.

## YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

## END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# Table of Contents

# About This Guide

Use this guide to understand the features and tasks that you can configure and perform from the Contrail Insights (formerly known as AppFormix) GUI. Contrail Insights manages intent-driven operations, visibility, and reporting for Multicloud and Network Functions Virtualization (NFV).

1

CHAPTER

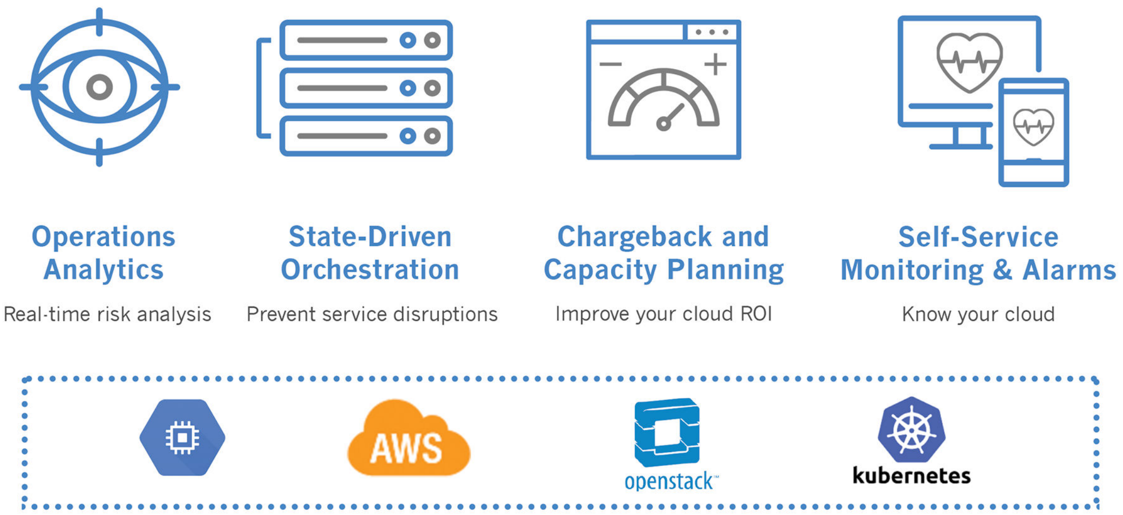
# Introduction

# Contrail Insights Overview

Contrail Insights enables operators to control and visualize how infrastructure resources are utilized by workloads, and plan adequate capacity to ensure application performance. Using Contrail Insights, operators of software-defined data centers have a toolset for visibility into operational performance and infrastructure resources.

Figure 1 on page 2 shows the product modules for the Contrail Insights optimization and management software platform.

**Figure 1: Product Modules**


Operations Analytics
Real-time risk analysis
State-Driven Orchestration
Prevent service disruptions
Chargeback and Capacity Planning
Improve your cloud ROI
Self-Service Monitoring & Alarms
Know your cloud
AWS
openstack
kubernetes

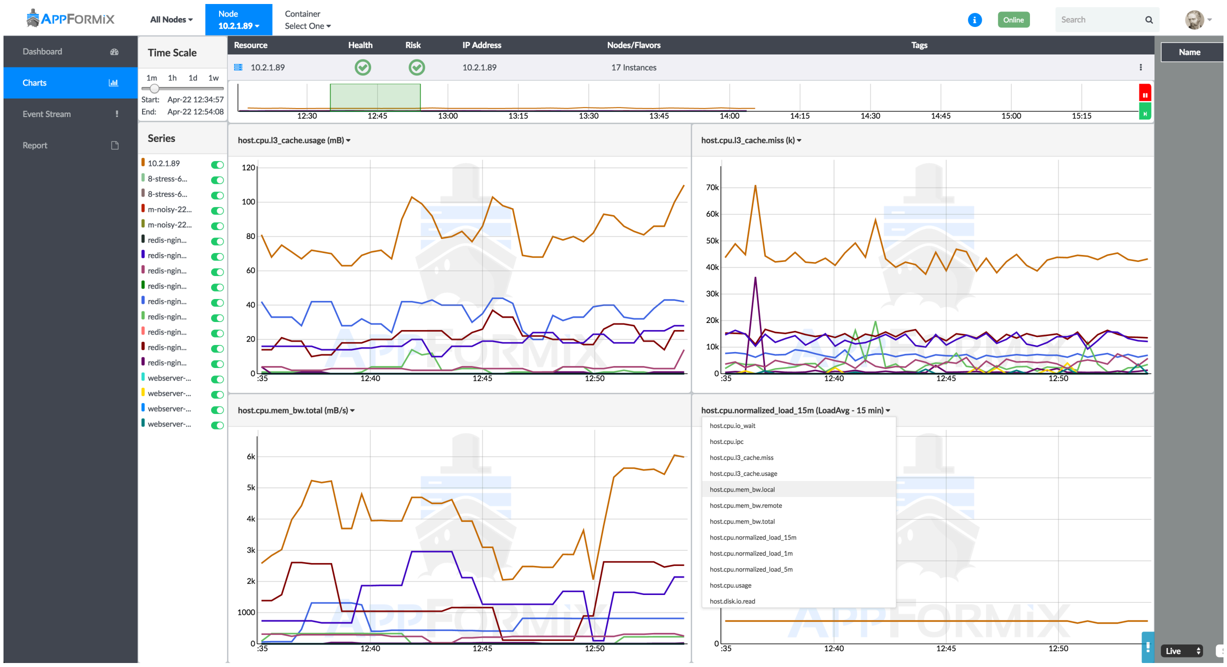
Juniper Networks Contrail Insights® is a cloud service optimization tool that provides advanced monitoring, scheduling, and performance management for software-defined infrastructure, where containers and virtual machines (VMs) can have life cycles much shorter than in traditional development environments.

The Contrail Insights software leverages big-data analytics and machine learning in a distributed architecture that puts the power of self-driving infrastructure at the core of almost any cloud. Contrail Insights redefines the state-of-the-art in telemetry and management across software-defined infrastructure and application software layers. In addition, real-time and historic monitoring, performance visibility and dynamic optimization features improve cloud orchestration, security, accounting, and planning. The following video provides an overview of the Contrail Insights infrastructure dashboard.

**Video:** Contrail Insights Dashboard

Contrail Insights operates in private enterprise cloud environments built on platforms such as OpenStack and Kubernetes. Contrail Insights accommodates both containers and virtual machines to support multitenant, dynamic, and constantly evolving enterprise clouds. Figure 2 on page 3 shows real-time CPU utilization in chart format for a specified host.

**Figure 2: Real-Time CPU Utilization Chart for a Specified Host**

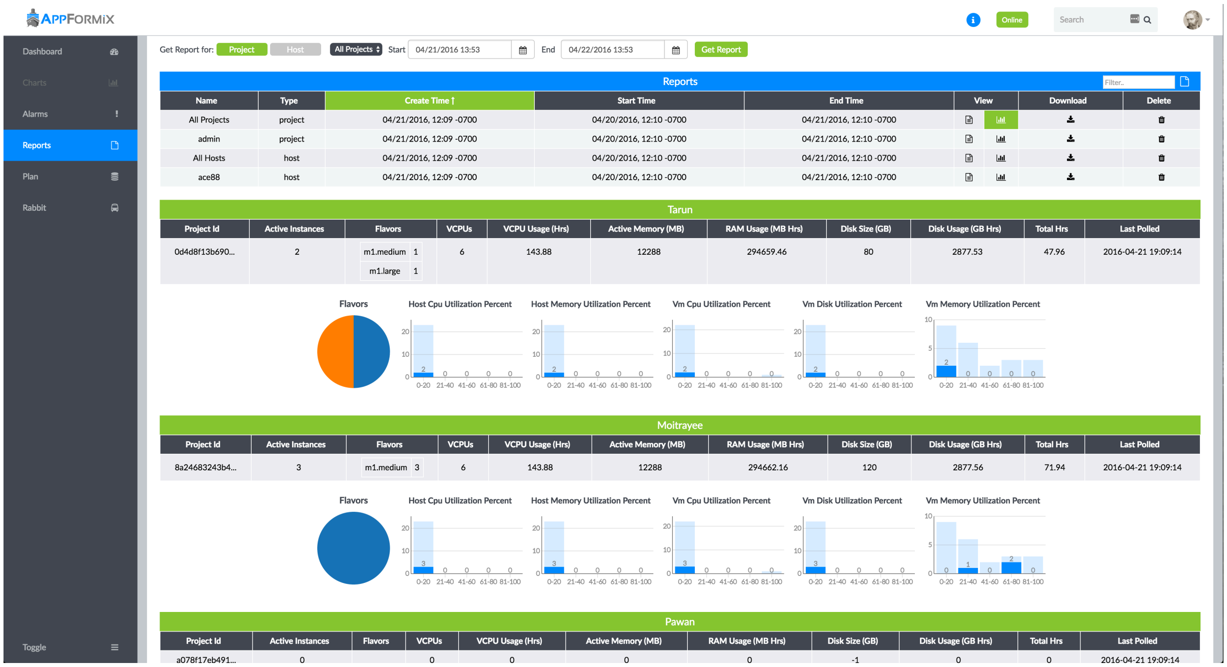


| Resource | Health | Risk | IP Address | Nodes/Flavors | Tags |
|---|---|---|---|---|---|
| 10.2.1.89 | ✓ | ✓ | 10.2.1.89 | 17 Instances | |

Time Scale: 1m 1h 1d 1w
Start: Apr-22 12:34:57
End: Apr-22 12:54:08

Contrail Insights analyzes metrics in real-time across all aspects of shared infrastructure—compute, storage, and networking—and associates resource consumption to containers and virtual machines. Figure 3 on page 4 shows a report that charts the project CPU and memory utilization percentage for specified dates.

## Figure 3: Report Showing Percentage of Project CPU and Memory Utilization


APPFORMIX
Online
Search
Dashboard
Charts
Alarms
Reports
Plan
Rabbit
Toggle
Get Report for: Project Host All Projects
Get Report
Filter..
Flavors
Host Cpu Utilization Percent
Host Memory Utilization Percent
Vm Cpu Utilization Percent
Vm Disk Utilization Percent
Vm Memory Utilization Percent
0-20 21-40 41-60 61-80 81-100

Start 04/21/2016 13:53 End 04/22/2016 13:53

Reports

| Name | Type | Create Time ↑ | Start Time | End Time | View | Download | Delete |
|---|---|---|---|---|---|---|---|
| All Projects | project | 04/21/2016, 12:09 -0700 | 04/20/2016, 12:10 -0700 | 04/21/2016, 12:10 -0700 | | | |
| admin | project | 04/21/2016, 12:09 -0700 | 04/20/2016, 12:10 -0700 | 04/21/2016, 12:10 -0700 | | | |
| All Hosts | host | 04/21/2016, 12:09 -0700 | 04/20/2016, 12:10 -0700 | 04/21/2016, 12:10 -0700 | | | |
| ace88 | host | 04/21/2016, 12:09 -0700 | 04/20/2016, 12:10 -0700 | 04/21/2016, 12:10 -0700 | | | |

Tarun

| Project Id | Active Instances | Flavors | VCPUs | VCPU Usage (Hrs) | Active Memory (MB) | RAM Usage (MB Hrs) | Disk Size (GB) | Disk Usage (GB Hrs) | Total Hrs | Last Polled |
|---|---|---|---|---|---|---|---|---|---|---|
| 0d4d8f13b690... | 2 | m1.medium 1<br>m1.large 1 | 6 | 143.88 | 12288 | 294659.46 | 80 | 2877.53 | 47.96 | 2016-04-21 19:09:14 |

Moitrayee

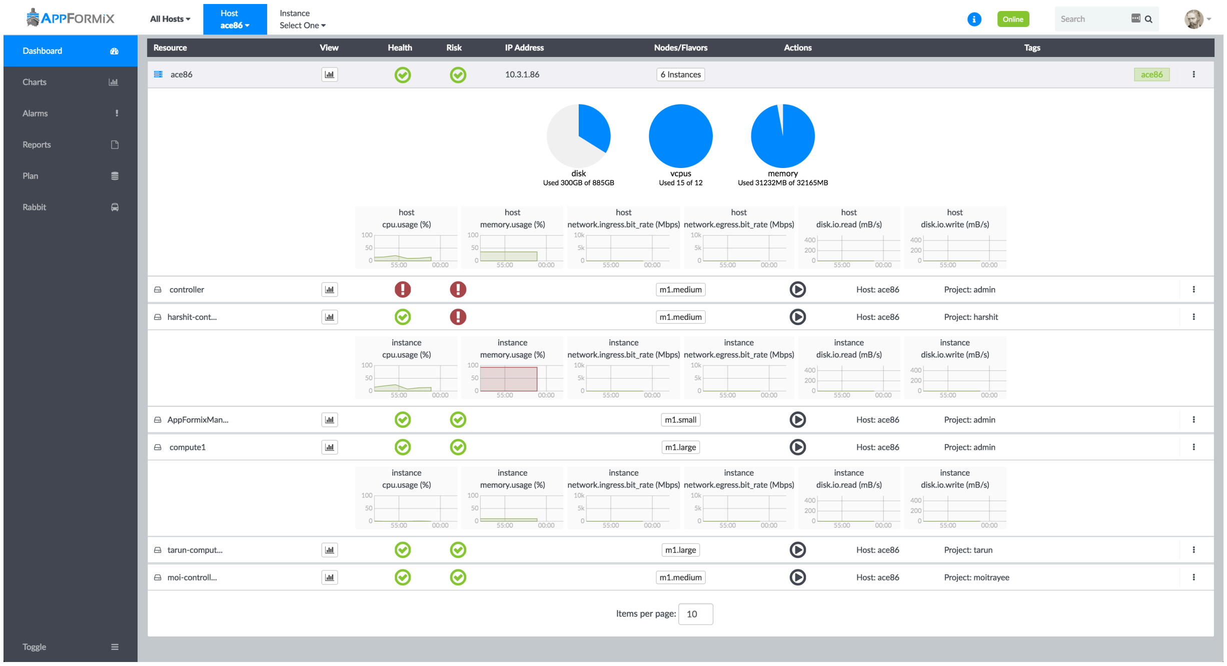
| Project Id | Active Instances | Flavors | VCPUs | VCPU Usage (Hrs) | Active Memory (MB) | RAM Usage (MB Hrs) | Disk Size (GB) | Disk Usage (GB Hrs) | Total Hrs | Last Polled |
|---|---|---|---|---|---|---|---|---|---|---|
| 8a24683243b4... | 3 | m1.medium 3 | 6 | 143.88 | 12288 | 294662.16 | 120 | 2877.56 | 71.94 | 2016-04-21 19:09:14 |

Pawan

| Project Id | Active Instances | Flavors | VCPUs | VCPU Usage (Hrs) | Active Memory (MB) | RAM Usage (MB Hrs) | Disk Size (GB) | Disk Usage (GB Hrs) | Total Hrs | Last Polled |
|---|---|---|---|---|---|---|---|---|---|---|
| a078f17eb491... | 0 | | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 2016-04-21 19:09:14 |

Figure 4 on page 4 shows the instances for host resources at a glance in the dashboard.

## Figure 4: Host Resources at a Glance in Dashboard


APPFORMIX
All Hosts
Host ace86
Instance Select One
Online
Search
Dashboard
Charts
Alarms
Reports
Plan
Rabbit
Toggle
disk
vcpus
memory
host cpu.usage (%)
host memory.usage (%)
host network.ingress.bit_rate (Mbps)
host network.egress.bit_rate (Mbps)
host disk.io.read (mB/s)
host disk.io.write (mB/s)
instance cpu.usage (%)
instance memory.usage (%)
instance network.ingress.bit_rate (Mbps)
instance network.egress.bit_rate (Mbps)
instance disk.io.read (mB/s)
instance disk.io.write (mB/s)
55:00
00:00

Used 300GB of 885GB

Used 15 of 12

Used 31232MB of 32165MB

| Resource | View | Health | Risk | IP Address | Nodes/Flavors | Actions | Tags |
|---|---|---|---|---|---|---|---|
| ace86 | | ✓ | ✓ | 10.3.1.86 | 6 Instances | | ace86 |
| controller | | ! | ! | | m1.medium | ▶ | Host: ace86<br>Project: admin |
| harshit-cont... | | ✓ | ! | | m1.medium | ▶ | Host: ace86<br>Project: harshit |
| AppFormixMan... | | ✓ | ✓ | | m1.small | ▶ | Host: ace86<br>Project: admin |
| compute1 | | ✓ | ✓ | | m1.large | ▶ | Host: ace86<br>Project: admin |
| tarun-comput... | | ✓ | ✓ | | m1.large | ▶ | Host: ace86<br>Project: tarun |
| moi-controll... | | ✓ | ✓ | | m1.medium | ▶ | Host: ace86<br>Project: moitrayee |

Items per page: 10

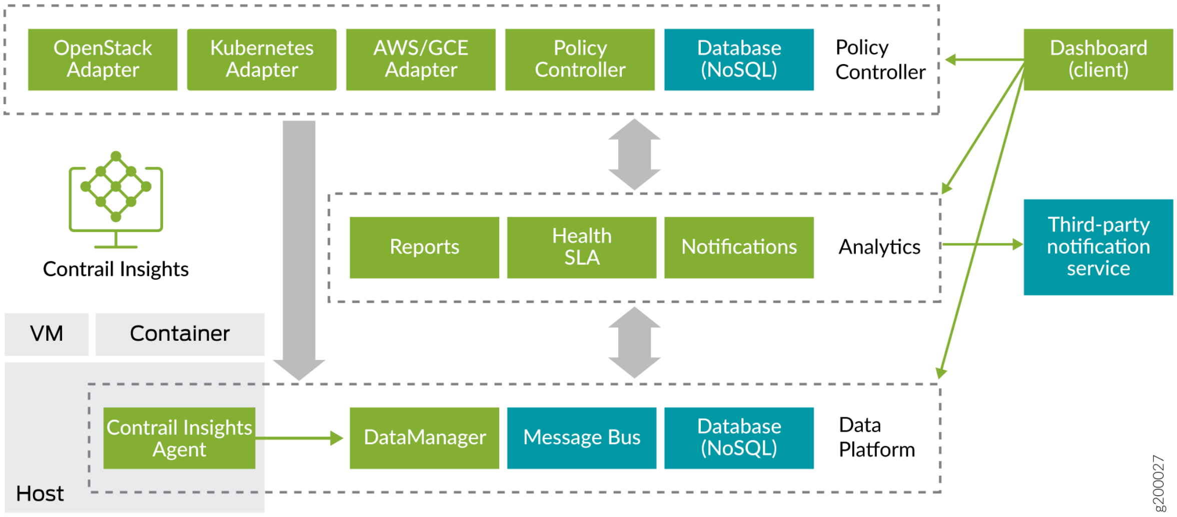
RELATED DOCUMENTATION

*Contrail Insights Architecture*

# Contrail Insights Architecture

Contrail Insights provides resource control and visibility for hosts, containers, and virtual machines in your cloud and network infrastructure. Figure 5 on page 5 shows the Contrail Insights architecture.

**Figure 5: Contrail Insights Architecture**



The software consists of multiple components:

- Contrail Insights Agent monitors resource usage on compute nodes.
- Contrail Insights controller offers REST APIs to configure the system.
- Contrail Insights DataManager stores data from multiple agents.
- Contrail Insights Dashboard provides a Web-based user interface.
- An adapter discovers platform-specific resources and configures controller. Adapters exist for OpenStack, Kubernetes, and NorthStar.

The agent component runs on the lowest level "compute nodes" of the infrastructure that provide computational resources to execute application workload. A compute node can be a bare-metal host or a virtual machine.

The remaining components run on a class of infrastructure node(s) that execute services that power software-defined infrastructure, such as the OpenStack infrastructure service nodes. A host on which Contrail Insights control plane components execute is a Platform Host (likely a virtual machine). A Platform Host requires network connectivity to all of the compute nodes and to infrastructure services that manage the infrastructure.

## RELATED DOCUMENTATION

*Contrail Insights Overview*

# 2
CHAPTER

# Configuration

# Aggregate of Network Device Entities

IN THIS SECTION

Each network device has multiple entities such as interfaces, kernel, power, fan, and so on. Contrail Insights supports creation of aggregate of network device entities.

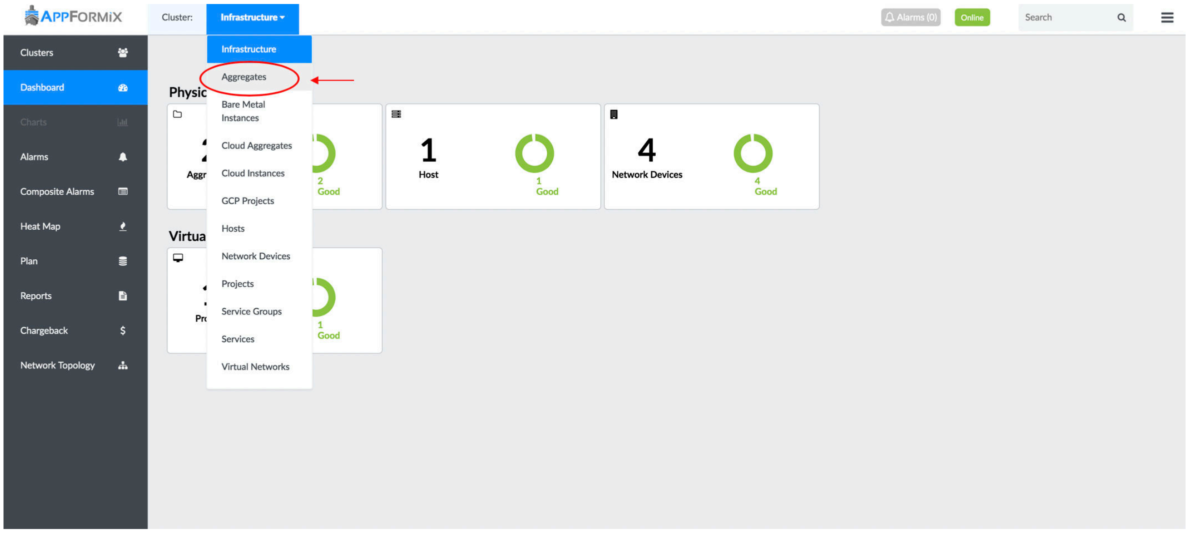
- Charts are viewable for the aggregate.
- Both static and dynamic rules are supported.
- SLA health and risk profiles for the aggregate are supported.

## Create an Aggregate of Network Device Entities

To create an aggregate of network device entities:

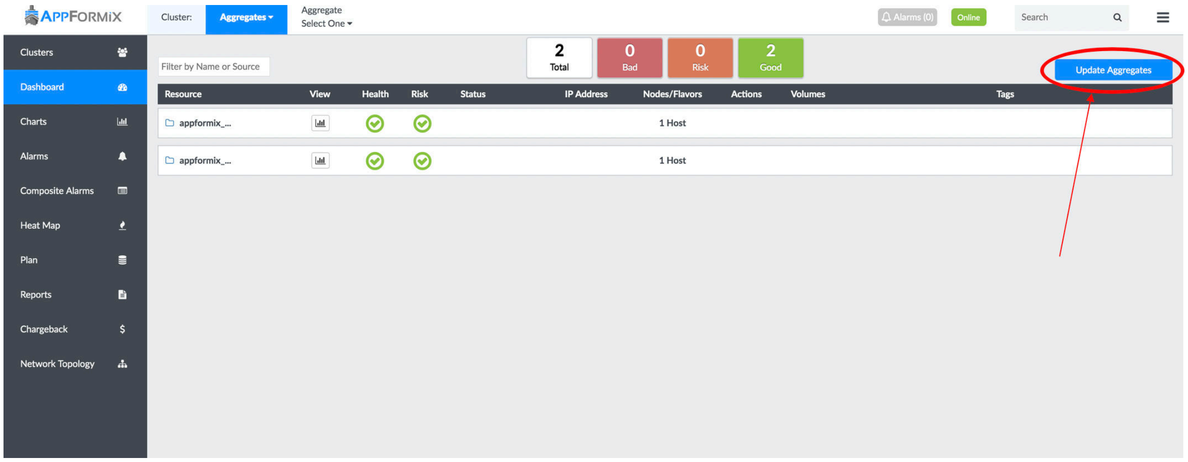
1. From the Contrail Insights Dashboard, select **Aggregates** from the drop-down list.

## Figure 6: Create Aggregates from Dashboard



2. Select **Update Aggregates**.

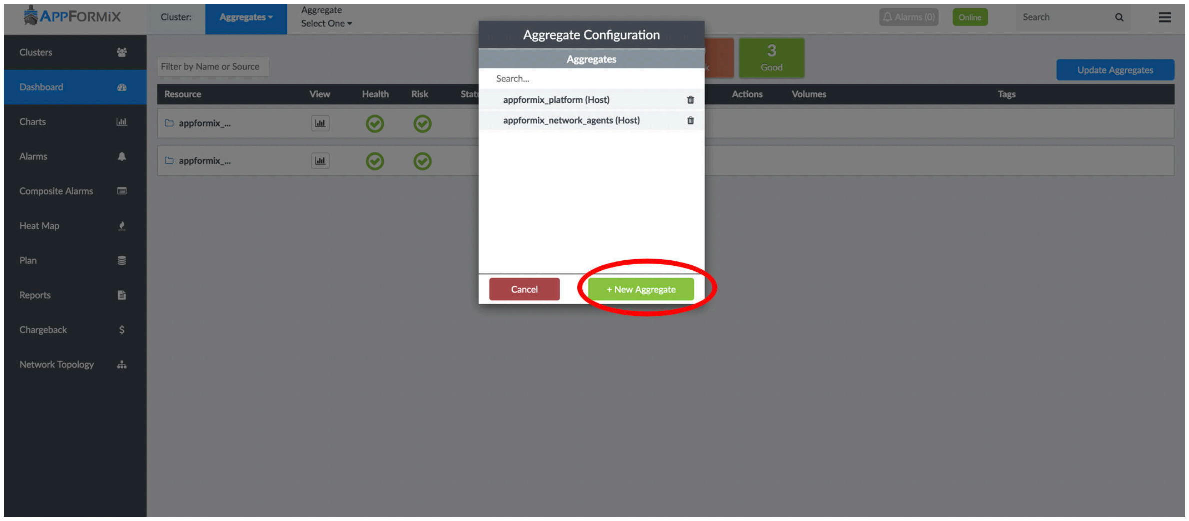
## Figure 7: Update Aggregates from Dashboard



| Resource | View | Health | Risk | Status | IP Address | Nodes/Flavors | Actions | Volumes | Tags |
|---|---|---|---|---|---|---|---|---|---|
| appformix_... | | ✓ | ✓ | | | 1 Host | | | |
| appformix_... | | ✓ | ✓ | | | 1 Host | | | |

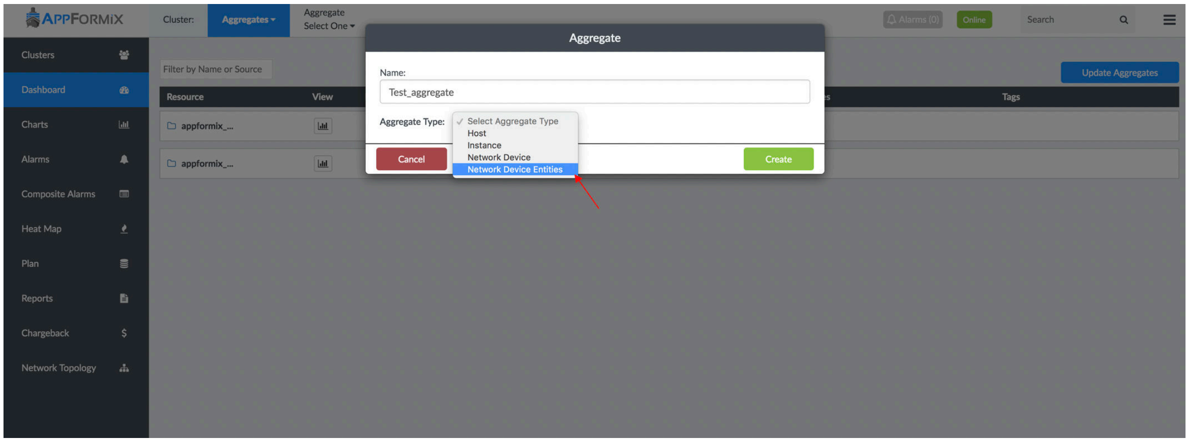
3. Select **New Aggregate**.

### Figure 8: Configure New Aggregate



4. Select the Aggregate type as **Network Device Entities**.

### Figure 9: Select Aggregate Type from Dashboard

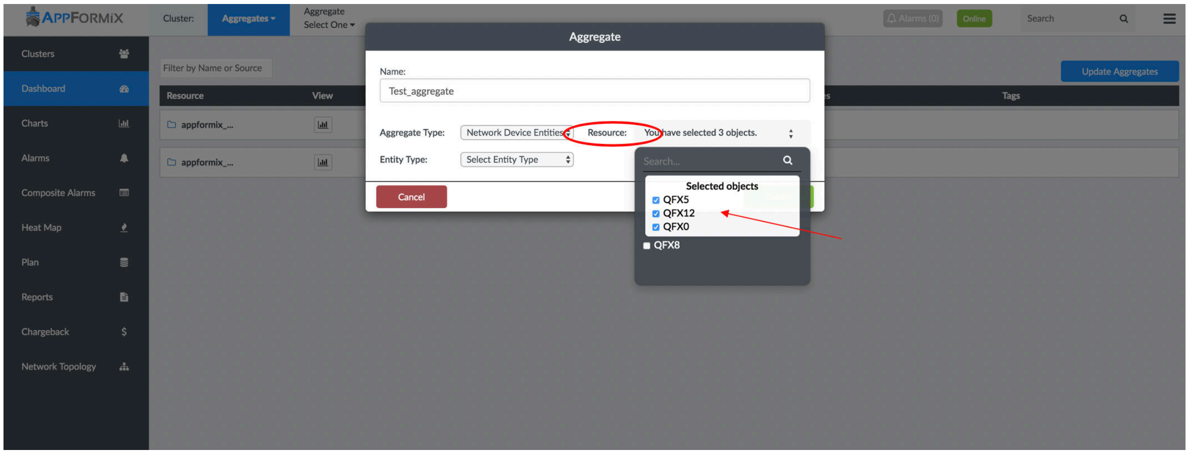


Name: Test_aggregate

Aggregate Type: Network Device Entities

5. Select the network devices you want to monitor by choosing from the Resource drop-down list.

### Figure 10: Select Network Devices to Monitor



Name: Test_aggregate

Aggregate Type: Network Device Entities

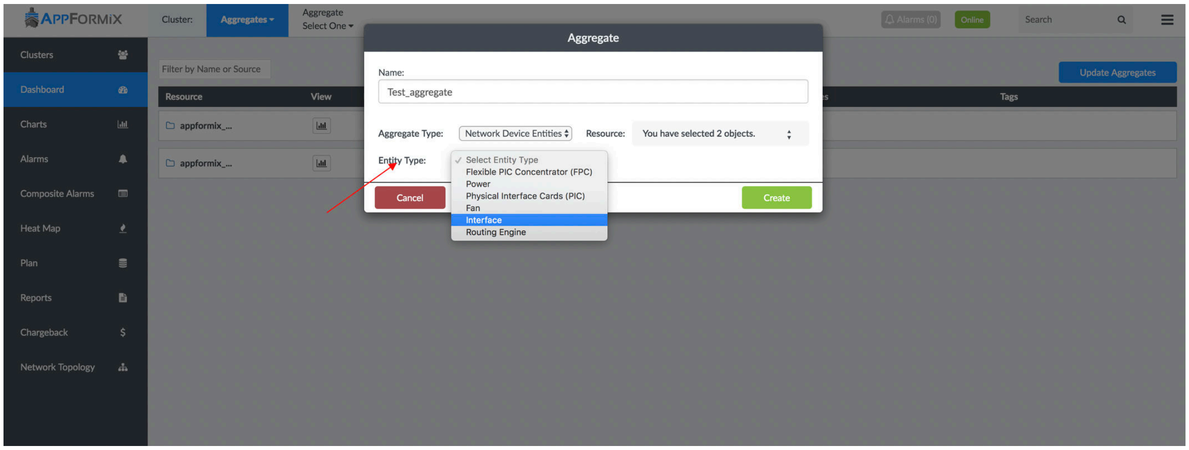
Resource: You have selected 3 objects.

Entity Type: Select Entity Type

☑ QFX5
☑ QFX12
☑ QFX0
☐ QFX8

6. Select the entity type to create an aggregate for.

### Figure 11: Aggregates Entity Types



Name: Test_aggregate

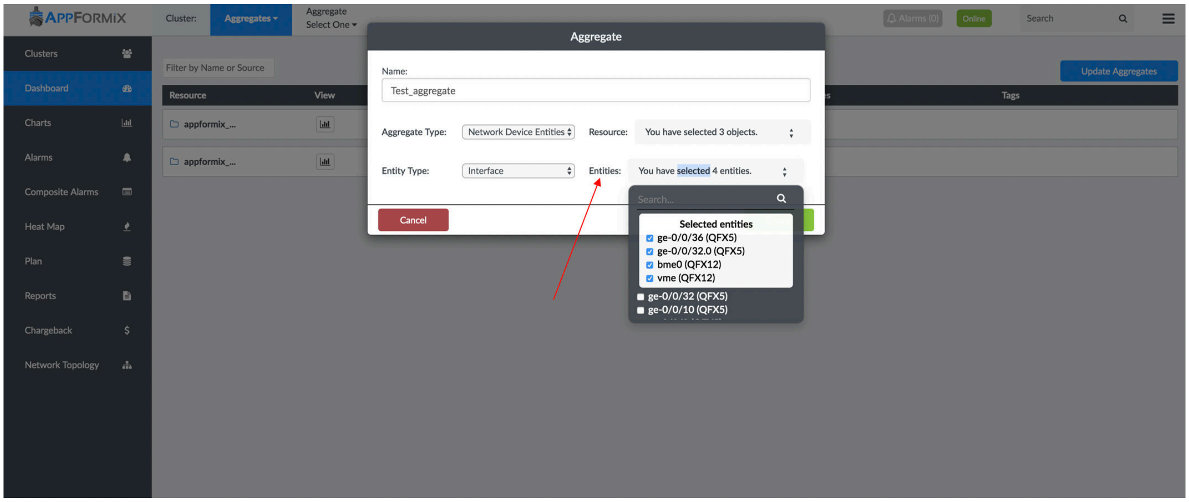
Aggregate Type: Network Device Entities

Resource: You have selected 2 objects.

Entity Type:
- ✓ Select Entity Type
- Flexible PIC Concentrator (FPC)
- Power
- Physical Interface Cards (PIC)
- Fan
- Interface
- Routing Engine

7. Select the entities that are to be monitored.

Figure 12: Select Entities to Monitor



Name: Test_aggregate

Aggregate Type: Network Device Entities

Resource: You have selected 3 objects.

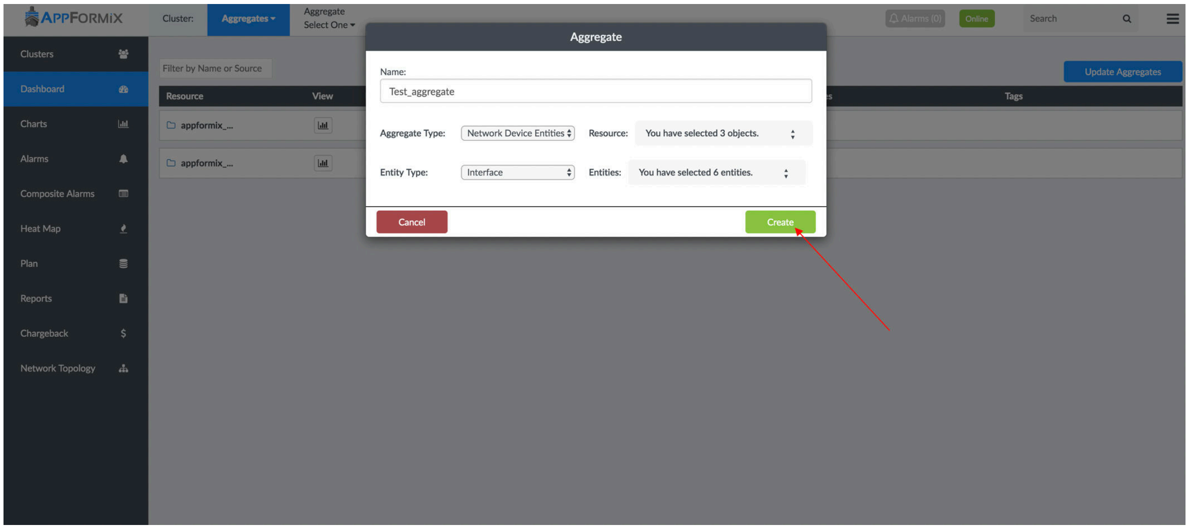
Entity Type: Interface

Entities: You have selected 4 entities.

Selected entities
- ☑ ge-0/0/36 (QFX5)
- ☑ ge-0/0/32.0 (QFX5)
- ☑ bme0 (QFX12)
- ☑ vme (QFX12)
- □ ge-0/0/32 (QFX5)
- □ ge-0/0/10 (QFX5)

8. Click **Create** to create the aggregate.

Figure 13: Create Aggregate - Dashboard View



Name: Test_aggregate

Aggregate Type: Network Device Entities

Resource: You have selected 3 objects.

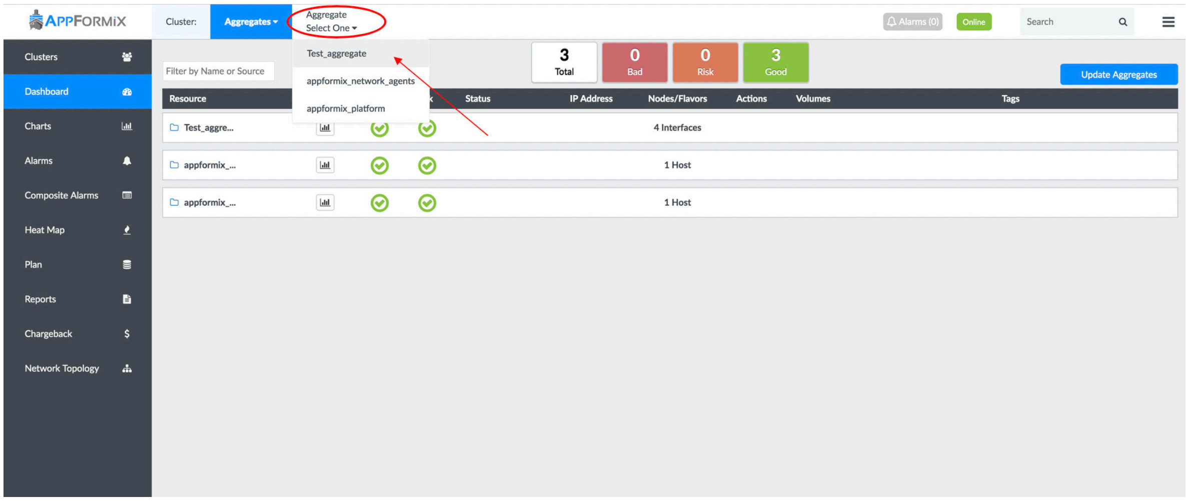
Entity Type: Interface

Entities: You have selected 6 entities.

## View Charts for Aggregate of Network Device Entities

1. From the Contrail Insights Dashboard, select **Infrastructure** > **Aggregates**. Then select the aggregate you created.

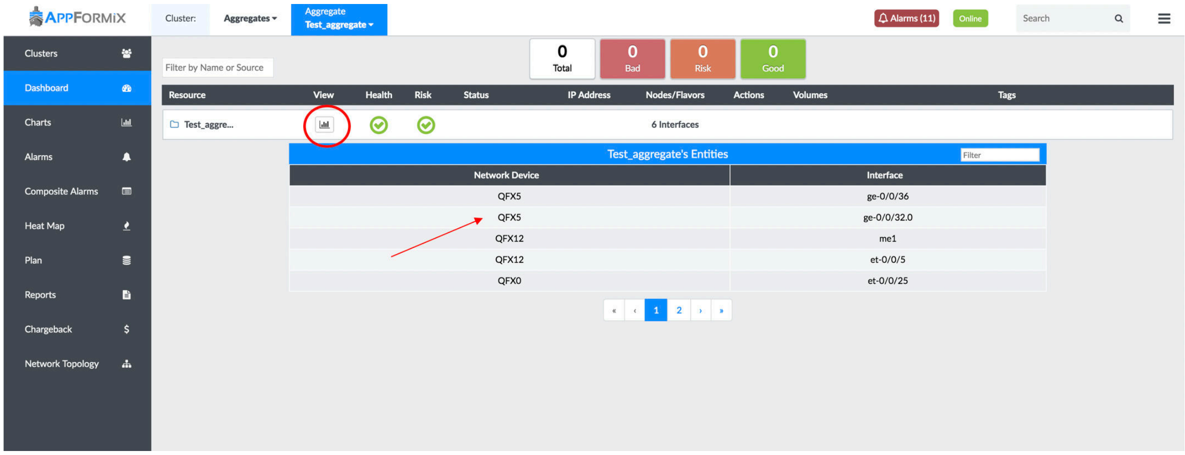
### Figure 14: Select a Created Aggregate



| Resource | Status | IP Address | Nodes/Flavors | Actions | Volumes | Tags |
|---|---|---|---|---|---|---|
| Test_aggre... | | | 4 Interfaces | | | |
| appformix_... | | | 1 Host | | | |
| appformix_... | | | 1 Host | | | |

2. The members of the aggregate are displayed. Select **Charts** to view charts.

### Figure 15: View Members of Aggregate



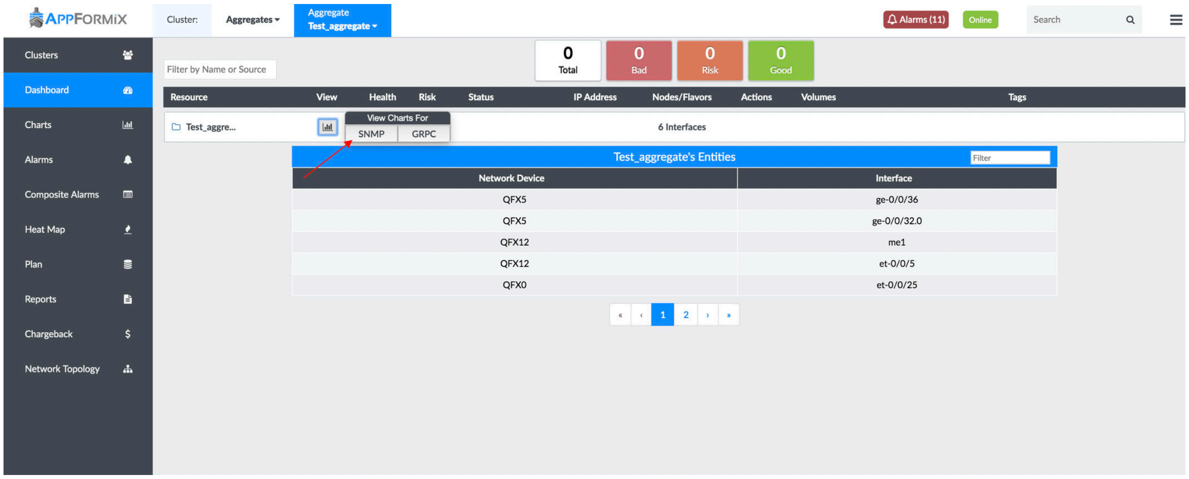
| Resource | View | Health | Risk | Status | IP Address | Nodes/Flavors | Actions | Volumes | Tags |
|---|---|---|---|---|---|---|---|---|---|
| Test_aggre... | | ✓ | ✓ | | | 6 Interfaces | | | |

Test_aggregate's Entities

| Network Device | Interface |
|---|---|
| QFX5 | ge-0/0/36 |
| QFX5 | ge-0/0/32.0 |
| QFX12 | me1 |
| QFX12 | et-0/0/5 |
| QFX0 | et-0/0/25 |

3. If there are multiple sources, for example, **SNMP**, **GRPC**, **JTI**, select the one you want to view charts for.

### Figure 16: Select Source for Chart Display

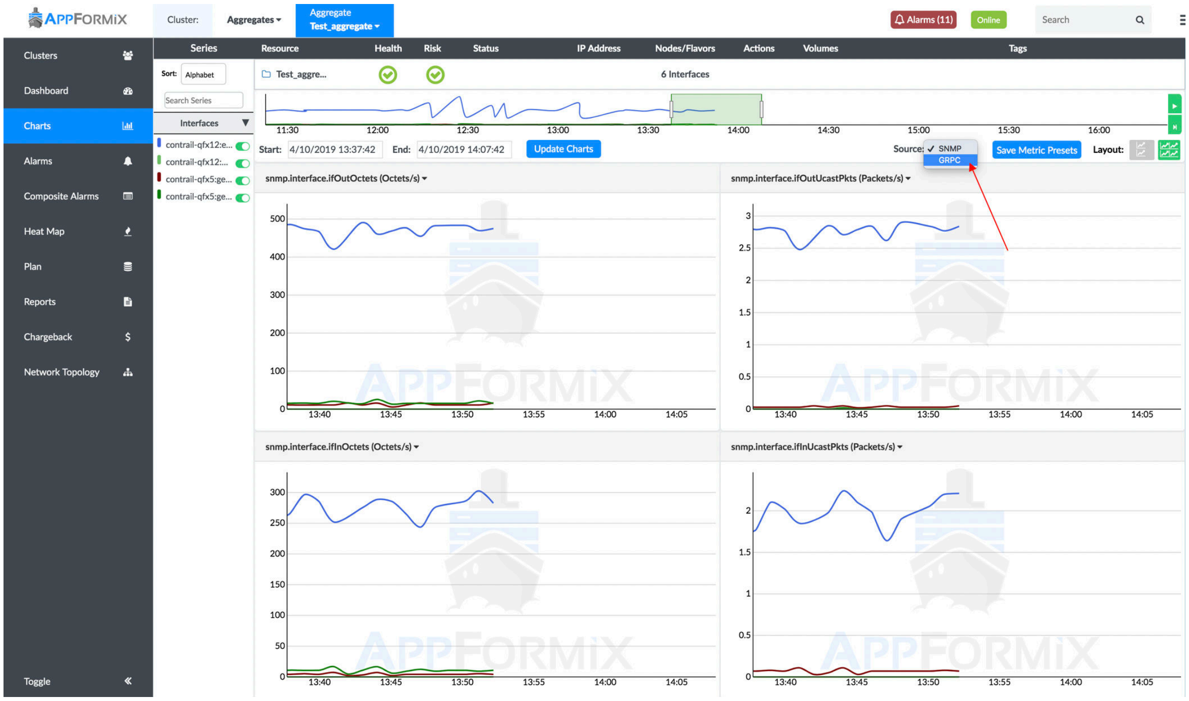


Test_aggregate's Entities

| Network Device | Interface |
|---|---|
| QFX5 | ge-0/0/36 |
| QFX5 | ge-0/0/32.0 |
| QFX12 | me1 |
| QFX12 | et-0/0/5 |
| QFX0 | et-0/0/25 |

4. After selecting a source, charts are displayed. To view charts for other sources, select the source from the drop down.

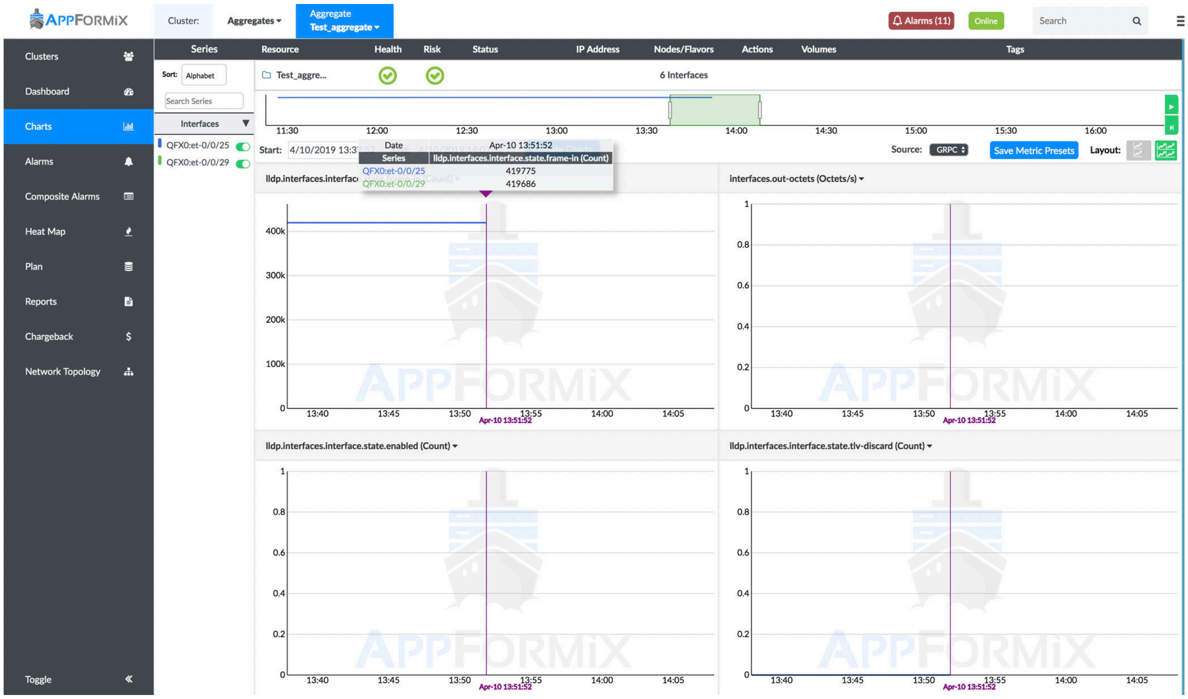
### Figure 17: Aggregate Charts



Start: 4/10/2019 13:37:42 End: 4/10/2019 14:07:42

Source: ✓ SNMP

**Figure 18: Select a Different Source from Charts**



## RELATED DOCUMENTATION

Metrics Collected by Contrail Insights | 106

# Aggregate Discovery and Alarms with OpenStack Heat Services

## IN THIS SECTION

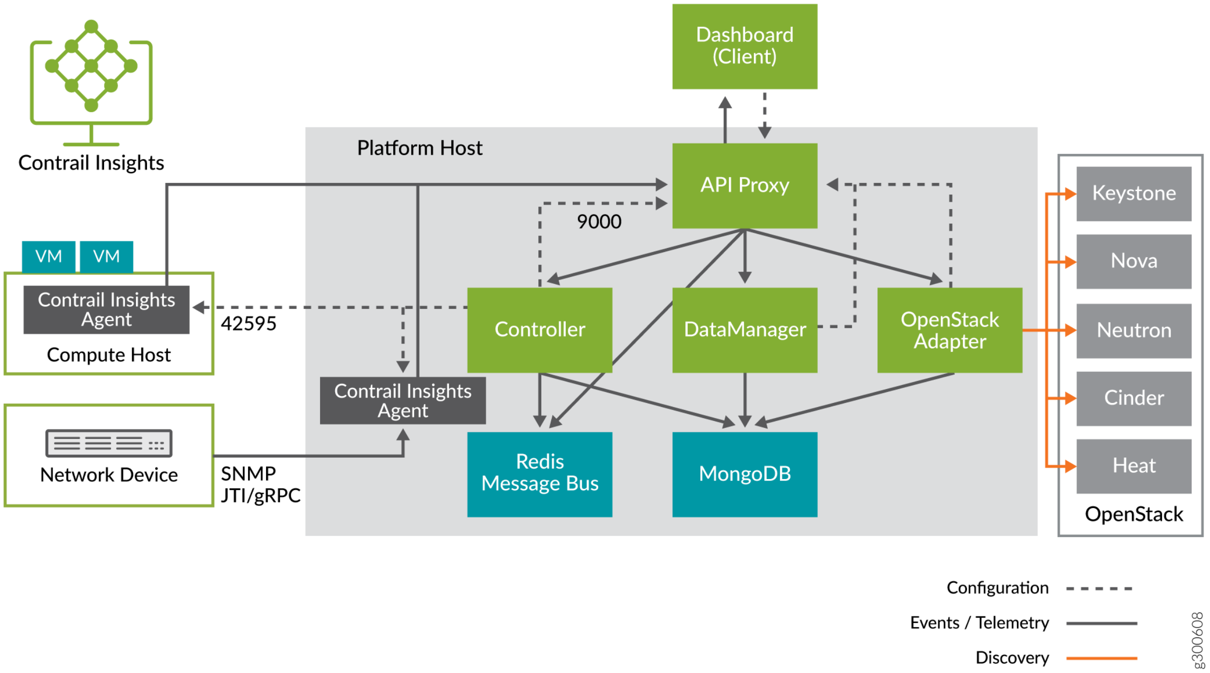
● Troubleshooting | 34

Contrail Insights integration with Heat has two independent aspects: discovery and alarms.

## Heat Stack Discovery

The Contrail Insights OpenStack Adapter uses Heat APIs to discover Heat stacks in an OpenStack cluster. Each stack is represented as an aggregate in Contrail Insights with the label **Heat**. When OpenStack Adapter discovers a Heat stack, OpenStack Adapter configures an aggregate in the Contrail Insights Platform, and adds any virtual machines and virtual networks defined by the Heat stack as members of the aggregate. See Figure 19 on page 16.

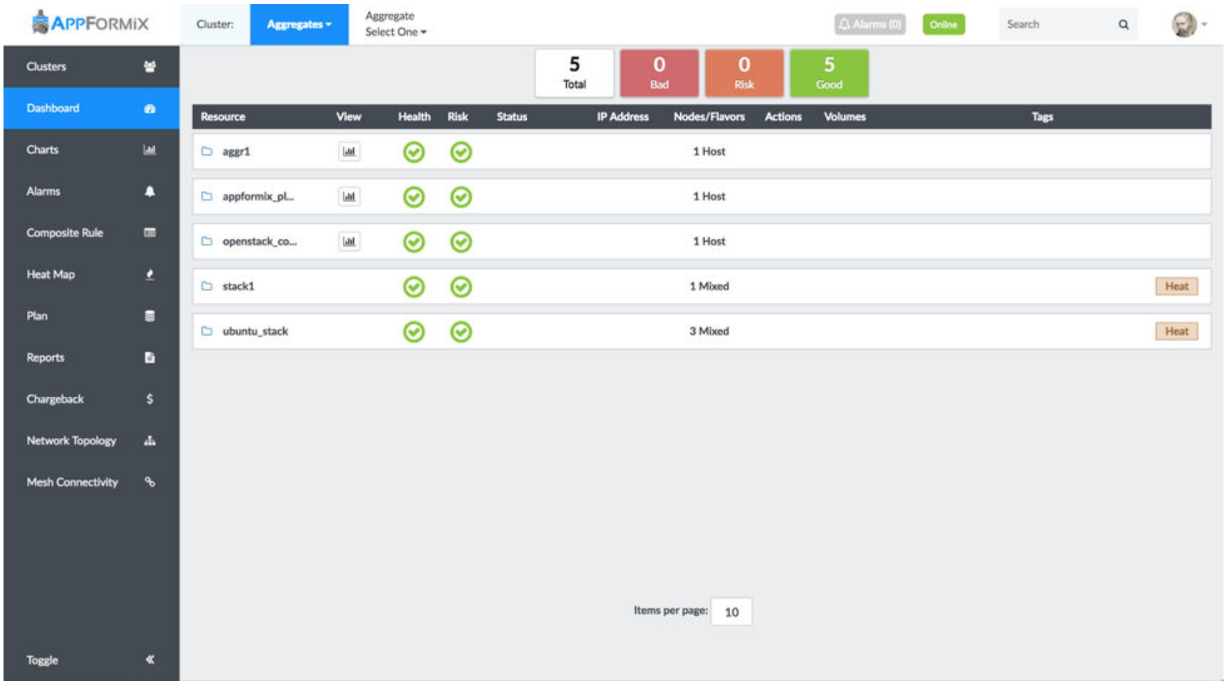
**Figure 19: Heat Stack Discovery in Contrail Insights**



Discovery functionality is provided by Contrail Insights OpenStack Adapter as part of the standard OpenStack integration. Discovery does not require any modifications to the OpenStack controller, OpenStack configuration, or Heat service, and does not require installation of the Contrail Insights Heat plug-in.

Figure 20 on page 17 shows two Heat stacks that were discovered by Contrail Insights: **stack1** and **ubuntu_stack**. Each stack is represented in Contrail Insights as a **Mixed** aggregate. A **Mixed** aggregate may contain entities of different types, such as virtual machine and virtual network. To view aggregate Heat stacks, select **Dashboard**, then from the Infrastructure tab, select **Aggregates**.

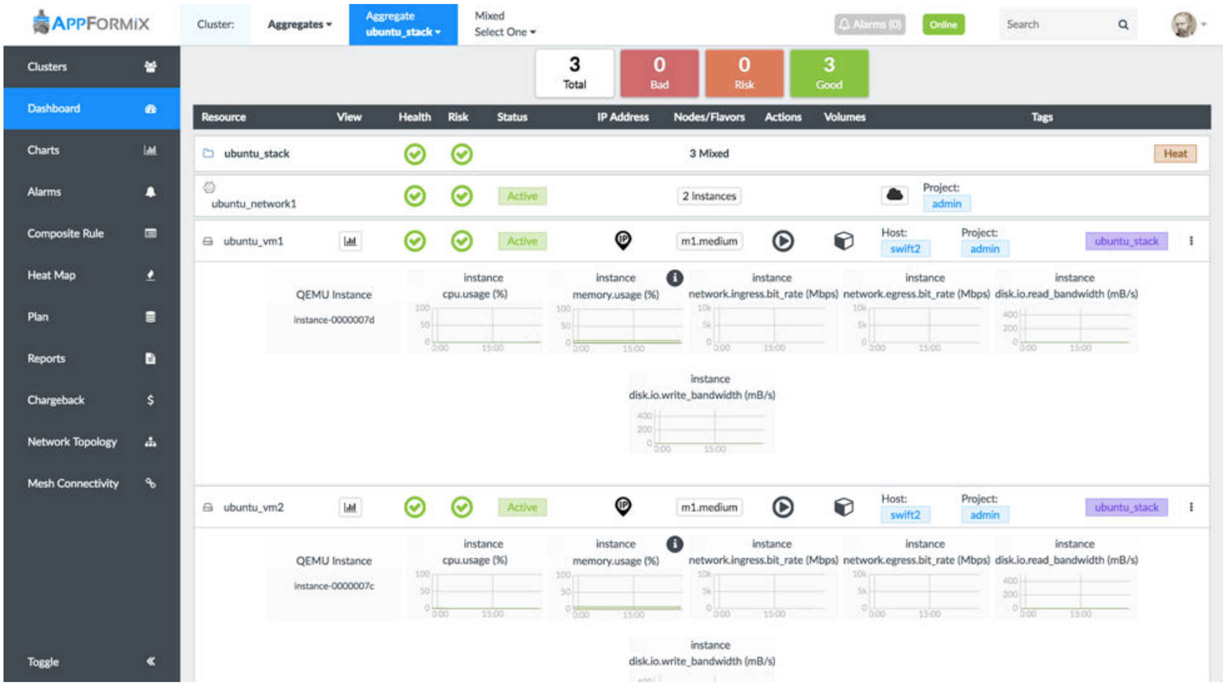
**Figure 20: Mixed Aggregate Heat Stacks in Contrail Insights**



| Resource | View | Health | Risk | Status | IP Address | Nodes/Flavors | Actions | Volumes | Tags |
|---|---|---|---|---|---|---|---|---|---|
| aggr1 | 📊 | ✓ | ✓ | | | 1 Host | | | |
| appformix_pl... | 📊 | ✓ | ✓ | | | 1 Host | | | |
| openstack_co... | 📊 | ✓ | ✓ | | | 1 Host | | | |
| stack1 | | ✓ | ✓ | | | 1 Mixed | | | Heat |
| ubuntu_stack | | ✓ | ✓ | | | 3 Mixed | | | Heat |

Figure 21 on page 18 shows the entities in the Heat stack **ubuntu_stack**. There are two virtual machines, **ubuntu_vm1** and **ubuntu_vm2**, and one virtual network, **ubuntu_network1**. To view this page, select **Dashboard**, then from the Infrastructure tab, select **Aggregates**. From the **Aggregate Select One** tab, select the entity to view.

Figure 21: Virtual Machines in Heat Stacks in Contrail Insights



## Contrail Insights Alarm Resource Types

In addition to discovery of Heat stacks, Contrail Insights provides a Heat plug-in that defines two new resource types for Heat templates.

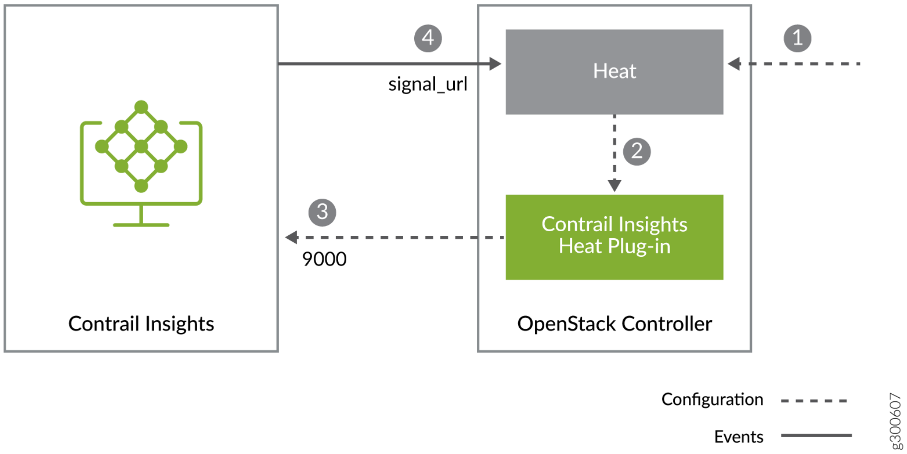
OS::AppFormix::Alarm
: This resource type is used to define single alarms for monitoring resources in a Heat stack.

OS::AppFormix::CompositeAlarm
: This resource type is used to define composite alarms for monitoring resources in a Heat stack.

The alarms and composite alarms are configured in Contrail Insights when a Heat stack is created from the template, and are evaluated by the Contrail Insights stream-based, distributed analysis engine.

You benefit by maintaining monitoring configuration in the same template that defines the resources (for example, virtual machines) to be monitored. Contrail Insights alarms and composite alarms are configured at the time that resources in the Heat stack are instantiated. Further, these alarms can be used to trigger scale-up and scale-down of Heat AutoScaling policies, which enables Heat to react more quickly and accurately due to the responsive and fine-grained Contrail Insights alarms.

**Figure 22: Interaction Sequence between Heat, Contrail Insights Heat Plug-In, and Contrail Insights Platform**



When a Heat stack is created using a Heat template that has an `OS::AppFormix::Alarm` or `OS::AppFormix::CompositeAlarm` resource, Heat will pass the resource properties to Contrail Insights Heat plug-in to configure the alarm in Contrail Insights. Figure 22 on page 19 illustrates the interaction between Heat, Contrail Insights Heat plug-in, and Contrail Insights Platform in the following sequence of events.

1. User instantiates a Heat stack from a template with an `OS::AppFormix::Alarm` or `OS::AppFormix::CompositeAlarm` resource.

2. Heat passes the alarm properties to Contrail Insights Heat plug-in.

3. When the resource `OS::AppFormix::Alarm` is used, Contrail Insights Heat plug-in configures an alarm in Contrail Insights Platform using the Contrail Insights REST API. The URL for Contrail Insights Platform is a configuration parameter (see “Install Contrail Insights Heat Plug-in”). The alarm is configured with mode set to `Event`. The alarm will generate a notification for each interval in which the condition for the alarm is satisfied.

4. When the resource `OS::AppFormix::CompositeAlarm` is used, Contrail Insights Heat plug-in configures a composite alarm in Contrail Insights Platform using the Contrail Insights REST API. The user can define multiple individual alarms in the composite alarm. The state of the composite alarm is a combination of the states of the individual alarms. The user can define weights for the individual alarms and a threshold for the composite alarm. The composite alarm is active when the sum of the weights of the active alarms equals or exceeds the user-defined threshold (see “Example: Heat Autoscaling with OS::AppFormix::Alarm”). A notification will be generated every 60 seconds for as long as the composite alarm is active.

5. When the alarm or composite alarm triggers, a notification is delivered by HTTP/HTTPS POST to an endpoint specified in the `notification_url` property of the alarm. To enable auto-scaling, a Heat

template can specify the `signal_url` of a Heat `ScalingPolicy` resource as the `notification_url`. In that case, the notification is sent to Heat for processing.

To make the `OS::AppFormix::Alarm` and `OS::AppFormix::CompositeAlarm` resource types available to Heat templates, the Contrail Insights Heat plug-in must be installed and configured on the OpenStack controller host(s). See the following section, "Install Contrail Insights Heat Plug-in."

For more information about the extensible design of Heat resources using plug-ins, refer to Heat documentation.

## Install Contrail Insights Heat Plug-In

To install and configure the Heat plug-in on the OpenStack controller host:

1. Copy `appformix-openstack` package.

   Copy the `appformix_openstack` Python package from the release bundle to the OpenStack controller host on which Heat service is running. This package is provided as either a Python wheel or an RPM package in the `pkg` directory of the release bundle.

2. Install `appformix-openstack` package.

   On the OpenStack controller host that runs the Heat service, install the `appformix-openstack` package. The latest version of `appformix-openstack` package is 0.6.2.

   ```
   $ pip install appformix_openstack-0.6.2-py2-none-any.whl
   ```

   By default, this will install the resources in:

   ```
    /usr/local/lib/python2.7/dist-packages/appformix/heat
   ```

   If the OpenStack services are running in containers, the resources should be installed in a directory that is accessible to the Heat containers. Use the `--target` option with `pip install` to install the resources in a different directory. For example:

   ```
   $ pip install --target=/var/lib/docker/volumes/opt_plugin/_data appformix_openstack-0.6.1-py2-
   none-any.whl
   ```

3. Modify the Heat configuration file.

- Define a variable called `appformix_controller_url` in the `[DEFAULT]` section and set it to the base URL of the Contrail Insights Platform.

- Add the installation directory to the list of plug-in directories. Look for the `plugin_dirs` entry in the `[DEFAULT]` section and add the installation directory to the end of the list. If the OpenStack services are running in containers, specify the mount path of the installation directory inside the Heat containers.

- If desired, define variables called `appformix_task_num_iterations` and `appformix_task_wait_milliseconds` to control how many times and how frequently the Heat plug-in checks the status of an Contrail Insights API request before declaring that the operation has timed out. Both variables accept Integer values. If these variables are undefined, they default to the following values:

  ```
  appformix_task_num_iterations = 10
  appformix_task_wait_milliseconds = 200
  ```

This is what the `heat.conf` file should look like after modification:

```
[DEFAULT]
    ...
appformix_controller_url = <base URL, e.g., http://appformix_platform_host:9000>
plugin_dirs = [...], <e.g. /usr/local/lib/python2.7/dist-packages/appformix/heat>
appformix_task_num_iterations = 10
appformix_task_wait_milliseconds = 200
```

If the OpenStack services are running in containers, make sure the changes are made in the `heat.conf` files in all the Heat containers.

4. Restart all the OpenStack Heat services.

```
service heat-api restart
service heat-api-cfn restart
service heat-engine restart
```

If the OpenStack services are running in containers, restart all the Heat containers.

```
docker restart heat_engine
docker restart heat_api
docker restart heat_api_cfn
```

## OS::AppFormix::Alarm Configuration in Heat Template

The OS::AppFormix::Alarm resource type can be used in Heat templates to create a Contrail Insights alarm. The resource type has the following input parameters:

**Table 1: OS::AppFormix::Alarm Resource Type Input Parameters**

| Parameter | Description |
|---|---|
| alarm_name | A name that identifies the alarm. |
| alarm_metric | Metric to evaluate in the alarm.<br>To see a list of choices, use the Contrail Insights API endpoint /describe/alarms and look for the following list in the output: output['EventRuleParams']['MetricTypeMap'][0]['static']['instance']. Use the value in the Value key as the metric name.<br>output['EventRuleParams']['MetricTypeMap'][0]['static']['instance']<br>Use the value in the Value key as the metric name. |
| threshold | Value by which to compare a metric measurement. Units for the threshold depend on the value of alarm_metric. |
| aggregation_function | Operation to use for combining measured values before comparison. Choices are:<br>• average<br>• max<br>• min<br>• std-dev<br>• sum |

Table 1: OS::AppFormix::Alarm Resource Type Input Parameters *(Continued)*

| Parameter | Description |
|---|---|
| comparison_function | Operation to use for comparing measured values to the threshold. Choices are:<br>• below<br>• equal<br>• above<br>• increasing-at-a-minimum-rate-of<br>• decreasing-at-a-minimum-rate-of |
| duration | Number of seconds for which sample values will be collected before being combined. |
| num_intervals | Number of intervals of length `duration` for which data will be collected before comparison. |
| num_exception_intervals | Number of intervals of length `duration` for which the alarm condition has to be true for the alarm to be considered active. |
| project_id | (Optional) ID of a project that contains the instances on which the alarm should be evaluated. |
| aggregate_id | (Optional) ID of an aggregate that contains the instances on which the alarm should be evaluated.<br>Use the following syntax to indicate that alarm should be evaluated on instances in the current Heat stack:<br>`aggregate_id: { get_param: "OS::stack_id" }`<br>Either `project_id` or `aggregate_id` must be specified in the template. |

Table 1: OS::AppFormix::Alarm Resource Type Input Parameters *(Continued)*

| Parameter | Description |
|---|---|
| notification_url | URL to which a notification will be sent when the alarm is active. This is any URL prepared to receive notification from Contrail Insights (refer to *Notifications*). For Heat templates that use Contrail Insights Alarms to trigger autoscaling, this URL should be set to the `signal_url` of the scaling policy (see example in "Example: Heat Autoscaling with OS::AppFormix::Alarm"). |

## Example: Heat Autoscaling with OS::AppFormix::Alarm

With Contrail Insights Heat plug-in, Contrail Insights Alarms can be used in Heat Autoscaling templates in place of Ceilometer Alarms. The following Heat template uses `OS::AppFormix::Alarm` to automatically scale the number of running instances based on CPU utilization:

```
heat_template_version: 2014-10-16
description: Example auto scale group, policy and alarm
resources:
  scaleup_group:
    type: OS::Heat::AutoScalingGroup
    properties:
      cooldown: 60
      desired_capacity: 1
      max_size: 5
      min_size: 1
      resource:
        type: OS::Nova::Server
        properties:
          key_name: heat_key
          image: 8e571a43-25c7-4eb1-bbb6-13e446e99e8a
          flavor: m1.tiny
          name: "test_vm"
          networks:
            - network: afx-net

  scaleup_policy:
    type: OS::Heat::ScalingPolicy
    properties:
      adjustment_type: change_in_capacity
      auto_scaling_group_id: { get_resource: scaleup_group }
      cooldown: 60
      scaling_adjustment: 1

  scaledown_policy:
    type: OS::Heat::ScalingPolicy
    properties:
      adjustment_type: change_in_capacity
      auto_scaling_group_id: { get_resource: scaleup_group }
      cooldown: 60
      scaling_adjustment: -1

  cpu_alarm_high:
    type: OS::AppFormix::Alarm
    properties:
      alarm_name: 'cpu_alarm_high'
      alarm_metric: 'cpu.usage'
      aggregation_function: 'average'
      comparison_function: 'above'
      duration: 60
    num_intervals: 1
      num_exception_intervals: 1
      threshold: 80
      aggregate_id: { get_param: "OS::stack_id" }
      notification_url: { get_attr: [scaleup_policy, signal_url] }

  cpu_alarm_low:
    type: OS::AppFormix::Alarm
    properties:
      alarm_name: 'cpu_alarm_low'
      alarm_metric: 'cpu.usage'
      aggregation_function: 'average'
      comparison_function: 'below’
      duration: 300
    num_intervals: 1
      num_exception_intervals: 1
      threshold: 10
      aggregate_id: { get_param: "OS::stack_id" }
      notification_url: { get_attr: [scaledown_policy, signal_url] }
```

The following sequence describes what happens when a Heat stack is created from the “Heat Autoscaling with OS::AppFormix::Alarm” template.

- When a heat stack is deployed using this template, a single instance `test_vm` is initially created.
- The two Contrail Insights alarms `cpu_alarm_high` and `cpu_alarm_low` are used to monitor CPU utilization on the instance. They can be defined to monitor any metric that Contrail Insights collects.
- When the CPU utilization on the instance goes above 80 percent, the alarm `cpu_alarm_high` is triggered. This results in the execution of the `scaleup_policy` which increases the number of running instances by 1 every 60 seconds, for as long as the alarm is active. The `scaleup_policy` stops executing when the number of running instances equals the value in `max_size`.
- When the CPU utilization on the instance drops below 10 percent, the alarm `cpu_alarm_low` is triggered. This results in the execution of the `scaledown_policy` which decreases the number of running instances by 1 every 300 seconds, for as long as the alarm is active. The `scaledown_policy` stops executing when the number of running instances equals the value in `min_size`.

### Create a Heat Stack for the Auto-Scaling Template using OS::AppFormix::Alarm

Now, let's create a Heat stack from the template “Example: Heat Autoscaling with OS::AppFormix::Alarm,” and observe what happens when we add CPU load on the VM to trigger the Contrail Insights Alarm.

1. Save the template defined in “Example: Heat Autoscaling with OS::AppFormix::Alarm” in a file named `appformix_autoscaling.yaml`. Enter appropriate values in the `project_id`or `aggregate_id` fields. Then create a Heat stack using the template:

```
$ heat stack-create -f appformix_autoscaling.yaml stack1
+--------------------------------------+------------+--------------------
+---------------------+--------------+
| id                                   | stack_name | stack_status | creation_time       |
updated_time |
+--------------------------------------+------------+--------------------
+---------------------+--------------+
| 753e8bfd-047e-4297-aaef-3d1a68d36b24 | stack1     | CREATE_IN_PROGRESS |
2017-09-10T19:08:34 | None   |
+--------------------------------------+------------+--------------------
+---------------------+--------------+
$ heat stack-list
+--------------------------------------+------------+-----------------+---------------------
+--------------+
| id                                   | stack_name | stack_status | creation_time       |
updated_time |
+--------------------------------------+------------+-----------------+---------------------
+--------------+
| eb9b7dd3-c1a6-4f5d-9039-8c5968b88775 | stack1     | CREATE_COMPLETE | 2017-09-10T19:17:28 |
None |
+--------------------------------------+------------+-----------------+---------------------
+--------------+
```

2. Check that there is a single `test_vm` instance running.

```
$ nova list
+--------------------------------------+------------+--------+------------+-------------
+----------------------+
| ID                                   | Name | Status | Task State | Power State |
Networks       |
+--------------------------------------+------------+--------+------------+-------------
+----------------------+
| 11b00a5b-fa62-407d-a155-e3b65b2436ca | test_vm    | ACTIVE | - | Running | afx-
net=192.168.10.3 |
+--------------------------------------+------------+--------+------------+-------------
+----------------------+
```

3. Generate some load on `test_vm`. Watch for the `cpu_alarm_high` alarm to become active on the Contrail Insights Dashboard.

4. When the alarm is active, check the running instances on the cluster. There should now be two running instances called `test_vm`.

```
$ nova list
+--------------------------------------+------------+--------+------------+-------------
+----------------------+
| ID                                   | Name | Status | Task State | Power State |
Networks       |
+--------------------------------------+------------+--------+------------+-------------
+----------------------+
| 0389529f-ae05-4677-99c9-fb79d27eb9e9 | test_vm    | ACTIVE | - | Running | afx-
net=192.168.10.4 |
| 11b00a5b-fa62-407d-a155-e3b65b2436ca | test_vm    | ACTIVE | - | Running | afx-
net=192.168.10.3 |
+--------------------------------------+------------+--------+------------+-------------
+----------------------+
```

5. Stop the load generator on `test_vm`. Watch for the `cpu_alarm_low` alarm to become active on the Contrail Insights Dashboard.

6. When the alarm is active, check the running instances on the cluster. There should now be a single `test_vm` instance running.

## OS::AppFormix::CompositeAlarm Configuration in Heat Template

You can define multiple individual Alarms and combine them in a Composite Alarm. The state of the Composite Alarm is a combination of the states of the individual Alarms. You can define weights for the individual Alarms and a threshold for the Composite Alarm. The Composite Alarm is active when the sum of the weights of the active Alarms equals or exceeds the user-defined threshold.

The `OS::AppFormix::CompositeAlarm` resource type can be used in Heat templates to create a Contrail Insights Composite Alarm. The resource type has the following input parameters:

**Table 2: OS::AppFormix::CompositeAlarm Resource Type Input Parameters**

| Parameter | Description |
|---|---|
| composite_alarm_name | A name that identifies the composite alarm. |
| project_id | (Optional) ID of a project that contains the instances on which the composite alarm should be evaluated. |
| aggregate_id | (Optional) ID of an aggregate that contains the instances on which the composite alarm should be evaluated.<br>Use the following syntax to indicate that the alarm should be evaluated on instances in the current Heat stack:<br>`aggregate_id: { get_param: "OS::stack_id" }`<br>Either `project_id` or `aggregate_id` must be specified in the template. |

**Table 2: OS::AppFormix::CompositeAlarm Resource Type Input Parameters *(Continued)***

| Parameter | Description |
|---|---|
| notification_url | URL to which a notification will be sent when the alarm is active. This is any URL prepared to receive notification from Contrail Insights (refer to *Notifications*). For Heat templates that use Contrail Insights Alarms to trigger autoscaling, this URL should be set to the `signal_url` of the scaling policy (see “Example: Heat Autoscaling with OS::AppFormix::CompositeAlarm”). |
| composite_alarm_rules | List of individual alarms that make up the composite alarm. Each individual alarm has the same input parameters as the `OS::AppFormix::Alarm` resource, except for `notification_url`, `project_id` and `aggregate_id`. These parameters are defined once for the entire composite alarm. Each individual alarm also has an input parameter called `alarm_weight`, which is a value between 0 and 1. The weights of all active alarms are summed and compared to the value of `composite_alarm_threshold`to determine if the composite alarm is active. |
| composite_alarm_threshold | Value between 0 and 1 used to determine if composite alarm is active. If the sum of weights of all active rules equals or exceeds the threshold, the composite alarm is determined to be active. |

## Example: Heat Autoscaling with OS::AppFormix::CompositeAlarm

With Contrail Insights Heat plug-in, Contrail Insights Composite Alarms can be used in Heat Autoscaling templates in place of Ceilometer Alarms. The following Heat template uses `OS::AppFormix::CompositeAlarm` to automatically scale the number of running instances based on CPU utilization and memory utilization.

```
heat_template_version: 2014-10-16
description: Example auto scale group, policy and alarm
resources:
  scaleup_group:
    type: OS::Heat::AutoScalingGroup
    properties:
      cooldown: 120
      desired_capacity: 1
      max_size: 5
      min_size: 1
      resource:
        type: OS::Nova::Server
        properties:
          key_name: heat_key
          image: 8e571a43-25c7-4eb1-bbb6-13e446e99e8a
          flavor: m1.tiny
          name: "test_vm"
          networks:
            - network: afx-net

  scaleup_policy:
    type: OS::Heat::ScalingPolicy
    properties:
      adjustment_type: change_in_capacity
      auto_scaling_group_id: { get_resource: scaleup_group }
      cooldown: 120
      scaling_adjustment: 1

  scaledown_policy:
    type: OS::Heat::ScalingPolicy
    properties:
      adjustment_type: change_in_capacity
      auto_scaling_group_id: { get_resource: scaleup_group }
      cooldown: 120
      scaling_adjustment: -1

  composite_alarm_high:
    type: OS::AppFormix::CompositeAlarm
    properties:
      composite_alarm_name: 'composite_alarm_high'
      aggregate_id: { get_param: "OS::stack_id" }
      notification_url: { get_attr: [scaleup_policy, signal_url] }
      composite_alarm_threshold: 0.5
      composite_alarm_rules:
        - alarm_name: 'rule1'
          alarm_metric: 'cpu.usage'
          aggregation_function: 'average'
          comparison_function: 'above'
```

```
          duration: 180
          num_intervals: 1
          num_exception_intervals: 1
          threshold: 80
          alarm_weight: 0.5
        - alarm_name: 'rule2'
          alarm_metric: 'memory.usage'
          aggregation_function: 'average'
          comparison_function: 'above'
          duration: 180
          num_intervals: 1
          num_exception_intervals: 1
          threshold: 80
          alarm_weight: 0.5

  composite_alarm_low:
    type: OS::AppFormix::CompositeAlarm
    properties:
      composite_alarm_name: 'composite_alarm_low'
      aggregate_id: { get_param: "OS::stack_id" }
      notification_url: { get_attr: [scaledown_policy, signal_url] }
      composite_alarm_threshold: 1.0
      composite_alarm_rules:
        - alarm_name: 'rule3'
          alarm_metric: 'cpu.usage'
          aggregation_function: 'average'
          comparison_function: 'below'
          duration: 300
          num_intervals: 1
          num_exception_intervals: 1
          threshold: 10
          alarm_weight: 0.5
        - alarm_name: 'rule4'
          alarm_metric: 'memory.usage'
          aggregation_function: 'average'
          comparison_function: 'below'
          duration: 300
          num_intervals: 1
          num_exception_intervals: 1
          threshold: 10
          alarm_weight: 0.5
```

The following sequence describes what happens when a Heat stack is created from the above template.

- When a heat stack is deployed using this template, a single instance `test_vm` is initially created.

- The two Contrail Insights composite alarms `composite_alarm_high` and `composite_alarm_low` are used to monitor resource utilization on the instances in the Heat stack. They are comprised of two individual alarms to monitor CPU utilization and memory utilization on the instances. They can be defined to monitor any metric that Contrail Insights collects.

- The individual alarms in `composite_alarm_high` are defined with weights of 0.5 each and the composite alarm is defined with a threshold of 0.5. This means that the composite alarm will be considered active when any of the individual alarms is active.

- The individual alarms in `composite_alarm_low` are defined with weights of 0.5 each and the composite alarm is defined with a threshold of 1.0. This means that the composite alarm will be considered active only when both the individual alarms are active.

- When either the CPU utilization or memory utilization on the instance `test_vm` goes above 80 percent, the composite alarm `composite_alarm_high` is triggered. This results in the execution of the `scaleup_policy` which increases the number of running instances by 1 every 60 seconds for as long as the composite alarm is active. The `scaleup_policy` stops executing when the number of running instances equals the value in `max_size`.

- When both the CPU utilization and memory utilization on the instance `test_vm` drops below 10 percent, the composite alarm `composite_alarm_low` is triggered. This results in the execution of the `scaledown_policy` which decreases the number of running instances by 1 every 60 seconds, for as long as the composite alarm is active. The `scaledown_policy` stops executing when the number of running instances equals the value in `min_size`.

### Create a Heat Stack for the Auto-Scaling Template using OS::AppFormix::CompositeAlarm

Now, let's create a Heat stack from the template “Example: Heat Autoscaling with OS::AppFormix::CompositeAlarm” and observe what happens when we add CPU load on the VM to trigger the Contrail Insights Composite Alarm.

1. Save the template defined in “Example: Heat Autoscaling with OS::AppFormix::CompositeAlarm” in a file called `appformix_composite_autoscaling.yaml`. Enter appropriate values in the `project_id` or `aggregate_id` fields. Then create a Heat stack using the template:

```
$ heat stack-create -f appformix_composite_autoscaling.yaml composite1
+--------------------------------------+------------+--------------------
+----------------------+--------------+
| id                                   | stack_name | stack_status | creation_time        |
updated_time |
+--------------------------------------+------------+--------------------
+----------------------+--------------+
| f2bc3282-1d8b-4230-a3ef-a589f3527188 | composite1 | CREATE_IN_PROGRESS |
2018-10-30T03:12:08Z | None         |
+--------------------------------------+------------+--------------------
+----------------------+--------------+
$ heat stack-list
+--------------------------------------+------------+-----------------+----------------------
+--------------+
| id                                   | stack_name | stack_status | creation_time        |
updated_time |
+--------------------------------------+------------+-----------------+----------------------
+--------------+
| f2bc3282-1d8b-4230-a3ef-a589f3527188 | composite1 | CREATE_COMPLETE | 2018-10-30T03:12:08Z
| None         |
+--------------------------------------+------------+-----------------+----------------------
+--------------+
```

2. Check that there is a single `test_vm` instance running.

```
$ nova list
+--------------------------------------+---------+--------+------------+-------------
+------------------------+
| ID                                   | Name | Status | Task State | Power State |
Networks               |
+--------------------------------------+---------+--------+------------+-------------
+------------------------+
| e9bc550d-084e-410c-8154-4c590f504a69 | test_vm | ACTIVE | -          | Running |
private-2=192.168.27.3 |
+--------------------------------------+---------+--------+------------+-------------
+------------------------+
```

3. Generate some load on `test_vm`. Watch for the `composite_alarm_high` composite alarm to become active on the Contrail Insights Dashboard.

4. When the composite alarm is active, check the running instances on the cluster. There should now be two running instances called `test_vm`.

```
$ nova list
+--------------------------------------+---------+--------+------------+-------------
+------------------------+
| ID                                   | Name | Status | Task State | Power State |
Networks        |
+--------------------------------------+---------+--------+------------+-------------
+------------------------+
| e9bc550d-084e-410c-8154-4c590f504a69 | test_vm | ACTIVE | -          | Running |
private-2=192.168.27.3 |
| f7feb43b-221d-4738-9092-476fa2e4b3aa | test_vm | ACTIVE | -          | Running |
private-2=192.168.27.8 |
+--------------------------------------+---------+--------+------------+-------------
+------------------------+
```

5. Stop the load generator on `test_vm`. Watch for the `composite_alarm_low` composite alarm to become active on the Contrail Insights Dashboard.

6. When the alarm is active, check the running instances on the cluster. There should now be a single `test_vm` instance running.

## Troubleshooting

For debugging, enable the `verbose` and `debug` options by adding them to the `[DEFAULT]` section in `heat.conf`.

```
verbose = True
debug = True
```

Then restart the Heat services or containers. Detailed logs will appear in `/var/log/heat/heat-engine.log`.

### RELATED DOCUMENTATION

*Notifications*

*Alarms*

# Application Event Ingestion

IN THIS SECTION

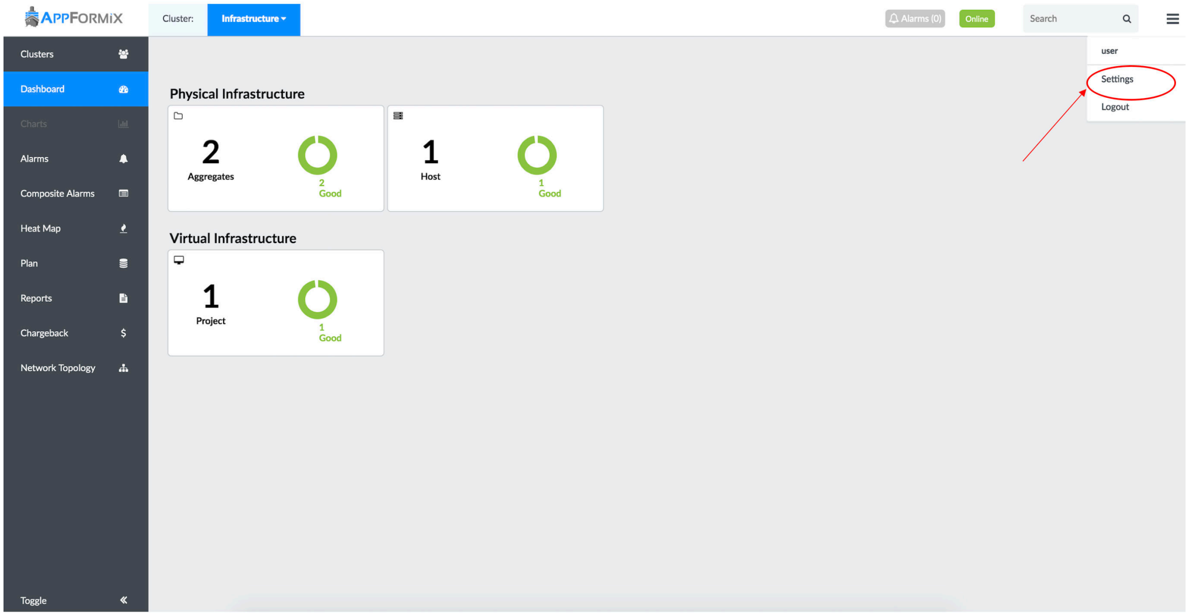
Contrail Insights can ingest events from a registered application and perform alarms on them. You can register an application with Contrail Insights and specify the event IDs for which the application will be posting data. Upon successful registration, a token is given to the application. The application uses that token to post events to Contrail Insights for any of the event IDs registered. Alarms can be configured for these events.

## Register an Application

To register an application:

1. Select **Settings** in the top right of the Dashboard.

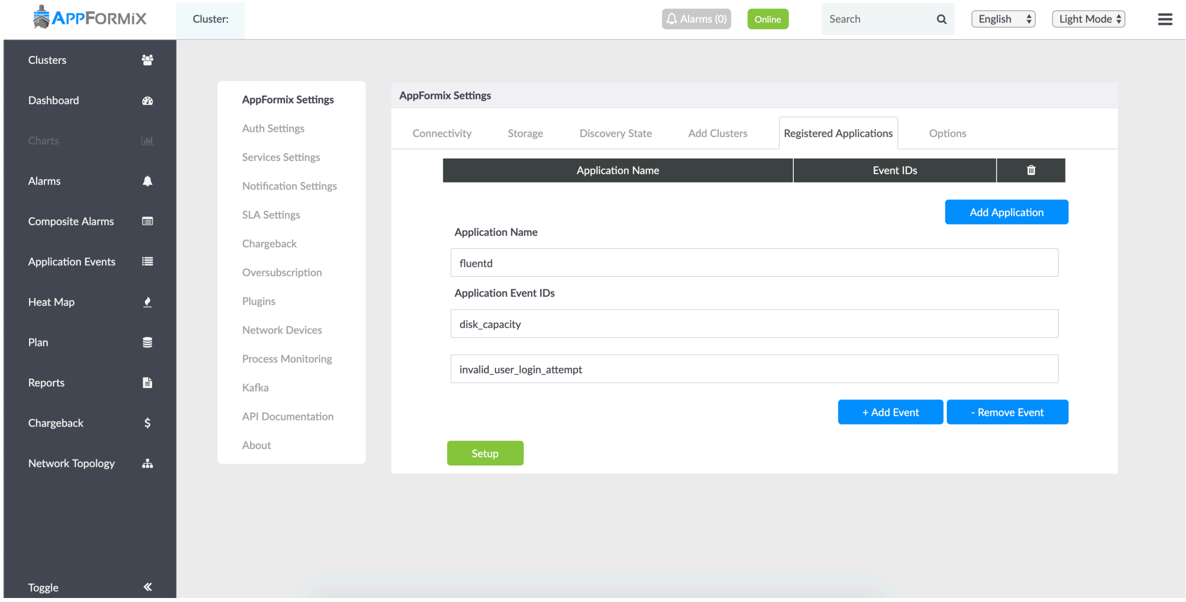
Figure 23: Select Settings in the Dashboard



2. Select **AppFormix Settings**, then click the **Registered Applications** tab. Click **Add Application**.

3. Provide the **Application Name** and add all the Application Event IDs for which the application will be posting data by clicking **+Add Event**. Then click **Setup**.

Figure 24: AppFormix Settings for Adding an Application and Application Event IDs



Application Name

fluentd

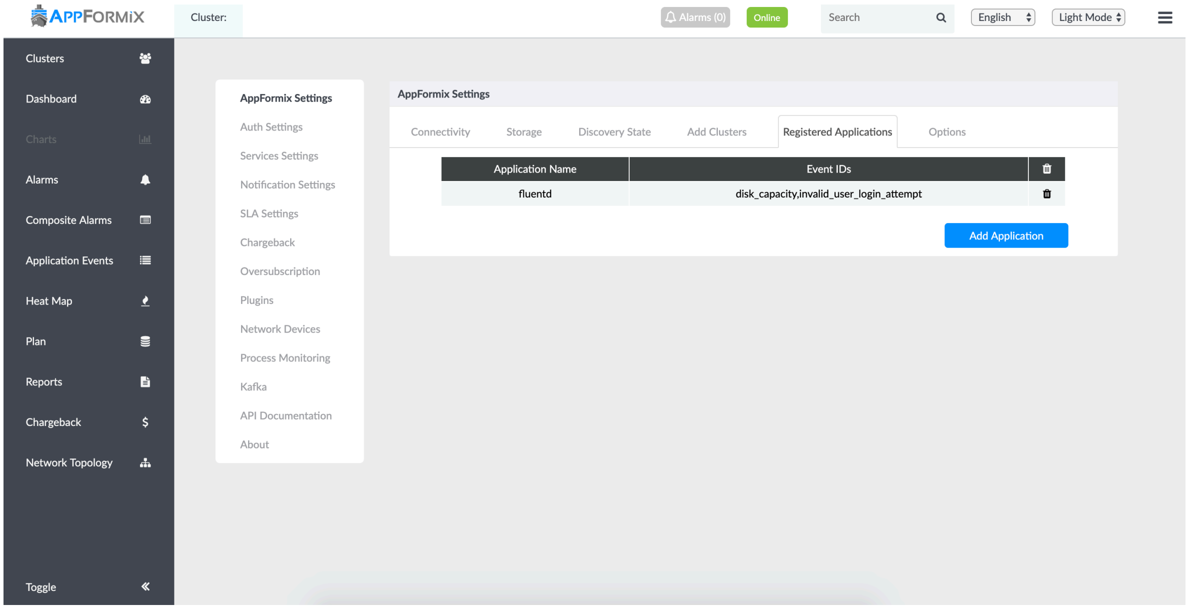
Application Event IDs

disk_capacity

invalid_user_login_attempt

4. The Application appears as successfully added. It can be deleted by clicking the Trash icon.

**Figure 25: Successfully Added Application**



| Application Name | Event IDs | 🗑 |
|---|---|---|
| fluentd | disk_capacity,invalid_user_login_attempt | 🗑 |

Application registration can also be achieved using the API:

**Request:**

*url*:

```
POST http://<appformix_controller:port>/appformix/v1.0/application_registration
```

*headers*:

```
"Content-Type": application/json,
"X-Auth-Token": <>, (required)
"X-Auth-Type": <> (required)
```

*data*:

```
{

       "ApplicationName": "fluentd",
        "ApplicationEventIds": ["disk_capacity", "invalid_user_login_attempt"]
}
```

**Response:**

```
{
        "ApplicationName": "fluentd",
        "ApplicationId": "567854a8-a9ea-11e9-ab42-0242ac120005",
        "ApplicationToken": "abc8902cd17459fe73839494bde39310506380220"
        "ApplicationEventIds": ["disk_capacity", "invalid_user_login_attempt"]
}
```

## Post Events for a Registered Application

After an application is configured, it can post events to Contrail Insights.

The data should be in the following format:

**Request:**

*url*:

```
POST http://<appformix_controller:port>/appformix/v1.0/analytics/application_event
```

*headers*:

```
{
        "Content-Type": application/json,
        "X-Auth-Token": <>, (required, provide the ApplicationToken)
        "X-Auth-Type": 'appformix' (required)
}
```

*data*:

```
{
        "ApplicationId": "567854a8-a9ea-11e9-ab42-0242ac120005",
        "EventId": "disk_capacity", # One of the event IDs registered for the application
        "Metric": 80,
        "Metadata": {
        <variable dictionary, not used for alarming>
        }
}
```

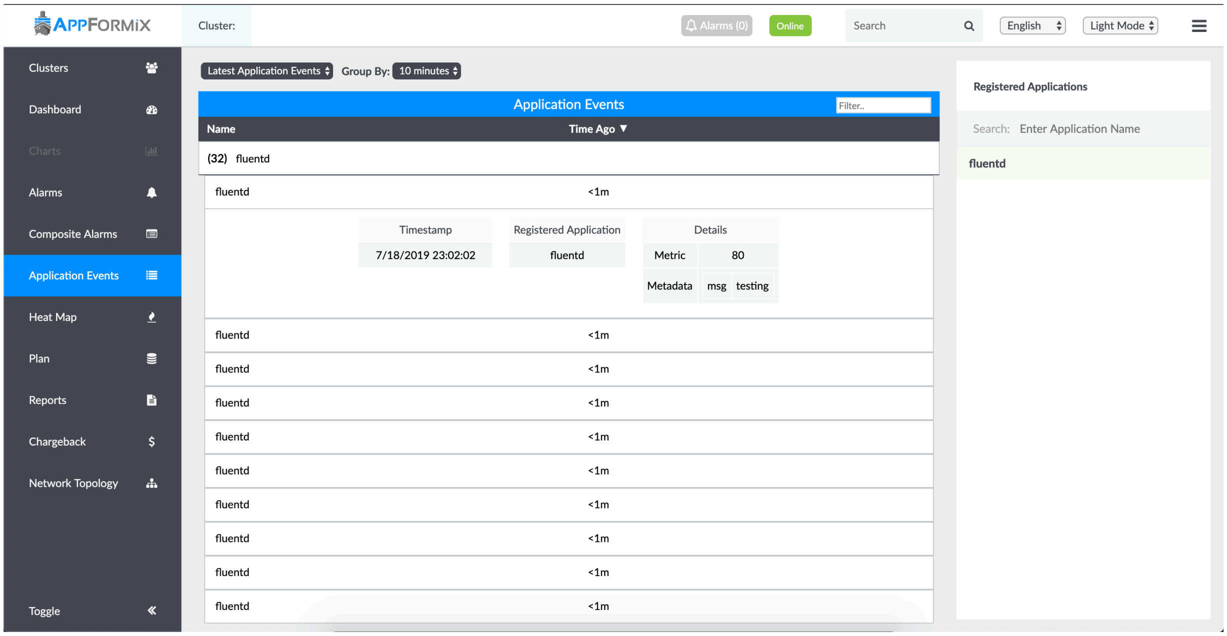
**Response:**

*status code*:

```
200: Success
401: Authentication failure(ApplicationToken Missing/Invalid)
```

All the posted events are displayed on the UI in the Application Events page. From this page, in the right panel, select any application to toggle the displaying and hiding of events from that application.

**Figure 26: Viewing Latest Application Events**



Application Events

| Name | Time Ago ▼ |
|---|---|
| (32) fluentd | |
| fluentd | <1m |
| fluentd | <1m |
| fluentd | <1m |
| fluentd | <1m |
| fluentd | <1m |
| fluentd | <1m |
| fluentd | <1m |
| fluentd | <1m |
| fluentd | <1m |
| fluentd | <1m |

| Timestamp | Registered Application | Details | | |
|---|---|---|---|---|
| 7/18/2019 23:02:02 | fluentd | Metric | 80 | |
| | | Metadata | msg | testing |

Registered Applications

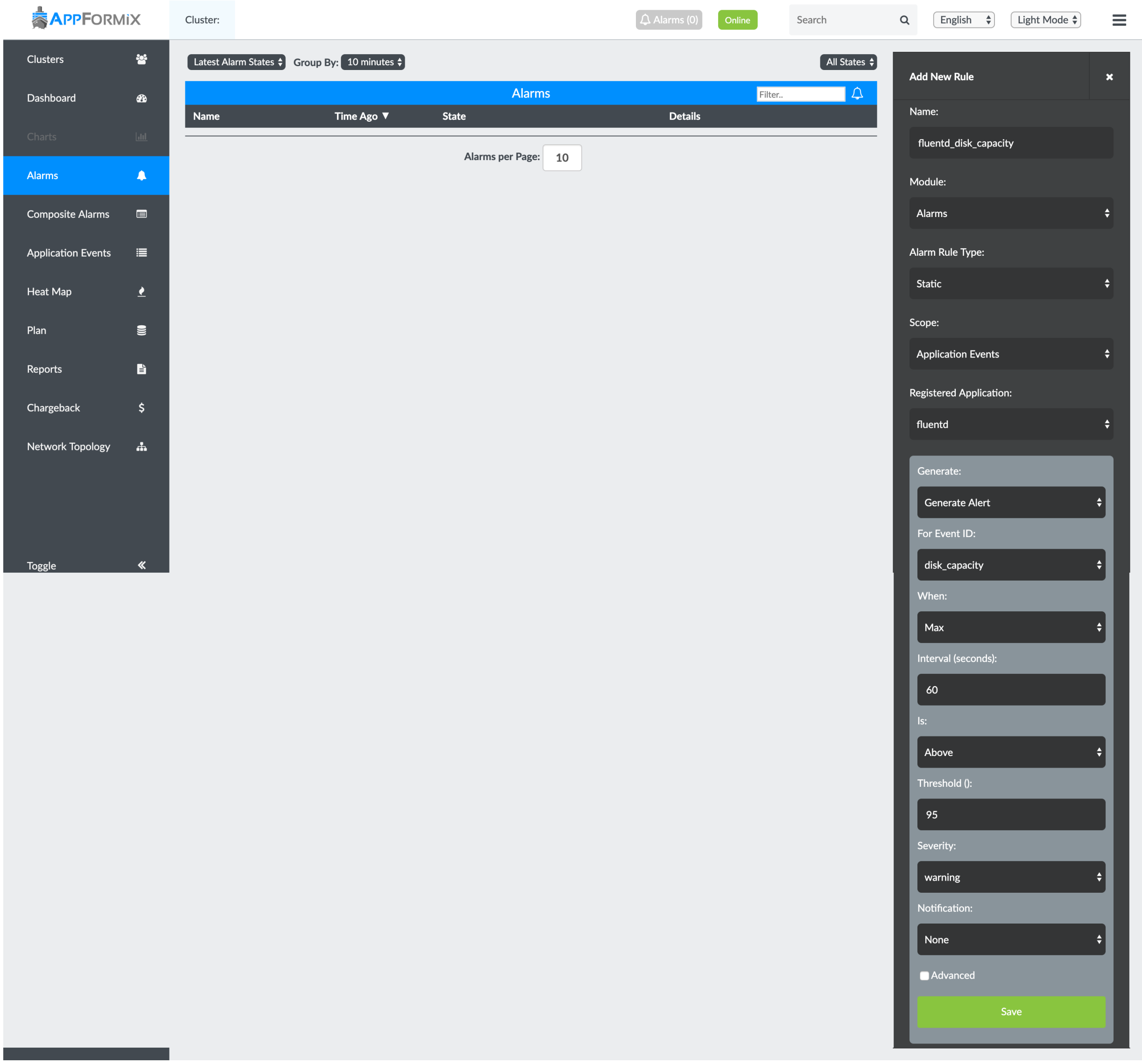
Search: Enter Application Name

fluentd

## Alarms for Application Events

Alarms can be configured for any of the event IDs registered for the application.

Figure 27: Configuring Alarms for Application Events



Name: fluentd_disk_capacity

Module: Alarms

Alarm Rule Type: Static

Scope: Application Events

Registered Application: fluentd

Generate: Generate Alert

For Event ID: disk_capacity

When: Max

Interval (seconds): 60

Is: Above

Threshold (): 95

Severity: warning

Notification: None

Alarm configuration using the API:

*url*:

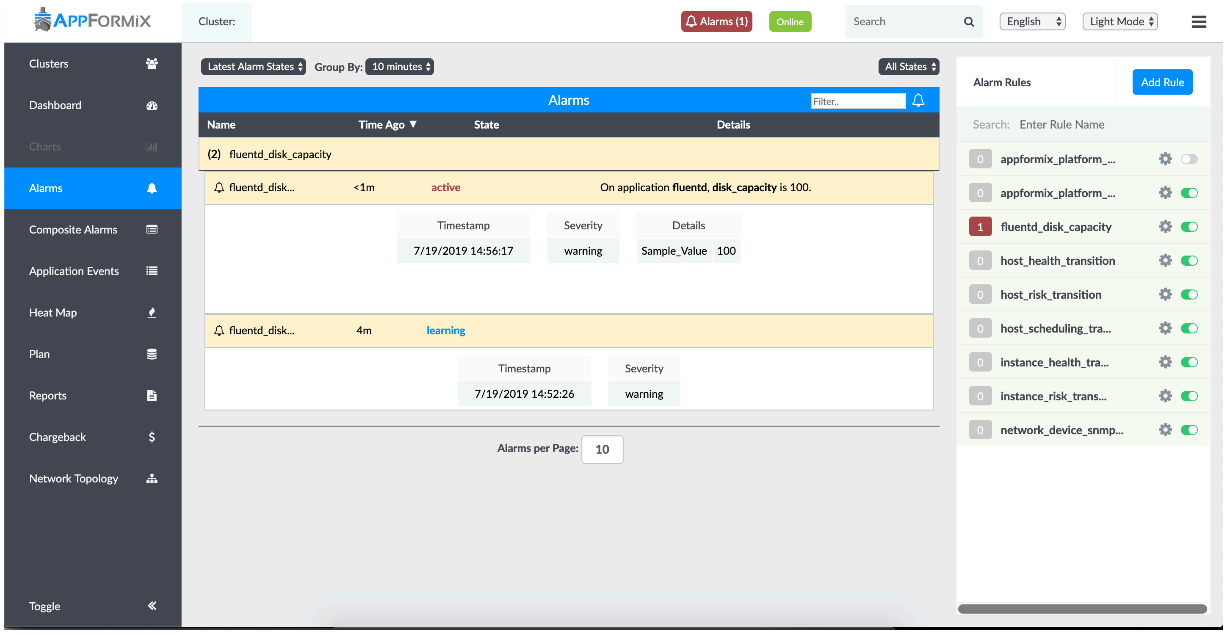
```
POST http://<appformix_controller:port>/appformix/v1.0/analytics/application_event
```

*data*:

```
{
  "Name": "fluentd_disk_capacity",
  "ApplicationId": "567854a8-a9ea-11e9-ab42-0242ac120005",
  "MetricType": "disk_capacity",
  "AggregationFunction": "max",
  "ComparisonFunction": "above",
  "Threshold": 95,                # This value compared to "Metric" in an event
  "IntervalsWithException": 1,
  "IntervalDuration": "60s",
  "IntervalCount": 1,
  "Mode": "alert",
  "Severity": "warning",
  "EventRuleScope": "application_events",
  "CreatedBy": "user",
  "DisplayEvent": true,
  "Module": "alarms",
  "EventRuleType": "static",
  "EntityType":""
}
```

When the threshold configured in the alarm is exceeded, the triggered alarm is shown on the Alarms page in the UI.

**Figure 28: Viewing Event Alarm on Dashboard**



| Name | Time Ago ▼ | State | Details |
|---|---|---|---|
| (2) fluentd_disk_capacity | | | |
| fluentd_disk... | <1m | active | On application **fluentd**, **disk_capacity** is 100. |

| Timestamp | Severity | Details | |
|---|---|---|---|
| 7/19/2019 14:56:17 | warning | Sample_Value | 100 |

| Name | Time Ago | State |
|---|---|---|
| fluentd_disk... | 4m | learning |

| Timestamp | Severity |
|---|---|
| 7/19/2019 14:52:26 | warning |

Alarms per Page: 10

| | |
|---|---|
| 0 | appformix_platform_... |
| 0 | appformix_platform_... |
| 1 | fluentd_disk_capacity |
| 0 | host_health_transition |
| 0 | host_risk_transition |
| 0 | host_scheduling_tra... |
| 0 | instance_health_tra... |
| 0 | instance_risk_trans... |
| 0 | network_device_snmp... |

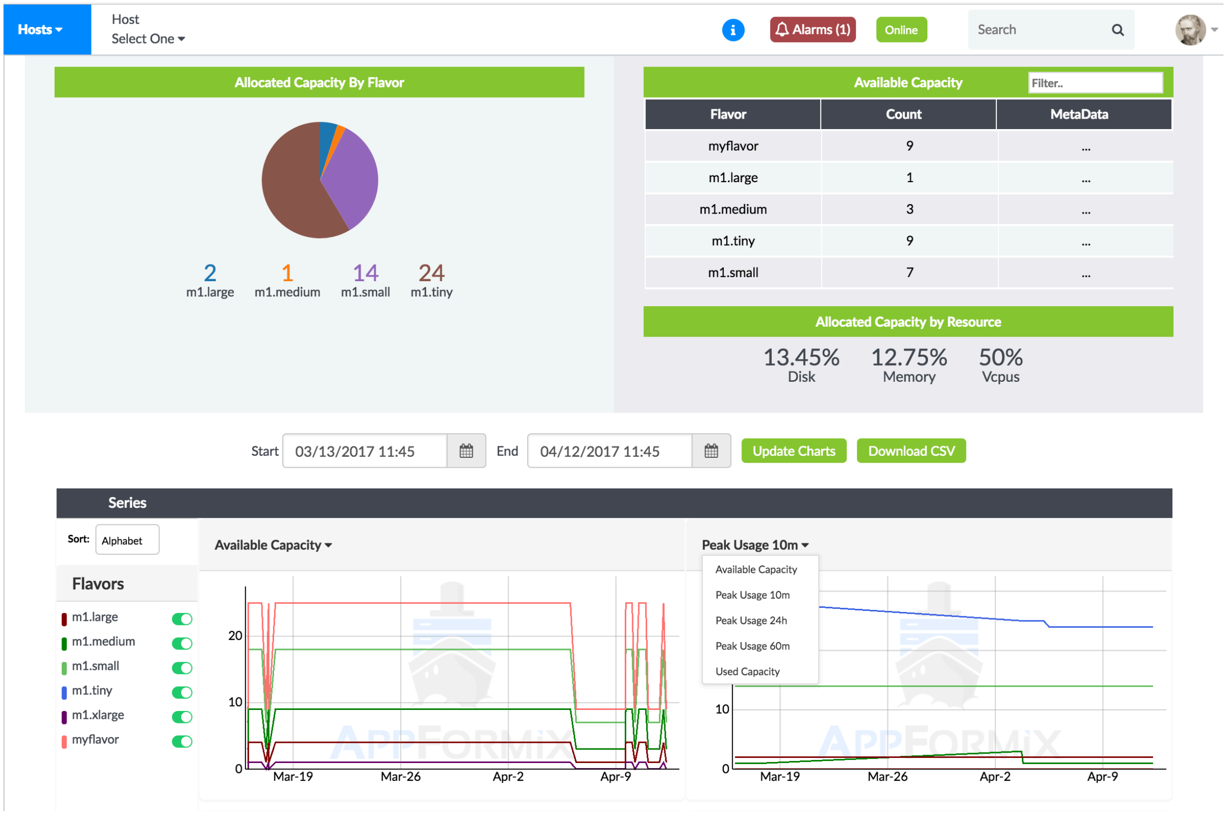
The alarm is also sent to Kafka with the topic being the alarm's name. For more information, see *Contrail Insights with Kafka*.

# Capacity Planning

IN THIS SECTION

Contrail Insights Plan helps you understand, plan, and model the capacity of your infrastructure. Figure 29 on page 42 shows the allocated capacity charts and availabile capacity table.

Figure 29: Allocated Capacity Charts and Available Capacity Table



| Flavor | Count | MetaData |
|---|---|---|
| myflavor | 9 | ... |
| m1.large | 1 | ... |
| m1.medium | 3 | ... |
| m1.tiny | 9 | ... |
| m1.small | 7 | ... |

Allocated Capacity by Resource

13.45% Disk · 12.75% Memory · 50% Vcpus

Start 03/13/2017 11:45 · End 04/12/2017 11:45

## Allocated Capacity

In OpenStack, the unit of compute resource allocation is an instance of a particular flavor. A flavor defines the amount of virtual central processing units (vCPUs), memory, and local storage allocations for an instance.

In the Plan pane of the Dashboard, a pie chart indicates the current number of allocated instances on a per-flavor basis. This provides an operator with an understanding of the types of resources requested by users of the infrastructure.

To understand the change in capacity over time, line charts show the history of allocated and available capacity in terms of the number of instances of each flavor type. Using the drop-down list, you can choose to see the used, available, or peak capacities of the infrastructure. Each line on the chart represents one flavor type. The time period displayed by the charts is configurable to view trends over long or short time horizons, and plan for appropriate resource capacity.

Allocated capacity is also displayed on a per-resource basis. The allocated percentage of capacity is displayed for compute, memory, and storage resources.

## Available Capacity

The available capacity table shows the number of instances of each flavor type that can be allocated presently. The available capacity takes into consideration the resource requirements of each flavor type, the current unused capacity of the physical infrastructure, and the scheduler policy that constrains which sets of hosts can be used to allocate an instance of a particular flavor.

### Modeling Oversubscription Policy

The oversubscription policy of the OpenStack Nova scheduler affects the available capacity of the infrastructure. In **Settings > Oversubscription**, Contrail Insights has a configuration for oversubscription ratios. Contrail Insights uses these ratios when calculating the available capacity. The ratios configured in Contrail Insights do not affect the configuration of the scheduler.

You can modify the ratios to model how an oversubscription policy will affect the available capacity of the infrastructure. When the ratios are modified, the available capacity table will update to show how many instances of each flavor type may be allocated. By configuring different modeling ratios, an administrator can see the impact of potential changes to the oversubscription policy of the scheduler, or understand how increasing physical capacity in the areas of compute, memory, and storage will address the demands of users.

Upon initial installation, the oversubscription ratios in Contrail Insights are set to 1 (that is, no oversubscription). When not modeling a policy change, we recommend configuring the ratios to match the configuration of the OpenStack Nova scheduler policy so the available capacity table reflects the actual capacity of the infrastructure.

RELATED DOCUMENTATION

*OpenStack Nova Scheduler Service*

*Alarms*

*Chargeback*

*Charts*

*Health Monitor*

*Metrics Collected by Contrail Insights*

*Notifications*

*Extensibility Using Plug-Ins*

*Reports*

*Service Monitoring from the UI*

# Contrail Insights with Kafka

IN THIS SECTION

## Set Up Kafka

**NOTE**: Contrail Insights does not explicitly create Kafka topics. The Kafka broker cluster should be configured to auto-create topics. Alternatively, you can manually manage the topic creation. If you already have Kafka running, you can skip "Set Up Kafka" on page 45 and go directly to "Set Up Contrail Insights with Kafka" on page 47.

Setting up Kafka as a Docker container:

1. Create a Docker network for this Kafka container and its dependencies to be connected to by running the following command:

```
docker network create AppformixKafka
```

2. Next, bring up Zookeeper for Kafka to work:

```
docker run -d \
       --name appformix-zookeeper \
       --net AppformixKafka \
       -e ZOOKEEPER_TICK_TIME=2000 \
       -e ZOOKEEPER_CLIENT_PORT=2181 \
       -p 2181:2181 \
       --restart always \
       confluent/zookeeper
```

3. Bring up the Kafka container by running the following. The variable `ip_address` must be specified appropriately.

```
docker run -d \
       --net=AppformixKafka \
       --name=appformix-kafka \
       -p 9092:9092 \
       -e KAFKA_BROKER_ID=2 \
       -e KAFKA_ZOOKEEPER_CONNECT=appformix-zookeeper:2181 \
       -e KAFKA_ADVERTISED_HOST_NAME=appformix-kafka \
       -e KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://<ip_address>:9092 \
       -e KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR=1 \
        confluentinc/cp-kafka:latest
```

4. If Kafka with SSL is required, then additional parameters are required:

```
docker run -d \
       --net=AppformixKafkaSSL \
       --name=appformix-kafka-ssl \
       -p 9092:9092 \
       -e KAFKA_BROKER_ID=2 \
       -e KAFKA_ZOOKEEPER_CONNECT=appformix-zookeeper-ssl:2181 \
       -e KAFKA_ADVERTISED_HOST_NAME=appformix-kafka-ssl \
       -e KAFKA_ADVERTISED_LISTENERS=SSL://$ipaddr:9092 \
       -e KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR=1 \
       -e KAFKA_SECURITY_INTER_BROKER_PROTOCOL=SSL \
       -e KAFKA_SSL_KEYSTORE_FILENAME=kafka.broker.keystore.jks \
       -e KAFKA_SSL_KEYSTORE_CREDENTIALS=broker_keystore_creds \
       -e KAFKA_SSL_KEY_CREDENTIALS=broker_sslkey_creds \
       -e KAFKA_SSL_TRUSTSTORE_FILENAME=kafka.broker.truststore.jks \
       -e KAFKA_SSL_TRUSTSTORE_CREDENTIALS=broker_truststore_creds \
       -e KAFKA_SSL_ENDPOINT_IDENTIFICATION_ALGORITHM=" " \
       -e KAFKA_SSL_CLIENT_AUTH=requested \
       -v <secret_files_path>:/etc/kafka/secrets \
        confluentinc/cp-kafka:latest
```

The `secret_files_path` should be replaced where all of the keystore and truststore files are present. For an example of how to create the above keystores and truststores, reference: https://github.com/confluentinc/cp-docker-images/blob/5.2.1-post/examples/kafka-cluster-ssl/secrets/create-certs.sh.

**NOTE**: If Kafka with SSL is required, then all of the hosts monitored by Contrail Insights must have at least Python version of 2.7.9. It is also required that the Certificate Authority (CA) used for the certificates for the Kafka broker(s) be a trusted CA on all of the hosts monitored by Contrail Insights. In order for Contrail Insights containers to communicate with the Kafka broker(s), the CA file must be set as a `group_vars/all` variable `appformix_kafka_ssl_ca` at installation time.

Now that Kafka is set up, next you can configure Contrail Insights with Kafka.

## Set Up Contrail Insights with Kafka

To configure Contrail Insights with Kafka, a POST request must be sent to an Contrail Insights Platform API:

```
http://<controller_ip>:9000/appformix/controller/v2.0/kafka_config
```

The following fields must be sent in this request:

**Name** The name of the Kafka cluster, which can be anything.

**BootstrapServers** A list of host/port pairs to use for establishing the initial connection to the Kafka cluster. Each item in the list is a string in the format `host:port`.

To send a POST body request using Ansible:

Run the POST body request, which is similar to the following:

```
{
    "Name": "Kafka Config",
    "BootstrapServers": ["10.X.X.1:9092"]
}
```

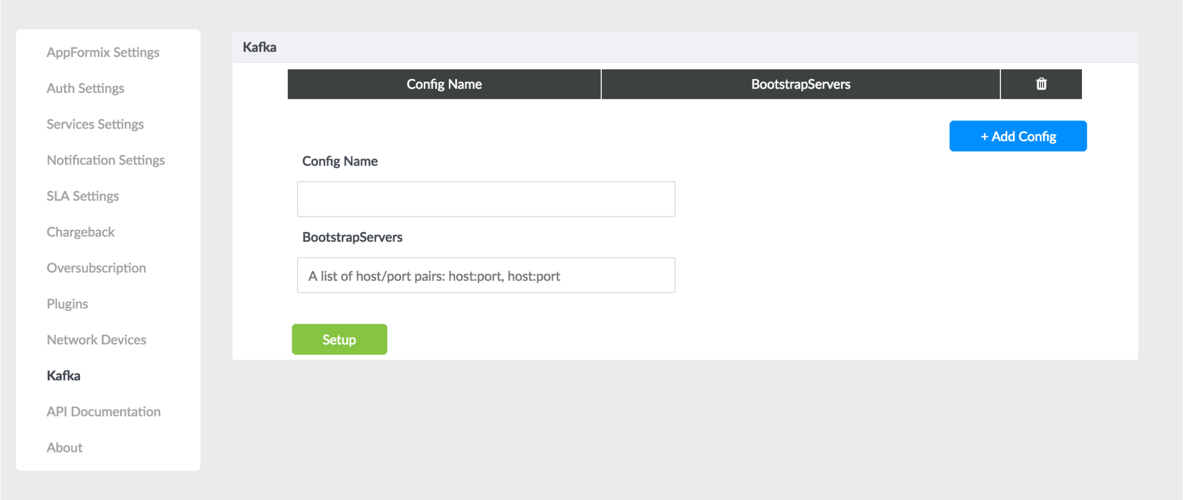
If Kafka has been set up with SSL, then an additional field is needed:

```
{
    "Name": "Kafka Config",
    "BootstrapServers": ["10.X.X.1:9092"],
    "SecurityProtocol": "SSL"
}
```

To send a POST body request from the Contrail Insights Dashboard:

1. Select **Settings** in the upper right corner, then select **AppFormix Settings** > **Kafka**. Next, click **+ Add Config**.

Figure 30: AppFormix Settings for Kafka Page



Config Name

BootstrapServers

A list of host/port pairs: host:port, host:port

2. Enter a name for the Kafka configuration and list the BootstrapServers as a comma separated list of strings with each string in the `host:port` format.

3. Click **Setup** after the fields have been populated.

> **NOTE**: The following steps are for streaming network telemetry data to Kafka. All Contrail Insights alarms are automatically sent to Kafka once Kafka has been configured as stated in the earlier procedures above. There are no additional steps needed for alarms. See "Contrail Insights Alarms With Kafka" on page 49.

4. Click **+ Add Subscription** to create a subscription.

5. Next create a Topic, select devices, and then select which Sensors/MIBs you want sent to Kafka. The specified data will then be sent to Kafka under the specified topic. Click **Create Subscription** after the fields are populated.

## Messages from Contrail Insights to Kafka

After configuration, messages from Contrail Insights are received by an appropriate Kafka consumer.

In the following command, `bootstrap_server` is one of the bootstrap servers specified in the `BootstrapServers` variable above and `topic` is the topic that was specified in the subscription created. :

```
/usr/bin/kafka-console-consumer --bootstrap-server <bootstrap_server> --topic <topic>  --from-
beginning
```

This command outputs messages to standard output. Output for topic `grpc-components` with sensor `/components/` selected will look something like:

```
[
    {
        "AgentId": uuid,
        "Timestamp": 1533915694346,
        "RoomKey": "QFX0:Routing Engine0",
        "ResourcePath": "/components/",
        "Data": {
            "cpu-utilization-background": 0,
            "cpu-utilization-user": 1,
            "temperature": 36,
            "temperature-cpu": 36
        },
    },
]
```

## Contrail Insights Alarms With Kafka

Contrail Insights Alarms are configured to automatically send alerts to Kafka, if Kafka has been configured in Contrail Insights. See "Set Up Contrail Insights with Kafka" on page 47. Contrail Insights sends alarms with the topic as the alarm's name. For example, Alarm name `host_cpu` is sent to Kafka with topic `host_cpu`.

# OpenStack Nova Scheduler Service

IN THIS SECTION

The OpenStack `nova-scheduler` service supports plug-ins to filter which hosts are eligible candidates on which to schedule a virtual machine. The `AppFormixFilter` queries the Contrail Insights Platform to determine if a host is compliant with the Host Scheduling service-level agreement (SLA). If a host is not compliant with the SLA policy, then the host is filtered from the list of eligible hosts.

If the `AppFormixFilter` fails to request SLA status of a host, then the host remains in the eligible pool.

## Installation

To use `AppFormixFilter` in OpenStack Nova Scheduler, you must first install the `appformix_openstack` package and then modify the configuration in `/etc/nova/nova.conf`.

### Install appformix_openstack Package

First, on the OpenStack controller host that executes the `nova-scheduler` service, install the AppFormix Python package that contains OpenStack filter scheduler plug-ins:

```
pip install appformix_openstack-0.6.1-py2-none-any.whl
```

By default, this command installs the resources in:

```
/usr/local/lib/python2.7/dist-packages/appformix
```

To install in a different directory, run the following:

```
pip install --target <dir>
```

**NOTE**: The directory specified must be part of the `PYTHONPATH` environment variable in order for the `nova-scheduler` to be able to find the plug-in.

If the OpenStack services are running in containers, the resources should be installed in a directory that is accessible to the Nova containers.

## Modify the Nova Configuration File

Then modify the `/etc/nova/nova.conf` file. If the OpenStack services are running in containers, make sure the changes are made in the `nova.conf` files in all the Nova containers. Following are instructions for different OpenStack releases.

### For OpenStack Releases Juno to Newton

Add the following lines in the `[DEFAULT]` section of `/etc/nova/nova.conf`:

```
  #
    # Configure Contrail Insights Platform URL used by AppFormixFilter.
    #
    # If `appformix_controller_url` has HTTPS as its protocol, and the host
    # has a self-signed certificate, then set `appformix_verify_cert` to
    # false to ignore verification of the certificate.  By default,
    # `appformix_verify_cert` is True.
    #
    # Set 'appformix_api_token' to the value of 'TokenId' from
    # the file /opt/appformix/etc/appformix_token.rst on the Contrail Insights Platform host.
    #
    appformix_controller_url = <base URL, e.g., http://appformix_platform_host:9000/>
appformix_verify_cert = False
appformix_api_token = <AppFormix token from /opt/appformix/etc/appformix_token.rst>

#
    # Adding AppFormixFilter to `scheduler_available_filters` makes it
    # available as a choice to configure in `scheduler_default_filters`.
    # The appformix-openstack Python package must be installed on the host
    # that executes nova-scheduler service.
    # The following are sample configuration values for nova-scheduler to
    # use the FilterScheduler.  The key addition is to include
    # AppFormixFilter in the list of `scheduler_available_filters`.
    #
    scheduler_driver_task_period = 60
    scheduler_driver = nova.scheduler.filter_scheduler.FilterScheduler
    scheduler_available_filters = appformix.openstack.nova_filters.AppFormixFilter
    scheduler_available_filters = nova.scheduler.filters.all_filters
    scheduler_default_filters = AppFormixFilter, DiskFilter, RetryFilter, CoreFilter,
AvailabilityZoneFilter, RamFilter, ComputeFilter, ComputeCapabilitiesFilter,
ImagePropertiesFilter, ServerGroupAntiAffinityFilter, ServerGroupAffinityFilter
```

### For OpenStack Ocata and Later Releases

1. Add the following lines in the `[DEFAULT]` section of `/etc/nova/nova.conf`:

```
 #
    # Configure Contrail Insights Platform URL used by AppFormixFilter.
    #
    # If `appformix_controller_url` has HTTPS as its protocol, and the host
    # has a self-signed certificate, then set `appformix_verify_cert` to
    # false to ignore verification of the certificate.  By default,
    # `appformix_verify_cert` is True.
    #
    # Set 'appformix_api_token' to the value of 'TokenId' from
    # the file /opt/appformix/etc/appformix_token.rst on the AppFormix Platform host.
    #
    appformix_controller_url = <base URL, e.g., http://appformix_platform_host:9000/>
appformix_verify_cert = False
appformix_api_token = <AppFormix token from /opt/appformix/etc/appformix_token.rst>
```

2. Add the following lines under the `[scheduler]` section:

```
periodic_task_interval = 60
    driver = filter_scheduler
```

3. Add the following lines under the `[filter_scheduler]` section:

```
#
    # Adding AppFormixFilter to `available_filters` makes it
    # available as a choice to configure in `enabled_filters`.
    # The appformix-openstack Python package must be installed on the host
    # that executes nova-scheduler service.
    # The other variables are sample configuration values for nova-scheduler
    # to use the FilterScheduler.  The key addition is to include
    # AppFormixFilter in the list of `enabled_filters`.
    #
    available_filters = nova.scheduler.filters.all_filters
    available_filters = appformix.openstack.nova_filters.AppFormixFilter
    enabled_filters = AppFormixFilter, RetryFilter, AvailabilityZoneFilter, ComputeFilter,
ComputeCapabilitiesFilter, ImagePropertiesFilter, ServerGroupAntiAffinityFilter,
ServerGroupAffinityFilter
```

## Restart the OpenStack Nova Scheduler Services

Run the following command to restart the Nova Scheduler services:

```
service nova-scheduler restart
```

If the OpenStack services are running in containers, restart all of the Nova containers.

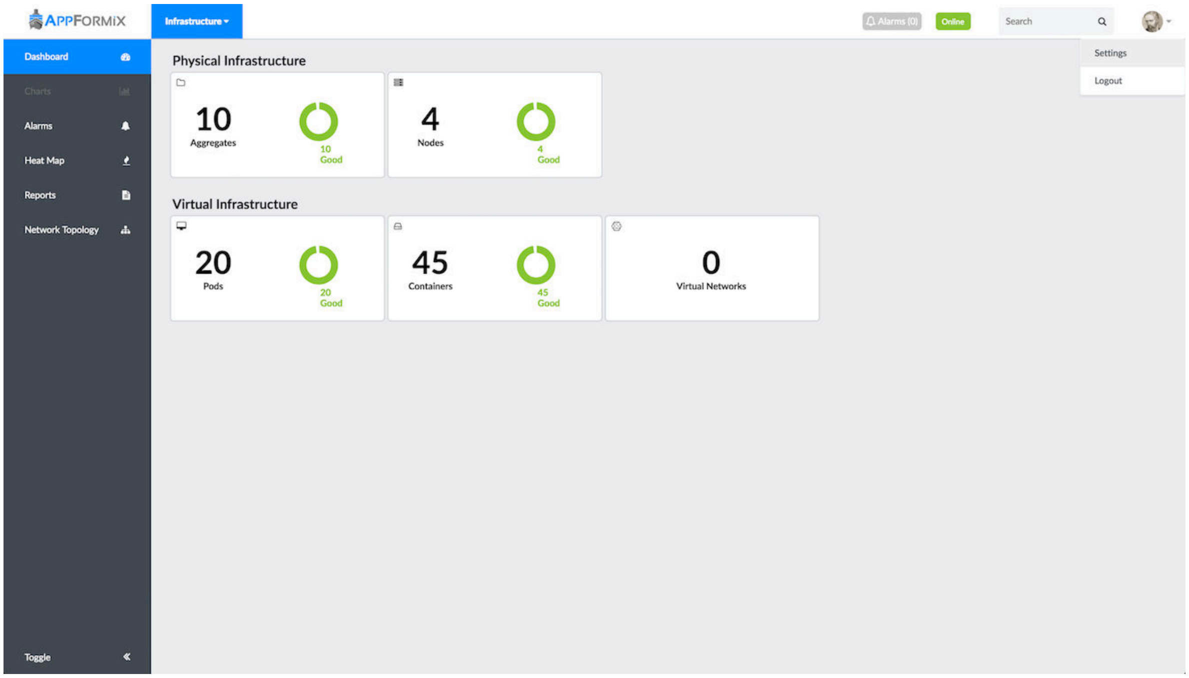
## Using the Contrail Insights Nova Scheduler Plug-In

The Contrail Insights `nova-scheduler` plug-in uses a Scheduling SLA to filter hosts. This SLA is comprised of user-defined Contrail Insights Alarms. Contrail Insights ships with a default Scheduling SLA that includes alarms for missed heartbeat, high CPU load, and high memory usage.

To change the alarms in the Scheduling SLA, do the following:

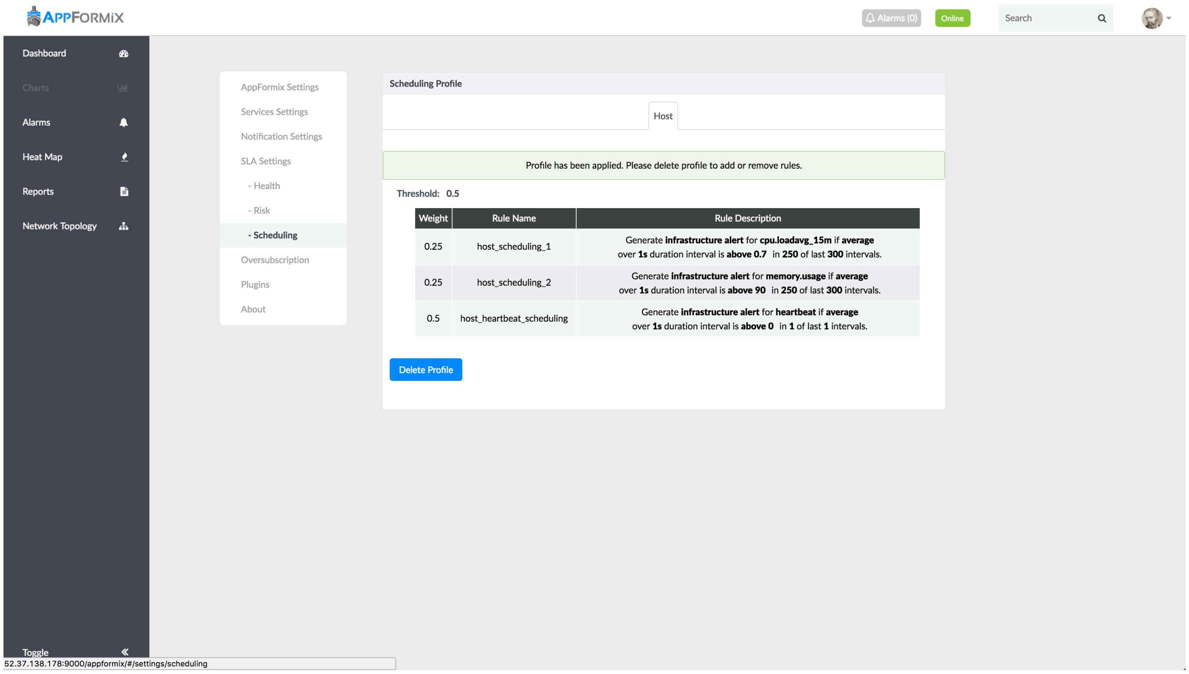
1. Select **Settings** from the list in the top right of the Dashboard, then select **SLA Settings** > **Scheduling**.

**Figure 31: Settings in Dashboard**



2. Click **Delete Profile** to delete the existing profile.

**Figure 32: Delete Scheduling Profile**



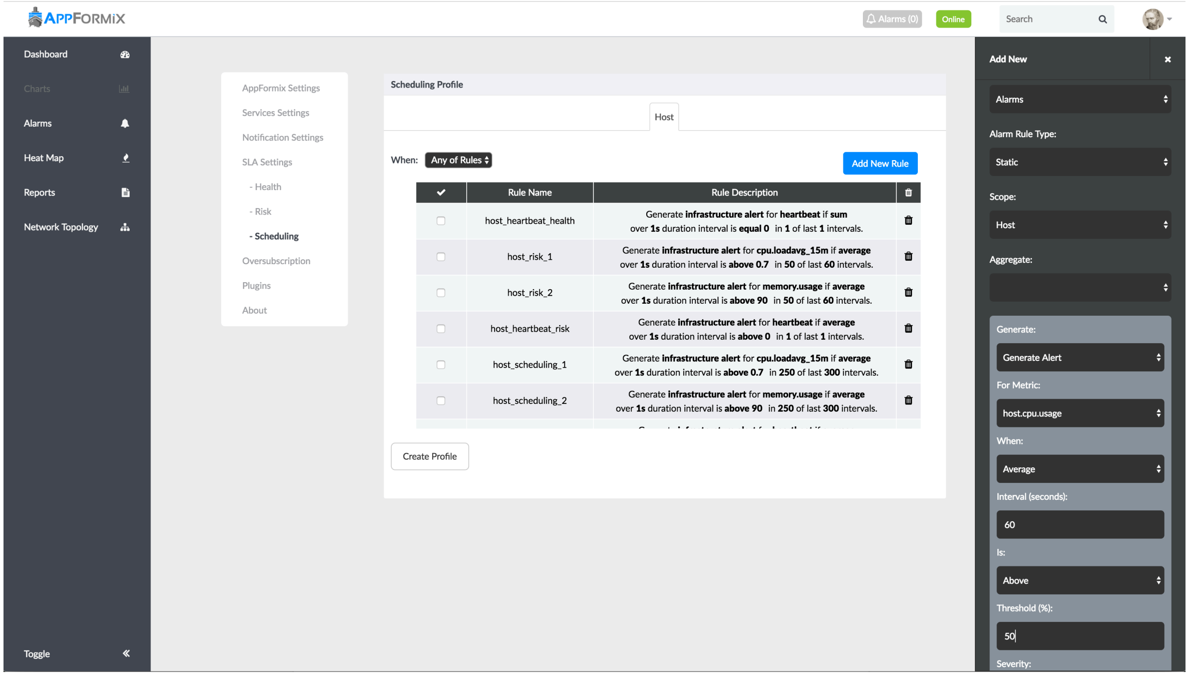
Profile has been applied. Please delete profile to add or remove rules.

Threshold: 0.5

| Weight | Rule Name | Rule Description |
|---|---|---|
| 0.25 | host_scheduling_1 | Generate **infrastructure alert** for **cpu.loadavg_15m** if **average** over **1s** duration interval is **above 0.7** in **250** of last **300** intervals. |
| 0.25 | host_scheduling_2 | Generate **infrastructure alert** for **memory.usage** if **average** over **1s** duration interval is **above 90** in **250** of last **300** intervals. |
| 0.5 | host_heartbeat_scheduling | Generate **infrastructure alert** for **heartbeat** if **average** over **1s** duration interval is **above 0** in **1** of last **1** intervals. |

3. Click **Add New Rule** and define a new alarm.

**Figure 33: Add New Rule in Scheduling Profile**



Scheduling Profile

When: Any of Rules

| ✓ | Rule Name | Rule Description | 🗑 |
|---|---|---|---|
| □ | host_heartbeat_health | Generate **infrastructure alert** for **heartbeat** if **sum** over **1s** duration interval is **equal 0** in **1** of last **1** intervals. | 🗑 |
| □ | host_risk_1 | Generate **infrastructure alert** for **cpu.loadavg_15m** if **average** over **1s** duration interval is **above 0.7** in **50** of last **60** intervals. | 🗑 |
| □ | host_risk_2 | Generate **infrastructure alert** for **memory.usage** if **average** over **1s** duration interval is **above 90** in **50** of last **60** intervals. | 🗑 |
| □ | host_heartbeat_risk | Generate **infrastructure alert** for **heartbeat** if **average** over **1s** duration interval is **above 0** in **1** of last **1** intervals. | 🗑 |
| □ | host_scheduling_1 | Generate **infrastructure alert** for **cpu.loadavg_15m** if **average** over **1s** duration interval is **above 0.7** in **250** of last **300** intervals. | 🗑 |
| □ | host_scheduling_2 | Generate **infrastructure alert** for **memory.usage** if **average** over **1s** duration interval is **above 90** in **250** of last **300** intervals. | 🗑 |

Add New

Alarms

Alarm Rule Type: Static

Scope: Host

Aggregate:

Generate: Generate Alert

For Metric: host.cpu.usage

When: Average

Interval (seconds): 60

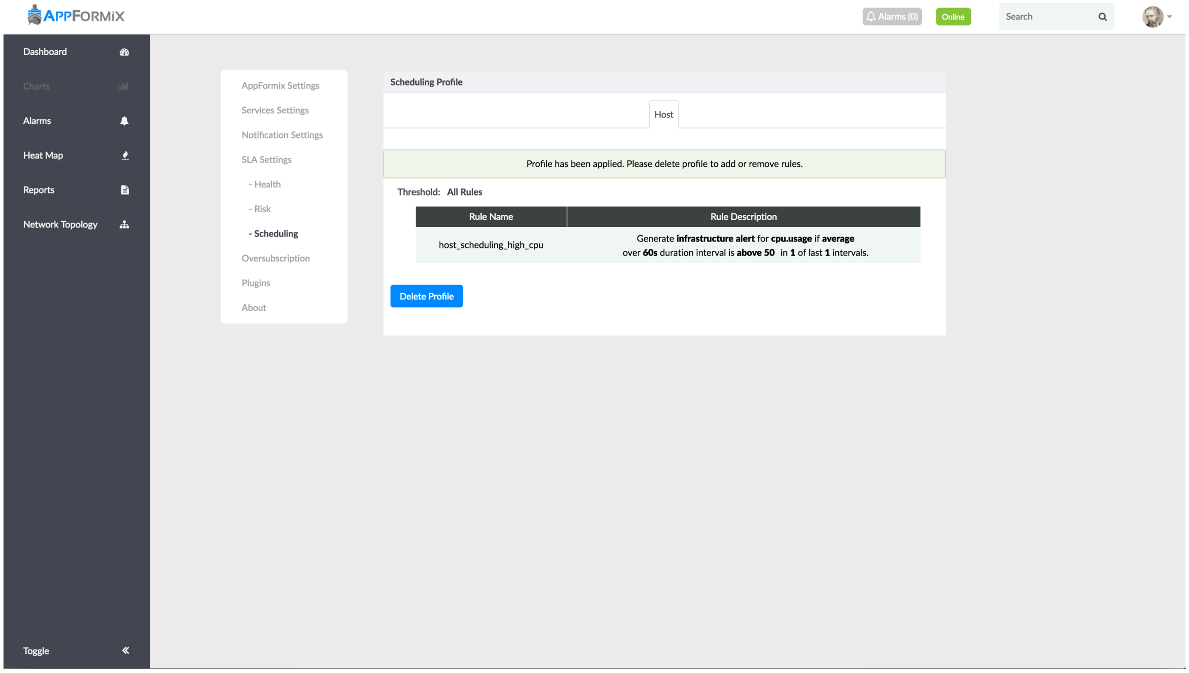
Is: Above

Threshold (%): 50

Severity:

4. Select the newly created alarm from the list of available alarms and click **Create Profile**. You can add several alarms with custom weights to the SLA profile.

**Figure 34: Create Profile in Scheduling SLA**



Scheduling Profile

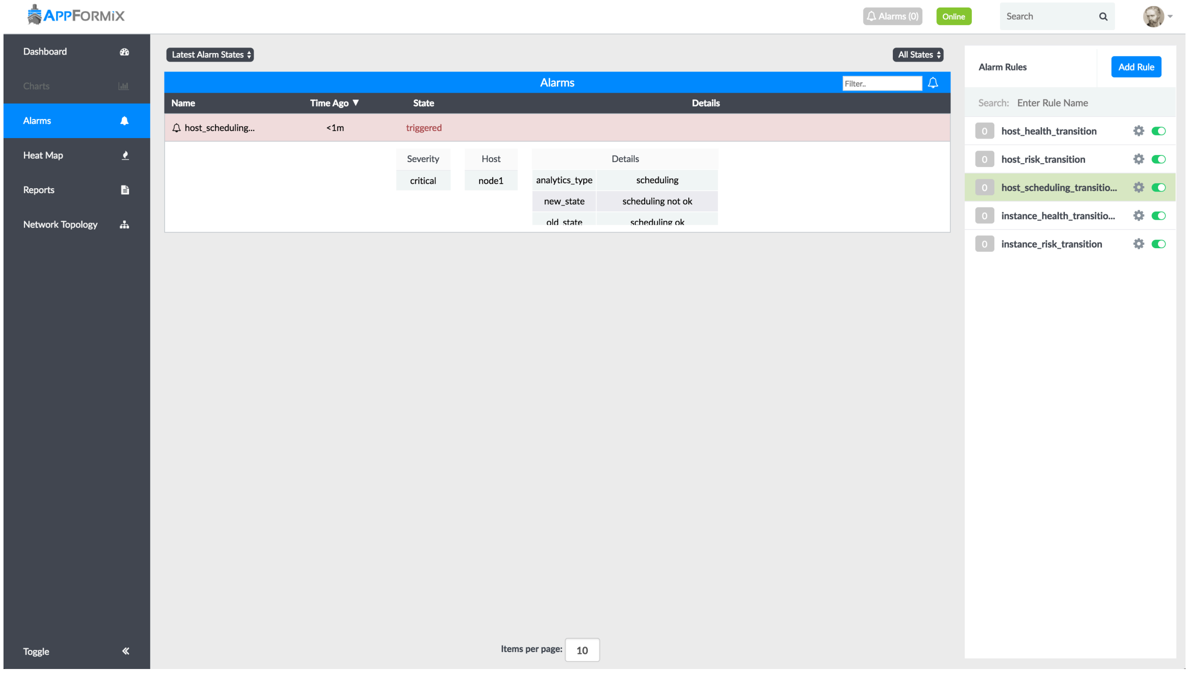
Profile has been applied. Please delete profile to add or remove rules.

Threshold: All Rules

| Rule Name | Rule Description |
|---|---|
| host_scheduling_high_cpu | Generate **infrastructure alert** for **cpu.usage** if **average** over **60s** duration interval is **above 50** in **1** of last **1** intervals. |

5. To see the plug-in in action, generate some load on one of the `nova-compute` hosts so that the Scheduling SLA is violated. Check the status of the SLA from the Alarms page.

**Figure 35: Violated Scheduling SLA in Alarms page**



Alarms

| Name | Time Ago ▼ | State | Details |
|---|---|---|---|
| host_scheduling... | <1m | triggered | |

| Severity | Host |
|---|---|
| critical | node1 |

| Details | |
|---|---|
| analytics_type | scheduling |
| new_state | scheduling not ok |
| old_state | scheduling ok |

| | |
|---|---|
| 0 | host_health_transition |
| 0 | host_risk_transition |
| 0 | host_scheduling_transitio... |
| 0 | instance_health_transitio... |
| 0 | instance_risk_transition |

Then create some new virtual machines and check which host they get scheduled on. The host that is violating the SLA will not have any new virtual machines scheduled on it. This will be enforced until the host starts complying with the SLA.

## Troubleshooting

For debugging, enable the `verbose` and `debug` options in `nova.conf` by adding them to the `[DEFAULT]` section in `/etc/nova/nova.conf`.

```
verbose = True
debug = True
```

Then restart the `nova-scheduler` service or all Nova containers. Detailed logs will appear in `/var/log/nova/nova.log`.

### RELATED DOCUMENTATION

*Capacity Planning*

# Extensibility Using Plug-Ins

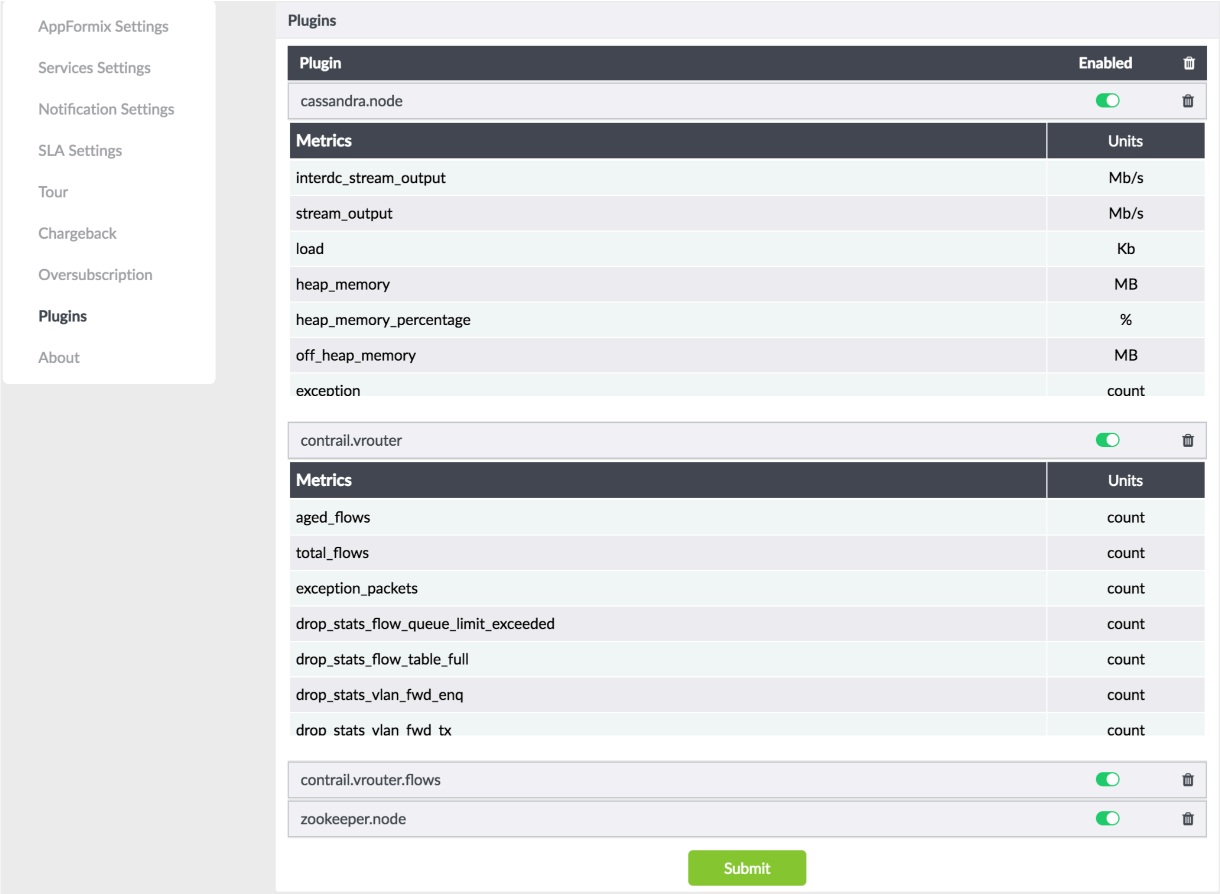
**IN THIS SECTION**

Plug-ins provide a framework for adding user-defined metrics to Contrail Insights. Metrics provided by a plug-in are available for charting and alarming. Plug-ins can be configured using the Ansible playbooks, as described in *Contrail Insights User-Defined Plug-Ins*.

## Plug-In Configuration Panel

After a plug-in is added to the Contrail Insights Platform, you can make modifications to the plug-in configuration from the settings panel on the dashboard. Figure 36 on page 58 shows the plug-in configuration panel.

**Figure 36: Plug-In Configuration Panel**



Plugins

| Plugin | Enabled |
|---|---|
| cassandra.node | ● |

| Metrics | Units |
|---|---|
| interdc_stream_output | Mb/s |
| stream_output | Mb/s |
| load | Kb |
| heap_memory | MB |
| heap_memory_percentage | % |
| off_heap_memory | MB |
| exception | count |

| Plugin | Enabled |
|---|---|
| contrail.vrouter | ● |

| Metrics | Units |
|---|---|
| aged_flows | count |
| total_flows | count |
| exception_packets | count |
| drop_stats_flow_queue_limit_exceeded | count |
| drop_stats_flow_table_full | count |
| drop_stats_vlan_fwd_enq | count |
| drop_stats_vlan_fwd_tx | count |

| Plugin | Enabled |
|---|---|
| contrail.vrouter.flows | ● |
| zookeeper.node | ● |

## Plug-In Grammar

A plug-in of type `command` is a Nagios-Style plug-in that outputs metrics as a string. A command plug-in may be any executable that outputs a string in the following format.

```
OK - <plugin_name_suffix>: <metric1.value><metric1.units> <metric1.name>, ...
<metricN.value><metricN.units> <metricN.name>
```

Table 3 on page 59 describes the fields.

**Table 3: Command Plug-In Fields**

| Field | Description |
|---|---|
| metric value | Must contain only digits and optional decimal point: `[0-9]+\.?[0-9]` |
| metric.units | Must be a valid string that starts with a letter. |
| metric.name | Must be a valid string that starts with a letter. |

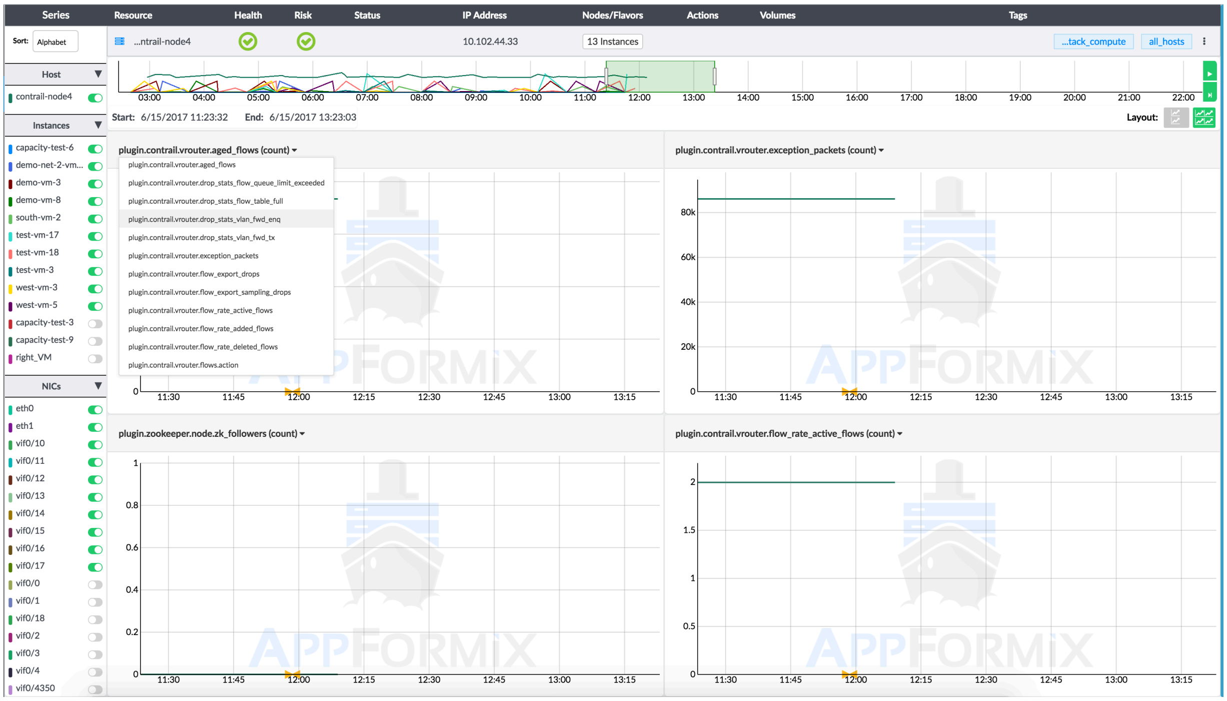
For example:

```
$ run_app1_performance.sh
OK - app1.performance: 102586MB/s bandwidth, 102610reqs/s queries_per_sec, 10count
active_connections, 5% capacity
```

## Plug-In Metrics Charts

After a plug-in is installed, you can navigate to the host chart page and the metrics will be visible in the charting panel on the hosts where the plug-in is enabled. Figure 37 on page 60 shows a plug-in metric chart displaying instances and NICs, color-coded and sorted by start and end date, as well a times.

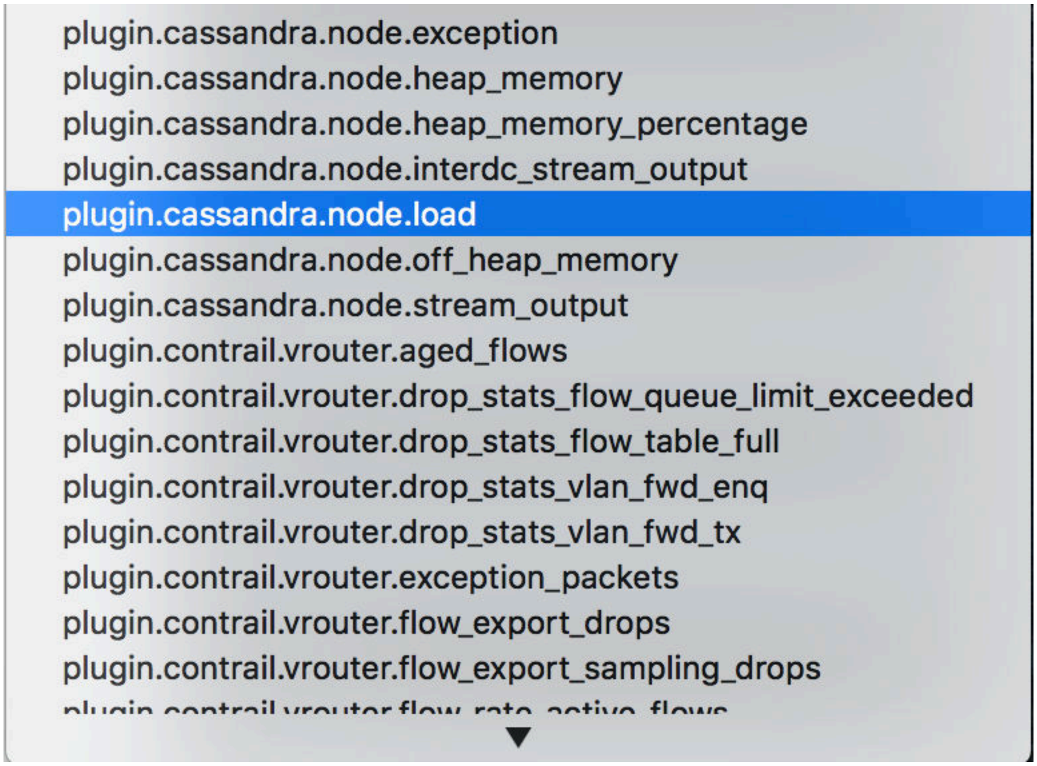
**Figure 37: Plug-In Metrics Chart**


Series
Resource
Health
Risk
Status
IP Address
Nodes/Flavors
Actions
Volumes
Tags
Sort: Alphabet
...ntrail-node4
10.102.44.33
13 Instances
...tack_compute
all_hosts
Host
contrail-node4
Instances
capacity-test-6
demo-net-2-vm...
demo-vm-3
demo-vm-8
south-vm-2
test-vm-17
test-vm-18
test-vm-3
west-vm-3
west-vm-5
capacity-test-3
capacity-test-9
right_VM
NICs
eth0
eth1
vif0/10
vif0/11
vif0/12
vif0/13
vif0/14
vif0/15
vif0/16
vif0/17
vif0/0
vif0/1
vif0/18
vif0/2
vif0/3
vif0/4
vif0/4350
Start: 6/15/2017 11:23:32
End: 6/15/2017 13:23:03
Layout:
plugin.contrail.vrouter.aged_flows (count)
plugin.contrail.vrouter.aged_flows
plugin.contrail.vrouter.drop_stats_flow_queue_limit_exceeded
plugin.contrail.vrouter.drop_stats_flow_table_full
plugin.contrail.vrouter.drop_stats_vlan_fwd_enq
plugin.contrail.vrouter.drop_stats_vlan_fwd_tx
plugin.contrail.vrouter.exception_packets
plugin.contrail.vrouter.flow_export_drops
plugin.contrail.vrouter.flow_export_sampling_drops
plugin.contrail.vrouter.flow_rate_active_flows
plugin.contrail.vrouter.flow_rate_added_flows
plugin.contrail.vrouter.flow_rate_deleted_flows
plugin.contrail.vrouter.flows.action
plugin.contrail.vrouter.exception_packets (count)
plugin.zookeeper.node.zk_followers (count)
plugin.contrail.vrouter.flow_rate_active_flows (count)
03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00
80k 60k 40k 20k 0
1 0.8 0.6 0.4 0.2 0
2 1.5 1 0.5 0
11:30 11:45 12:00 12:15 12:30 12:45 13:00 13:15
APPFORMIX

## Plug-In Metrics Alarms

An alarm can be configured for any plug-in metric from the panel for the alarm configuration. If the plug-in is enabled the plug-in metrics are also available on the alarm panel. Figure 38 on page 61 shows a list of these metric alarms.

**Figure 38: Plug-In Metric Alarms**



```
plugin.cassandra.node.exception
plugin.cassandra.node.heap_memory
plugin.cassandra.node.heap_memory_percentage
plugin.cassandra.node.interdc_stream_output
plugin.cassandra.node.load
plugin.cassandra.node.off_heap_memory
plugin.cassandra.node.stream_output
plugin.contrail.vrouter.aged_flows
plugin.contrail.vrouter.drop_stats_flow_queue_limit_exceeded
plugin.contrail.vrouter.drop_stats_flow_table_full
plugin.contrail.vrouter.drop_stats_vlan_fwd_enq
plugin.contrail.vrouter.drop_stats_vlan_fwd_tx
plugin.contrail.vrouter.exception_packets
plugin.contrail.vrouter.flow_export_drops
plugin.contrail.vrouter.flow_export_sampling_drops
plugin.contrail.vrouter.flow_rate_active_flows
```

## RELATED DOCUMENTATION

*Contrail Insights User-Defined Plug-Ins*

*Alarms*

*Capacity Planning*

*Chargeback*

*Charts*

*Health Monitor*

*Heat Map*

*Metrics Collected by Contrail Insights*

*Notifications*

*Reports*

*Service Monitoring from the UI*

# Configure Network Devices from the UI

IN THIS SECTION

Starting with Contrail Insights v2.18, Contrail Insights has a dedicated view for adding, modifying, or deleting network devices from the UI.

## Configure Network Devices

Select **Settings** in the top right of the Dashboard, then select **Network Devices**. A list of all devices monitored by Contrail Insights in your cluster displays as shown in No Link Title.

## Add a Network Device

To add a network device:

1. Select **Settings** in the top right of the Dashboard, then select **Network Devices**.
2. Click **+Add Device**.
3. Follow the wizard instructions to add your SNMP, JTI, or gRPC devices. One device can have multiple sources (SNMP, JTI, and gRPC).
   - It might take several minutes for Contrail Insights to discover the device's name and interfaces.

- If an error was made during configuration, you can modify the existing device by clicking on the gear icon of the device you want to edit.

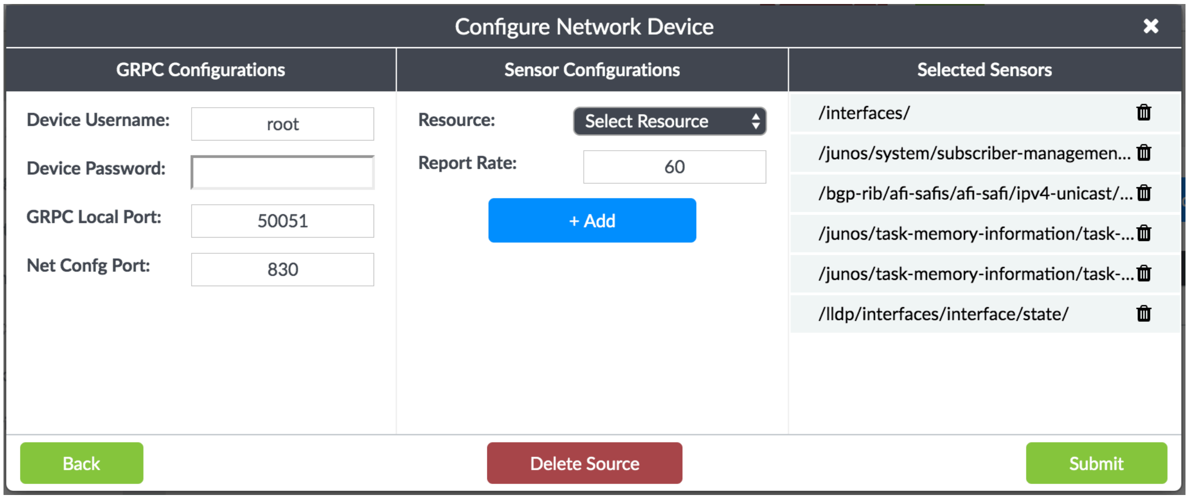
4. To add multiple devices with the same configuration, see the following section "Copying an Existing Device's Configurations."

## Edit an Existing Network Device

To edit an existing network device:

1. Select **Settings** in the top right of the Dashboard, then select **Network Devices**.
2. Click the gear icon next to the target device on the Network Devices page. See No Link Title.
3. Continue with the wizard to edit your network device. You can choose the configuration of sources (SNMP, JTI, or gRPC) to edit. You can also add new sources for this device.
4. In the individual source page, you can change credentials, add/edit/delete MIBs and sensors on this device, or delete this source from this device. See Figure 39 on page 63.

**Figure 39: Individual Device Page to add, edit, or delete MIBs and Sensors**



Configure Network Device

GRPC Configurations

Device Username: root

Device Password:

GRPC Local Port: 50051

Net Confg Port: 830

Sensor Configurations

Resource: Select Resource

Report Rate: 60

Selected Sensors

/interfaces/

/junos/system/subscriber-managemen...

/bgp-rib/afi-safis/afi-safi/ipv4-unicast/...

/junos/task-memory-information/task-...

/junos/task-memory-information/task-...

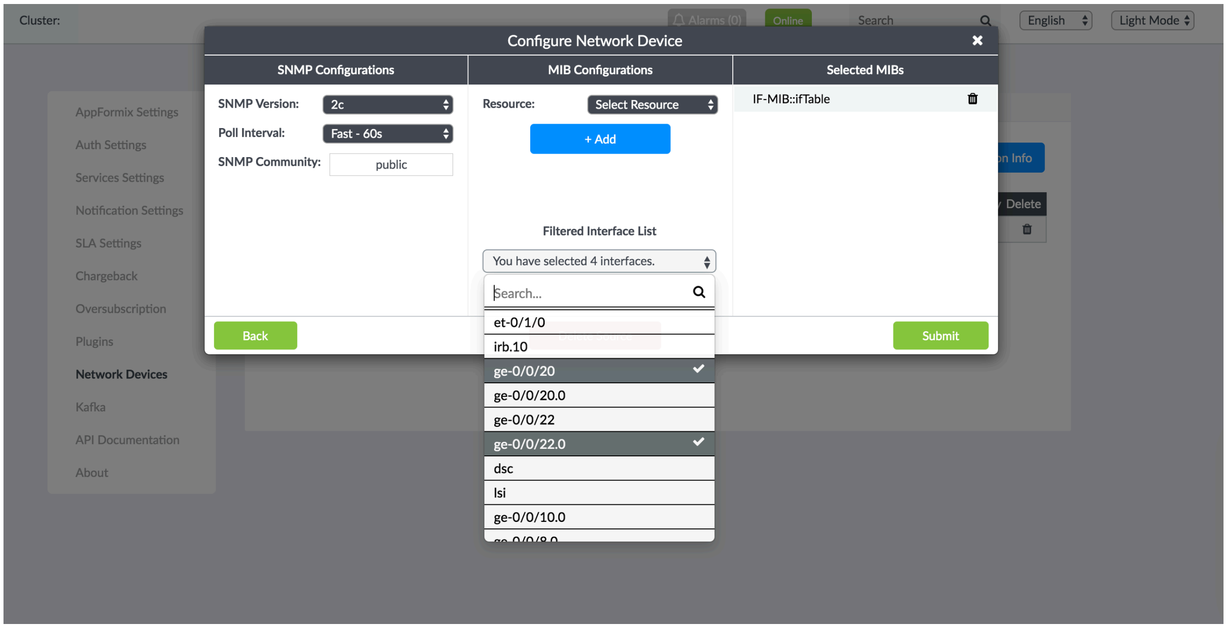
/lldp/interfaces/interface/state/

## Add Filtered Interface List to SNMP Device

Polling network devices for SNMP data at regular intervals adds load on the device and affects the query performance. In some scenarios, you might be interested in monitoring only a certain set of interfaces from a device. Contrail Insights allows you to select a subset of interfaces to monitor. Contrail Insights will only run `snmpwalk` against those interfaces, which reduces the device load and makes the Contrail Insights SNMP query faster.

> **NOTE**: You can select a subset of interfaces to monitor only after the device is added and interface list is discovered.

**Figure 40: Selecting a Subset of Interfaces to Monitor**



Configure Network Device

SNMP Version: 2c

Poll Interval: Fast - 60s

SNMP Community: public

Resource: Select Resource

Filtered Interface List: You have selected 4 interfaces.

Selected MIBs: IF-MIB::ifTable

## Copy an Existing Device’s Configurations

To copy an existing device’s configurations:

1. Select **Settings** in the top right of the Dashboard, then select **Network Devices**.

2. Click the clipboard icon next to the target device you want to copy.

3. Select a management IP from the list of currently available devices or choose to add a new device by clicking **Create New Device**.

4. Add a single device or add multiple devices by selecting the radio button at the top.

To add multiple IP addresses, indicate the range of the IP addresses separated by a "-". For example, to add 10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4, type **10.1.1.1 - 4** in the **IP Range** field.

5. Click **Save** after all devices are added to the list on the right.

## Configure Connection Information Between Devices

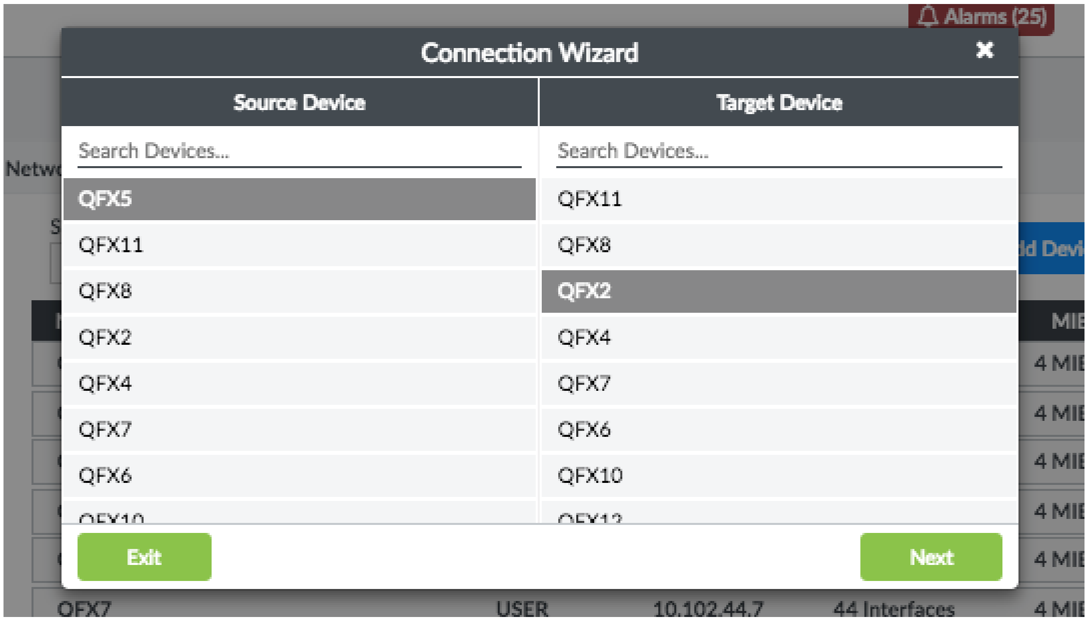
To configure connection information between devices:

1. Select **Settings** in the top right of the Dashboard, then select **Network Devices**.

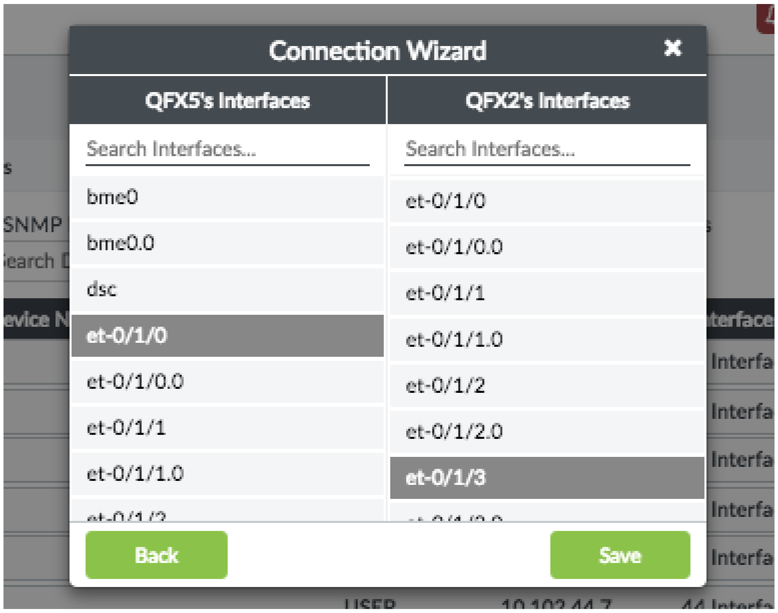
2. Click **Edit Connection Info**. See No Link Title.

3. Continue with the wizard or navigate to the topology view to configure the devices visually.

**Figure 41: Configure the Connection Between the Source Device and Target Device**



4. Click **Next** to continue and configure the connection between the source and target interfaces.



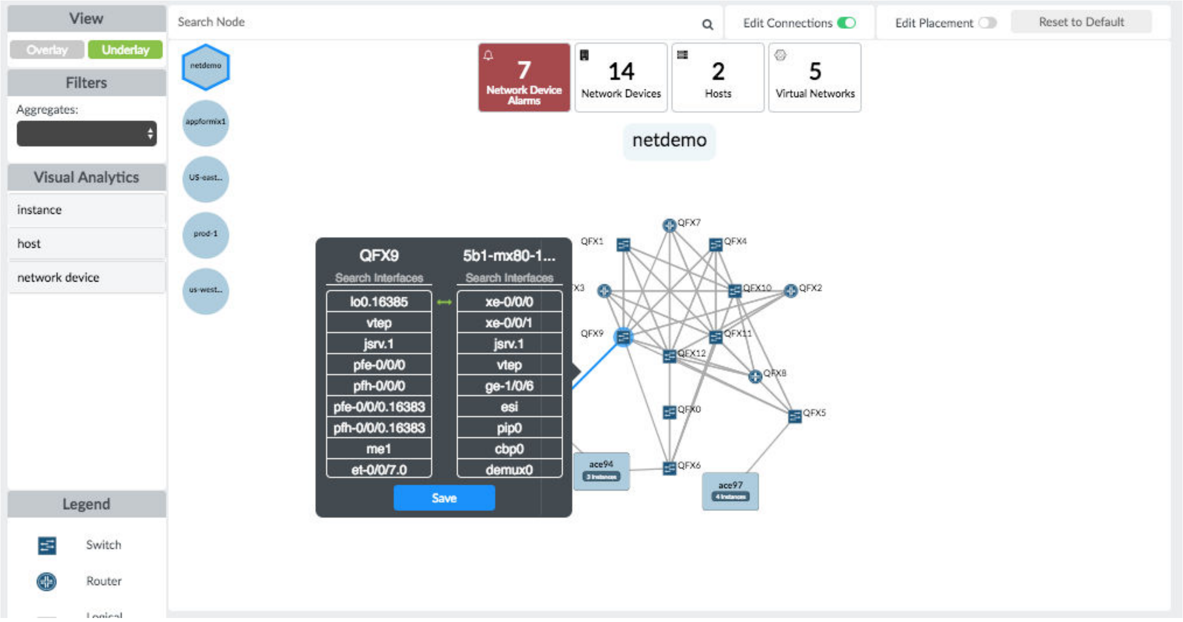
5. Click **Save** to confirm.

### Edit from Topology View

To edit from topology view:

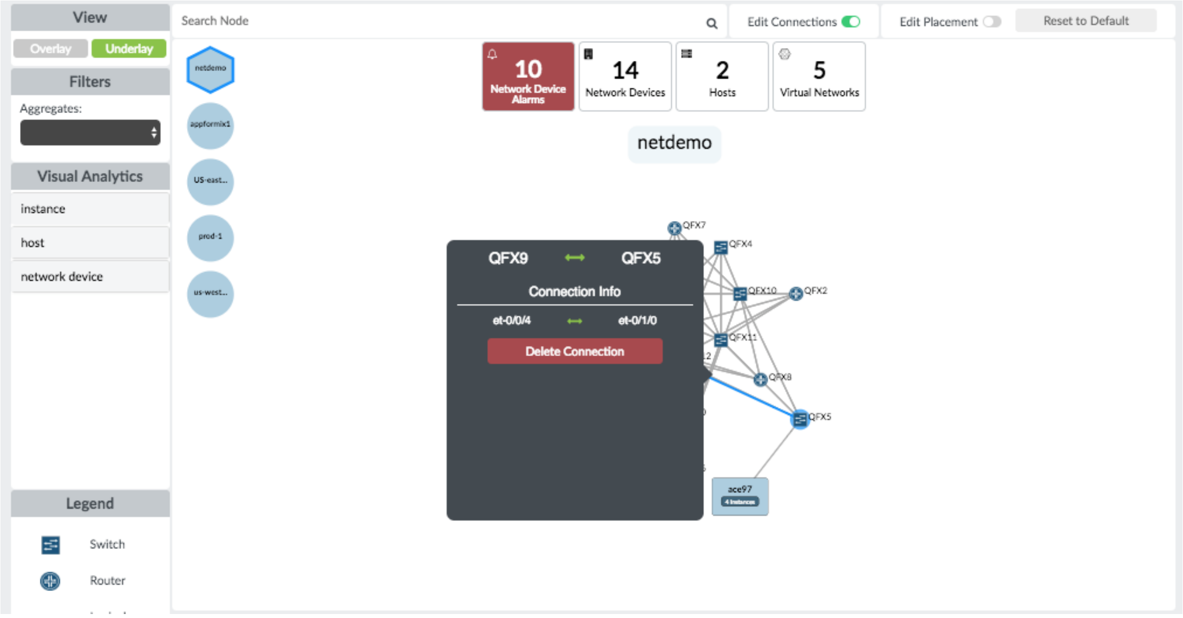
1. Select **Settings** in the top right of the Dashboard, then select **Network Devices**.
2. Click **Edit Connection Info**. See No Link Title.
3. Select the two devices that you want to either add or remove links between them. The devices selected will be highlighted in blue.
4. Select the desired interface from its respective device and click **Save** to confirm.

Figure 42: Selecting the Devices and Interfaces to Edit from Topology View



5. (Optional) To delete a connection, select the link between two connected devices and click **Delete Connection**.

Figure 43: Deleting a Connection from Topology View



# Contrail Insights Auto Discovery of Network Devices from Contrail Networking

IN THIS SECTION

Juniper® Contrail Networking is a software-defined networking (SDN) platform based on the open-source network virtualization project, OpenContrail. The Contrail Networking platform automates and orchestrates the creation of highly scalable virtual networks.

Contrail Insights provides monitoring and orchestration for the Contrail Service. See Service Monitoring Ansible Variables for how to configure Contrail monitoring. See "Contrail Monitoring" on page 165 for how Contrail monitoring works in Contrail Insights.

## Network Device Discovery from Contrail

You can add network devices from the Contrail UI to Contrail and Contrail Insights has the capability to auto discovery all the network devices you added to Contrail. Contrail Insights will discovery all of the network devices' IP address, Chassis Type, and Connection Information.

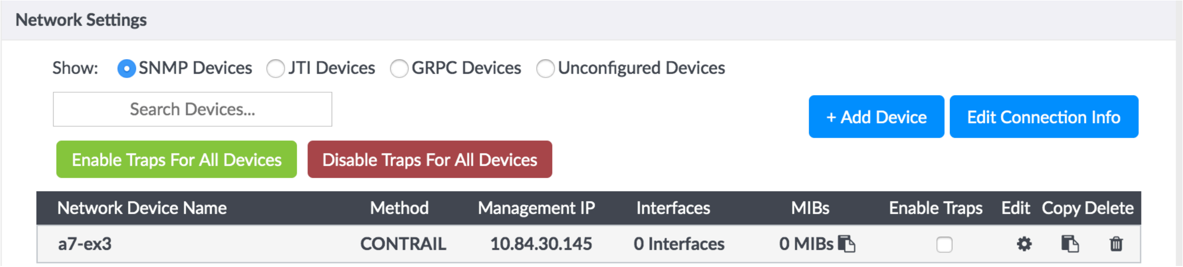
### Before Contrail Insights Version 3.1.9

To enable Contrail Insights monitoring on these added devices:

1. From the Contrail Insights Dashboard, select **Settings > Network Devices** to edit the network devices discovered by Contrail Insights.

2. Add Source **SNMP**, **gRPC**, or **JTI** and the corresponding configurations (such as credentials, sensors, MIBs, and so on) in the Contrail Insights Dashboard and Contrail Insights will automatically start monitoring those devices.

Figure 44: Enable Network Monitoring on Added Network Devices



Network Settings

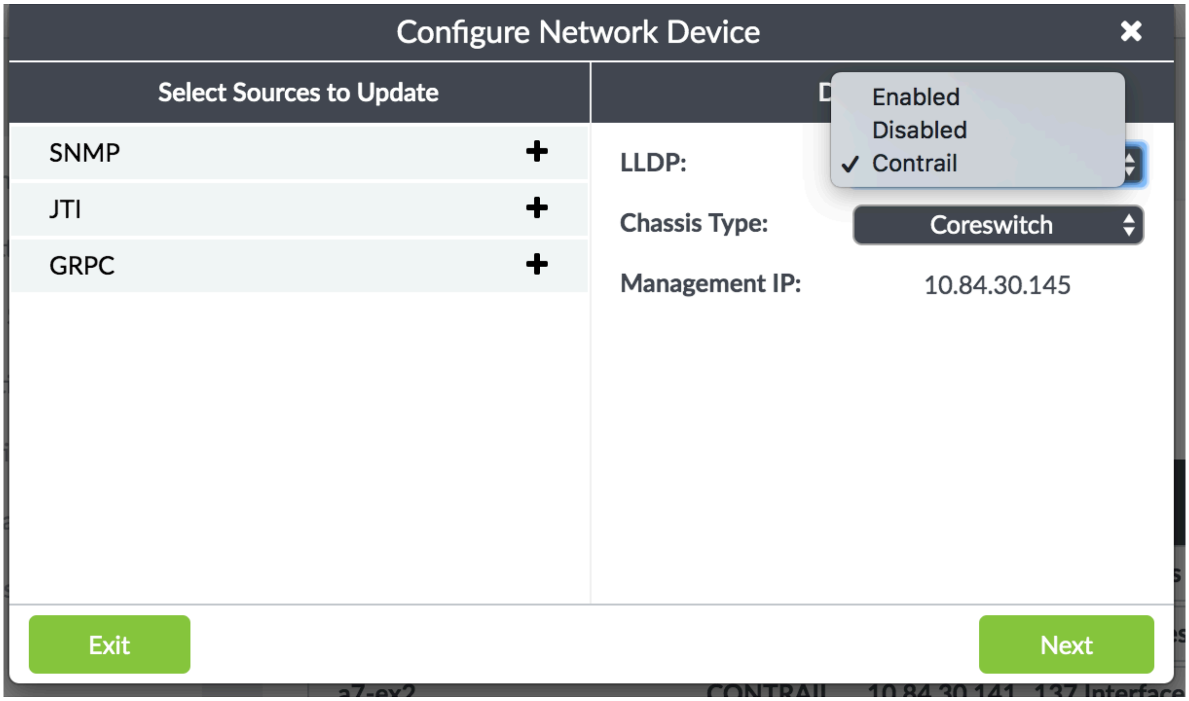
Show: ● SNMP Devices ○ JTI Devices ○ GRPC Devices ○ Unconfigured Devices

Search Devices...

| Network Device Name | Method | Management IP | Interfaces | MIBs | Enable Traps | Edit | Copy | Delete |
|---|---|---|---|---|---|---|---|---|
| a7-ex3 | CONTRAIL | 10.84.30.145 | 0 Interfaces | 0 MIBs | □ | | | |

By default, all the network devices discovered from Contrail are associated with Method **Contrail**. This means Contrail Insights will get the Chassis Type and Connection Information of devices from Contrail and honor the data from Contrail. If any of the devices' Connections get updated, Contrail Insights will automatically discover the newest Connection from Contrail and update it in Contrail Insights. However, you can also change the Method to **USER** or **LLDP** in the Contrail Insights Dashboard Settings page.

Figure 45: Changing Method to LLDP Enabled or Disabled



Configure Network Device

Select Sources to Update

LLDP: Contrail

Chassis Type: Coreswitch

Management IP: 10.84.30.145

If you change Method to LLDP **Enabled** in Contrail Insights, Contrail Insights will start running LLDP periodically on that device. Contrail Insights will build the Connection Information of this device on its own and will no longer get this device Connection Information from Contrail.

If you change Method to LLDP **Disabled** in Contrail Insights, Contrail Insights will assume user will take care of the device Connection Information themselves. User can add, edit, or delete connections of this device from Contrail Insights Dashboard either in Settings page or Topology page. Contrail Insights will no longer get this device Connection Information from Contrail.

In both LLDP and USER mode, deleting or updating this device in Contrail will no longer take effect on the device configuration on Contrail Insights.

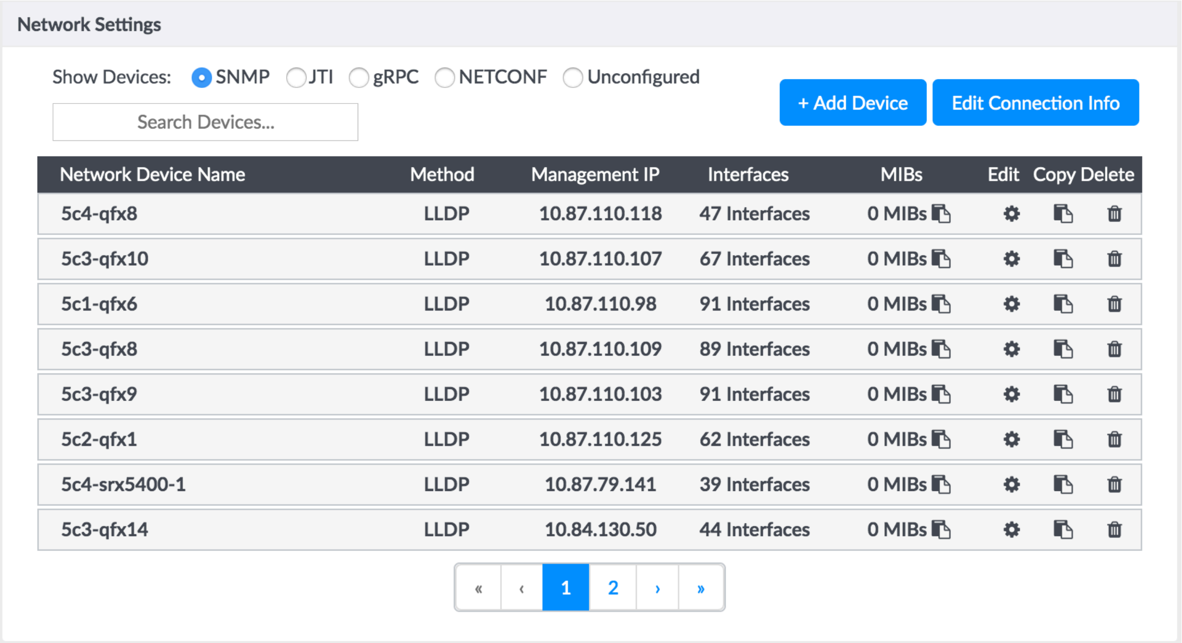
### Starting from Contrail Insights Version 3.1.9

To enable Contrail Insights monitoring on these added devices:

1. From the Contrail Insights Dashboard, select **Settings > Network Devices** to edit the network devices discovered by Contrail Insights.

2. Add Source **SNMP**, **gRPC**, or **JTI** and the corresponding configurations, such as credentials, sensors, MIBs, and so on) in the Contrail Insights Dashboard and Contrail Insights will automatically start monitoring those devices.

**Figure 46: Network Devices Discovered from Contrail Associated with Method LLDP**



Show Devices: ● SNMP ○ JTI ○ gRPC ○ NETCONF ○ Unconfigured

| Network Device Name | Method | Management IP | Interfaces | MIBs | Edit | Copy | Delete |
|---|---|---|---|---|---|---|---|
| 5c4-qfx8 | LLDP | 10.87.110.118 | 47 Interfaces | 0 MIBs | ⚙ | ⎘ | 🗑 |
| 5c3-qfx10 | LLDP | 10.87.110.107 | 67 Interfaces | 0 MIBs | ⚙ | ⎘ | 🗑 |
| 5c1-qfx6 | LLDP | 10.87.110.98 | 91 Interfaces | 0 MIBs | ⚙ | ⎘ | 🗑 |
| 5c3-qfx8 | LLDP | 10.87.110.109 | 89 Interfaces | 0 MIBs | ⚙ | ⎘ | 🗑 |
| 5c3-qfx9 | LLDP | 10.87.110.103 | 91 Interfaces | 0 MIBs | ⚙ | ⎘ | 🗑 |
| 5c2-qfx1 | LLDP | 10.87.110.125 | 62 Interfaces | 0 MIBs | ⚙ | ⎘ | 🗑 |
| 5c4-srx5400-1 | LLDP | 10.87.79.141 | 39 Interfaces | 0 MIBs | ⚙ | ⎘ | 🗑 |
| 5c3-qfx14 | LLDP | 10.84.130.50 | 44 Interfaces | 0 MIBs | ⚙ | ⎘ | 🗑 |

By default, all the network devices discovered from Contrail are associated with Method **LLDP**. It means Contrail Insights will automatically discover the Connection between network devices and hosts. However, you can also change the Method to **USER** in Contrail Insights Dashboard Settings page so that you can edit the connection manually. Note that the Connection between network devices and hosts will be discovered only when the network devices are Juniper QFX series or EX series.

Contrail Insights will still synchronize the device ChassisType and SNMP credentials with Contrail. Any update of these fields in Contrail will be reflected here. If you delete devices from Contrail, those deleted devices will also be removed from Contrail Insights.

If you change the Method to **LLDP Disabled** in Contrail Insights, Contrail Insights will assume user will take care of the device Connection Information themselves. User can add, edit, or delete connections of this device from Contrail Insights Dashboard in the Settings page or Topology page.

RELATED DOCUMENTATION

Contrail Insights Network Device Monitoring Common Issues | 233

Contrail Insights JTI (UDP) Monitoring | 214

Contrail Insights JTI (gRPC) Monitoring | 222

Contrail Insights SNMP Monitoring | 227

# SNMP Traps in Contrail Insights

IN THIS SECTION

Contrail Insights supports monitoring of SNMP traps sent from network devices. Traps are unsolicited messages sent from an SNMP agent to remote network management systems or trap receivers.

# Configuring Devices to Forward SNMP Traps

For Contrail Insights to listen to SNMP traps from devices, you need to configure the devices to forward the traps because they are not forwarded by default. This can be done either manually from the Junos OS CLI of the device or through Contrail Insights software development kit (SDK).

## Check the SNMP Trap Configuration on Device

**Before Contrail Insights Version 3.1**:

In Contrail Insights version 3.0, Contrail Insights only supports SNMPv2 traps. After the device is configured to forward SNMP traps, you can verify the configuration by logging into the Junos OS CLI and running the following command:

```
show snmp
```

The output should be similar to the following example:

```
trap-options {
    source-address {device_ip};
}
trap-group snmp-trap-metallb-test {
    version v2;
    destination-port 42597;
    categories {
        link;
        authentication;
    }
    targets {
        {collector_1_ip};
        {collector_2_ip};
    }
}
```

For additional details regarding configuration, refer to *Configuring SNMP Traps*.

**After Contrail Insights Version 3.1**:

When Contrail Insights version 3.1 or later is installed, SNMP trap configuration on the device needs to be updated because the configuration for the devices on Contrail Insights version 3.0 is no longer valid. In Contrail Insights version 3.1 or later, Contrail Insights supports both SNMPv2 and SNMPv3. You need to configure the device using the following sample configuration so that Contrail Insights will collect the SNMP traps.

After the device is configured to forward SNMP traps, you can verify the configuration by logging into Junos OS CLI and running the following command:

```
show snmp v3
```

The output should be similar to the following example for SNMPv2:

**NOTE**: The `security-name public` in the following configuration refers to the SNMPv2 community name you set in your device. Set the SNMPv2 community name before you add this SNMP trap configuration.

```
...
SNMP v2c Configuration
...
target-address appformix_snmp_v2 {
    address x.x.x.x;
    port 42597;
    tag-list appformix_snmp_v2;
    target-parameters appformix_snmp_v2;
}
target-parameters appformix_snmp_v2 {
    parameters {
        message-processing-model v2c;
        security-model v2c;
        security-level none;
        security-name public; //this is the snmp v2c community name
    }
    notify-filter appformix_snmp_v2;
}
notify appformix_snmp_v2 {
    type trap;
    tag appformix_snmp_v2;
}
notify-filter appformix_snmp_v2 {
    oid .1 include;
}
```

The output should be similar to the following example for SNMPv3:

```
...
SNMP v3 Configuration
...
target-address appformix_snmp_v3 {
    address x.x.x.x;
    port 42597;
    tag-list appformix_snmp_v3;
    target-parameters appformix_snmp_v3;
}
target-parameters appformix_snmp_v3 {
    parameters {
        message-processing-model v3;
        security-model usm;
        security-level authentication;
        security-name acelio;
    }
    notify-filter appformix_snmp_v3;
}
notify appformix_snmp_v3 {
    type trap;
    tag appformix_snmp_v3;
}
notify-filter appformix_snmp_v3 {
    oid .1 include;
}
```

For additional details regarding configuration, refer to *Configuring SNMPv3 Traps on a Device Running Junos OS*.

The variables `security-model`, `security-level`, and `security-name` are related to the SNMPv3 configuration you set in this device. Configure the device with SNMPv3 credentials before you enable SNMPv3 traps.

## Configuring Contrail Insights to Enable SNMP Traps Monitoring from Network Devices

### Enable Listening to SNMP Traps for Network Devices

In Contrail Insights Dashboard, **Settings > Network Devices**, you can add or edit SNMP device configuration and enable Contrail Insights to collect the SNMP traps for those configured devices. As long as you have posted the `snmp_trap_network_device` plug-in from Ansible, Contrail Insights will automatically start listening on SNMP traps from all SNMP network devices configured in Contrail Insights.

> **NOTE**: The field `SnmpEngineId` is needed when you want to enable SNMPv3 traps for a device. This field is not required for normal SNMP polling.

### Create Network Device JSON File for SNMPv2c

The list of network devices that needs to be monitored should be added to a JSON file with the following format. There can be multiple devices in the JSON file.

```
{
  "NetworkDeviceList": [
    {
      "NetworkDevice": {
        "MetaData": {
          "SnmpConfig": {
            "Version": "2c",
            "OIDList": ["TCP-MIB::tcp",
                        "IF-MIB::ifTable",
                        "enterprises.2636.3.1.13.1"],
            "Community": "public"}
        },
        "Name": "QFX0",
        "NetworkDeviceId": "QFX0",
        "ManagementIp": "x.x.x.x",
        "ChassisType": "tor",
        "Source": ["user.snmp"],
        "InterfaceList": [
        ],
        "ConnectionInfo": []
      }
    }
  ]
}
```

The `user.snmp` needs to be included in `Source` field. Contrail Insights automatically starts monitoring the traps sent from all `user.snmp` devices configured in Contrail Insights. For more details about other fields and how to post network devices using Ansible, refer to *Configure Network Device from JSON File*.

### Create Network Device JSON File for SNMPv3

The list of network devices that needs to be monitored should be added to a JSON file using the following format. There can be multiple devices in the JSON file. For SNMPv3 traps, you need to specify the `SnmpEngineId` for `SnmpConfig`.

```
{
  "NetworkDeviceList": [
    {
      "NetworkDevice": {
        "MetaData": {
          "SnmpConfig": {
            "Version": "3",
            "Password": "pwd",
            "Level": "authPriv",
            "PrivKey": "privkey",
            "PrivProtocol": "DES",
            "Protocol": "MD5",
            "SnmpEngineId": "80000a4c010a574478",
            "OIDList": ["TCP-MIB::tcp",
                        "IF-MIB::ifTable",
                        "enterprises.2636.3.1.13.1"],
            "Username": "user"}
        },
        "Name": "QFX0",
        "NetworkDeviceId": "QFX0",
        "ManagementIp": "x.x.x.x",
        "ChassisType": "tor",
        "Source": ["user.snmp"],
        "InterfaceList": [
        ],
        "ConnectionInfo": []
      }
    }
  ]
}
```

## Configuring Contrail Insights Network Device Monitoring Plug-Ins

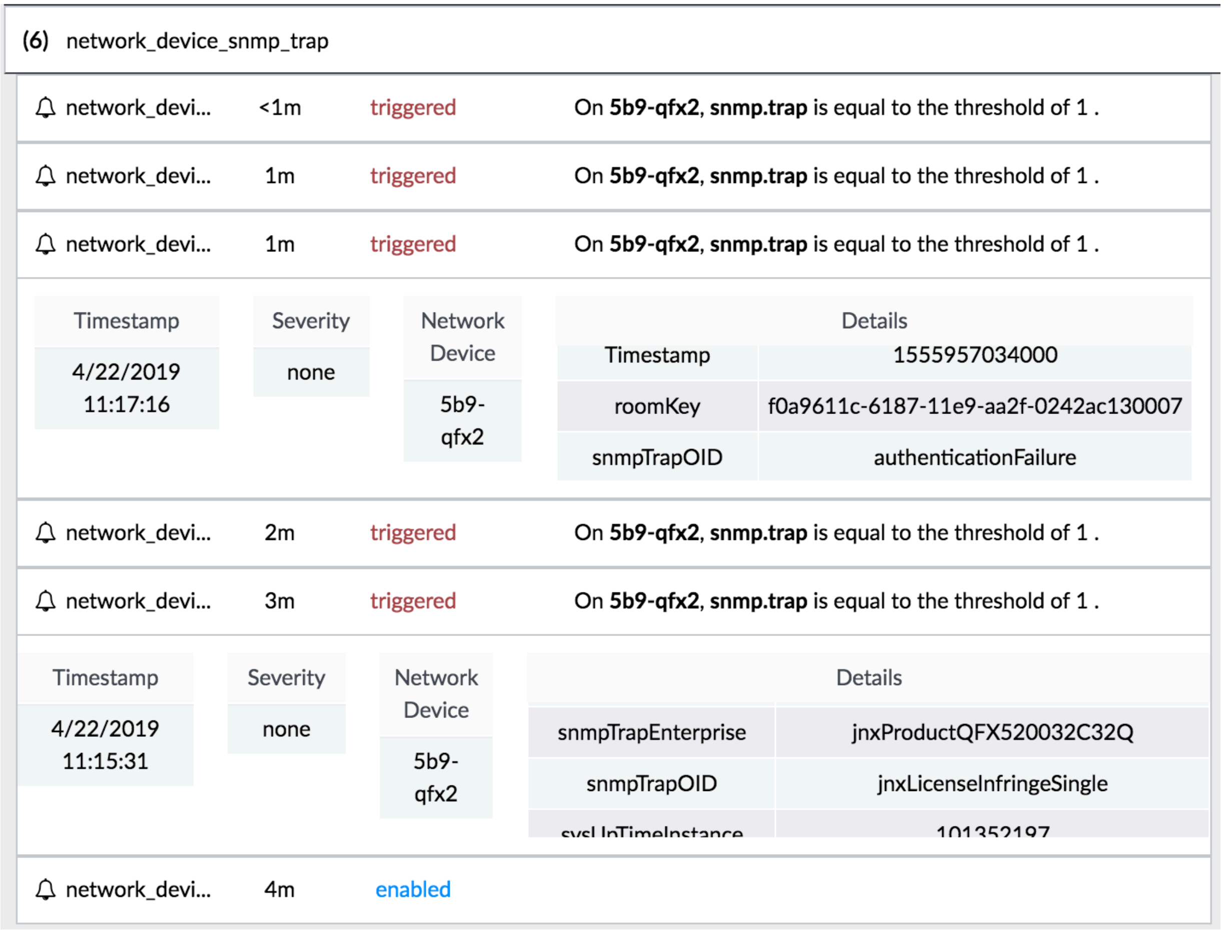
Contrail Insights needs to be configured at the time of installation to enable the SNMP trap plug-in. Contrail Insights has a built-in SNMP trap plug-in in the `certified_plugins` folder in the Ansible installation directory. This needs to be included in the plug-in descriptor in the `appformix_plugins` variable in `group_vars/all`.

```
# network_device_file_name is optional, if you want to add devices from UI, then you don't need
it
network_device_file_name: <path_to_above_json_file>
appformix_plugins:
- { plugin_info: certified_plugins/snmp_trap_network_device.json }
```

## Enable SNMP Trap to Show in Contrail Insights Dashboard

There is a built-in SNMP trap rule that is configured in `profiles/network_device_snmp_trap_profile.json` to enable SNMP trap pop-up in the Contrail Insights Dashboard Alarm page. The trap is posted to Contrail Insights by default and every time traps are sent to Contrail Insights Agent, the traps appear in the Contrail Insights Dashboard Alarm page and display detailed information about the traps Contrail Insights receives.

Figure 47: SNMP Traps Enabled and Displayed in Contrail Insights Dashboard.



(6) network_device_snmp_trap

network_devi... <1m triggered On **5b9-qfx2**, **snmp.trap** is equal to the threshold of 1 .

network_devi... 1m triggered On **5b9-qfx2**, **snmp.trap** is equal to the threshold of 1 .

network_devi... 1m triggered On **5b9-qfx2**, **snmp.trap** is equal to the threshold of 1 .

| Timestamp | Severity | Network Device | Details | |
|---|---|---|---|---|
| 4/22/2019 11:17:16 | none | 5b9-qfx2 | Timestamp | 1555957034000 |
| | | | roomKey | f0a9611c-6187-11e9-aa2f-0242ac130007 |
| | | | snmpTrapOID | authenticationFailure |

network_devi... 2m triggered On **5b9-qfx2**, **snmp.trap** is equal to the threshold of 1 .

network_devi... 3m triggered On **5b9-qfx2**, **snmp.trap** is equal to the threshold of 1 .

| Timestamp | Severity | Network Device | Details | |
|---|---|---|---|---|
| 4/22/2019 11:15:31 | none | 5b9-qfx2 | snmpTrapEnterprise | jnxProductQFX520032C32Q |
| | | | snmpTrapOID | jnxLicenseInfringeSingle |
| | | | sysUpTimeInstance | 101352197 |

network_devi... 4m enabled

## SNMP Trap Data for External Notification

When Contrail Insights receives a SNMP trap, Contrail Insights displays the trap in **Dashboard > Alarms** as rule `network_device_snmp_trap` and sends it to Apache Kafka, if Kafka has been configured in Contrail Insights. You can associate the rule `network_device_snmp_trap` with external notifiers such as PagerDuty, ServiceNow, Slack, Custom Notifier, and so on.

Following is an example JSON file sent to external notifiers for SNMP trap:

```
{'status': {
   'description': 'NetworkDevice sample_device: SNMP Trap Received for OID=linkUp',
    'timestamp': 1555549001000,
    'entityType': 'network_device',
    'state': 'triggered',
    'entityDetails': {},
    'entityId': 'sample_device',
    'metaData': {
      'snmpTrapOID': 'linkUp',
      'Timestamp': 1555548996000,
      'ifAdminStatus': '1',
      'roomKey': 'sample_device',
      'ifIndex': '545',
      'ifName': 'irb.20',
      'ifOperStatus': '1',
      'sysUpTimeInstance': '1028117810'}
  },
  'kind': 'Alarm',
  'spec': {
    'aggregationFunction': 'sum',
    'intervalDuration': 1,
    'severity': 'none',
    'module': 'alarms',
    'intervalCount': 1,
    'metricType': 'snmp.trap',
    'name': 'network_device_snmp_trap',
    'eventRuleId': 'NETWORK_DEVICE_SNMP_TRAP',
    'mode': 'event',
    'intervalsWithException': 1,
    'threshold': 1,
    'comparisonFunction': 'equal'},
'apiVersion': 'v2'}
```

You can find a brief description of the SNMP trap in the `status` > `description` field, detailed information of the SNMP trap in the `status` > `metaData` field, and `status` > `entityId` tells you which network device this trap belongs to.

## Install MIBs in Contrail Insights Network Agents

When Contrail Insights receives the traps from devices, Contrail Insights might not be able to decode the OID into a proper user understandable string if corresponding MIBs are not installed in your Contrail Insights Agents. You need to download the MIBs and either manually copy all of the `*.txt` MIB files to all network agents `/usr/share/snmp/mibs/` or use Contrail Insights Ansible to deploy the MIB files.

To install MIBs from Ansible see *Custom SNMP Plug-Ins*.

## RELATED DOCUMENTATION

*Contrail Insights SNMP Monitoring*

Configuring SNMPv3 Traps on a Device Running Junos OS

*Custom SNMP Plug-Ins*

3

CHAPTER

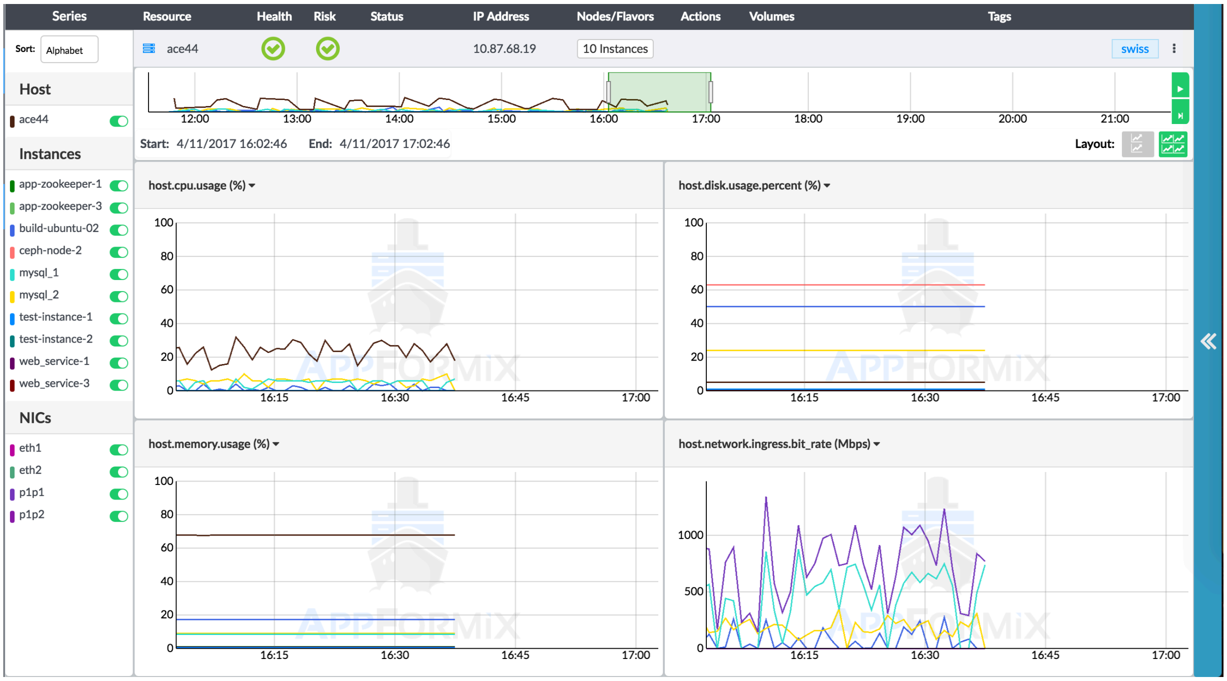
# Monitoring

# Charts

## IN THIS SECTION

With Contrail Insights Charts, you can view real-time and historical values of all metrics that Contrail Insights monitors. Charts provide you with a way to view metrics for multiple entities across layers and organized by physical host, project, or aggregate. The charts update with the latest data streamed from the Contrail Insights Platform without needing to refresh. You can select which entities to display on the charts, and select the time period that is displayed. When you hover over the charts, a pop-up box shows the actual values for the selected entities at a specific point in time. Figure 48 on page 82 shows real-time metric values streamed from the Contrail Insights Platform.

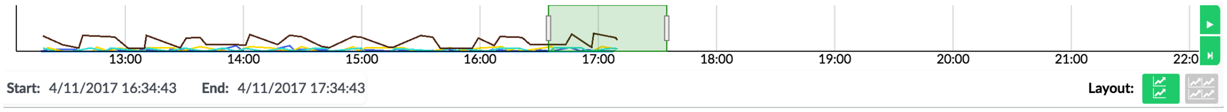
**Figure 48: Real-Time Metric Values Streamed from the Contrail Insights Platform**



## Timeline

The Timeline at the top of the page provides navigation to a specific point in time that you want to view. The green rectangle on the Timeline can be dragged left or right, or resized to change the time window displayed in the charts. To the right of the Timeline, the play/pause button (top button) allows you to pause and start the charts from moving. The live button (bottom button) resets the view to the current time. Figure 49 on page 83 shows navigation using the green rectangle for a timeline from 16:43:43 to 17:34:43.

**Figure 49: Chart Timeline For Viewing Metrics for Specific Times**

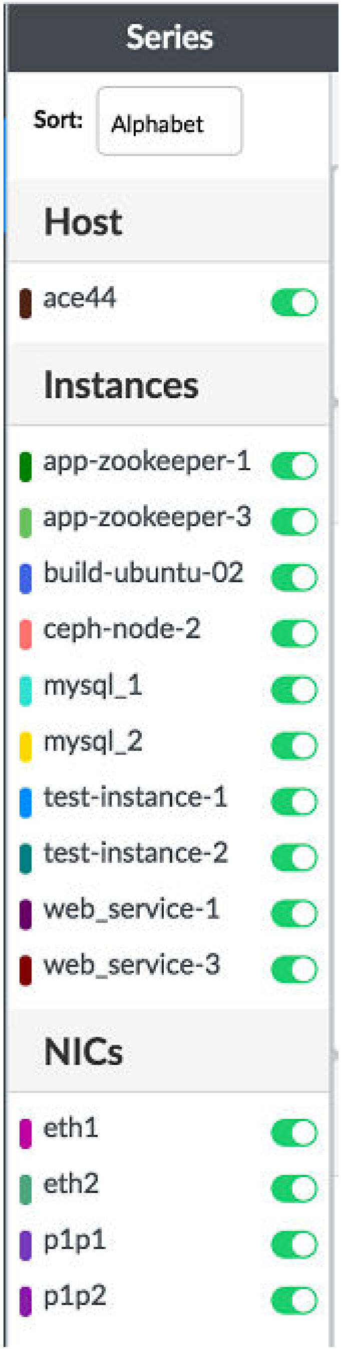


Start: 4/11/2017 16:34:43 End: 4/11/2017 17:34:43

## Chart Legend

The chart legend shows which entities are currently being displayed in the charts. You can select a subset of entities to display to improve the clarity of the charts and focus on specific entities. By default, the entities are sorted alphabetically, but they can be sorted by a specific metric as well. Figure 50 on page 84 shows the chart legend.

**Figure 50: Chart Legend Showing Entities Currently Displayed**



Series

Sort: Alphabet

Host

- ace44

Instances

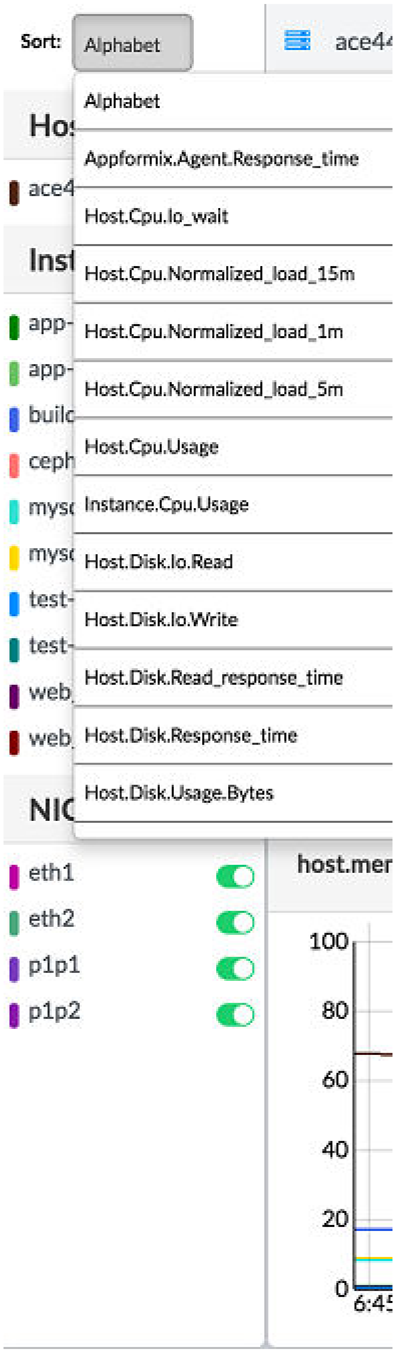
- app-zookeeper-1
- app-zookeeper-3
- build-ubuntu-02
- ceph-node-2
- mysql_1
- mysql_2
- test-instance-1
- test-instance-2
- web_service-1
- web_service-3

NICs

- eth1
- eth2
- p1p1
- p1p2

When selecting a metric by which to sort, the top 10 entities will be selected, as shown in Figure 51 on page 85.

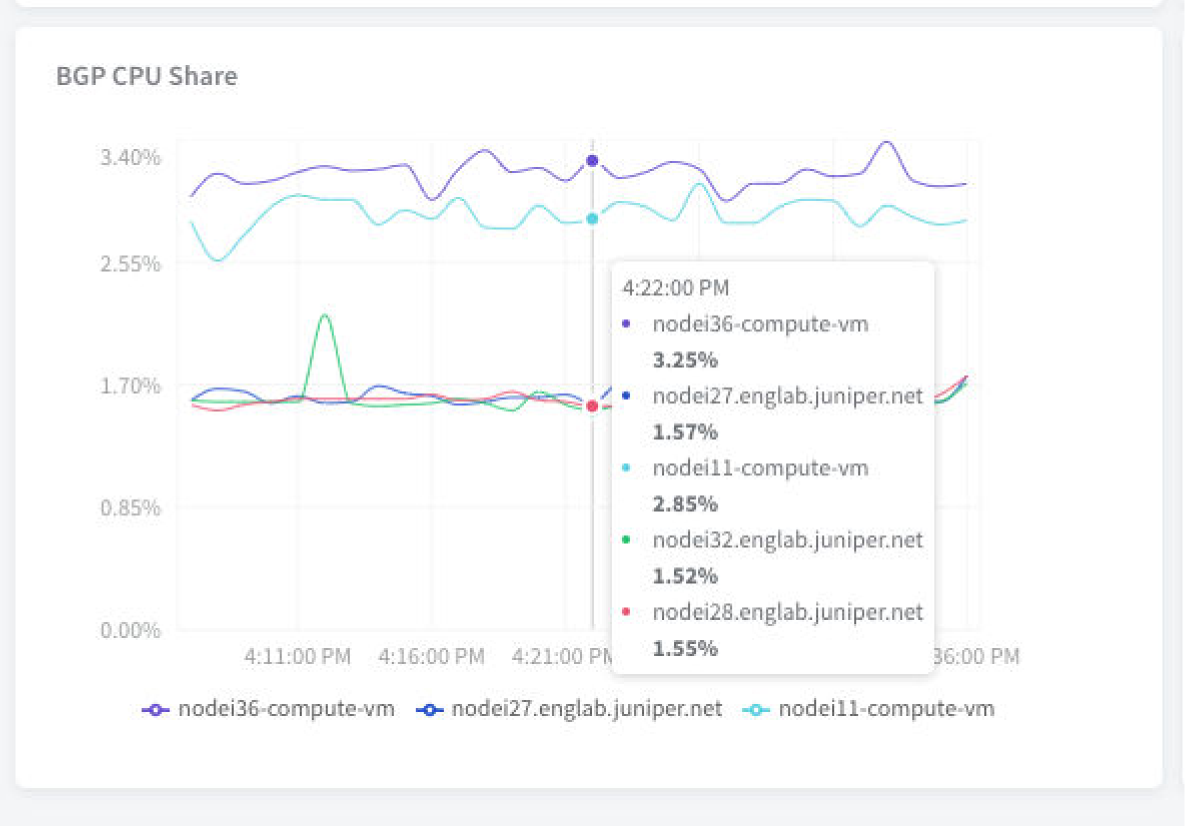
**Figure 51: Chart Legend Sort by Metrics and Selected Entities**



## Chart Data Values

At the center of the page, the charts show the latest data for up to four different metrics, updating in real-time from a stream of data from the Contrail Insights Platform. When the cursor is positioned over the charts, a pop-up box shows the data values at that particular time. Charts can be zoomed in or out using the mouse scroll wheel. You can choose to display two or four charts at a time. Figure 52 on page 86 shows the chart data values pop-up box for a particular time.

**Figure 52: Chart Data Values Pop-Up Box for a Particular Time**

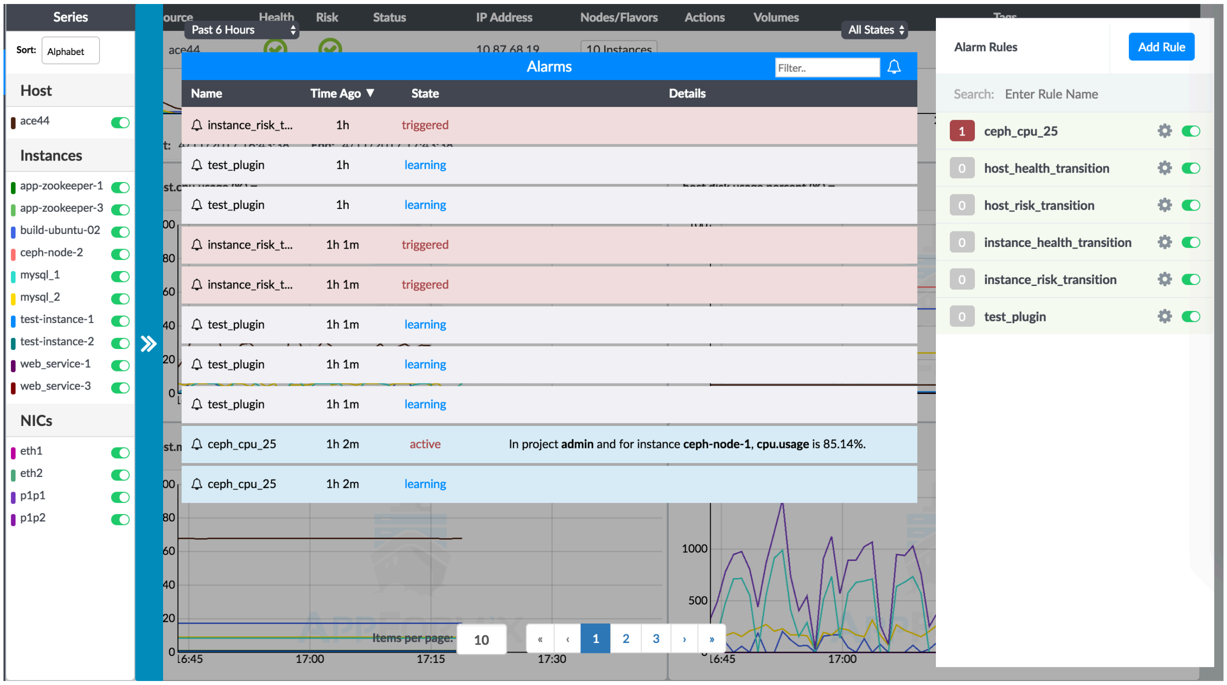


## Alarms on Charts

Alarms can be viewed without navigating away from the charts. There is a blue expand button to the right side of the charts that overlays the alarms history and configuration on top of the charts view.

Any alarms that occur while on the page will display as symbols on the chart. A circle appears at the time a new alarm enters learning state. A triangle pointing to the right indicates the time at which an alarm became active. A triangle pointing to the left indicates the time at which an alarm became inactive. If any symbol is clicked, then a pop-up box will display the details about the alarm that fired. Figure 53 on page 87 shows the alarms history and state from the charts view.

### Figure 53: Alarms History and State from the Charts View



Alarms

| Name | Time Ago ▼ | State | Details |
|---|---|---|---|
| instance_risk_t... | 1h | triggered | |
| test_plugin | 1h | learning | |
| test_plugin | 1h | learning | |
| instance_risk_t... | 1h 1m | triggered | |
| instance_risk_t... | 1h 1m | triggered | |
| test_plugin | 1h 1m | learning | |
| test_plugin | 1h 1m | learning | |
| test_plugin | 1h 1m | learning | |
| ceph_cpu_25 | 1h 2m | active | In project **admin** and for instance **ceph-node-1**, **cpu.usage** is 85.14%. |
| ceph_cpu_25 | 1h 2m | learning | |

Alarm Rules

Add Rule

Search: Enter Rule Name

| | |
|---|---|
| 1 | ceph_cpu_25 |
| 0 | host_health_transition |
| 0 | host_risk_transition |
| 0 | instance_health_transition |
| 0 | instance_risk_transition |
| 0 | test_plugin |

## RELATED DOCUMENTATION

*Alarms*

*Capacity Planning*

*Chargeback*

*Health Monitor*

*Heat Map*

*Metrics Collected by Contrail Insights*

*Notifications*

*Extensibility Using Plug-Ins*

*Reports*

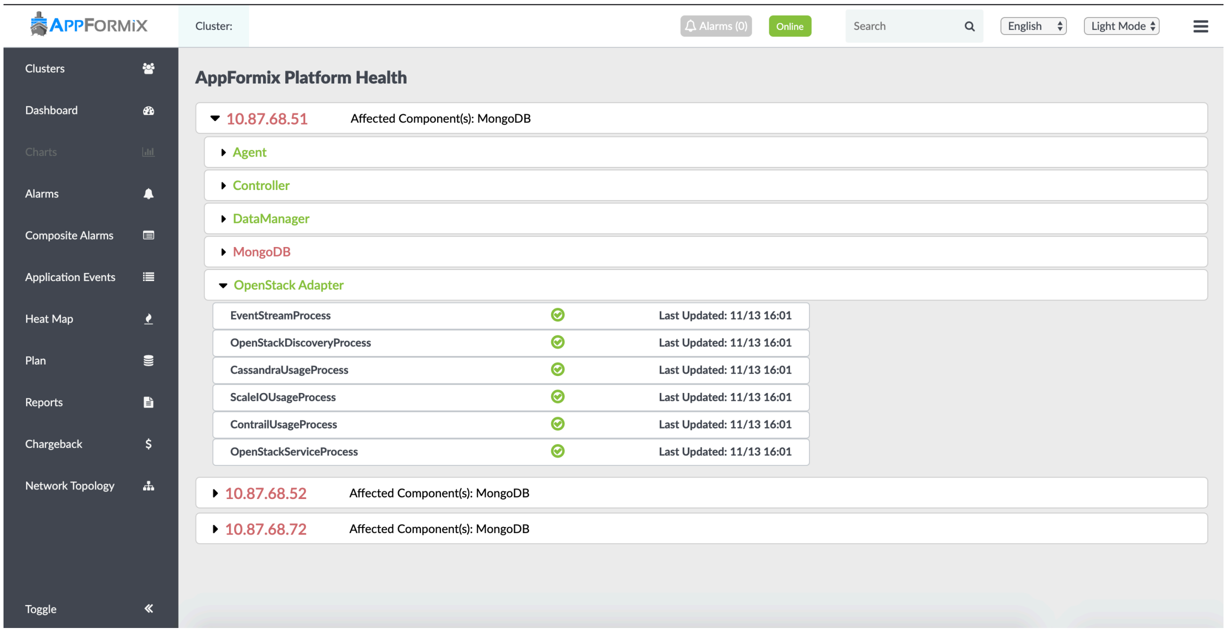
*Service Monitoring from the UI*

# Contrail Insights Platform Health

IN THIS SECTION

All the Contrail Insights Platform components can be monitored from the Contrail Insights Platform Health page. To access this page, click the menu in the top right corner, and from the drop down list, select **Platform Health**. This page provides useful data such as connection statuses, usage statistics, and errors that provide an overview of the health of the components.

**Figure 54: Contrail Insights Platform Health Page**



AppFormix Platform Health

▾ 10.87.68.51 Affected Component(s): MongoDB

▸ Agent

▸ Controller

▸ DataManager

▸ MongoDB

▾ OpenStack Adapter

| | | |
|---|---|---|
| EventStreamProcess | ✓ | Last Updated: 11/13 16:01 |
| OpenStackDiscoveryProcess | ✓ | Last Updated: 11/13 16:01 |
| CassandraUsageProcess | ✓ | Last Updated: 11/13 16:01 |
| ScaleIOUsageProcess | ✓ | Last Updated: 11/13 16:01 |
| ContrailUsageProcess | ✓ | Last Updated: 11/13 16:01 |
| OpenStackServiceProcess | ✓ | Last Updated: 11/13 16:01 |

▸ 10.87.68.52 Affected Component(s): MongoDB

▸ 10.87.68.72 Affected Component(s): MongoDB

This page shows relevant health statistics for each of the Contrail Insights Platform components, namely, Controller, OpenStack Adapter, Agent, DataManager, Mongo, Redis, and HAProxy.

In addition to the UI, these health statistics can also be obtained using APIs.

## Contrail Insights Controller

Health panel for Contrail Insights controller shows the RedisConnectionStatus, MongoConnectionStatus, ProcessStatuses, CeleryTaskStatus. For ProcessStatuses, the time the process last sent an update is tracked, hence checking liveness of the process, and errors logged by the process since last update.

API:

```
http://<appformix-vip>:<appformix-port>/appformix/controller/v2.0/controller_health
```

The response is a `task_id`.

Using this `task_id`, call the following endpoint to get the result:

```
http://<appformix-vip>:<appformix-port>/appformix/controller/v2.0/task/<task_id>/result
```

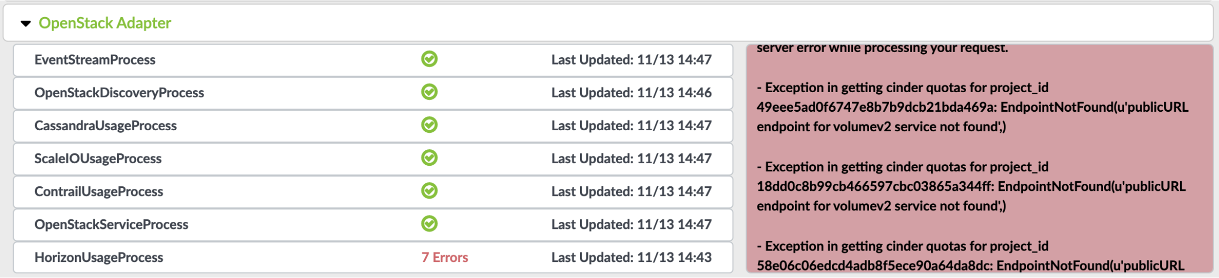
## Contrail Insights OpenStack Adapter

Similar to the Controller panel, OpenStack Adapter panel shows the statuses of various processes and any error logs.

API:

```
http://<appformix-vip>:<appformix-port>/appformix/openstack_adapter/v2.0/status
```

**Figure 55: OpenStack Adapter Panel**



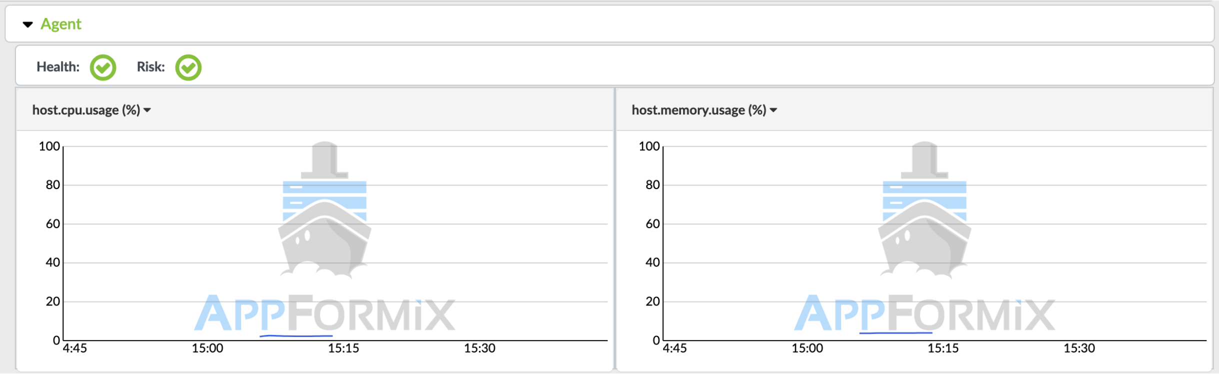
| | | |
|---|---|---|
| EventStreamProcess | ✓ | Last Updated: 11/13 14:47 |
| OpenStackDiscoveryProcess | ✓ | Last Updated: 11/13 14:46 |
| CassandraUsageProcess | ✓ | Last Updated: 11/13 14:47 |
| ScaleIOUsageProcess | ✓ | Last Updated: 11/13 14:47 |
| ContrailUsageProcess | ✓ | Last Updated: 11/13 14:47 |
| OpenStackServiceProcess | ✓ | Last Updated: 11/13 14:47 |
| HorizonUsageProcess | 7 Errors | Last Updated: 11/13 14:43 |

```
server error while processing your request.

- Exception in getting cinder quotas for project_id
49eee5ad0f6747e8b7b9dcb21bda469a: EndpointNotFound(u'publicURL
endpoint for volumev2 service not found',)

- Exception in getting cinder quotas for project_id
18dd0c8b99cb466597cbc03865a344ff: EndpointNotFound(u'publicURL
endpoint for volumev2 service not found',)

- Exception in getting cinder quotas for project_id
58e06c06edcd4adb8f5ece90a64da8dc: EndpointNotFound(u'publicURL
```

## Contrail Insights Agent

The Contrail Insights Agent panel shows all the regular host level metrics along with host's Health and Risk.

**Figure 56: Host Level Metrics with Health and Risk**

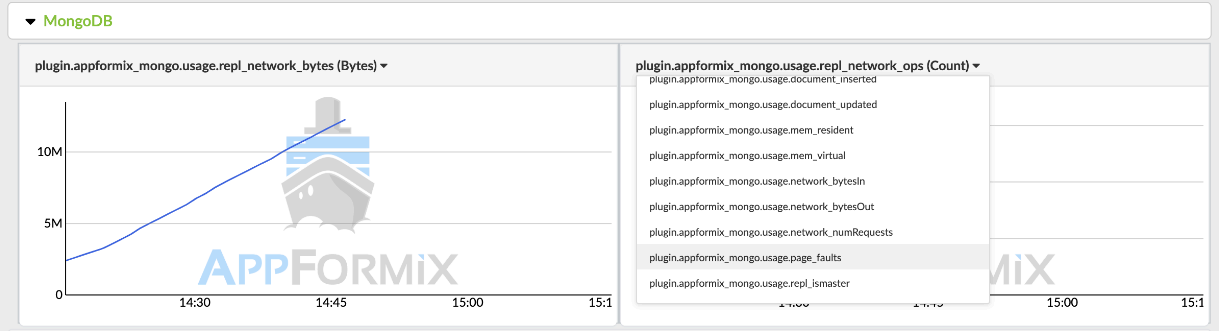


Health: ✓ Risk: ✓

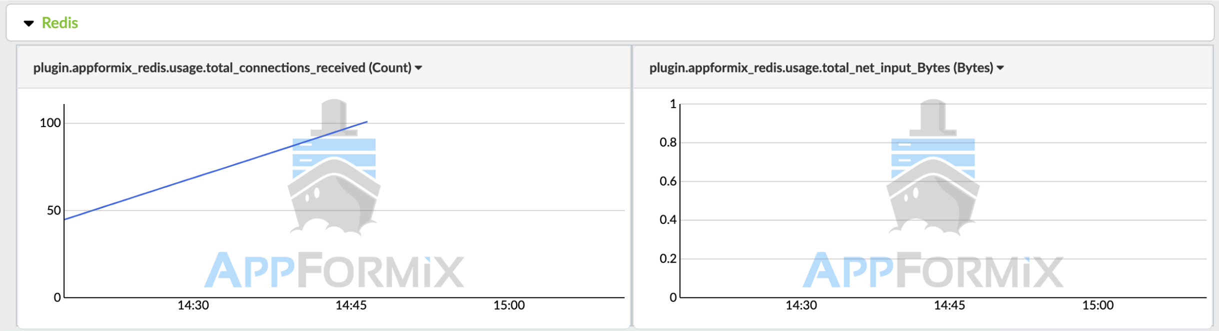
## Other Components

The captured metrics for other components can be seen on the UI, as shown in the following figures.

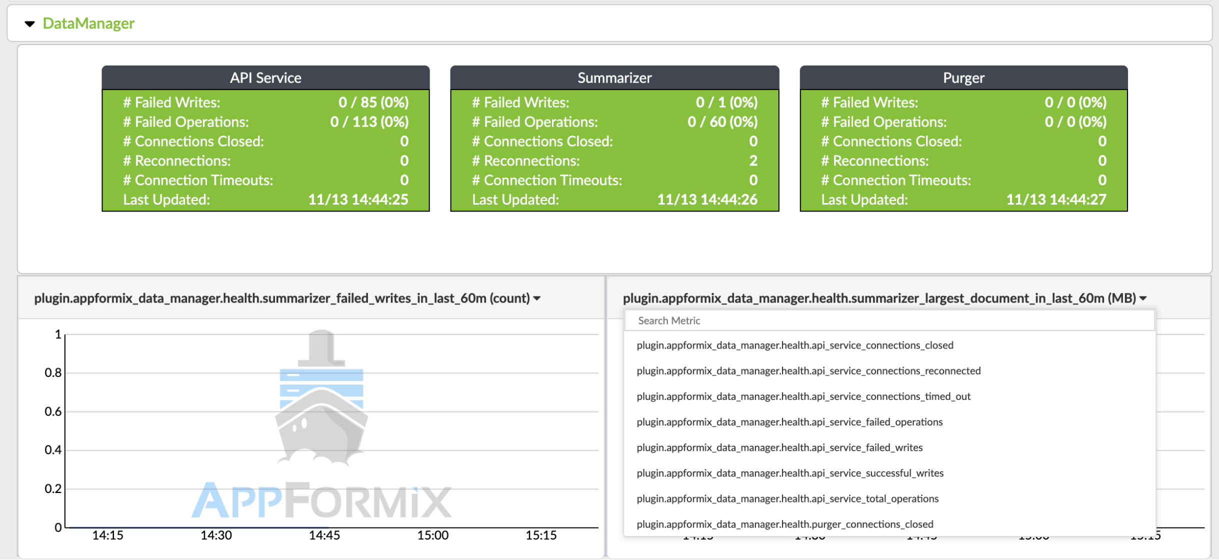
## Figure 57: MongoDB Metrics


MongoDB
plugin.appformix_mongo.usage.repl_network_bytes (Bytes)
10M
5M
0
14:30
14:45
15:00
15:1
plugin.appformix_mongo.usage.repl_network_ops (Count)
plugin.appformix_mongo.usage.document_inserted
plugin.appformix_mongo.usage.document_updated
plugin.appformix_mongo.usage.mem_resident
plugin.appformix_mongo.usage.mem_virtual
plugin.appformix_mongo.usage.network_bytesIn
plugin.appformix_mongo.usage.network_bytesOut
plugin.appformix_mongo.usage.network_numRequests
plugin.appformix_mongo.usage.page_faults
plugin.appformix_mongo.usage.repl_ismaster
15:00
15:1

## Figure 58: Redis Metrics


Redis
plugin.appformix_redis.usage.total_connections_received (Count)
100
50
0
14:30
14:45
15:00
plugin.appformix_redis.usage.total_net_input_Bytes (Bytes)
1
0.8
0.6
0.4
0.2
0
14:30
14:45
15:00

## Figure 59: Data Manager Metrics



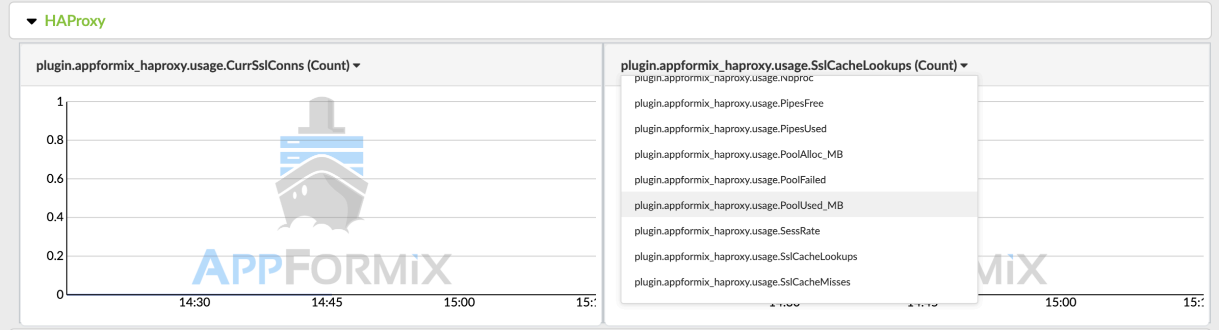
| API Service | |
|---|---|
| # Failed Writes: | 0 / 85 (0%) |
| # Failed Operations: | 0 / 113 (0%) |
| # Connections Closed: | 0 |
| # Reconnections: | 0 |
| # Connection Timeouts: | 0 |
| Last Updated: | 11/13 14:44:25 |

| Summarizer | |
|---|---|
| # Failed Writes: | 0 / 1 (0%) |
| # Failed Operations: | 0 / 60 (0%) |
| # Connections Closed: | 0 |
| # Reconnections: | 2 |
| # Connection Timeouts: | 0 |
| Last Updated: | 11/13 14:44:26 |

| Purger | |
|---|---|
| # Failed Writes: | 0 / 0 (0%) |
| # Failed Operations: | 0 / 0 (0%) |
| # Connections Closed: | 0 |
| # Reconnections: | 0 |
| # Connection Timeouts: | 0 |
| Last Updated: | 11/13 14:44:27 |

plugin.appformix_data_manager.health.summarizer_failed_writes_in_last_60m (count)
1
0.8
0.6
0.4
0.2
0
14:15
14:30
14:45
15:00
15:15
plugin.appformix_data_manager.health.summarizer_largest_document_in_last_60m (MB)
Search Metric
plugin.appformix_data_manager.health.api_service_connections_closed
plugin.appformix_data_manager.health.api_service_connections_reconnected
plugin.appformix_data_manager.health.api_service_connections_timed_out
plugin.appformix_data_manager.health.api_service_failed_operations
plugin.appformix_data_manager.health.api_service_failed_writes
plugin.appformix_data_manager.health.api_service_successful_writes
plugin.appformix_data_manager.health.api_service_total_operations
plugin.appformix_data_manager.health.purger_connections_closed

Figure 60: HAProxy Metrics


HAProxy
plugin.appformix_haproxy.usage.CurrSslConns (Count)
1
0.8
0.6
0.4
0.2
0
14:30
14:45
15:00
15::
APPFORMIX
plugin.appformix_haproxy.usage.SslCacheLookups (Count)
plugin.appformix_haproxy.usage.Nbproc
plugin.appformix_haproxy.usage.PipesFree
plugin.appformix_haproxy.usage.PipesUsed
plugin.appformix_haproxy.usage.PoolAlloc_MB
plugin.appformix_haproxy.usage.PoolFailed
plugin.appformix_haproxy.usage.PoolUsed_MB
plugin.appformix_haproxy.usage.SessRate
plugin.appformix_haproxy.usage.SslCacheLookups
plugin.appformix_haproxy.usage.SslCacheMisses
MIX
15:00
15::

API:

Use the `/data/metrics` API to collect all the data for the platform node for the above plug-ins. To get data for a specific platform node, note its `host_id`. Then enter the following in your browser:

```
http://<appformix-vip>:<appformix-port>/appformix/controller/v2.0/data/metrics?
start=<start_time_in_ms>&end=<end_time_in_ms>&entity_type=host&entity_id=<host_id>
```

The DataManager statistics can also be queried using:

```
http://<appformix-vip>:<appformix-port>/version/2.0/health_status
```

# Health Monitor

Contrail Insights Health Monitor indicates the health and risk for a resource in the infrastructure. Health is an indicator that a resource is currently operating outside of user-specified performance policy. Risk is an indicator that a resource may be unhealthy in the future.

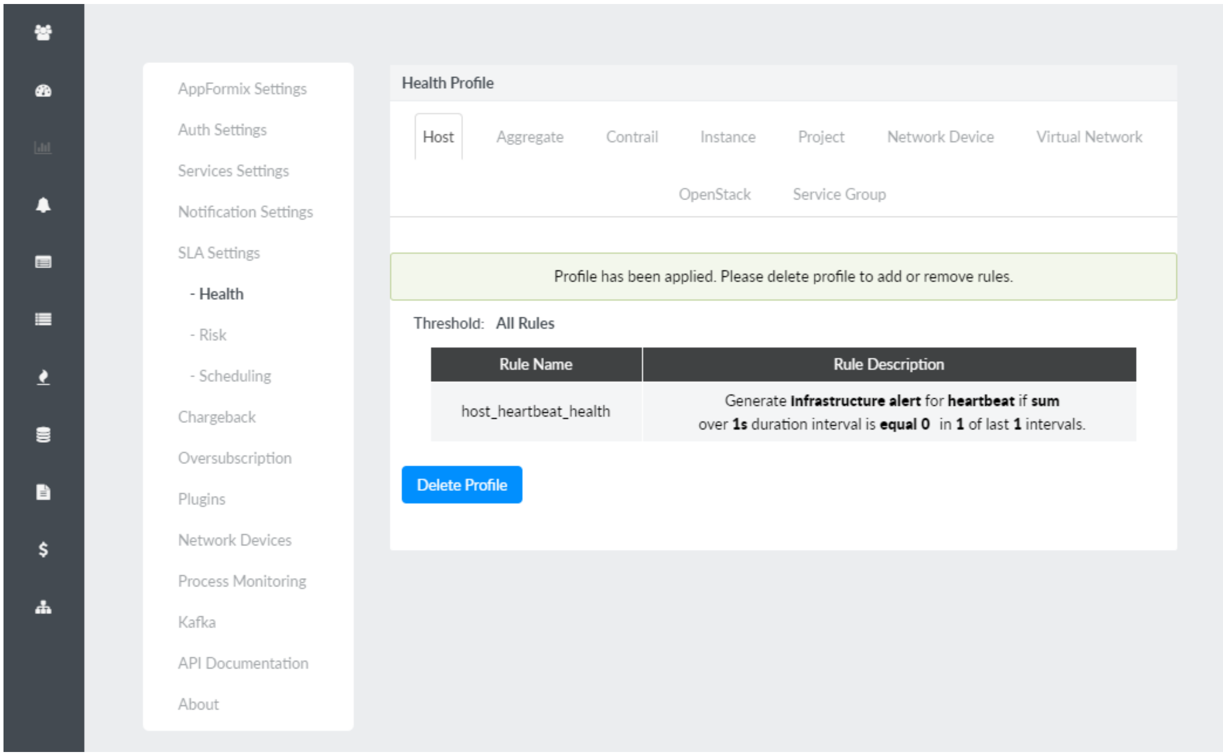
For example, if the Contrail Insights Platform is not receiving heartbeats from a host, then that host and all of its instances are marked as unhealthy. The reason for the unhealthy state is indicated as *missed heartbeat*. The following video provides an overview of the Contrail Insights health analysis.

**Video:** Contrail Insights Analytics

The health and risk are determined by monitoring alarms. Contrail Insights supplies default health and risk profiles. You can modify the health or risk profile to suit your environment.

In the Settings page, select **SLA Settings > Health**. A health profile can be configured separately for hosts and instances. Similarly, a risk profile can be configured separately for hosts and instances. The health and risk profiles can only be configured by an administrator. The profiles apply globally across all users. Figure 61 on page 93 shows the health and risk profile.

**Figure 61: Health Profile**



Health Profile

Profile has been applied. Please delete profile to add or remove rules.

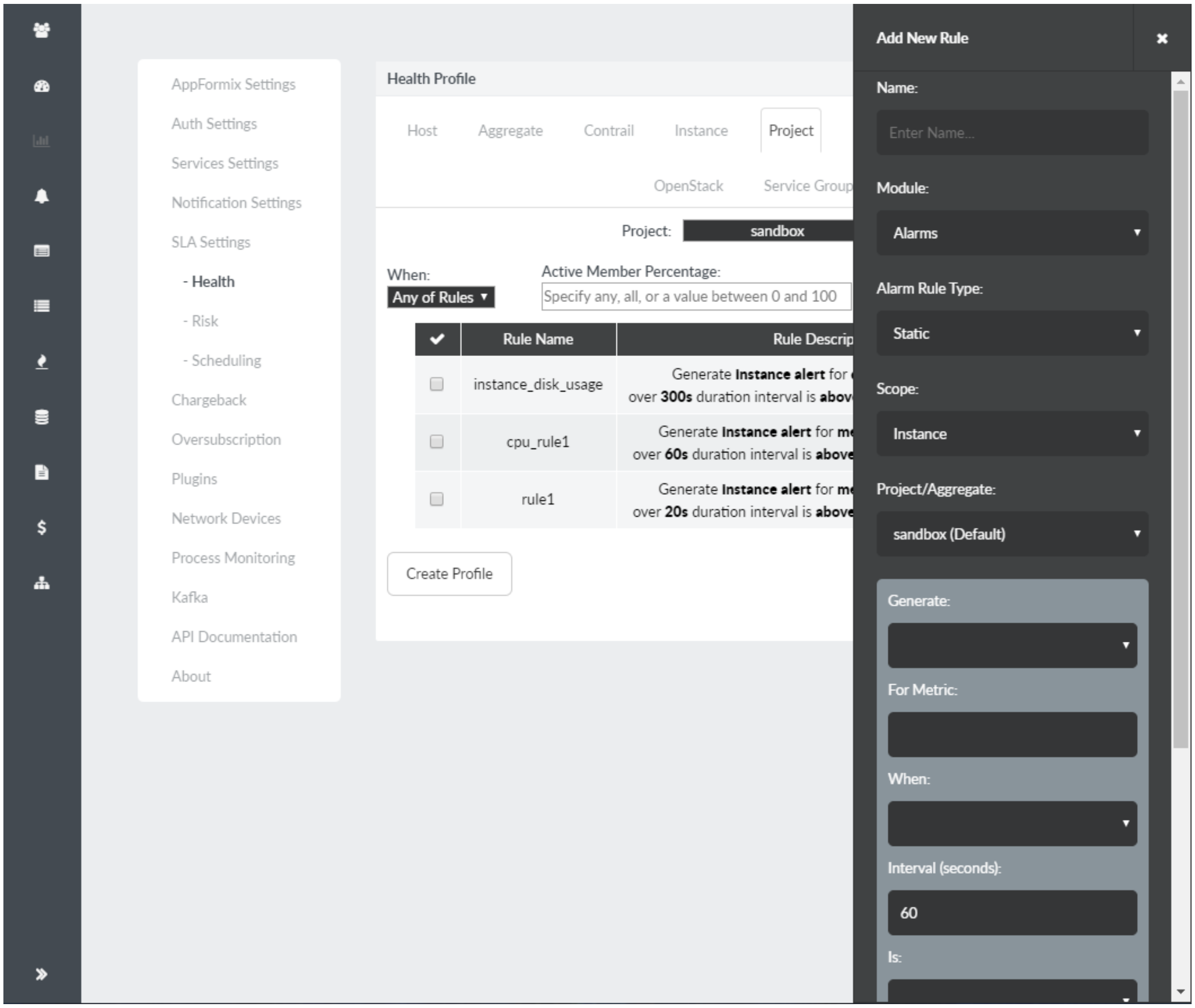
Threshold: All Rules

| Rule Name | Rule Description |
|---|---|
| host_heartbeat_health | Generate **Infrastructure alert** for **heartbeat** if **sum** over **1s** duration interval is **equal 0** in **1** of last **1** intervals. |

To configure a new health or risk profile, first delete the existing profile by clicking **Delete Profile**, as shown in Figure 61 on page 93. After a profile is deleted, select the add button to specify a new set of rules that constitute the profile.

A profile consists of multiple rules that are defined by clicking **Add New Rule**, as shown in Figure 62 on page 94. Each rule specifies conditions that are monitored by Contrail Insights. Select **Any of Rules** or **All of Rules** to specify how multiple rules in a profile are combined. Click **Create Profile** to save the profile. Figure 62 on page 94 shows the Add New rule side pane.

### Figure 62: Health Monitor Profile Configuration Using the Add New Rule Pane



Project: sandbox

When: Any of Rules ▾

Active Member Percentage: Specify any, all, or a value between 0 and 100

| ✔ | Rule Name | Rule Descrip |
|---|---|---|
| □ | instance_disk_usage | Generate **Instance alert** for<br>over **300s** duration interval is **abov** |
| □ | cpu_rule1 | Generate **Instance alert** for **me**<br>over **60s** duration interval is **above** |
| □ | rule1 | Generate **Instance alert** for **me**<br>over **20s** duration interval is **above** |

Name: Enter Name...

Module: Alarms ▾

Alarm Rule Type: Static ▾

Scope: Instance ▾

Project/Aggregate: sandbox (Default) ▾

Generate: ▾

For Metric:

When: ▾

Interval (seconds): 60

Is:

## RELATED DOCUMENTATION

*Alarms*

*Capacity Planning*

*Chargeback*

*Charts*

*Heat Map*

*Metrics Collected by Contrail Insights*

*Notifications*

*Extensibility Using Plug-Ins*

*Reports*

*Service Monitoring from the UI*

# Heat Map

IN THIS SECTION

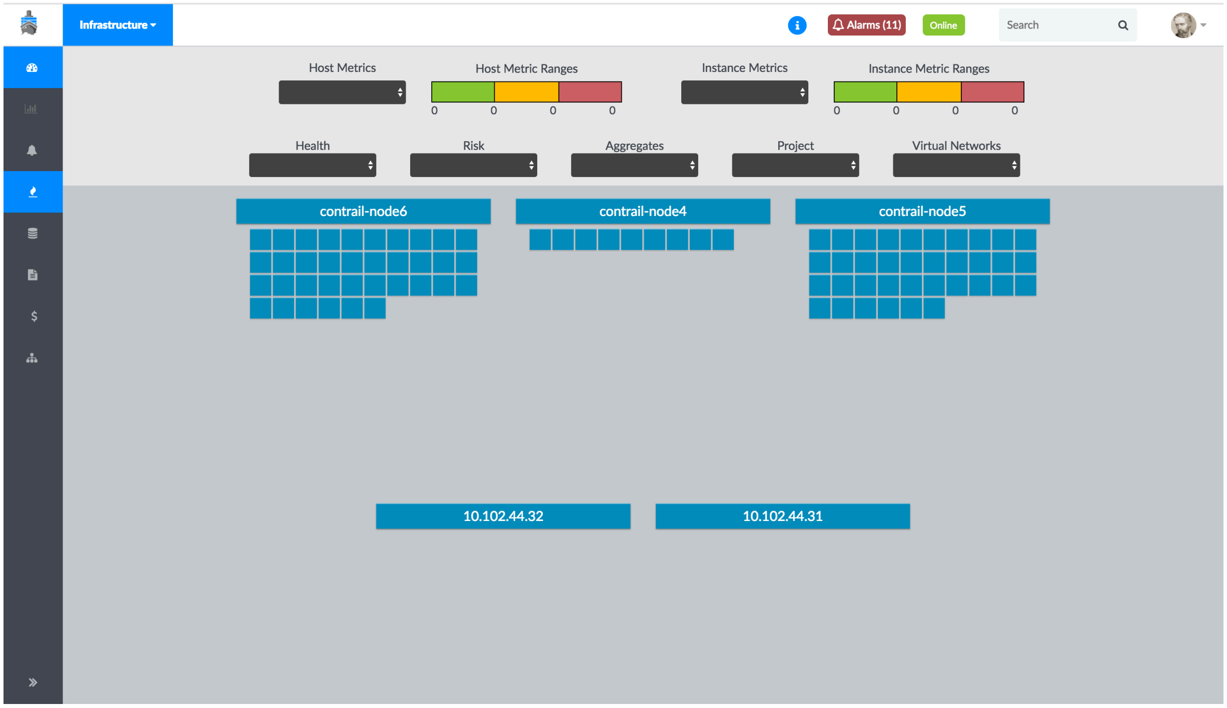
Contrail Insights provides a real-time Heat Map of resources. Heat Map is a visual depiction of the relationship between hosts and instances that allows you to understand infrastructure performance at a glance.

Heat Map can also be thought of as a tool to understand usage patterns of physical infrastructure components that provide metric correlation for an ever-changing virtual infrastructure. In addition, one can consider it as a tool for visualizing the usage patterns of entities of the virtual infrastructure itself.

## Using the Heat Map

Use the top context menu to select the scope of entities to display. In the following example, **Infrastructure** is selected, which displays all hosts. In Figure 63 on page 96 there are three hosts, each represented by a rectangle. Under each host rectangle is a square for each virtual machine executing on the host. Figure 63 on page 96 shows a heat map of infrastructure components displaying usage patterns.

Figure 63: Heat Map of Infrastructure Components Displaying Usage Patterns



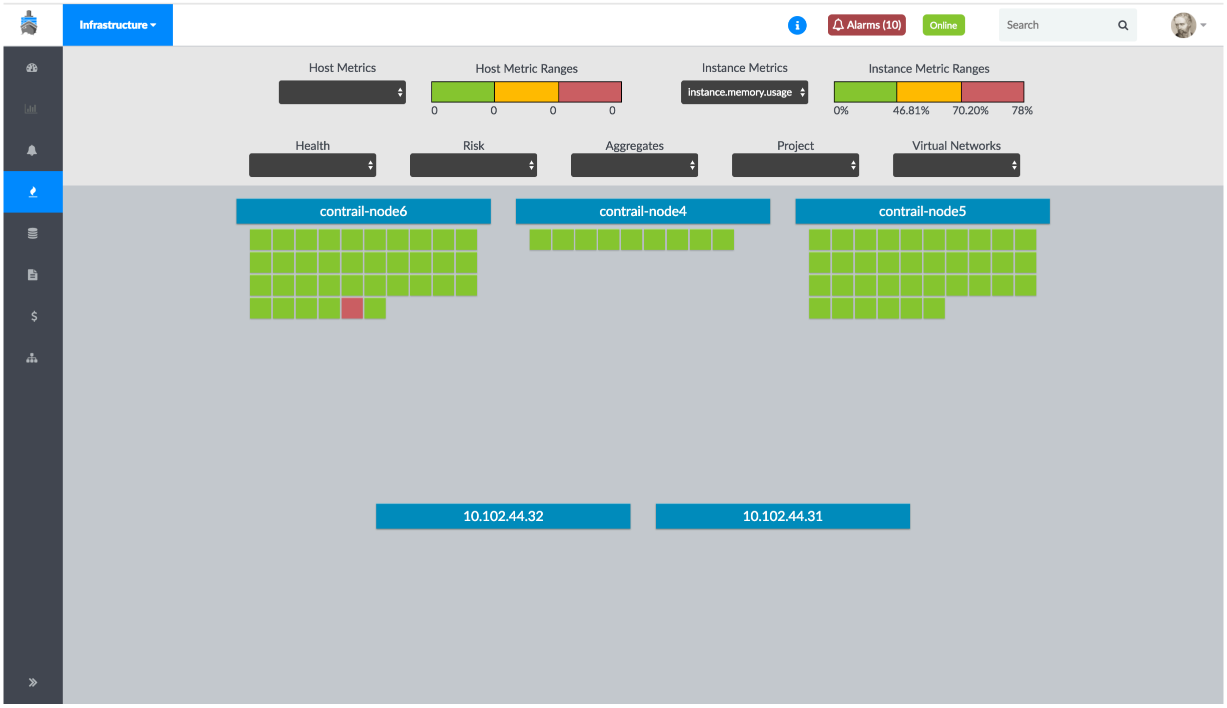
## Temperature Scale

The *temperature* of an entity is displayed for a metric selected from the drop-down lists for host and instance. The temperature scale is automatically determined by Contrail Insights using machine learning that evaluates values of the metric across the infrastructure.

### Example: Using the Heat Map

In Figure 64 on page 97, the **instance.memory.usage** metric is selected.

**Figure 64: Heat Map Showing Instance Memory Usage Metrics**



Instance Metrics: instance.memory.usage

Each instance is colored according to its memory usage and the temperature scale determined by Contrail Insights. The temperature scale is displayed in metric range at the top. In Figure 64 on page 97, instances are colored according to the following scale:

**Green** Using between 0-46.8% of memory capacity.

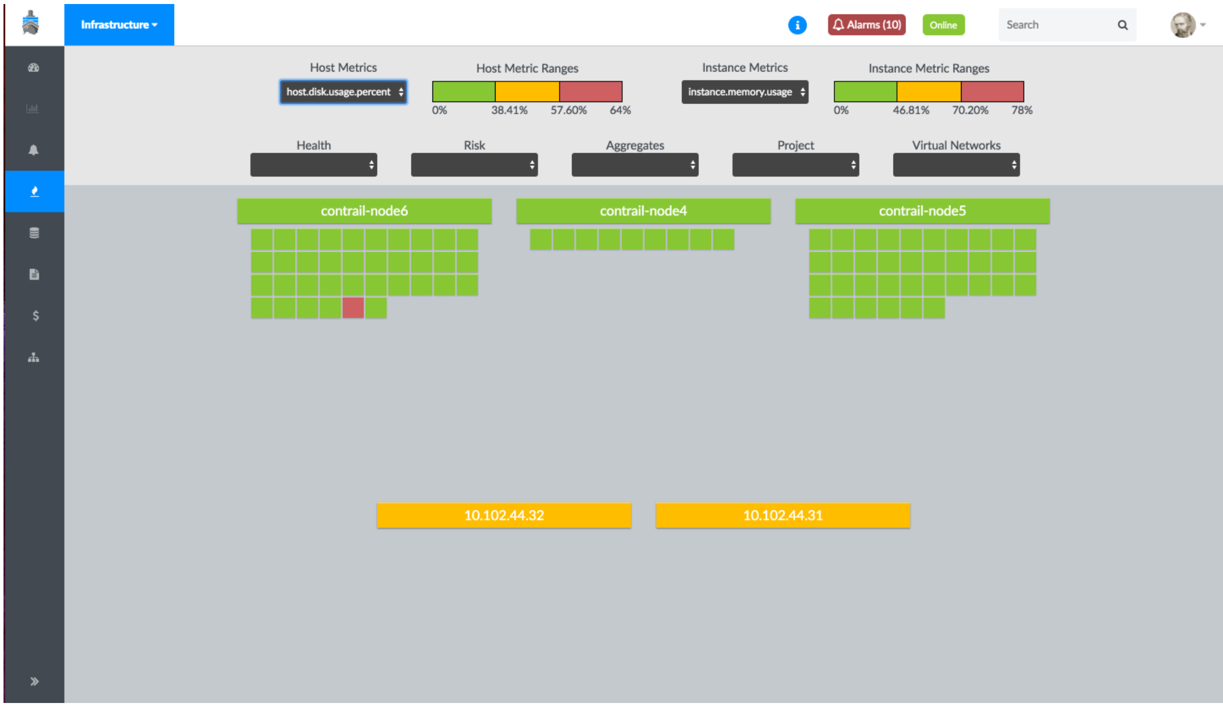
**Yellow** Using between 46.81-70.19% of memory capacity.

**Red** Using between 70.20-78% of memory capacity.

The range ends at 78% in this example because that is the maximum value from the last hour across all instances. The scale changes according to the recent resource consumption learned by Contrail Insights.

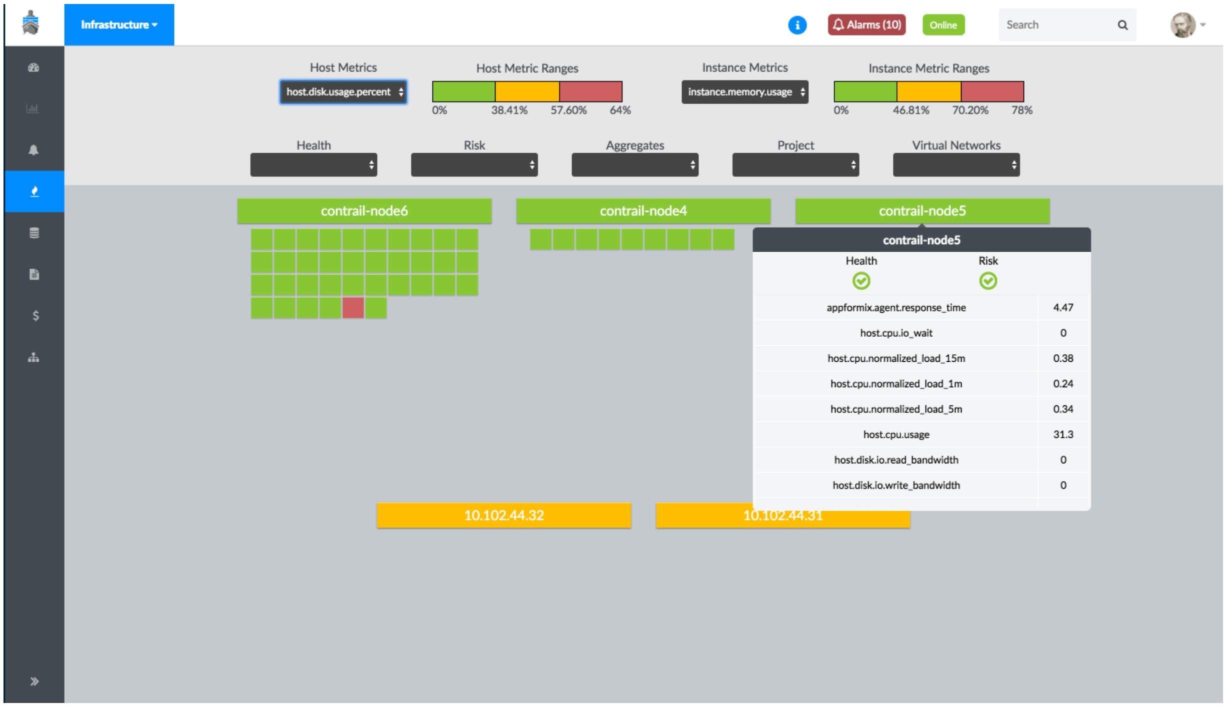
Simultaneously, a host metric can be selected to show the temperature of hosts, as shown in Figure 65 on page 98.

Figure 65: Selected Host Metric Showing Temperature of Hosts



To display the exact values of metrics for an entity, place the cursor over the entity. A pop-up box displays a metric table. Scroll to the metric to view its last reported value. In Figure 66 on page 99, the mouse cursor is hovering over a host to display its metric table.

Figure 66: Heat Map Host Metric Details Table



Host Metrics: host.disk.usage.percent

Instance Metrics: instance.memory.usage

contrail-node5

Health ✓ Risk ✓

| | |
|---|---|
| appformix.agent.response_time | 4.47 |
| host.cpu.io_wait | 0 |
| host.cpu.normalized_load_15m | 0.38 |
| host.cpu.normalized_load_1m | 0.24 |
| host.cpu.normalized_load_5m | 0.34 |
| host.cpu.usage | 31.3 |
| host.disk.io.read_bandwidth | 0 |
| host.disk.io.write_bandwidth | 0 |

## Filtering Entities

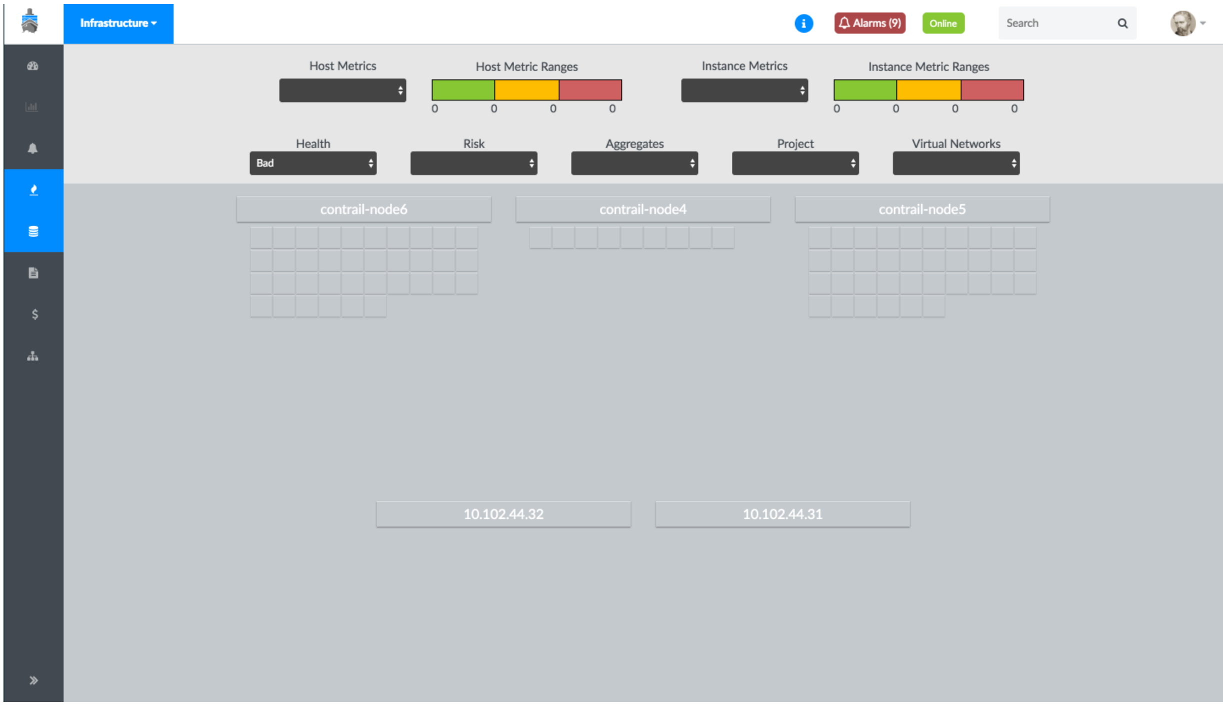
Entities in the view can have multiple filters applied to them. These filters help you:

- View the temperature of a subset of entities in a logical group.
- Visualize how a subset of instances are distributed across hosts.

To filter resources, select a value for any given filter in the row of filters. Entities not selected by the filter will be depicted in gray. Entities that are selected by the filter will be colored according to the temperature scale.

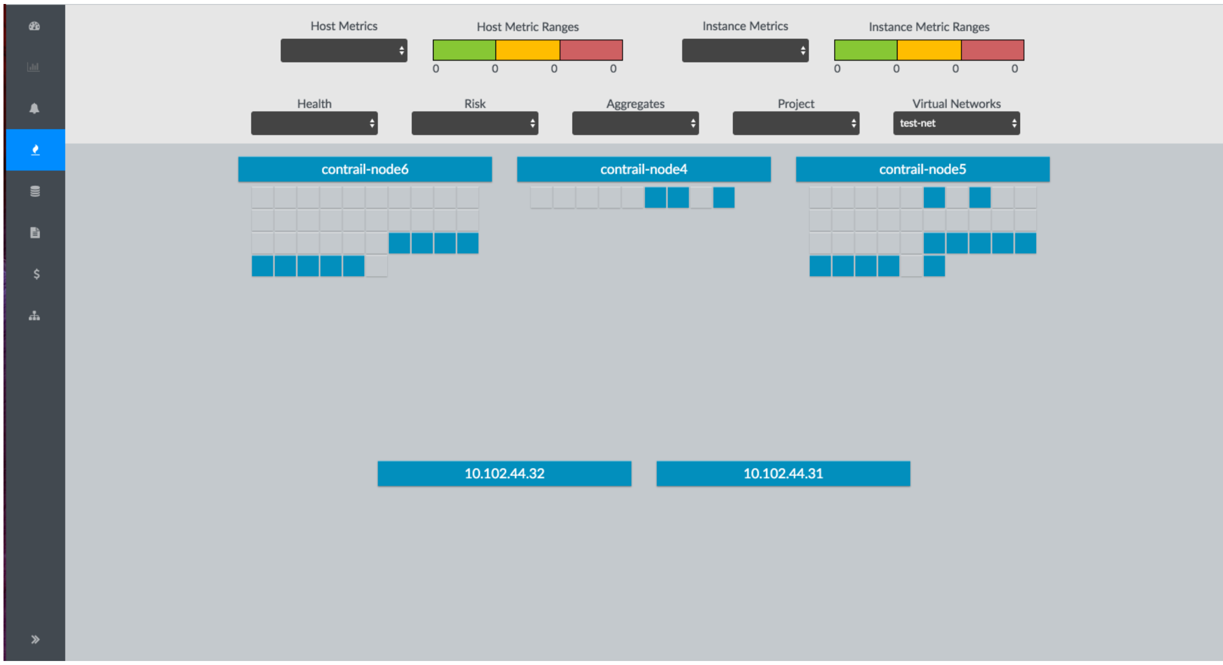
In Figure 67 on page 100, **Bad** is selected from the Health filter to display any entity that has bad health, according to user-defined health profiles.

Figure 67: Heat Map Health Filter to Identify Any Entity with Bad Health



The Heat Map can be further filtered to see all of the resources that belong to a specific virtual network. This can be done by selecting a virtual network from the Virtual Network filter. In Figure 68 on page 101, instances attached to **test-net** are colored blue because an Instance Metric has not been selected.

Figure 68: Heat Map Virtual Network Filter



Host Metrics

Host Metric Ranges

Instance Metrics

Instance Metric Ranges

Health

Risk

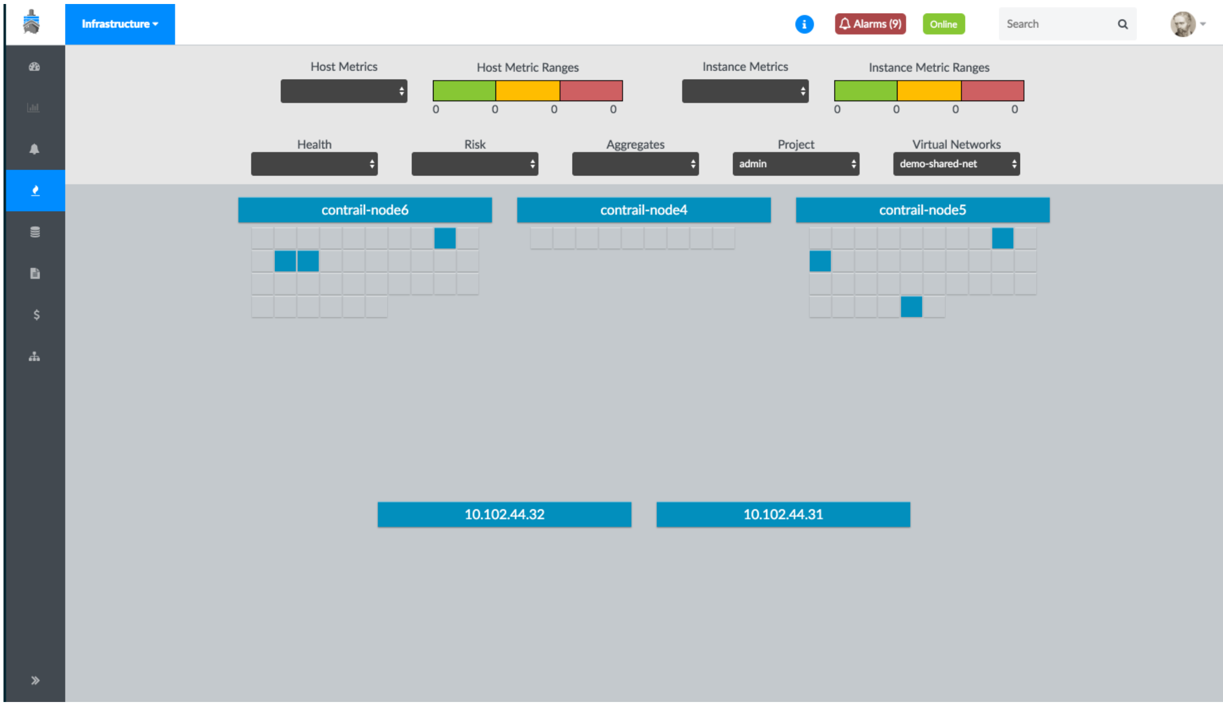
Aggregates

Project

Virtual Networks: test-net

Multiple filters can be applied at the same time. In Figure 69 on page 101, instances that belong to both **admin** project and **demo-shared-net** virtual network are colored blue. All other instances are gray.

Figure 69: Heat Map Using Multiple Filters



Host Metrics

Host Metric Ranges

Instance Metrics

Instance Metric Ranges

Health

Risk

Aggregates

Project: admin

Virtual Networks: demo-shared-net

RELATED DOCUMENTATION

*Alarms*

*Capacity Planning*

*Chargeback*

*Charts*

*Health Monitor*

*Metrics Collected by Contrail Insights*

*Notifications*

*Extensibility Using Plug-Ins*

*Reports*

*Service Monitoring from the UI*

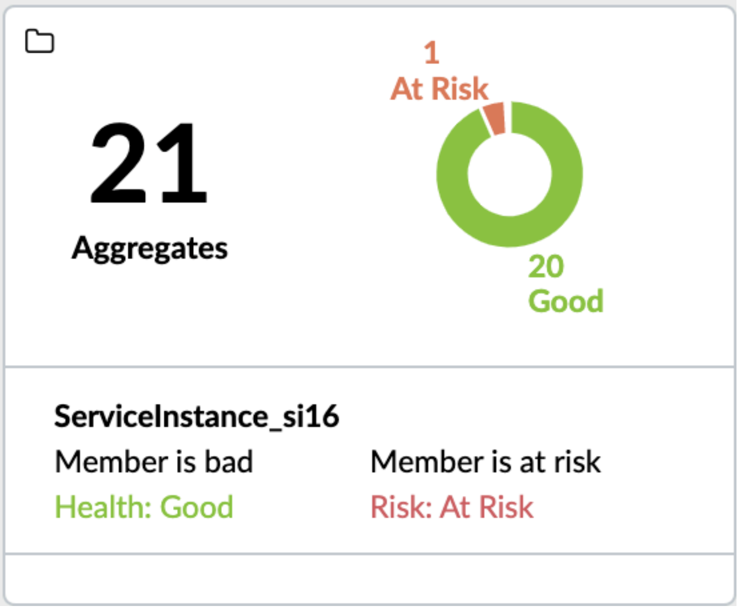
# Monitor Service Instances

IN THIS SECTION

Starting in Contrail Insights Release 3.3.7, you can monitor service instances that are created by using the Contrail Command user interface (UI), from the Contrail Insights UI. A service instance is used to launch a VNF or PNF device as part of a service chain. For more information, see Example: Creating a Transparent Service Chain by Using Contrail Command.

Contrail Insights uses Contrail Server Sent Events (SSE) to fetch service instances from the Contrail API server that are created by using Contrail Command. Contrail Insights discovers service instances that have port tuples configured and records these service instances only if the virtual machines are present in the Contrail Insights database. After the service instances are recorded, the service instance is represented as an instance aggregate on the Contrail Insights Dashboard with the prefix `ServiceInstance_`.

**Figure 70: Service Instance and Aggregates displayed on the Dashboard**



ServiceInstance_si16

Member is bad

Health: Good

Member is at risk

Risk: At Risk

## Viewing Service Instances on Contrail Insights UI

Follow these steps to view and to monitor service instances from the Contrail Insights user interface (UI).

1. From the **Dashboard** view, select **Aggregates** from the **Context** menu.

   The **Aggregate** list appears.

2. Select **ServiceInstance_*< service instance name>*** from the **Aggregate** list.

   The service instance and the associated virtual machines are displayed.

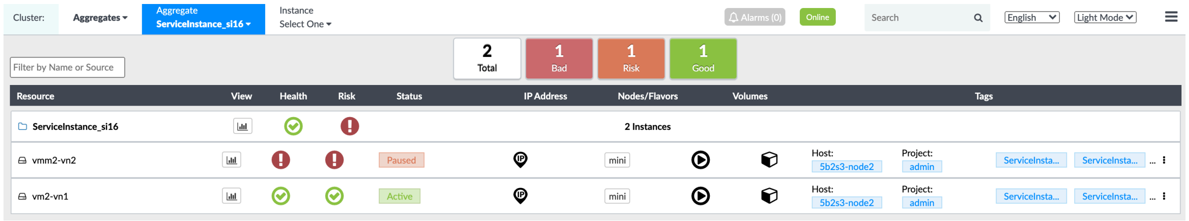
## Monitoring Service Instance Status (Analytics Profile)

After you have navigated to the Aggregate page, you can view and monitor health, risk, and status of all virtual machines of a service instance.

A health and risk profile is added against every instance aggregate when the service instance is created. If any of the virtual machines in the aggregate is marked **NOT ACTIVE**, then the profile is marked ’at risk’. ’At risk’ represents a **PARTIALLY_ACTIVE** state. If all the virtual machines in the aggregate are bad, then the profile health is marked 'BAD’. ’BAD’ represents an **INACTIVE** state.

1. `PARTIALLY_ACTIVE`**State—vmm2-vn2 in Paused State** image shows the **ServiceInstance_si16** instance aggregate is at risk with virtual machine **vmm2-vn2** in the paused state.

### Figure 71: vmm2-vn2 in Paused State



Cluster: Aggregates ▾ | Aggregate ServiceInstance_si16 ▾ | Instance Select One ▾ | Alarms (0) | Online | Search | English | Light Mode

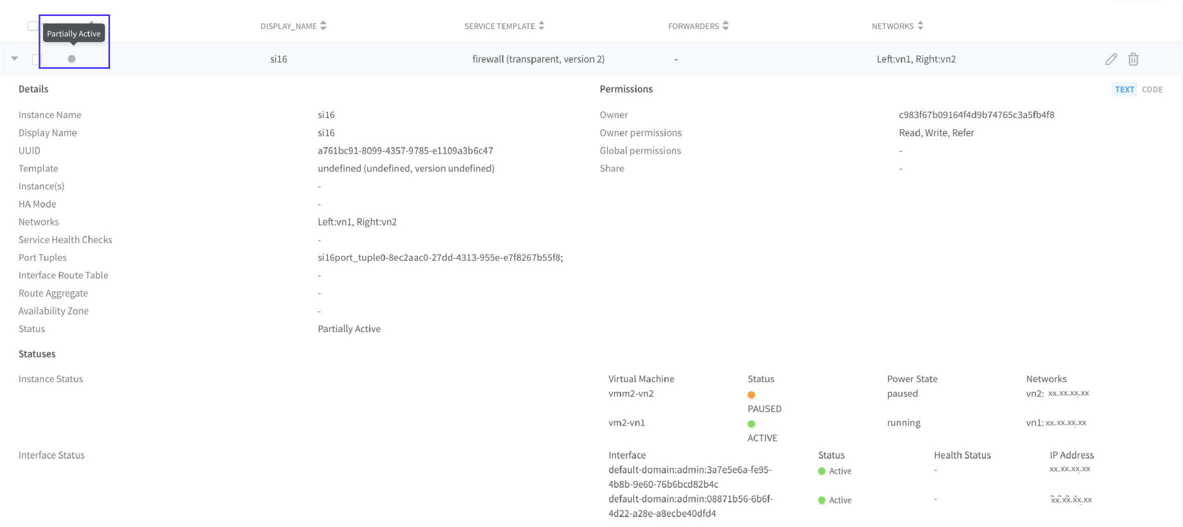
Filter by Name or Source

2 Total | 1 Bad | 1 Risk | 1 Good

| Resource | View | Health | Risk | Status | IP Address | Nodes/Flavors | Volumes | | | Tags |
|---|---|---|---|---|---|---|---|---|---|---|
| ServiceInstance_si16 | | ✓ | ! | | | 2 Instances | | | | |
| vmm2-vn2 | | ! | ! | Paused | | mini | | Host: 5b2s3-node2 | Project: admin | ServiceInsta... ServiceInsta... ... |
| vm2-vn1 | | ✓ | ✓ | Active | | mini | | Host: 5b2s3-node2 | Project: admin | ServiceInsta... ServiceInsta... ... |

**Service Instance in the Partially Active State** image shows an example of the **si16** service instance in **Partially Active** state in the Contrail Command UI.

### Figure 72: Service Instance in the Partially Active State



| Partially Active | DISPLAY_NAME | SERVICE TEMPLATE | FORWARDERS | NETWORKS |
|---|---|---|---|---|
| ● | si16 | firewall (transparent, version 2) | - | Left:vn1, Right:vn2 |

TEXT CODE

**Details**

| | |
|---|---|
| Instance Name | si16 |
| Display Name | si16 |
| UUID | a761bc91-8099-4357-9785-e1109a3b6c47 |
| Template | undefined (undefined, version undefined) |
| Instance(s) | - |
| HA Mode | - |
| Networks | Left:vn1, Right:vn2 |
| Service Health Checks | - |
| Port Tuples | si16port_tuple0-8ec2aac0-27dd-4313-955e-e7f8267b55f8; |
| Interface Route Table | - |
| Route Aggregate | - |
| Availability Zone | - |
| Status | Partially Active |

**Permissions**

| | |
|---|---|
| Owner | c983f67b09164f4d9b74765c3a5fb4f8 |
| Owner permissions | Read, Write, Refer |
| Global permissions | - |
| Share | - |

**Statuses**

Instance Status

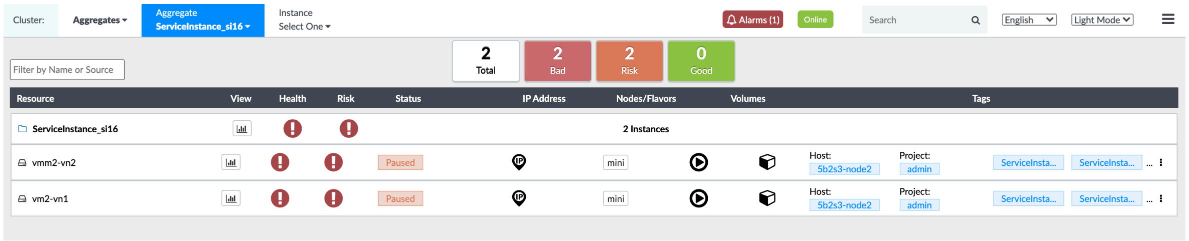
| Virtual Machine | Status | Power State | Networks |
|---|---|---|---|
| vmm2-vn2 | ● PAUSED | paused | vn2: xx.xx.xx.xx |
| vm2-vn1 | ● ACTIVE | running | vn1: xx.xx.xx.xx |

Interface Status

| Interface | Status | Health Status | IP Address |
|---|---|---|---|
| default-domain:admin:3a7e5e6a-fe95-4b8b-9e60-76b6bcd82b4c | ● Active | - | xx.xx.xx.xx |
| default-domain:admin:08871b56-6b6f-4d22-a28e-a8ecbe40dfd4 | ● Active | - | xx.xx.xx.xx |

2. `INACTIVE`**State—vm2-vn1 in Paused State** image shows the **ServiceInstance_si16** instance aggregate is at risk and in bad health with virtual machine **vm2-vn1** in the paused state.

Figure 73: vm2-vn1 in Paused State



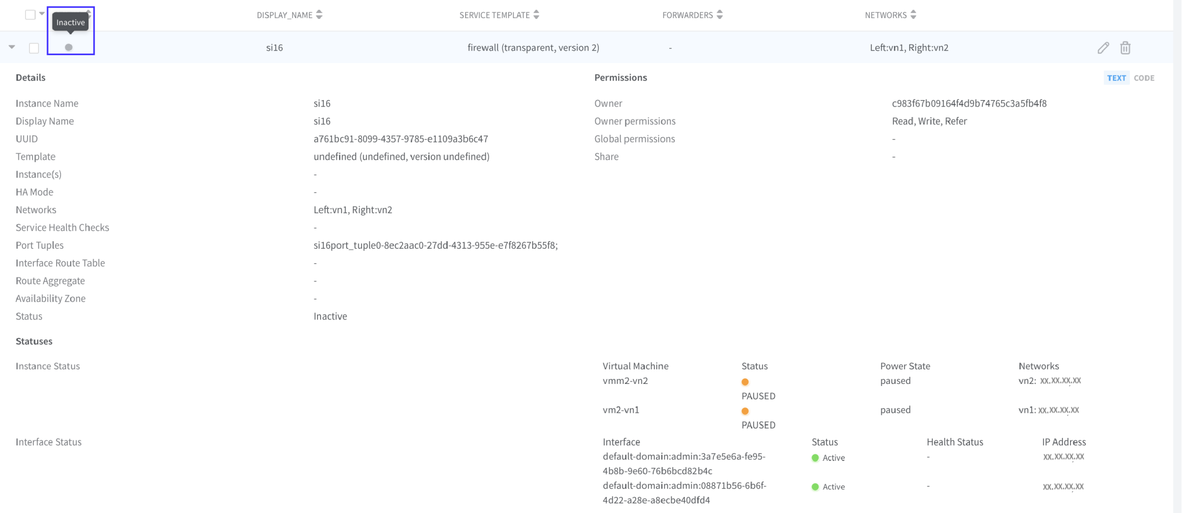
| Resource | View | Health | Risk | Status | IP Address | Nodes/Flavors | Volumes | | | | Tags |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ServiceInstance_si16 | | | | | | 2 Instances | | | | | |
| vmm2-vn2 | | | | Paused | | mini | | | Host: 5b2s3-node2 | Project: admin | ServiceInsta... ServiceInsta... ... |
| vm2-vn1 | | | | Paused | | mini | | | Host: 5b2s3-node2 | Project: admin | ServiceInsta... ServiceInsta... ... |

**Service Instance in the Inactive State** image shows an example of the si16 service instance in the Inactive state in the Contrail

Command

UI.

Figure 74: Service Instance in the Inactive State



| | DISPLAY_NAME | SERVICE TEMPLATE | FORWARDERS | NETWORKS |
|---|---|---|---|---|
| | si16 | firewall (transparent, version 2) | - | Left:vn1, Right:vn2 |

**Details**

| | |
|---|---|
| Instance Name | si16 |
| Display Name | si16 |
| UUID | a761bc91-8099-4357-9785-e1109a3b6c47 |
| Template | undefined (undefined, version undefined) |
| Instance(s) | - |
| HA Mode | - |
| Networks | Left:vn1, Right:vn2 |
| Service Health Checks | - |
| Port Tuples | si16port_tuple0-8ec2aac0-27dd-4313-955e-e7f8267b55f8; |
| Interface Route Table | - |
| Route Aggregate | - |
| Availability Zone | - |
| Status | Inactive |

**Permissions**

| | |
|---|---|
| Owner | c983f67b09164f4d9b74765c3a5fb4f8 |
| Owner permissions | Read, Write, Refer |
| Global permissions | - |
| Share | - |

**Statuses**

Instance Status

| Virtual Machine | Status | Power State | Networks |
|---|---|---|---|
| vmm2-vn2 | ● PAUSED | paused | vn2: xx.xx.xx.xx |
| vm2-vn1 | ● PAUSED | paused | vn1: xx.xx.xx.xx |

Interface Status

| Interface | Status | Health Status | IP Address |
|---|---|---|---|
| default-domain:admin:3a7e5e6a-fe95-4b8b-9e60-76b6bcd82b4c | ● Active | - | xx.xx.xx.xx |
| default-domain:admin:08871b56-6b6f-4d22-a28e-a8ecbe40dfd4 | ● Active | - | xx.xx.xx.xx |

## Alarm and Notification

You can setup a notification service for Contrail Insights to forward triggered alarms from the **Settings**>**Notification Settings** page. You can create an alarm by selecting **Instance** as **Scope**, and selecting **ServiceInstance_<** ***service instance name*****>** as the **Instance Aggregate**.

The triggered alarm notification will then be sent to the configured notification service with information on metric_name, value, and details on the service instance.

# Metrics Collected by Contrail Insights

IN THIS SECTION

A *metric* is a measured value for an element in the infrastructure. Contrail Insights Agent collects and calculates metrics for hosts and instances. Contrail Insights metrics are organized into hierarchical categories based on the type of metric.

Some metrics are s of total capacity. In such cases, the category of the metric determines the total capacity by which the is computed. For instance, host.cpu.usage indicates the percentage of CPU consumed relative to the total CPU available on a host. In contrast, instance.cpu.usage is the percentage of CPU consumed relative to the total CPU available to an instance. As an example, consider an instance that is using 50% of one core on a host with 20 cores. The instance's host.cpu.usage will be 2.5%. If the instance has been allocated two cores, then its instance.cpu.usage will be 25%.

*Alarms* can be configured for any metric. Many metrics can also be displayed in charts. When an alarm triggers for a metric, the alarm is plotted on charts at the time of the event. In this way, metrics that cannot be plotted directly as a chart are still visually correlated in time with other metrics.

Contrail Insights Agent collects both raw metrics and calculated metrics. Raw metrics are values read directly from the underlying infrastructure. Calculated metrics are metrics that Contrail Insights Agent derives from raw metrics.

## Host CPU Data Metrics

Table 4 on page 107 lists the calculated metrics available for the host CPU data.

**Table 4: Host CPU Data Metrics**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| host.cpu.usage | % | x | x |
| host.cpu.io_wait | % | x | x |
| host.cpu.per_core.usage | % | — | x |
| host.cpu.per_core.user.usage | % | — | x |
| host.cpu.temperature | degree | — | x |
| host.cpu.normalized_load_1m | loadavg | x | x |
| host.cpu.normalized_load_5m | loadavg | x | x |
| host.cpu.normalized_load_15m | loadavg | x | x |
| host.cpu.cores.state_transition | 0 or 1 | — | x |
| host.disk.smart.predict_failure | 0 or 1 | — | x |
| host.heartbeat | 0 or 1 | — | x |

**host.cpu.normalized_load**

Normalized load is calculated as a ratio of the number of running and ready-to-run threads to the number of CPU cores. This family of metrics indicate the level of demand for CPU. If the value exceeds 1, then more threads are ready to run than exists CPU cores to perform the execution. Normalized load is a provided as an average over 1-minute, 5-minute, and 15-minute intervals.

**host.cpu.temperature**

CPU temperature is derived from multiple temperature sensors in the processor(s) and chassis. This temperature provides a general indicator of temperature in degrees Celsius inside a physical host.

**host.disk.smart.predict_failure**

Contrail Insights Agent calculates *predict_failure* using multiple S.M.A.R.T. counters provided by disk hardware. The agent will set *predict_failure* to true (value=1) when it determines from a combination of S.M.A.R.T. counters that a disk is likely to fail. An alarm triggered for this metric contains the disk identifier in the metadata.

**host.heartbeat**

The *host.heartbeat* indicates if Contrail Insights Agent is functioning on a host. Contrail Insights Platform periodically checks the status of each host by making a status request to Contrail Insights Agent. The *host.heartbeat* metric is incremented for each successful response. Alarms can be configured to detect missed heartbeats over a given interval.

## Host Disk Metrics

Table 5 on page 108 lists the raw metrics available for host disk.

**Table 5: Host Disk Metrics**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| host.disk.io.read | MBps | x | x |
| host.disk.io.write | MBps | x | x |
| host.disk.response_time | ms | x | x |
| host.disk.read_response_time | ms | x | x |
| host.disk.write_response_time | ms | x | x |
| host.disk.smart.hdd.command_timeout | count | — | x |

**Table 5: Host Disk Metrics *(Continued)***

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| host.disk.smart.hdd.current_pending_sector_count | count | — | x |
| host.disk.smart.hdd.offline_uncorrectable | count | — | x |
| host.disk.smart.hdd.reallocated_sector_count | count | — | x |
| host.disk.smart.hdd.reported_uncorrectable_errors | count | — | x |
| host.disk.smart.ssd.available_reserved_space | count | — | x |
| host.disk.smart.ssd.media_wearout_indicator | count | — | x |
| host.disk.smart.ssd.reallocated_sector_count | count | — | x |
| host.disk.smart.ssd.wear_leveling_count | count | — | x |
| host.disk.usage.bytes | GB | x | x |
| host.disk.usage.percent | % | x | x |

## Host Memory Usage

Table 6 on page 109 lists the raw metrics available for host memory usage.

**Table 6: Metrics for Host Memory Usage**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| host.memory.usage | % | x | x |

Table 6: Metrics for Host Memory Usage *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| host.memory.dirty.rate | dirty pages/s | x | x |
| host.memory.page_in_out.rate | dirty pages/s | x | x |
| host.memory.page_fault.rate | dirty pages/s | x | x |
| host.memory.swap.usage | dirty pages/s | x | x |

## Host Mount Metrics

Table 7 on page 110 lists the raw metrics available for host mount.

Table 7: Host Mount Metrics

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| host.mount.usage | % | x | x |
| host.mount.io.read | MBps | x | x |
| host.mount.io.write | MBps | x | x |
| host.mount.detect_change | 1 or 0 | — | x |
| host.mount.usage.bytes | GB | x | — |

## Host Network Data

Table 8 on page 111 lists the raw metrics available for host network data.

**Table 8: Host Network Data Metrics**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| host.network.ingress.bit_rate | Mbps | x | x |
| host.network.egress.bit_rate | Mbps | x | x |
| host.network.ingress.packet_rate | packets/s | x | x |
| host.network.egress.packet_rate | packets/s | x | x |
| host.network.ingress.errors | errors/s | x | x |
| host.network.egress.errors | errors/s | x | x |
| host.network.ingress.drops | drops/s | x | x |
| host.network.egress.drops | drops/s | x | x |
| host.network.ipv4tables.rule_count | count | x | x |
| host.network.ipv6tables.rule_count | count | x | x |
| openstack.host.disk_gb.allocated.count | count | x | x |
| openstack.host.disk_gb.allocated.percentage | percentage | — | x |
| openstack.host.memory_mb.allocated.count | count | x | x |
| openstack.host.memory_mb.allocated.percentage | percentage | — | x |

Table 8: Host Network Data Metrics *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| openstack.host.vcpus_allocated.count | count | x | x |
| openstack.host.vcpus_allocated.percentage | percentage | — | x |

## Instances

Table 9 on page 112 lists the raw metrics available for instances.

Table 9: Raw Metrics for Instances

| Metric | Chart | Alarm |
|---|---|---|
| instance.cpu.usage | x | x |
| instance.disk.io.read_bandwidth | x | x |
| instance.disk.io.read_iops | x | x |
| instance.disk.io.read_iosize | x | x |
| instance.disk.io.read_response_time | x | x |
| instance.disk.io.write_bandwidth | x | x |
| instance.disk.io.write_iops | x | x |
| instance.disk.io.write_iosize | x | x |
| instance.disk.io.write_response_time | x | x |

Table 9: Raw Metrics for Instances *(Continued)*

| Metric | Chart | Alarm |
|---|---|---|
| instance.disk.usage.bytes | x | x |
| instance.disk.usage.percentage | x | x |
| instance.memory.usage | x | x |
| instance.network.egress.bit_rate | x | x |
| instance.network.egress.drops | x | x |
| instance.network.egress.errors | x | x |
| instance.network.egress.packet_rate | x | x |
| instance.network.ingress.bit_rate | x | x |
| instance.network.ingress.drops | x | x |
| instance.network.ingress.errors | x | x |
| instance.network.ingress.packet_rate | x | x |

Table 10 on page 113 lists the calculated metric available for instances.

Table 10: Calculated Metrics for Instances

| Metric | Chart | Alarm |
|---|---|---|
| instance.heartbeat | — | x |

**instance.heartbeat** The *instance.heartbeat* indicates whether an instance is running. Contrail Insights Agent periodically checks the state of host processes associated with each instance. The **instance.heartbeat** metric is incremented for each successful status check. Alarms may be configured to detect missed heartbeats over a given interval.

## Network Device

Contrail Insights can collect network device metrics using SNMP or Juniper Telemetry Interface (JTI). See *Configure Network Devices from the UI* and *Configure Network Device from JSON File* for configuration and monitoring information.

Table 11 on page 114 lists some of the metrics available per interface with SNMP network device monitoring. For the complete list, refer to the files present in the `certified_plugins/` directory in the Contrail Insights installation TAR file. See *Contrail Insights SNMP Monitoring*.

**Table 11: Metrics Available per Interface with SNMP Network Device Monitoring**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| snmp.interface.out_discards | discards/s | x | x |
| snmp.interface.in_discards | discards/s | x | x |
| snmp.interface.in_errors | errors/s | x | x |
| snmp.interface.out_unicast_packets | packets/s | x | x |
| snmp.interface.in_octets | octets/s | x | x |
| snmp.interface.in_unicast_packets | packets/s | x | x |
| snmp.interface.out_packet_queue_length | count | x | x |
| snmp.interface.speed | bits/s | x | x |
| snmp.interface.out_octets | octets/s | x | x |

**Table 11: Metrics Available per Interface with SNMP Network Device Monitoring *(Continued)***

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| snmp.interface.in_unknown_protocol | packets/s | x | x |
| snmp.interface.in_non_unicast_packets | packets/s | x | x |
| snmp.interface.out_errors | errors/s | x | x |
| snmp.interface.out_non_unicast_packets | packets/s | x | x |

Table 12 on page 115 lists some of the metrics available per interface with JTI network device monitoring. For the complete list, refer to the file `jti_config_all_sensors.json` in the `certified_plugins/` directory of the Contrail Insights installation TAR file. See also *Contrail Insights JTI (UDP) Monitoring* *Contrail Insights JTI (gRPC) Monitoring* and *Custom Sensors for JTI, gRPC, and NETCONF*.

**Table 12: Metrics Available per Interface with JTI Network Device Monitoring**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| junos.system.linecard.interface.egress_errors.if_errors | errors/s | x | x |
| junos.system.linecard.interface.egress_errors.if_discard | discards/s | x | x |
| junos.system.linecard.interface.egress_stats.if_1sec_pkts | packets/s | x | x |
| junos.system.linecard.interface.egress_stats.if_octets | octets/s | x | x |
| junos.system.linecard.interface.egress_stats.if_mc_pkts | packets/s | x | x |
| junos.system.linecard.interface.egress_stats.if_bc_pkts | packets/s | x | x |
| junos.system.linecard.interface.egress_stats.if_1sec_octets | octets/s | x | x |
| junos.system.linecard.interface.egress_stats.if_pkts | packets/s | x | x |

Table 12: Metrics Available per Interface with JTI Network Device Monitoring *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| junos.system.linecard.interface.egress_stats.if_uc_pkts | packets/s | x | x |
| junos.system.linecard.interface.egress_stats.if_pause_pkts | packets/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_in_fifo_errors | errors/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_in_frame_errors | errors/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_in_l3_incompletes | packets/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_in_runts | packets/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_errors | errors/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_in_l2chan_errors | errors/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_in_resource_errors | errors/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_in_qdrops | drops/s | x | x |
| junos.system.linecard.interface.ingress_errors.if_in_l2_mismatch_timeouts | packets/s | x | x |
| junos.system.linecard.interface.ingress_stats.if_1sec_pkts | packets/s | x | x |
| junos.system.linecard.interface.ingress_stats.if_octets | octets/s | x | x |
| junos.system.linecard.interface.ingress_stats.if_mc_pkts | packets/s | x | x |
| junos.system.linecard.interface.ingress_stats.if_bc_pkts | packets/s | x | x |

**Table 12: Metrics Available per Interface with JTI Network Device Monitoring *(Continued)***

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| junos.system.linecard.interface.ingress_stats.if_1sec_octets | octets/s | x | x |
| junos.system.linecard.interface.ingress_stats.if_error | errors/s | x | x |
| junos.system.linecard.interface.ingress_stats.if_pkts | packets/s | x | x |
| junos.system.linecard.interface.ingress_stats.if_uc_pkts | packets/s | x | x |
| junos.system.linecard.interface.ingress_stats.if_pause_pkts | packets/s | x | x |

Table 13 on page 117 lists the metrics available per interface queue with JTI network device monitoring. For the complete list, refer to the file `jti_config_all_sensors.json` in the `certified_plugins/` directory of the Contrail Insights installation TAR file.

**Table 13: Metrics Available per Interface Queue with JTI Network Device Monitoring**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| junos.system.linecard.interface.egress_queue_info.bytes | bytes/s | x | x |
| junos.system.linecard.interface.egress_queue_info.packets | packets/s | x | x |
| junos.system.linecard.interface.egress_queue_info.allocated_buffer_size | bytes | x | x |
| junos.system.linecard.interface.egress_queue_info.avg_buffer_occupancy | bytes | x | x |
| junos.system.linecard.interface.egress_queue_info.cur_buffer_occupancy | bytes | x | x |
| junos.system.linecard.interface.egress_queue_info.peak_buffer_occupancy | bytes | x | x |
| junos.system.linecard.interface.egress_queue_info.red_drop_bytes | bytes/s | x | x |

**Table 13: Metrics Available per Interface Queue with JTI Network Device Monitoring *(Continued)***

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| junos.system.linecard.interface.egress_queue_info.red_drop_packets | packets/s | x | x |
| junos.system.linecard.interface.egress_queue_info.rl_drop_bytes | bytes/s | x | x |
| junos.system.linecard.interface.egress_queue_info.rl_drop_packets | packets/s | x | x |
| junos.system.linecard.interface.egress_queue_info.tail_drop_packets | packets/s | x | x |

## Contrail Release 5.0 vRouter Plug-In

Table 14 on page 118 lists metrics published by the Contrail Release 5.0 vRouter plug-in. See *Contrail Monitoring* to view and configure Contrail services from the GUI. See *Service Monitoring Ansible Variables* to configure Contrail monitoring using Ansible.

**Table 14: Metrics for Contrail Release 5.0 vRouter Plug-In**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| plugin.contrail.vrouter.v5. aged_flows | count | x | x |
| plugin.contrail.vrouter.v5. total_flows | count | x | x |
| plugin.contrail.vrouter.v5. exception_packets | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_flow_queue_limit_exceeded | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_flow_table_full | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vlan_fwd_enq | count | x | x |

| | | | |
|---|---|---|---|
| plugin.contrail.vrouter.v5. drop_stats_vlan_fwd_tx | count | x | x |
| plugin.contrail.vrouter.v5. flow_export_drops | count | x | x |
| plugin.contrail.vrouter.v5. flow_export_sampling_drops | count | x | x |
| plugin.contrail.vrouter.v5. flow_rate_active_flows | count | x | x |
| plugin.contrail.vrouter.v5. flow_rate_deleted_flows | count | x | x |
| plugin.contrail.vrouter.v5. flow_rate_added_flows | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_discard | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_pull | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_flow_no_memory | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_flow_invalid_protocol | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_flow_action_drop | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_interface_drop | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_duplicated | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_push | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_invalid_nh | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_invalid_protocol | count | x | x |
| plugin.contrail.vrouter.v5. drop_stats_vhost_ds_drop_pkts | count | x | x |

## Contrail vRouter on a Host

Table 15 on page 120 lists raw metrics available for an Contrail vRouter on a host. See *Service Monitoring Ansible Variables* to configure Contrail monitoring using Ansible.

**Table 15: Raw Metrics for Contrail vRouter**

| Metric | Chart | Alarm |
|---|---|---|
| plugin.contrail.vrouter.aged_flows | x | x |
| plugin.contrail.vrouter.total_flows | x | x |
| plugin.contrail.vrouter.exception_packets | x | x |
| plugin.contrail.vrouter.drop_stats_flow_queue_limit_exceeded | x | x |
| plugin.contrail.vrouter.drop_stats_flow_table_full | x | x |
| plugin.contrail.vrouter.drop_stats_vlan_fwd_enq | x | x |
| plugin.contrail.vrouter.drop_stats_vlan_fwd_tx | x | x |
| plugin.contrail.vrouter.flow_export_drops | x | x |
| plugin.contrail.vrouter.flow_export_sampling_drops | x | x |
| plugin.contrail.vrouter.flow_rate_active_flows | x | x |
| plugin.contrail.vrouter.flow_rate_added_flows | x | x |
| plugin.contrail.vrouter.flow_rate_deleted_flows | x | x |

## OpenStack Project in Chart View

Table 16 on page 121 lists the raw metrics available in the OpenStack Project Chart View. See *Contrail Insights Role-Based Access* to grant Contrail Insights permissions to read-only OpenStack users.

Table 16: Raw Metrics for OpenStack Project

| Metric | Chart | Alarm |
|---|---|---|
| openstack.project.active_instances.count | x | x |
| openstack.project.active_instances.percentage | – | x |
| openstack.project.floating_ip.allocated.count | x | x |
| openstack.project.floating_ip.allocated.percentage | – | x |
| openstack.project.ram.allocated.count | x | x |
| openstack.project.ram.allocated.percentage | – | x |
| openstack.project.security_group.allocated.count | x | x |
| openstack.project.security_group.allocated.percentage | – | x |
| openstack.project.total_disk_usage_gb_hours.count | – | x |
| openstack.project.total_hours.count | – | x |
| openstack.project.total_memory_usage_mb_hours.count | – | x |
| openstack.project.total_vcpu_usage_hours.count | – | x |
| openstack.project.vcpus.allocated.count | – | x |

**Table 16: Raw Metrics for OpenStack Project *(Continued)***

| Metric | Chart | Alarm |
|---|---|---|
| openstack.project.vcpus.allocated.percentage | – | x |
| openstack.project.virtual_network.allocated.count | x | x |
| openstack.project.virtual_network.allocated.percentage | – | x |
| openstack.project.volume.allocated.count | x | x |
| openstack.project.volume.allocated.percentage | – | x |
| openstack.project.volume_gb.allocated.count | x | x |
| openstack.project.volume_gb.allocated.percentage | – | x |

## RabbitMQ Service

Table 17 on page 122 lists the raw metrics available for RabbitMQ monitoring. See *RabbitMQ Monitoring* to view and configure RabbitMQ services from the GUI. See *Service Monitoring Ansible Variables* to configure RabbitMQ monitoring using Ansible.

**Table 17: Raw Metrics for RabbitMQ Monitoring**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| rabbit.cluster.connection_totals.blocked_connections | count | x | x |
| rabbit.cluster.connection_totals.blocked_connections_details | messages/s | x | x |
| rabbit.cluster.message_stats.ack | count | x | x |

Table 17: Raw Metrics for RabbitMQ Monitoring *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| rabbit.cluster.message_stats.ack_details | messages/s | x | x |
| rabbit.cluster.message_stats.deliver | count | x | x |
| rabbit.cluster.message_stats.deliver_details | messages/s | x | x |
| rabbit.cluster.message_stats.deliver_get | count | x | x |
| rabbit.cluster.message_stats.deliver_get_details | messages/s | x | x |
| rabbit.cluster.message_stats.get | count | x | x |
| rabbit.cluster.message_stats.get_details | messages/s | x | x |
| rabbit.cluster.message_stats.publish | count | x | x |
| rabbit.cluster.message_stats.publish_details | messages/s | x | x |
| rabbit.cluster.message_stats.redeliver | count | x | x |
| rabbit.cluster.message_stats.redeliver_details | messages/s | x | x |
| rabbit.cluster.object_totals.channels | count | x | x |
| rabbit.cluster.object_totals.connections | count | x | x |
| rabbit.cluster.object_totals.consumers | count | x | x |
| rabbit.cluster.object_totals.exchanges | count | x | x |

Table 17: Raw Metrics for RabbitMQ Monitoring *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| rabbit.cluster.object_totals.queues | count | x | x |
| rabbit.cluster.queue_totals.blocked_queues | count | x | x |
| rabbit.cluster.queue_totals.blocked_queues_details | messages/s | x | x |
| rabbit.cluster.queue_totals.consumer_utilisation_percent | count | x | x |
| rabbit.cluster.queue_totals.messages | count | x | x |
| rabbit.cluster.queue_totals.messages_details | messages/s | x | x |
| rabbit.cluster.queue_totals.messages_ready | count | x | x |
| rabbit.cluster.queue_totals.messages_ready_details | messages/s | x | x |
| rabbit.cluster.queue_totals.messages_unacknowledged | count | x | x |
| rabbit.cluster.queue_totals.messages_unacknowledged_details | messages/s | x | x |
| rabbit.queue.consumers | count | — | x |
| rabbit.queue.consumer_utilisation | count | — | x |
| rabbit.queue.messages | count | — | x |
| rabbit.queue.messages_ready | count | — | x |
| rabbit.queue.messages_ready_detail | count | — | x |

**Table 17: Raw Metrics for RabbitMQ Monitoring *(Continued)***

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| rabbit.queue.memory | count | – | x |
| rabbit.queue.messages_detail | count | – | x |
| rabbit.queue.messages_unacknowledged | count | – | x |
| rabbit.queue.messages_unacknowledged_detail | count | – | x |
| rabbit.queue.state | count | – | x |
| rabbit.node.sockets_total | count | x | x |
| rabbit.node.fd_total | count | x | x |
| rabbit.node.sockets_used_percent | count | x | x |
| rabbit.node.run_queue | count | x | x |
| rabbit.node.proc_used_percent | count | x | x |
| rabbit.node.proc_total | count | x | x |
| rabbit.node.mem_used_percent | count | x | x |
| rabbit.node.uptime | count | x | x |
| rabbit.node.disk_usage_ratio | count | x | x |
| rabbit.node.disk_free_alarm | count | x | x |

Table 17: Raw Metrics for RabbitMQ Monitoring *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| rabbit.node.fd_used_percent | count | x | x |
| rabbit.node.mem_limit | count | x | x |
| rabbit.node.mem_alarm | count | x | x |
| rabbit.node.disk_free | count | x | x |
| rabbit.node.sockets_used | count | x | x |
| rabbit.node.processors | count | x | x |
| rabbit.node.running | count | x | x |
| rabbit.node.disk_free_limit | count | x | x |
| rabbit.node.fd_used | count | x | x |
| rabbit.node.proc_used | count | x | x |
| rabbit.node.mem_used | count | x | x |
| rabbit.node.heartbeat | count | x | x |
| rabbit.node.latency | count | x | x |

## ScaleIO Service

Table 18 on page 127 lists the raw metrics available for ScaleIO monitoring. See *ScaleIO Monitoring* for viewing and configuring ScaleIO services from the GUI. See *Service Monitoring Ansible Variables* to configure ScaleIO monitoring using Ansible.

**Table 18: Raw Metrics for ScaleIO Monitoring**

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| numOfDevices | count | x | x |
| numOfProtectionDomains | count | x | x |
| numOfSdc | count | x | x |
| numOfSds | count | x | x |
| numOfStoragePools | count | x | x |
| numOfVtrees | count | x | x |
| numOfSnapshots | count | x | x |
| numOfVolumes | count | x | x |
| numOfThickBaseVolumes | count | x | x |
| numOfThinBaseVolumes | count | x | x |
| numOfVolumesInDeletion | count | x | x |
| numOfMappedToAllVolumes | count | x | x |
| numOfUnmappedVolumes | count | x | x |

Table 18: Raw Metrics for ScaleIO Monitoring *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| capacityAvailableForVolumeAllocationInKb | Kbyte | x | x |
| capacityInUseInKb | Kbyte | x | x |
| capacityLimitInKb | Kbyte | x | x |
| unusedCapacityInKb | Kbyte | x | x |
| spareCapacityInKb | Kbyte | x | x |
| protectedCapacityInKb | Kbyte | x | x |
| maxCapacityInKb | Kbyte | x | x |
| snapCapacityInUseInKb | Kbyte | x | x |
| thickCapacityInUseInKb | Kbyte | x | x |
| thinCapacityInUseInKb | Kbyte | x | x |
| bckRebuildReadBandwidth | Kbyte/sec | x | x |
| bckRebuildWriteBandwidth | Kbyte/sec | x | x |
| fwdRebuildReadBandwidth | Kbyte/sec | x | x |
| fwdRebuildWriteBandwidth | Kbyte/sec | x | x |
| normRebuildReadBandwidth | Kbyte/sec | x | x |

Table 18: Raw Metrics for ScaleIO Monitoring *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| normRebuildWriteBandwidth | Kbyte/sec | x | x |
| primaryReadBandwidth | Kbyte/sec | x | x |
| primaryWriteBandwidth | Kbyte/sec | x | x |
| rebalanceReadBandwidth | Kbyte/sec | x | x |
| rebalanceWriteBandwidth | Kbyte/sec | x | x |
| secondaryReadBandwidth | Kbyte/sec | x | x |
| secondaryWriteBandwidth | Kbyte/sec | x | x |
| totalReadBandwidth | Kbyte/sec | x | x |
| totalWriteBandwidth | Kbyte/sec | x | x |
| bckRebuildReadIops | IOPS | x | x |
| bckRebuildWriteIops | IOPS | x | x |
| fwdRebuildReadIops | IOPS | x | x |
| fwdRebuildWriteIops | IOPS | x | x |
| normRebuildReadIops | IOPS | x | x |
| normRebuildWriteIops | IOPS | x | x |

Table 18: Raw Metrics for ScaleIO Monitoring *(Continued)*

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| primaryReadIops | IOPS | x | x |
| primaryWriteIops | IOPS | x | x |
| rebalanceReadIops | IOPS | x | x |
| rebalanceWriteIops | IOPS | x | x |
| secondaryReadIops | IOPS | x | x |
| secondaryWriteIops | IOPS | x | x |
| totalReadIops | IOPS | x | x |
| totalWriteIops | IOPS | x | x |
| bckRebuildReadIosize | Kbyte | x | x |
| bckRebuildWriteIosize | Kbyte | x | x |
| fwdRebuildReadIosize | Kbyte | x | x |
| fwdRebuildWriteIosize | Kbyte | x | x |
| normRebuildReadIosize | Kbyte | x | x |
| normRebuildWriteIosize | Kbyte | x | x |
| primaryReadIosize | Kbyte | x | x |

**Table 18: Raw Metrics for ScaleIO Monitoring *(Continued)***

| Metric | Unit | Chart | Alarm |
|---|---|---|---|
| primaryWriteIosize | Kbyte | x | x |
| rebalanceReadIosize | Kbyte | x | x |
| rebalanceWriteIosize | Kbyte | x | x |
| secondaryReadIosize | Kbyte | x | x |
| secondaryWriteIosize | Kbyte | x | x |
| totalReadIosize | Kbyte | x | x |
| totalWriteIosize | Kbyte | x | x |

## gRPC Sensors

Table 19 on page 131 lists the available gRPC sensors. To enable these sensors, see *Custom Sensors for JTI, gRPC, and NETCONF*.

**NOTE**: These sensors are applicable only for Juniper network devices.

**Table 19: gRPC Sensors**

| Sensor | Chart | Alarm |
|---|---|---|
| /junos/services/label-switched-path/usage/ | x | x |
| /components/ | x | x |

Table 19: gRPC Sensors *(Continued)*

| Sensor | Chart | Alarm |
|---|---|---|
| /junos/system/subscriber-management/infra/sdb/statistics/ | x | x |
| /junos/task-memory-information/task-memory-overall-report/task-memory-stats-list/task-memory-stats/ | x | x |
| /junos/task-memory-information/task-memory-overall-report/task-size-block-list/task-size-block/ | x | x |
| /lldp/interfaces/interface/state/ | x | x |
| /interfaces/ | x | x |
| /bgp-rib/afi-safis/afi-safi/ipv4-unicast/loc-rib/ | x | x |
| /bgp-rib/afi-safis/afi-safi/ipv6-unicast/loc-rib/ | x | x |
| /bgp-rib/afi-safis/afi-safi/ipv4-unicast/neighbors/ | x | x |
| /bgp-rib/afi-safis/afi-safi/ipv6-unicast/neighbors/ | x | x |
| /junos/system/linecard/qmon/ | x | x |
| /junos/system/linecard/optics/ | x | x |
| /junos/system/linecard/packet/usage/ | x | x |
| /junos/system/linecard/firewall/ | x | x |
| /junos/rsvp-interface-information/ | x | x |

Table 19: gRPC Sensors *(Continued)*

| Sensor | Chart | Alarm |
|---|---|---|
| /junos/system/linecard/npu/memory | x | x |
| /junos/system/linecard/cpu/memory/ | x | x |
| /lacp/ | x | x |
| /network-instances/network-instance/protocols/protocol/isis/levels/level/ | x | x |
| /junos/services/segment-routing/interface/ingress/usage/ | x | x |
| /junos/services/segment-routing/interface/egress/usage/ | x | x |
| /lldp/ | x | x |
| /mpls/ | x | x |
| /nd6-information/ | x | x |
| /arp-information/ | x | x |
| /junos/system/subscriber-management/infra/network/ppp/ | x | x |
| /network-instances/network-instance/protocols/protocol/bgp/ | x | x |
| /network-instances/network-instance/protocols/protocol/isis/levels/level/ | x | x |

Table 19: gRPC Sensors *(Continued)*

| Sensor | Chart | Alarm |
|---|---|---|
| /junos/services/segment-routing/sid/usage/ | x | x |

RELATED DOCUMENTATION

*Alarms*

*Configure Network Devices from the UI*

*Configure Network Device from JSON File*

*Contrail Insights Auto Discovery of Network Devices from Contrail Networking*

*Contrail Insights JTI (UDP) Monitoring*

*Contrail Insights JTI (gRPC) Monitoring*

*Contrail Insights SNMP Monitoring*

*Contrail Insights Network Device Monitoring Common Issues*

*Service Monitoring from the UI*

# Reports

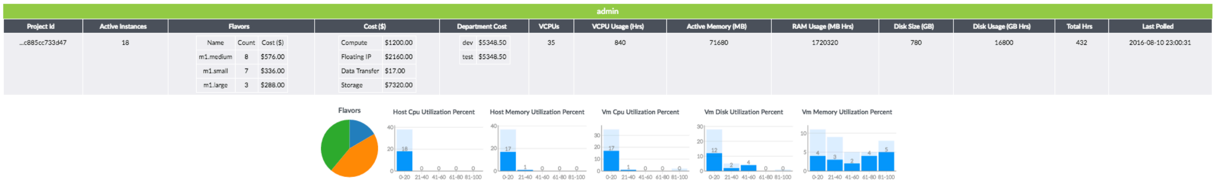
IN THIS SECTION

Contrail Insights Reports enable analysis of how infrastructure resources are consumed by instances over time. You can generate a report over a specified time period, organized by different scopes: project or host. In each case, the report shows the resource utilization by each instance that is in a project or scheduled on a host. Dashboard displays a report in both graphical or tabular formats. You can also download report data as a HTML-formatted report, raw comma-separated value (CSV) file, or JSONformatted data for further analysis. The following video provides an overview of the Contrail Insights reports that help you understand how resources are being used in an OpenStack cluster.

**Video:** Contrail Insights Reports

The graphical view provides a quick, visual overview of resource utilization by instance using histograms. The bins of the histogram represent the number of instances that used a given percentage of a resource, such as CPU utilization. Using the histograms, you can quickly identify patterns that indicate under-provisioned or over-provisioned instances. The dark blue bars of the histrogram depict the resource utilization by instances on a particular project or host. The light blue bars depict the total resource utilization across all hosts or projects, so that you can understand the resource utilization in context of the entire infrastructure. Figure 75 on page 135 shows a graphical view of resource utilization.

### Figure 75: Report Graphical View of Resource Utilization by Instance

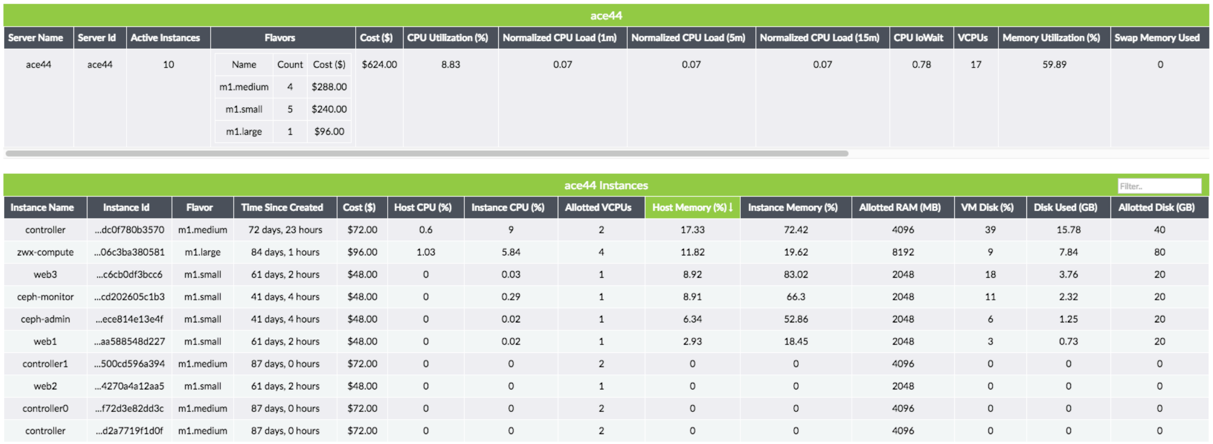


admin

| Project Id | Active Instances | Flavors | Cost ($) | Department Cost | VCPUs | VCPU Usage (Hrs) | Active Memory (MB) | RAM Usage (MB Hrs) | Disk Size (GB) | Disk Usage (GB Hrs) | Total Hrs | Last Polled |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ...c885cc733d47 | 18 | Name Count Cost ($)<br>m1.medium 8 $576.00<br>m1.small 7 $336.00<br>m1.large 3 $288.00 | Compute $1200.00<br>Floating IP $2160.00<br>Data Transfer $17.00<br>Storage $7320.00 | dev $5348.50<br>test $5348.50 | 35 | 840 | 71680 | 1720320 | 780 | 16800 | 432 | 2016-08-10 23:00:31 |

The tabular format shows additional detail in an interactive table that can be sorted and filtered. With the tabular display, as shown in Figure 76 on page 135 you can view resource utilization for a particular instance.

### Figure 76: Report Tabular View of Resource Utilization for a Particular Instance



ace44

| Server Name | Server Id | Active Instances | Flavors | Cost ($) | CPU Utilization (%) | Normalized CPU Load (1m) | Normalized CPU Load (5m) | Normalized CPU Load (15m) | CPU IoWait | VCPUs | Memory Utilization (%) | Swap Memory Used |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ace44 | ace44 | 10 | Name Count Cost ($)<br>m1.medium 4 $288.00<br>m1.small 5 $240.00<br>m1.large 1 $96.00 | $624.00 | 8.83 | 0.07 | 0.07 | 0.07 | 0.78 | 17 | 59.89 | 0 |

ace44 Instances

| Instance Name | Instance Id | Flavor | Time Since Created | Cost ($) | Host CPU (%) | Instance CPU (%) | Allotted VCPUs | Host Memory (%) ↓ | Instance Memory (%) | Allotted RAM (MB) | VM Disk (%) | Disk Used (GB) | Allotted Disk (GB) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| controller | ...dc0f780b3570 | m1.medium | 72 days, 23 hours | $72.00 | 0.6 | 9 | 2 | 17.33 | 72.42 | 4096 | 39 | 15.78 | 40 |
| zwx-compute | ...06c3ba380581 | m1.large | 84 days, 1 hours | $96.00 | 1.03 | 5.84 | 4 | 11.82 | 19.62 | 8192 | 9 | 7.84 | 80 |
| web3 | ...c6cb0df3bcc6 | m1.small | 61 days, 2 hours | $48.00 | 0 | 0.03 | 1 | 8.92 | 83.02 | 2048 | 18 | 3.76 | 20 |
| ceph-monitor | ...cd202605c1b3 | m1.small | 41 days, 4 hours | $48.00 | 0 | 0.29 | 1 | 8.91 | 66.3 | 2048 | 11 | 2.32 | 20 |
| ceph-admin | ...ece814e13e4f | m1.small | 41 days, 4 hours | $48.00 | 0 | 0.02 | 1 | 6.34 | 52.86 | 2048 | 6 | 1.25 | 20 |
| web1 | ...aa588548d227 | m1.small | 61 days, 2 hours | $48.00 | 0 | 0.02 | 1 | 2.93 | 18.45 | 2048 | 3 | 0.73 | 20 |
| controller1 | ...500cd596a394 | m1.medium | 87 days, 0 hours | $72.00 | 0 | 0 | 2 | 0 | 0 | 4096 | 0 | 0 | 0 |
| web2 | ...4270a4a12aa5 | m1.small | 61 days, 2 hours | $48.00 | 0 | 0 | 1 | 0 | 0 | 2048 | 0 | 0 | 0 |
| controller0 | ...f72d3e82dd3c | m1.medium | 87 days, 0 hours | $72.00 | 0 | 0 | 2 | 0 | 0 | 4096 | 0 | 0 | 0 |
| controller | ...d2a7719f1d0f | m1.medium | 87 days, 0 hours | $72.00 | 0 | 0 | 2 | 0 | 0 | 4096 | 0 | 0 | 0 |

In all views of the reports, you can also view costs charged for infastructure resource usage. The rate structure for resources is configurable in the Chargeback Settings.

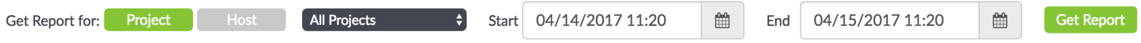
## Report Generation

To generate a report:

1. Select the type of report—**Project**, **Host**—and a context appropriate for the report type.

   For example, a project report can be generated for all projects or a single project. Figure 77 on page 136 shows the report configuration action bar.

   **Figure 77: Report Configuration Action Bar**

   All Projects

   Start 04/14/2017 11:20

   End 04/15/2017 11:20

2. Select a date range for the report. The report summarizes resource consumption and cost for the specified period.

3. Click **Get Report** to start generation of the report.

   After the report is generated, it is presented in a list of available reports.

4. (Optional) A report can be deleted by clicking the trash can icon.

## Project Report Generation

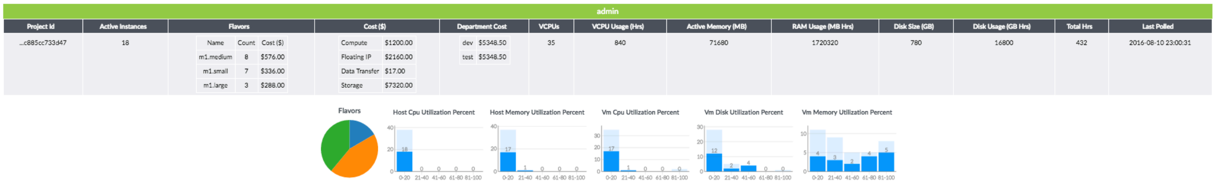
A project report may be generated for a single project or for all projects (provided you are authorized to access the project or all projects). A project report shows resource allocations, actual usage, and charges.

Resource allocation includes static allocations of resources, such as vCPUs, floating IP addresses, and storage volumes.

Actual resource usage is displayed for each instance in the project, and as the aggregate sum of usage by all instances in the project. Resource usage shows the actual physical resources consumed by an instance, such as CPU usage percentage, memory usage percentage, network I/O, and disk I/O.

The cost charged for resource usage is shown for each instance in the project. In addition, a cost breakdown by flavor type, and by resource type (compute, network, storage) is shown for the project as a whole. Figure 78 on page 137 and Figure 79 on page 137 show the graphical and tabular views for a project report.

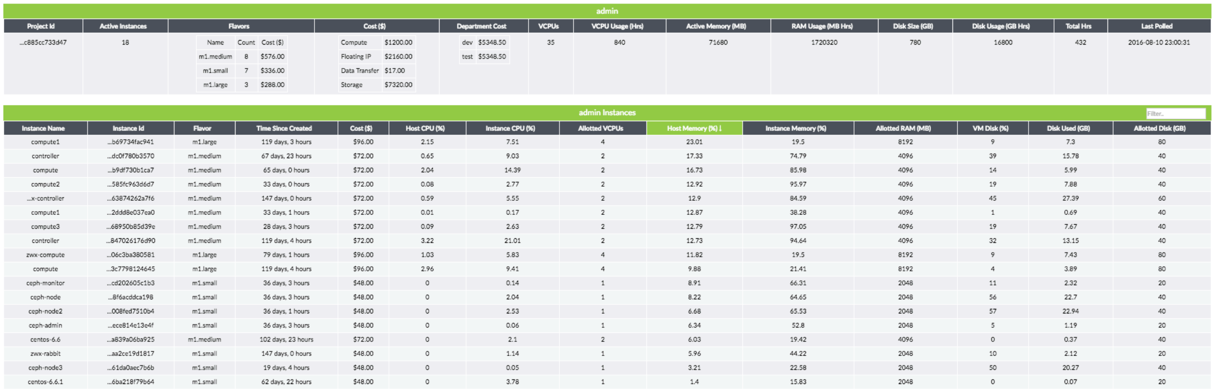
Figure 78: Project Report Graphical View for Admin and Admin Instances



admin

| Project Id | Active Instances | Flavors | Cost ($) | Department Cost | VCPUs | VCPU Usage (Hrs) | Active Memory (MB) | RAM Usage (MB Hrs) | Disk Size (GB) | Disk Usage (GB Hrs) | Total Hrs | Last Polled |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ...c885cc733d47 | 18 | Name Count Cost ($)<br>m1.medium 8 $576.00<br>m1.small 7 $336.00<br>m1.large 3 $288.00 | Compute $1200.00<br>Floating IP $2160.00<br>Data Transfer $17.00<br>Storage $7320.00 | dev $5348.50<br>test $5348.50 | 35 | 840 | 71680 | 1720320 | 780 | 16800 | 432 | 2016-08-10 23:00:31 |

Figure 79: Project Report Tabular View for Admin and Admin Instances



admin

| Project Id | Active Instances | Flavors | Cost ($) | Department Cost | VCPUs | VCPU Usage (Hrs) | Active Memory (MB) | RAM Usage (MB Hrs) | Disk Size (GB) | Disk Usage (GB Hrs) | Total Hrs | Last Polled |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ...c885cc733d47 | 18 | Name Count Cost ($)<br>m1.medium 8 $576.00<br>m1.small 7 $336.00<br>m1.large 3 $288.00 | Compute $1200.00<br>Floating IP $2160.00<br>Data Transfer $17.00<br>Storage $7320.00 | dev $5348.50<br>test $5348.50 | 35 | 840 | 71680 | 1720320 | 780 | 16800 | 432 | 2016-08-10 23:00:31 |

admin Instances

| Instance Name | Instance Id | Flavor | Time Since Created | Cost ($) | Host CPU (%) | Instance CPU (%) | Allotted VCPUs | Host Memory (%) ↓ | Instance Memory (%) | Allotted RAM (MB) | VM Disk (%) | Disk Used (GB) | Allotted Disk (GB) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| compute1 | ...b69734fac941 | m1.large | 119 days, 3 hours | $96.00 | 2.15 | 7.51 | 4 | 23.01 | 19.5 | 8192 | 9 | 7.3 | 80 |
| controller | ...dc0f780b3570 | m1.medium | 67 days, 23 hours | $72.00 | 0.65 | 9.03 | 2 | 17.33 | 74.79 | 4096 | 39 | 15.78 | 40 |
| compute | ...b9df730b1ca7 | m1.medium | 65 days, 0 hours | $72.00 | 2.04 | 14.39 | 2 | 16.73 | 85.98 | 4096 | 14 | 5.99 | 40 |
| compute2 | ...585fc963d6d7 | m1.medium | 33 days, 0 hours | $72.00 | 0.08 | 2.77 | 2 | 12.92 | 95.97 | 4096 | 19 | 7.88 | 40 |
| ...x-controller | ...63874262a7f6 | m1.medium | 147 days, 0 hours | $72.00 | 0.59 | 5.55 | 2 | 12.9 | 84.59 | 4096 | 45 | 27.39 | 60 |
| compute1 | ...2ddd8e037ea0 | m1.medium | 33 days, 1 hours | $72.00 | 0.01 | 0.17 | 2 | 12.87 | 38.28 | 4096 | 1 | 0.69 | 40 |
| compute3 | ...68950b85d39e | m1.medium | 28 days, 3 hours | $72.00 | 0.09 | 2.63 | 2 | 12.79 | 97.05 | 4096 | 19 | 7.67 | 40 |
| controller | ...847026176d90 | m1.medium | 119 days, 4 hours | $72.00 | 3.22 | 21.01 | 2 | 12.73 | 94.64 | 4096 | 32 | 13.15 | 40 |
| zwx-compute | ...06c3ba380581 | m1.large | 79 days, 1 hours | $96.00 | 1.03 | 5.83 | 4 | 11.82 | 19.5 | 8192 | 9 | 7.43 | 80 |
| compute | ...3c7798124645 | m1.large | 119 days, 4 hours | $96.00 | 2.96 | 9.41 | 4 | 9.88 | 21.41 | 8192 | 4 | 3.89 | 80 |
| ceph-monitor | ...cd202605c1b3 | m1.small | 36 days, 3 hours | $48.00 | 0 | 0.14 | 1 | 8.91 | 66.31 | 2048 | 11 | 2.32 | 20 |
| ceph-node | ...8f6acddca198 | m1.small | 36 days, 3 hours | $48.00 | 0 | 2.04 | 1 | 8.22 | 64.65 | 2048 | 56 | 22.7 | 40 |
| ceph-node2 | ...008fed7510b4 | m1.small | 36 days, 1 hours | $48.00 | 0 | 2.53 | 1 | 6.68 | 65.53 | 2048 | 57 | 22.94 | 40 |
| ceph-admin | ...ece814e13e4f | m1.small | 36 days, 3 hours | $48.00 | 0 | 0.06 | 1 | 6.34 | 52.8 | 2048 | 5 | 1.19 | 20 |
| centos-6.6 | ...a839a06ba925 | m1.medium | 102 days, 23 hours | $72.00 | 0 | 2.1 | 2 | 6.03 | 19.42 | 4096 | 0 | 0.37 | 40 |
| zwx-rabbit | ...aa2ce19d1817 | m1.small | 147 days, 0 hours | $48.00 | 0 | 1.14 | 1 | 5.96 | 44.22 | 2048 | 10 | 2.12 | 20 |
| ceph-node3 | ...61da0aec7b6b | m1.small | 19 days, 4 hours | $48.00 | 0 | 0.05 | 1 | 3.21 | 22.58 | 2048 | 50 | 20.27 | 40 |
| centos-6.6.1 | ...6ba218f79b64 | m1.small | 62 days, 22 hours | $48.00 | 0 | 3.78 | 1 | 1.4 | 15.83 | 2048 | 0 | 0.07 | 20 |

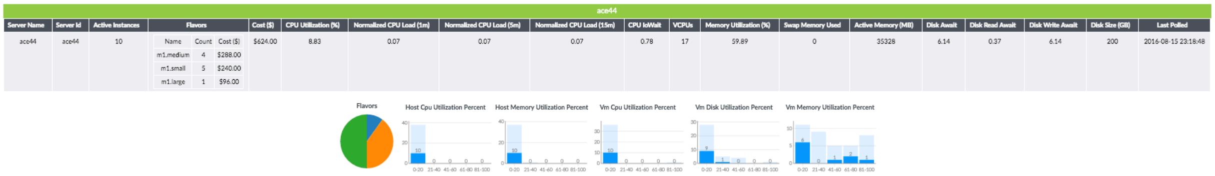
# Host Report Generation

A host report can be generated for all hosts or the set of hosts in a host aggregate. Only users with administrator role may generate a host report.

A host report shows the aggregate resource usage of a host, and a breakdown of resource usage by each instance scheduled on a host.

A host report also shows the cost charged for each instance on a host, as well as the total cost and total cost per flavor type. This provides an indication of the revenue generated by a host. Figure 80 on page 138 and Figure 81 on page 138 show the graphical and tabular views for a host report.

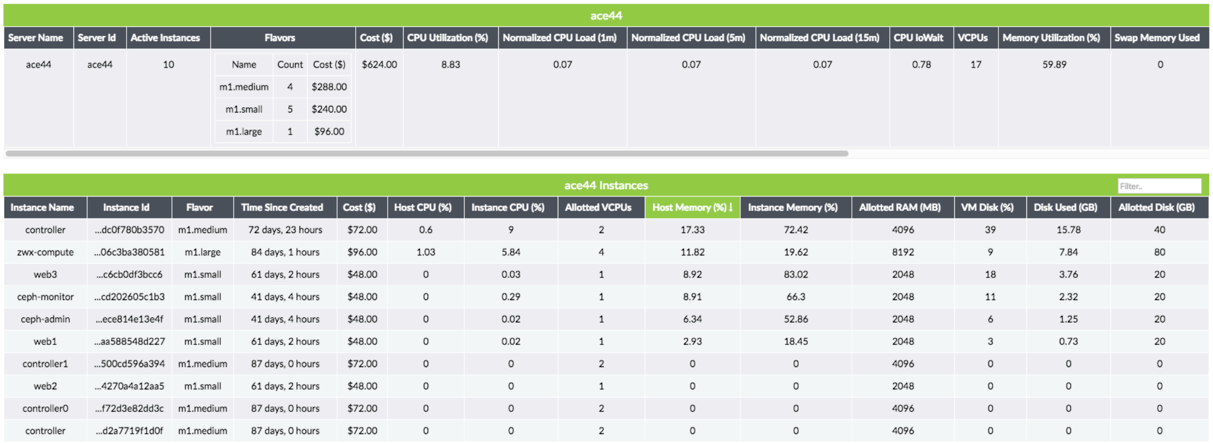
### Figure 80: Host Report Graphical View



ace44

| Server Name | Server Id | Active Instances | Flavors | Cost ($) | CPU Utilization (%) | Normalized CPU Load (1m) | Normalized CPU Load (5m) | Normalized CPU Load (15m) | CPU IoWait | VCPUs | Memory Utilization (%) | Swap Memory Used | Active Memory (MB) | Disk Await | Disk Read Await | Disk Write Await | Disk Size (GB) | Last Polled |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ace44 | ace44 | 10 | Name Count Cost ($)<br>m1.medium 4 $288.00<br>m1.small 5 $240.00<br>m1.large 1 $96.00 | $624.00 | 8.83 | 0.07 | 0.07 | 0.07 | 0.78 | 17 | 59.89 | 0 | 35328 | 6.14 | 0.37 | 6.14 | 200 | 2016-08-15 23:18:48 |

### Figure 81: Host Report Tabular View



ace44

| Server Name | Server Id | Active Instances | Flavors | Cost ($) | CPU Utilization (%) | Normalized CPU Load (1m) | Normalized CPU Load (5m) | Normalized CPU Load (15m) | CPU IoWait | VCPUs | Memory Utilization (%) | Swap Memory Used |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ace44 | ace44 | 10 | Name Count Cost ($)<br>m1.medium 4 $288.00<br>m1.small 5 $240.00<br>m1.large 1 $96.00 | $624.00 | 8.83 | 0.07 | 0.07 | 0.07 | 0.78 | 17 | 59.89 | 0 |

ace44 Instances

| Instance Name | Instance Id | Flavor | Time Since Created | Cost ($) | Host CPU (%) | Instance CPU (%) | Allotted VCPUs | Host Memory (%) ↓ | Instance Memory (%) | Allotted RAM (MB) | VM Disk (%) | Disk Used (GB) | Allotted Disk (GB) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| controller | ...dc0f780b3570 | m1.medium | 72 days, 23 hours | $72.00 | 0.6 | 9 | 2 | 17.33 | 72.42 | 4096 | 39 | 15.78 | 40 |
| zwx-compute | ...06c3ba380581 | m1.large | 84 days, 1 hours | $96.00 | 1.03 | 5.84 | 4 | 11.82 | 19.62 | 8192 | 9 | 7.84 | 80 |
| web3 | ...c6cb0df3bcc6 | m1.small | 61 days, 2 hours | $48.00 | 0 | 0.03 | 1 | 8.92 | 83.02 | 2048 | 18 | 3.76 | 20 |
| ceph-monitor | ...cd202605c1b3 | m1.small | 41 days, 4 hours | $48.00 | 0 | 0.29 | 1 | 8.91 | 66.3 | 2048 | 11 | 2.32 | 20 |
| ceph-admin | ...ece814e13e4f | m1.small | 41 days, 4 hours | $48.00 | 0 | 0.02 | 1 | 6.34 | 52.86 | 2048 | 6 | 1.25 | 20 |
| web1 | ...aa588548d227 | m1.small | 61 days, 2 hours | $48.00 | 0 | 0.02 | 1 | 2.93 | 18.45 | 2048 | 3 | 0.73 | 20 |
| controller1 | ...500cd596a394 | m1.medium | 87 days, 0 hours | $72.00 | 0 | 0 | 2 | 0 | 0 | 4096 | 0 | 0 | 0 |
| web2 | ...4270a4a12aa5 | m1.small | 61 days, 2 hours | $48.00 | 0 | 0 | 1 | 0 | 0 | 2048 | 0 | 0 | 0 |
| controller0 | ...f72d3e82dd3c | m1.medium | 87 days, 0 hours | $72.00 | 0 | 0 | 2 | 0 | 0 | 4096 | 0 | 0 | 0 |
| controller | ...d2a7719f1d0f | m1.medium | 87 days, 0 hours | $72.00 | 0 | 0 | 2 | 0 | 0 | 4096 | 0 | 0 | 0 |

## RELATED DOCUMENTATION

*Alarms*

*Capacity Planning*

*Chargeback*

*Charts*

*Health Monitor*

*Heat Map*

*Metrics Collected by Contrail Insights*

*Notifications*

*Extensibility Using Plug-Ins*

*Service Monitoring from the UI*

# Endpoint Monitoring with Service Groups

IN THIS SECTION

Service group is a logical collection of URL endpoints that can be monitored as a single entity. This feature performs continuous monitoring of all endpoints, monitors their reachability, and collects corresponding latency metrics. Contrail Insights supports charting of metrics collected for each endpoint.

There are two ways to add Service Groups in Contrail Insights:

1. Add service groups using the Contrail Insights Dashboard.

2. Add service groups using Ansible.

**NOTE**: Based on your system setup the REST calls being made can take more time. Use `appformix_agent_timeout_rest_client` in the `group_vars/all` file to set a timeout value suitable to your system. The default value is two (2) seconds.
Example:

```
appformix_agent_timeout_rest_client: 5
```

## Monitor Service Groups on User-Specified Agents

By default, service groups are monitored from the Contrail Insights Platform nodes. You can monitor service groups on specific Contrail Insights Agent by specifying them in the Ansible inventory.

Example:

In the Ansible inventory directory, edit the `inventory/hosts` file to add the following:

```
[appformix_service_group_agents]
172.16.70.220
172.16.70.221
```

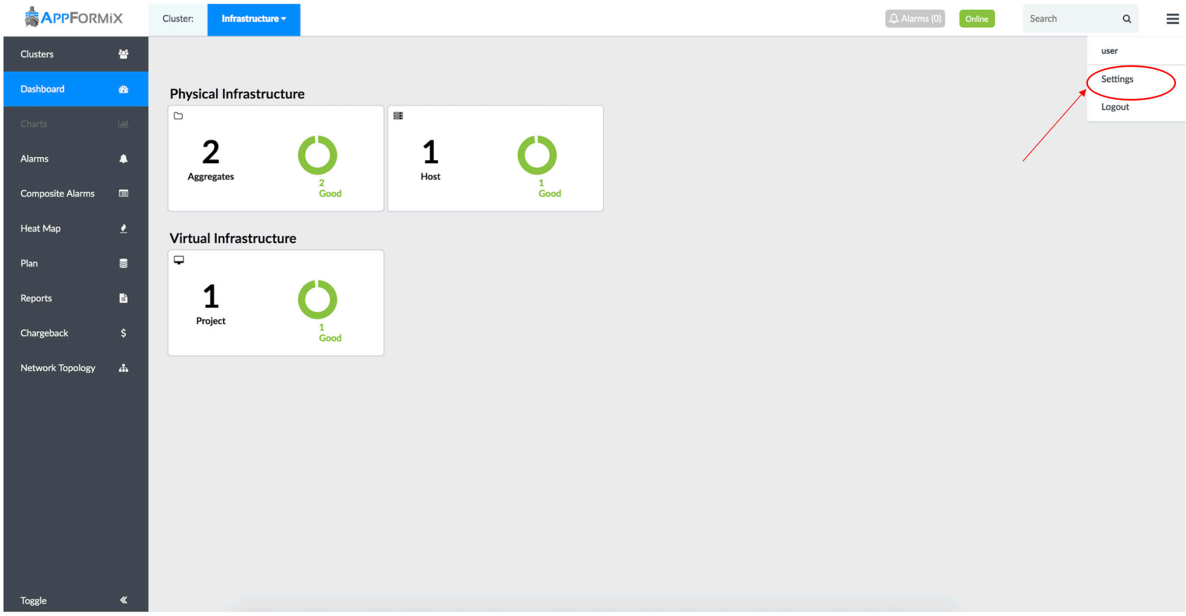
Now service groups are monitored on the Agents specified below the `appformix_service_group_agents` tag.

## Add Service Groups Using Contrail Insights Dashboard

To add a service group to the Dashboard:

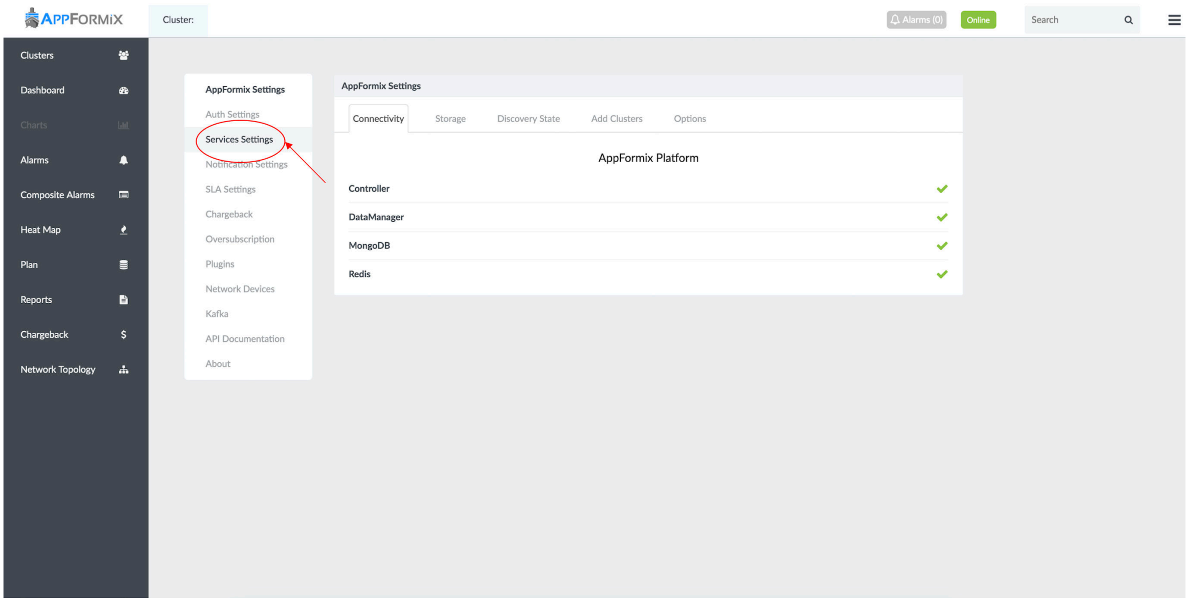
1. Select **Settings** in the top right of the Dashboard as shown in Figure 82 on page 140.

**Figure 82: Settings in Dashboard**



2. Select **Services Settings**.

**Figure 83: Services Settings for Service Groups**

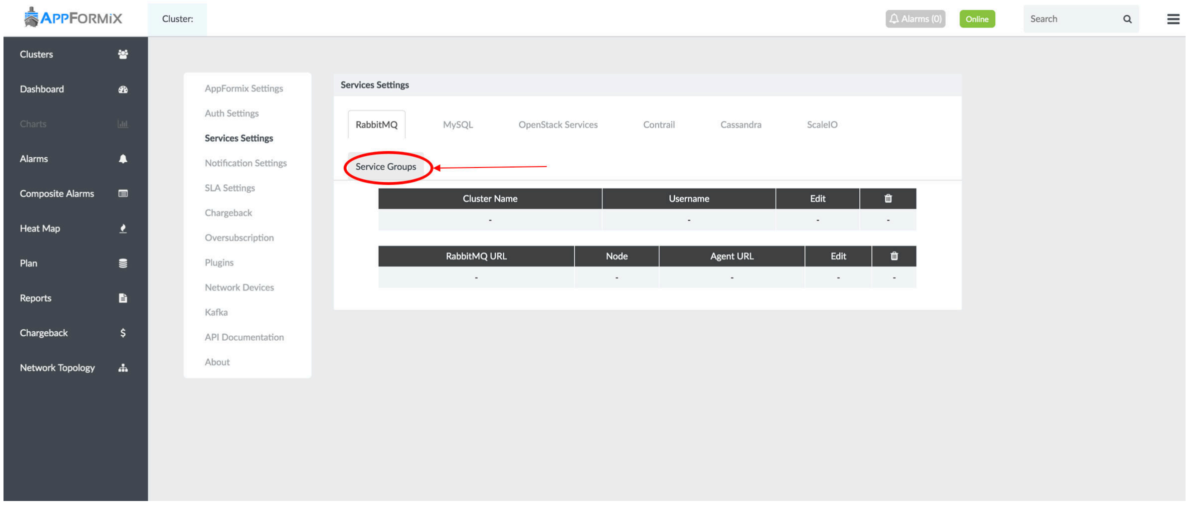


AppFormix Platform

| | |
|---|---|
| Controller | ✓ |
| DataManager | ✓ |
| MongoDB | ✓ |
| Redis | ✓ |

3. In the Services Settings panel, select the **Service Groups** tab and complete the necessary fields.

**Figure 84: Service Groups Tab**



| Cluster Name | Username | Edit | 🗑 |
|---|---|---|---|
| - | - | - | - |

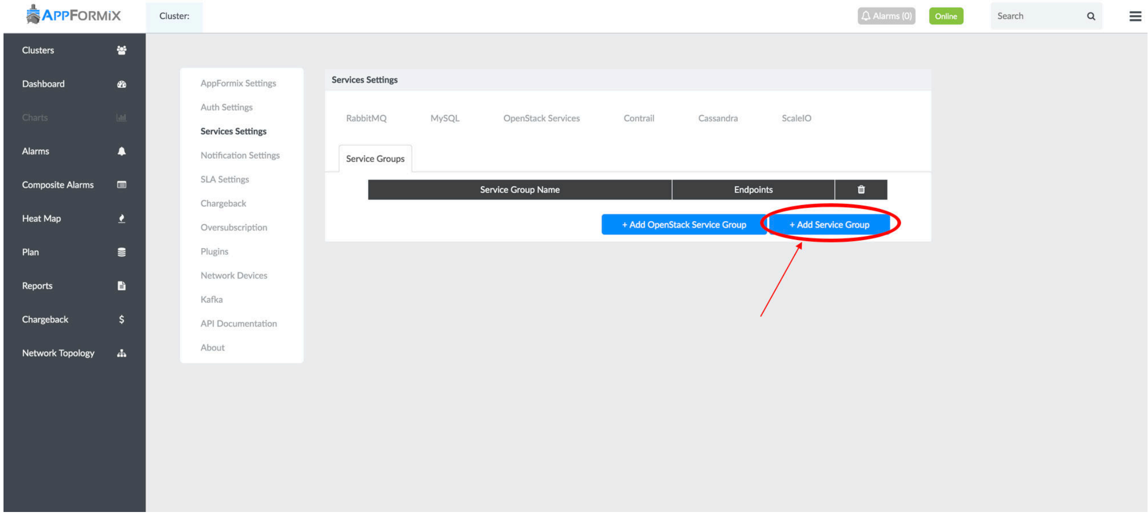
| RabbitMQ URL | Node | Agent URL | Edit | 🗑 |
|---|---|---|---|---|
| - | - | - | - | - |

4. Contrail Insights supports monitoring of OpenStack URLs, regular URLs, and ICMP (Internet Control Message Protocol) URLs.

To monitor ICMP endpoint URLs:

a. Click **Add Service Group**.

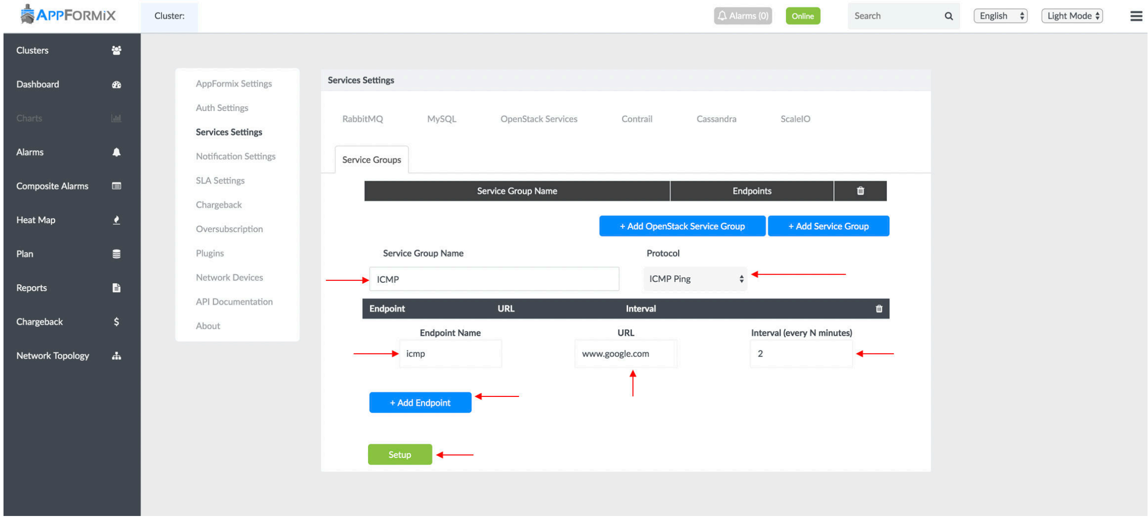
**Figure 85: Add Service Group to Monitor ICMP Endpoint URLs**



b. Complete the indicated fields, as shown in Figure 86 on page 142.

For Protocol, select **ICMP Ping** from the drop-down list.

**Figure 86: Add Service Group Details for ICMP**



Service Group Name: ICMP

Protocol: ICMP Ping

| Endpoint Name | URL | Interval (every N minutes) |
|---|---|---|
| icmp | www.google.com | 2 |

c. Enter the ICMP endpoint you want to monitor in the URL field and the interval at which it needs to be monitored.

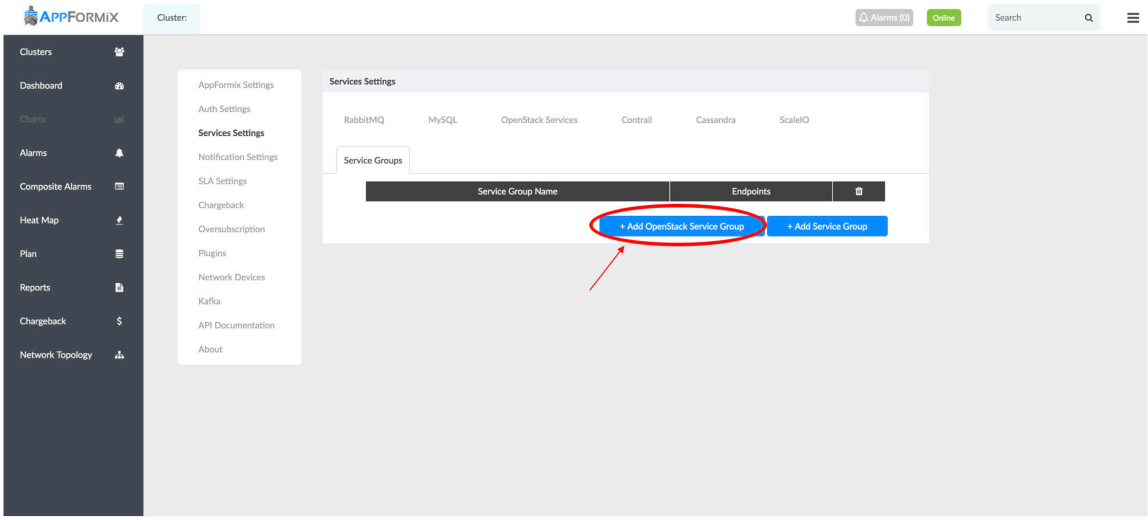
d. Click **Add Endpoint** when done configuring the endpoint. Multiple endpoints can be monitored under a single service group.

e. When done adding endpoints for this specific Service Group, click **Setup**.

To monitor OpenStack endpoint URLs:

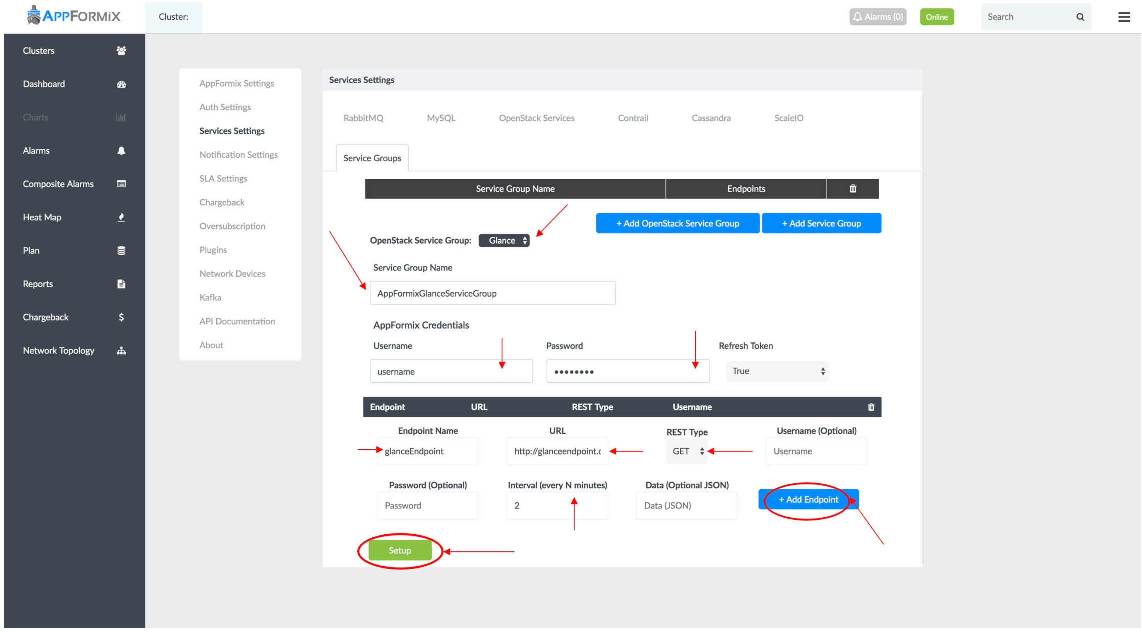
a. Click **Add Openstack Service Group**.

**Figure 87: Add OpenStack Service Group**



b. From the drop-down list, select the type of OpenStack service endpoint that needs to be monitored. This will autogenerate a service group name, which you can modify, if needed.

### Figure 88: Autogenerated Service Group Name



OpenStack Service Group: Glance

Service Group Name: AppFormixGlanceServiceGroup

AppFormix Credentials

Username: username

Password: ••••••••

Refresh Token: True

Endpoint Name: glanceEndpoint

URL: http://glanceendpoint.c

REST Type: GET

Username (Optional): Username

Password (Optional): Password

Interval (every N minutes): 2

Data (Optional JSON): Data (JSON)

c. Enter the Username and Password for the Contrail Insights Credentials.

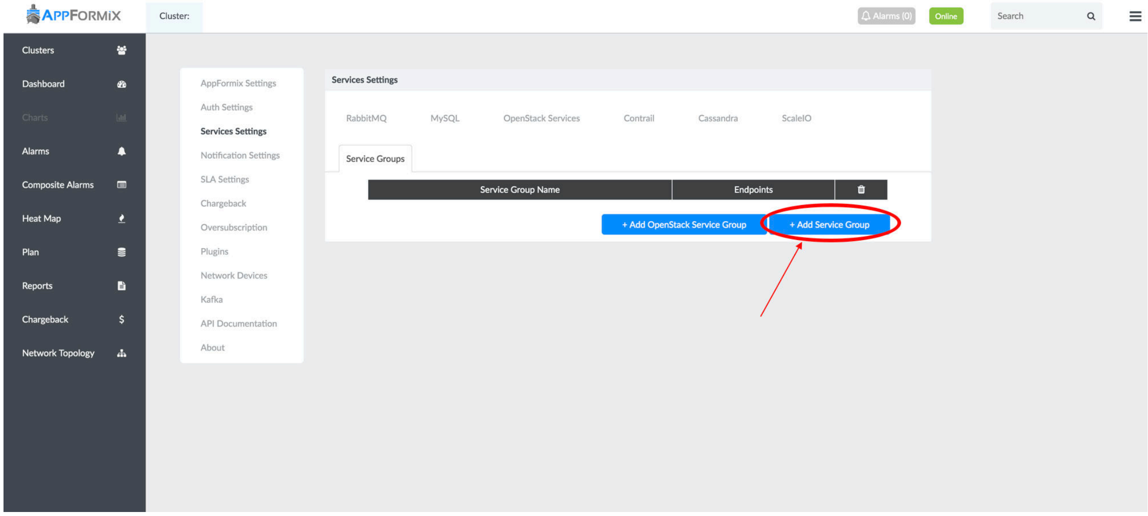
d. Enter the OpenStack endpoint you want to monitor in the URL field, the interval at which it needs to be monitored, and the type of REST call that needs to be made.

e. Click **Add Endpoint** when done configuring the endpoint. Multiple endpoints can be monitored under a single service group.

To monitor regular endpoint URLs:

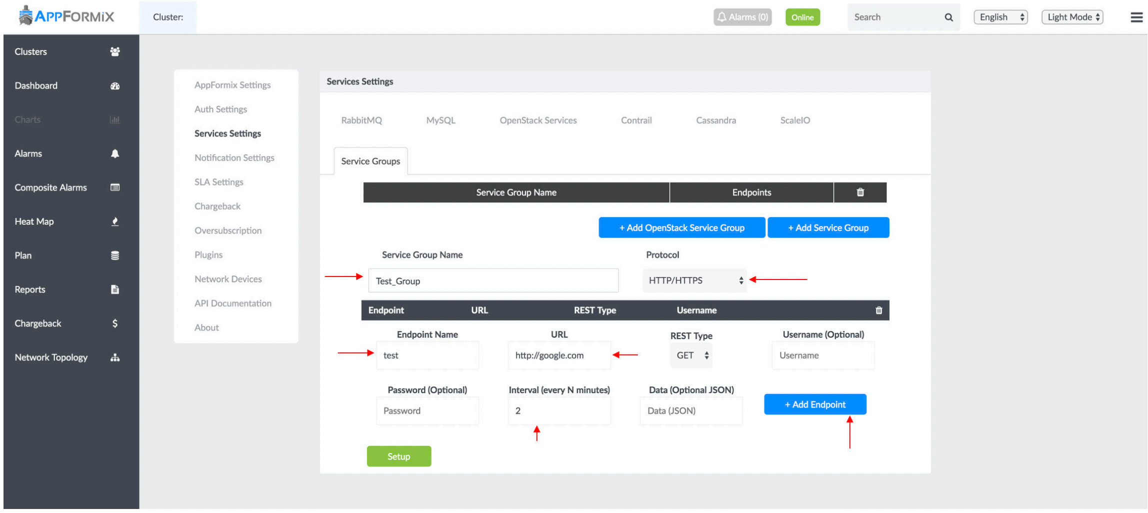
a. Click **Add Service Group**.

**Figure 89: Add Service Group to Monitor Regular Endpoint URLs**



b. Complete the indicated fields, as shown in Figure 90 on page 145. Default Protocol is **HTTP/ HTTPS**.

**Figure 90: Add Service Group Endpoint Details**



Service Group Name: Test_Group

Protocol: HTTP/HTTPS

Endpoint Name: test

URL: http://google.com

REST Type: GET

Username (Optional): Username

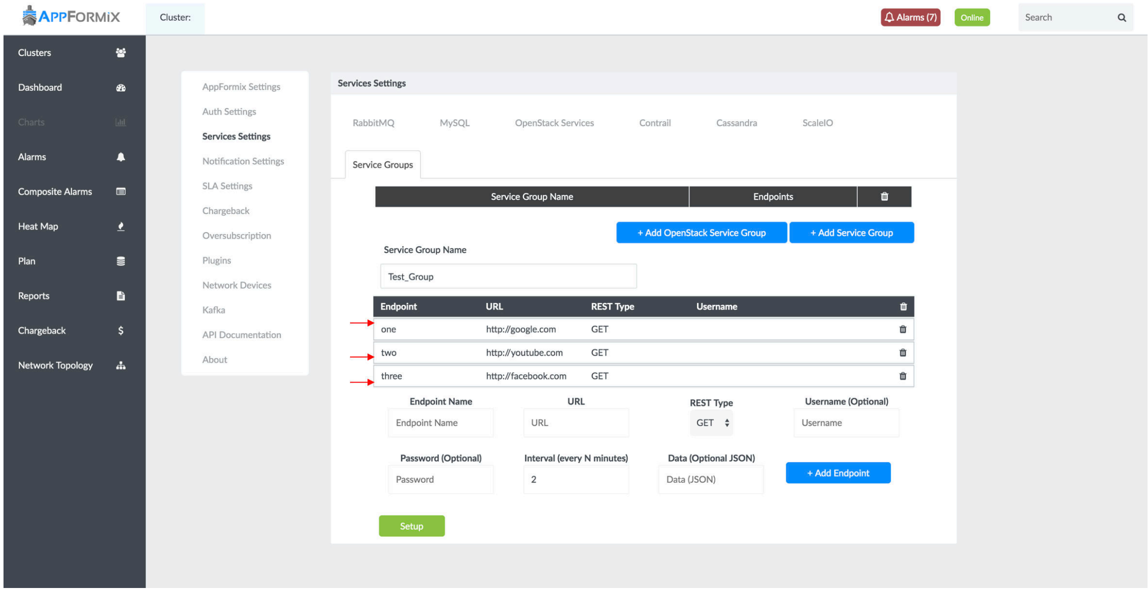
Password (Optional): Password

Interval (every N minutes): 2

Data (Optional JSON): Data (JSON)

c. To add more endpoints, click **Add Endpoint.** Following is an example where three endpoints are configured for one service group.

Figure 91: Add Three Endpoints for One Service Group



| Service Group Name | Endpoints |
|---|---|

Service Group Name: Test_Group

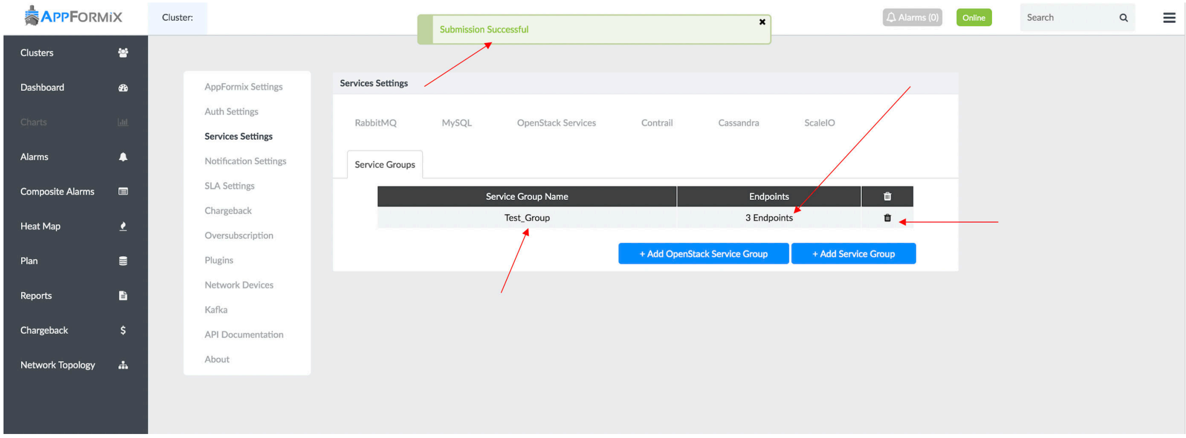
| Endpoint | URL | REST Type | Username |
|---|---|---|---|
| one | http://google.com | GET | |
| two | http://youtube.com | GET | |
| three | http://facebook.com | GET | |

Endpoint Name: Endpoint Name
URL: URL
REST Type: GET
Username (Optional): Username
Password (Optional): Password
Interval (every N minutes): 2
Data (Optional JSON): Data (JSON)

d. When done adding endpoints for this specific Service Group, click **Setup**.

5. The Service Group will show as successfully added. It can be deleted by clicking the Trash icon.

Figure 92: Add Service Group Successful



Submission Successful

| Service Group Name | Endpoints |
|---|---|
| Test_Group | 3 Endpoints |

## Add Service Groups Using Ansible

**Profile Overview**—In the directory **agent/tools/ansible/profiles/**, there are five profiles each pertaining to an OpenStack service. The prefix of each file is the name of the OpenStack service; either **cinder**,

**glance**, **keystone**, **neutron**, or **nova**. The suffix is ***_default_service_profile.json.j2**. For example, the profile for the OpenStack service Glance is named **glance_default_service_profile.json.j2**.

The default layout of the Glance profile is shown in the following example. The other profiles have an identical layout, just with the corresponding OpenStack service listed.

**Glance Profile Example**:

```
{
    "ServiceGroupName": "AppformixGlanceServiceGroup",
    "Protocol": "http_or_https",
    "Endpoints": [
        {
            "Url": "{{ glance_url }}",
            "EndpointName": "glanceEndpoint",
            "Method": "GET",
            "Interval": 2
        }
    ],
    "ServiceGroupId": "GlanceServiceGroupId",
    "RefreshTokenData": {
        "RefreshToken": "False",
        "Username": "admin",
        "AuthType": "openstack",
        "Password": "",
        "AuthUrl": "",
        "Project": ""
    }
}
```

**ICMP Profile Example**:

```
{
    "ServiceGroupId": "ICMP_service_group_id",
    "Protocol": "icmp_ping",
    "ServiceGroupName": "ICMP_test",
    "Endpoints": [{
        "EndpointName": "icmp",
        "Url": "127.0.0.1",
        "Interval": 2
    }]
}
```

**Add an HTTP Profile**—Profiles support unauthenticated and authenticated endpoints.

### Unauthenticated Endpoint

To add an unauthenticated endpoint:

1. Add the variable that the `Url` key is mapped to to your `group_vars/all` file.

2. Confirm this variable is mapped to a working endpoint.

   For example: In the `group_vars/all`, if you are using the Glance profile, add the `glance_url` variable as shown here:

   ```
   glance_url: "http://0.0.0.0:9292"
   ```

### Authenticated Endpoint

To add an endpoint that needs authentication, a RefreshToken is required. A RefreshToken enables access to endpoints that require authentication, as well as keeps that access by getting a new token when the current one is about to expire.

To obtain a refresh token:

1. Set the `RefreshToken` field in the `RefreshTokenData` dictionary to be `True`.

2. Then provide `Username`, `Password`, and `AuthUrl` in the same `RefreshTokenData`.

### GET and POST Examples for Refresh Token

**GET Example**:

```
{
    "ServiceGroupName": "AppformixGlanceServiceGroup",
    "Protocol": "http_or_https",
    "Endpoints": [
        {
            "Url": "{{ glance_url }}",
            "EndpointName": "glanceEndpoint",
            "Method": "GET",
            "Interval": 2
        }
    ],
```

```
    "ServiceGroupId": "GlanceServiceGroupId",
    "RefreshTokenData": {
        "RefreshToken": "True",
        "Username": "admin",
        "AuthType": "openstack",
        "Password": "password",
        "AuthUrl": "auth_url",
        "Project": ""
    }
}
```

**POST Example**:

```
{
    "ServiceGroupName": "AppformixGlanceServiceGroup",
    "Protocol": "http_or_https",
    "Endpoints": [
        {
            "Url": "{{ glance_url }}",
            "EndpointName": "glanceEndpoint",
            "Method": "POST",
            "Interval": 2,
            "Data": "{\"AuthType\":\"openstack\", \"UserName\": \"admin\", \"Password\":
\"password\"}"
        }
    ],
    "ServiceGroupId": "GlanceServiceGroupId",
    "RefreshTokenData": {
        "RefreshToken": "True",
        "Username": "admin",
        "AuthType": "openstack",
        "Password": "password",
        "AuthUrl": "auth_url",
        "Project": ""
    }
}
```

## Add a Profile or Multiple Profiles to Ansible

Using Ansible, a profile corresponding to a service group can be added to the Contrail Insights Dashboard during the installation.

To add a profile to the Contrail Insights Dashboard:

1. Add the variable `appformix_service_connectivity_profiles` to your `group_vars/all` file.

2. Map the variable to a list of dictionaries. Each dictionary in the list should only contain one key and one value.

   - The key should always be `connectivity_profiles`.

   - The value should be the path of the profile you want added to the Contrail Insights Dashboard during installation. An example follows:

     ```
     appformix_service_connectivity_profiles: [{ connectivity_profiles: 'profiles/
     glance_default_service_profile.json.j2' }]
     ```

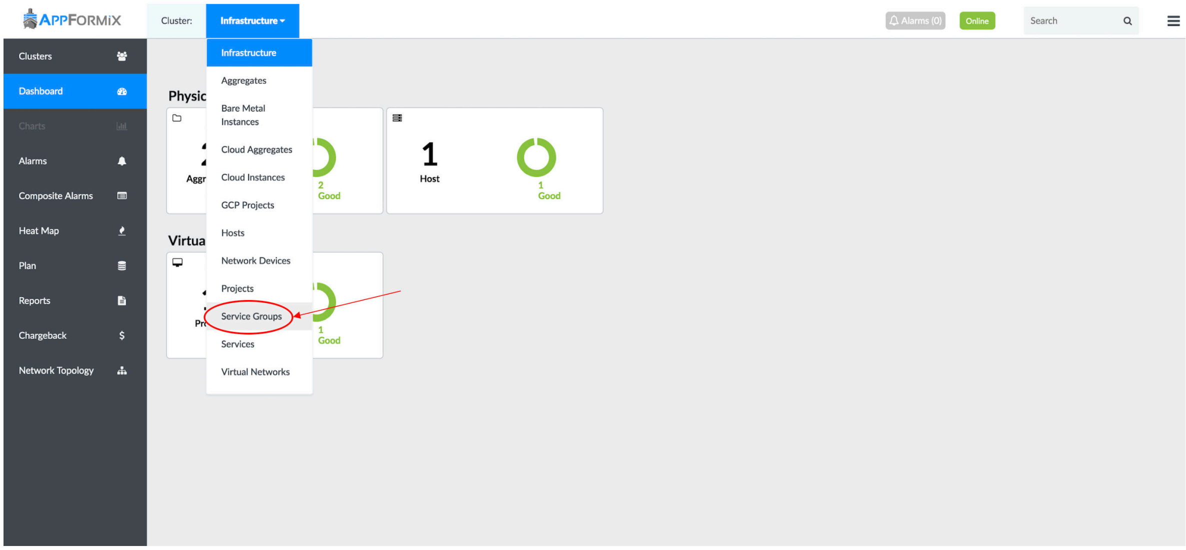
   To add multiple profiles, repeat these steps for as many profiles as needed.

## View Service Groups

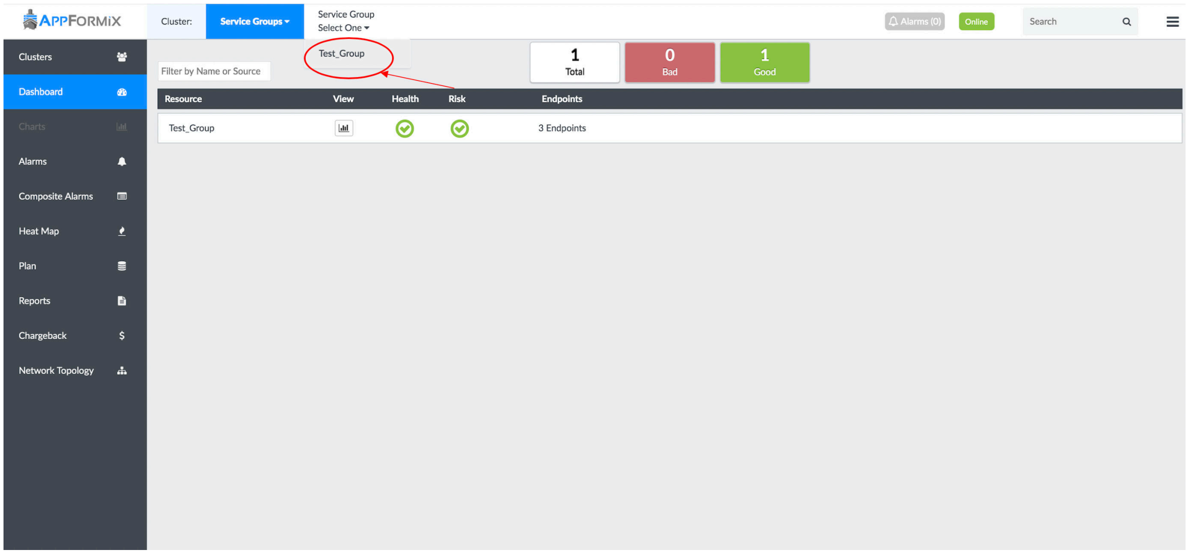
Successfully added service groups are viewable from the Dashboard.

To view service groups from the Dashboard:

1. Select **Infrastructure** > **Service Groups**.

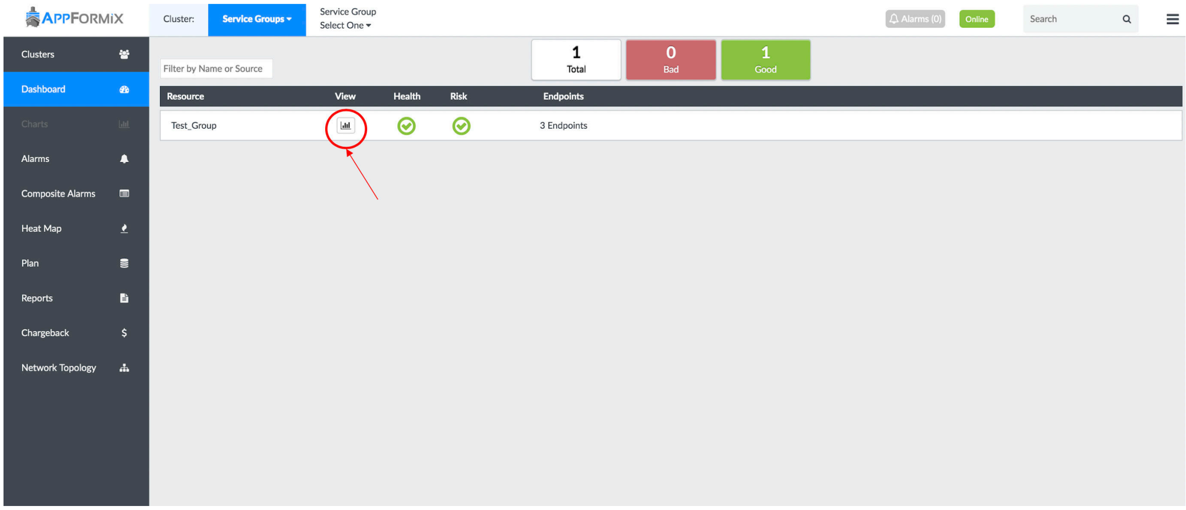


2. Select the Service Group you want to view.



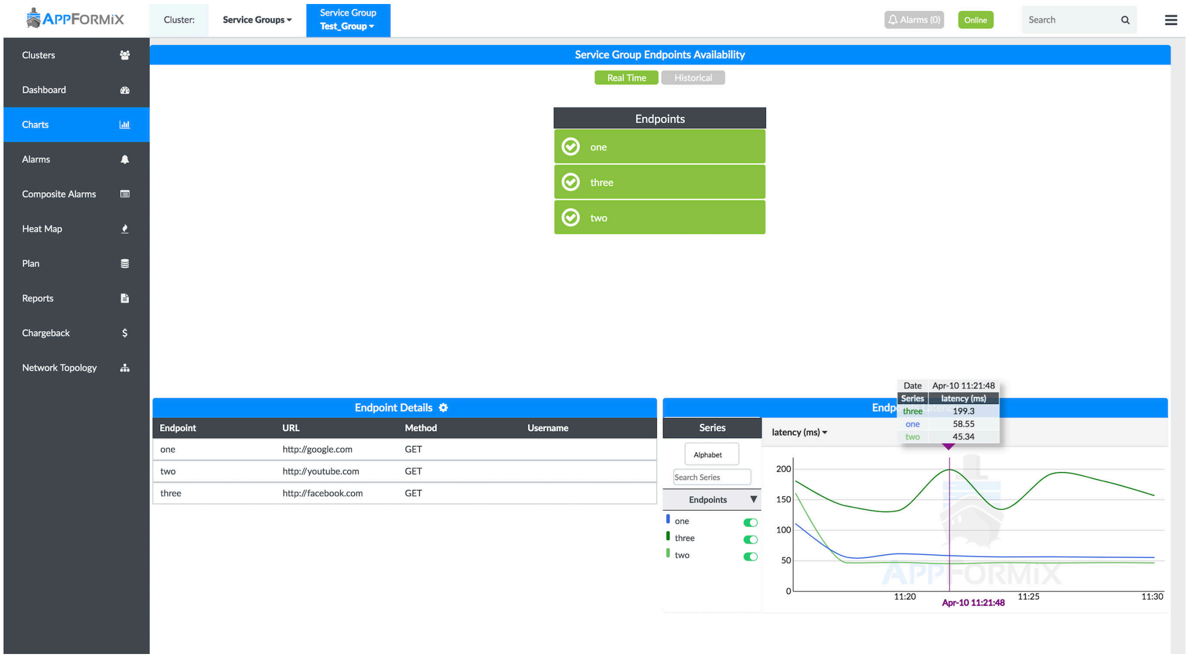
| Resource | View | Health | Risk | Endpoints |
|---|---|---|---|---|
| Test_Group | | ✓ | ✓ | 3 Endpoints |

3. Click **Charts** to view data being collected for this Service Group's endpoints.



| Resource | View | Health | Risk | Endpoints |
|---|---|---|---|---|
| Test_Group | | ✓ | ✓ | 3 Endpoints |

Select **Charts** to view the Charts display and endpoint details.



| Endpoint | URL | Method | Username |
|---|---|---|---|
| one | http://google.com | GET | |
| two | http://youtube.com | GET | |
| three | http://facebook.com | GET | |

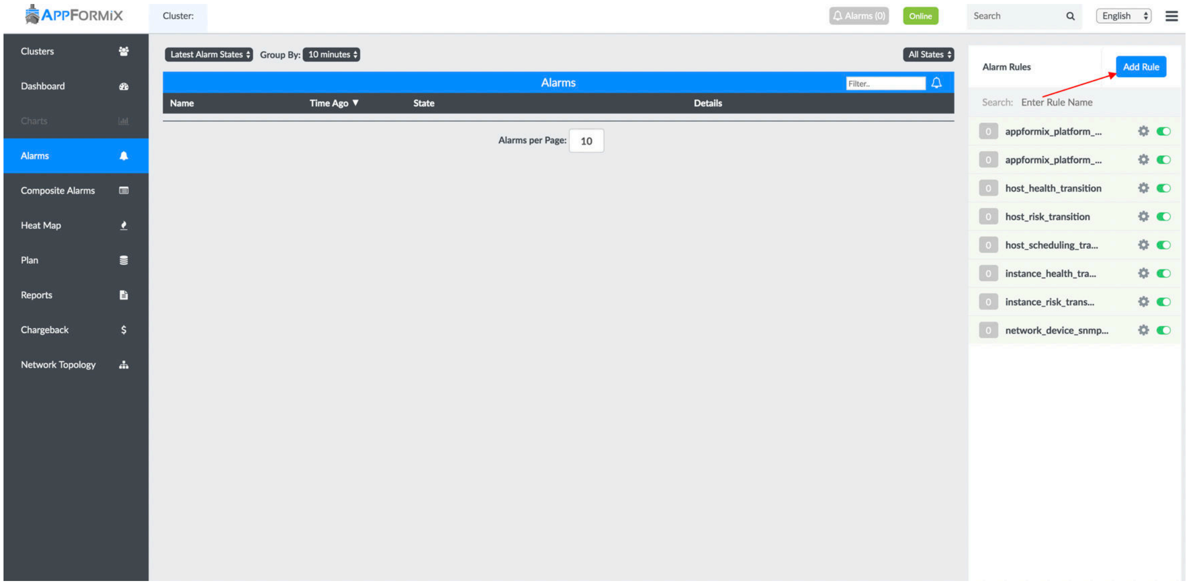
| Date | Apr-10 11:21:48 |
|---|---|
| **Series** | **latency (ms)** |
| three | 199.3 |
| one | 58.55 |
| two | 45.34 |

## Create Alarms for Service Groups

To create alarms for service groups:

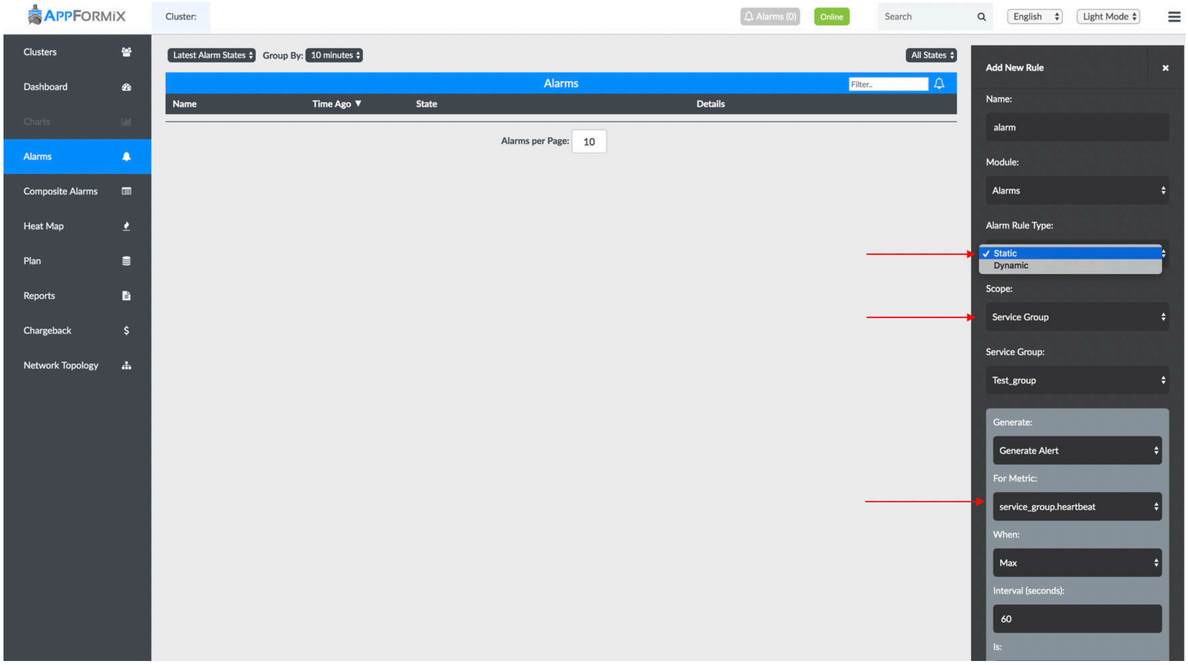
1. After a service group is created, navigate to the Alarms page and click **Add Rule**.

### Figure 93: Creating an Alarm for Service Group



2. For Scope, select **Service Group** and for Metric, select **service_group.heartbeat** (default). For Alarm Rule Type, both static and dynamic alarms are supported for service groups.

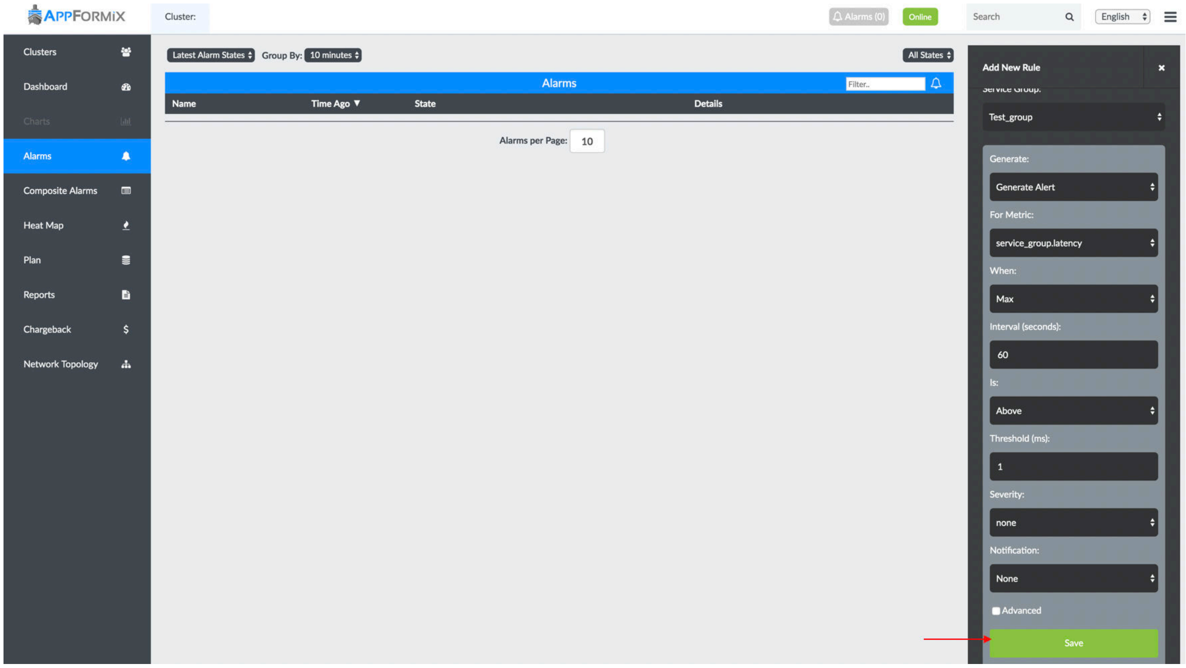
### Figure 94: Adding Alarm Rules for Service Group



Name: alarm
Module: Alarms
Alarm Rule Type: Static
Scope: Service Group
Service Group: Test_group
Generate: Generate Alert
For Metric: service_group.heartbeat
When: Max
Interval (seconds): 60
Is:

3. Complete any further details and click **Save** to confirm.

Figure 95: Saving Alarm Rules for Service Group



Service Group: Test_group

Generate: Generate Alert

For Metric: service_group.latency

When: Max

Interval (seconds): 60

Is: Above

Threshold (ms): 1

Severity: none

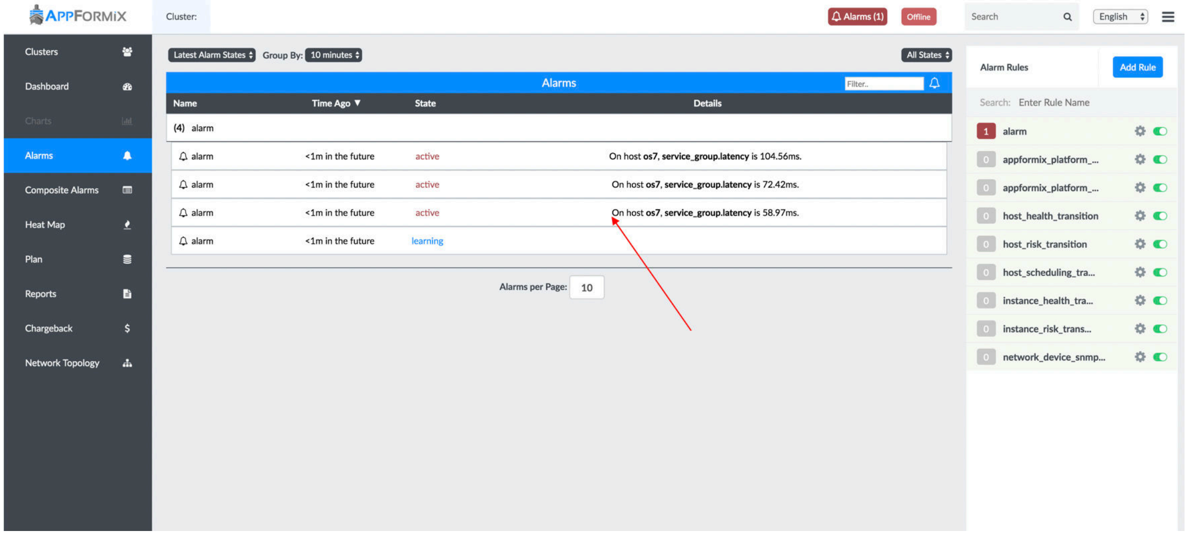
Notification: None

☐ Advanced

Save

4. After the alarms are triggered, they are visible on the Dashboard as active or inactive based on the rules set.

Figure 96: Service Group Triggered Alarm Visible on Dashboard



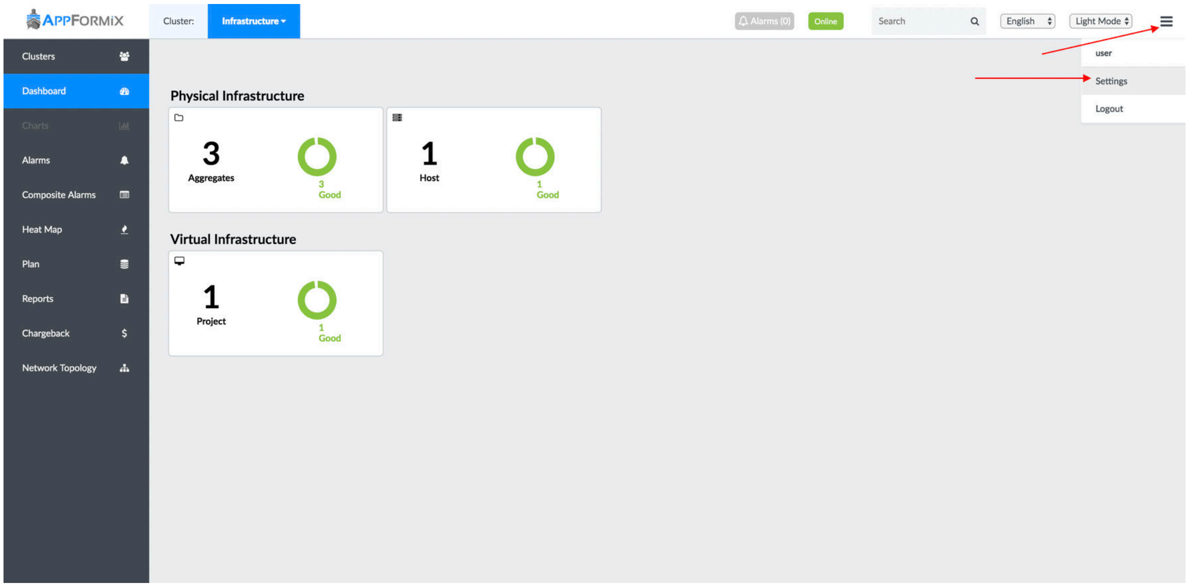
| Name | Time Ago ▼ | State | Details |
|---|---|---|---|
| (4) alarm | | | |
| alarm | <1m in the future | active | On host os7, service_group.latency is 104.56ms. |
| alarm | <1m in the future | active | On host os7, service_group.latency is 72.42ms. |
| alarm | <1m in the future | active | On host os7, service_group.latency is 58.97ms. |
| alarm | <1m in the future | learning | |

| | Alarm Rules |
|---|---|
| 1 | alarm |
| 0 | appformix_platform_... |
| 0 | appformix_platform_... |
| 0 | host_health_transition |
| 0 | host_risk_transition |
| 0 | host_scheduling_tra... |
| 0 | instance_health_tra... |
| 0 | instance_risk_trans... |
| 0 | network_device_snmp... |

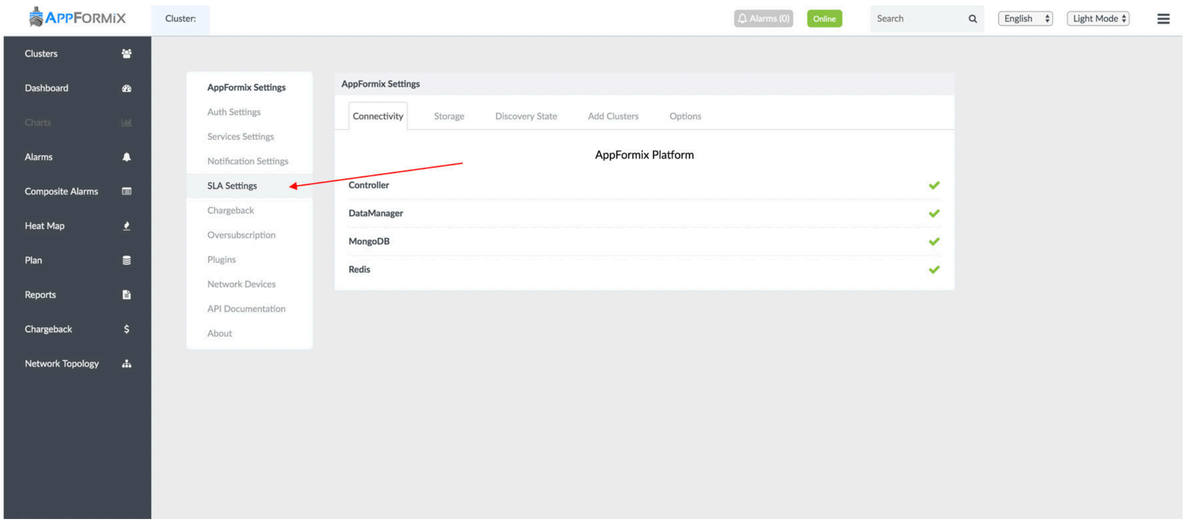
# Health and Risk SLA for Service Groups

To create health and risk service-level agreements (SLAs) for service groups:

1. Select Settings in the top right of the Dashboard.

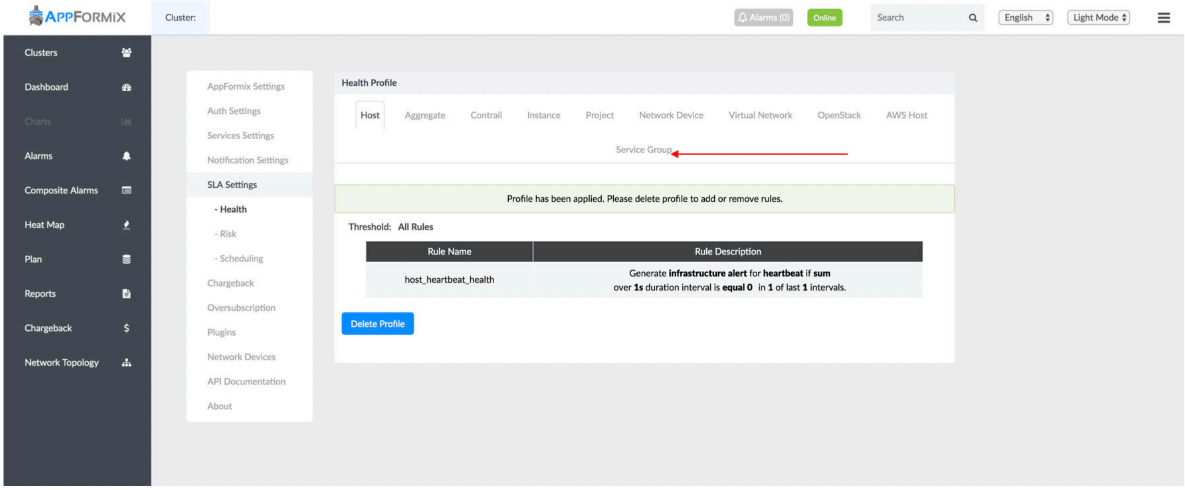


2. In Settings, select **SLA Settings**.



3. In Health Profile, click the **Service Group** tab.

### Figure 97: Health Profile Service Group Tab



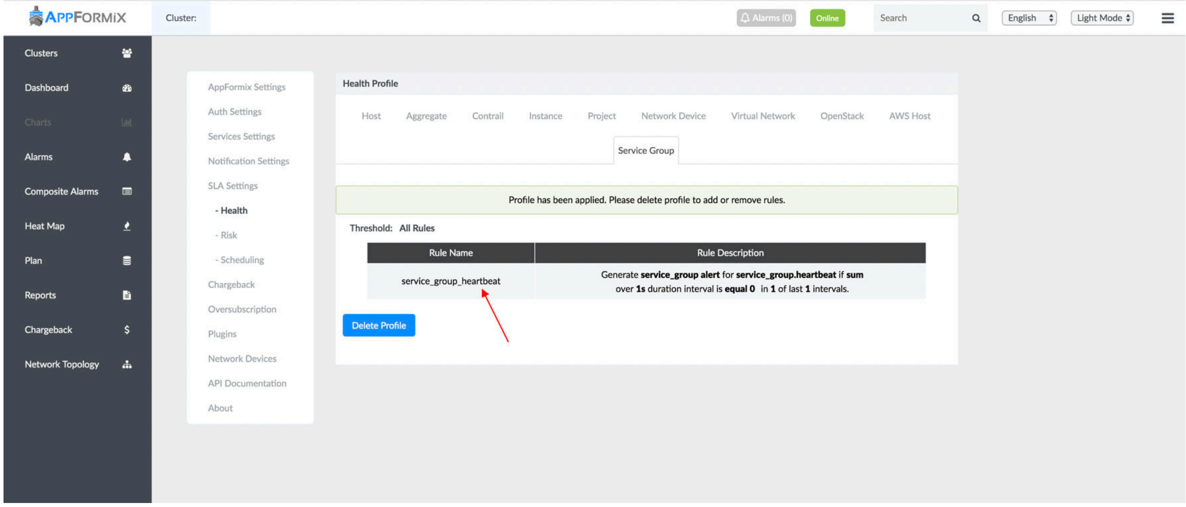
Profile has been applied. Please delete profile to add or remove rules.

Threshold: All Rules

| Rule Name | Rule Description |
|---|---|
| host_heartbeat_health | Generate **infrastructure alert** for **heartbeat** if **sum** over **1s** duration interval is **equal 0** in **1** of last **1** intervals. |

4. By default, Contrail Insights has a Health/Risk profile created for all the service groups. Click **Delete Profile** to add new profiles and set up a new SLA.

### Figure 98: Delete Profile in Service Groups to Add New Profile or New SLA



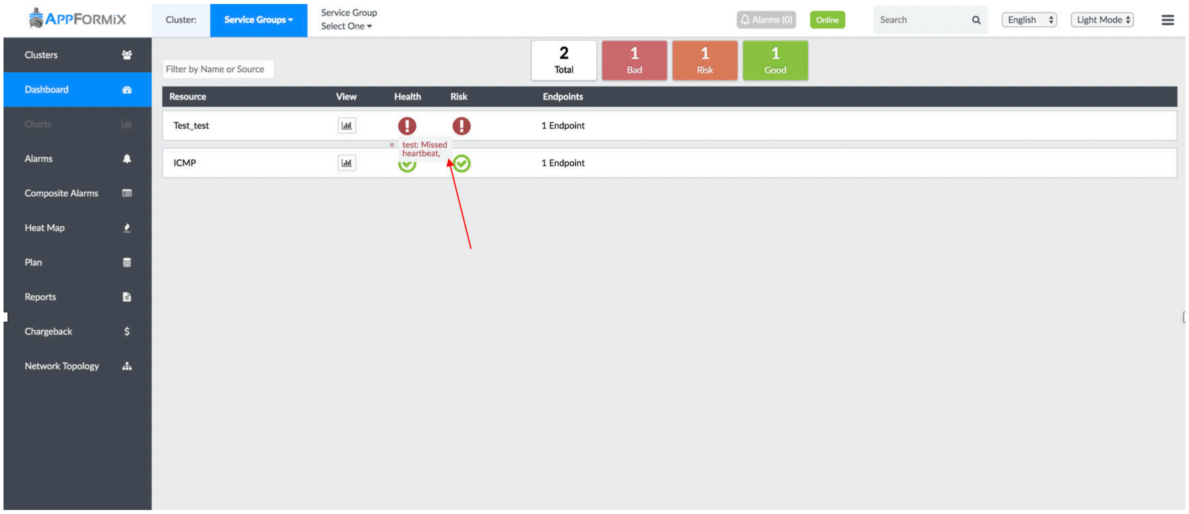
Profile has been applied. Please delete profile to add or remove rules.

Threshold: All Rules

| Rule Name | Rule Description |
|---|---|
| service_group_heartbeat | Generate **service_group alert** for **service_group.heartbeat** if **sum** over **1s** duration interval is **equal 0** in **1** of last **1** intervals. |

When a service group is in bad health, it is reflected on the service groups Dashboard view based on the profile.

Figure 99: Example Service Group Alert for Missed Heartbeat on Dashboard



| Resource | View | Health | Risk | Endpoints |
|---|---|---|---|---|
| Test_test |  | ! | ! | 1 Endpoint |
| ICMP |  | ✓ | ✓ | 1 Endpoint |

test: Missed heartbeat,

## Service Group Configuration Examples

Following are service group configuration examples.

**OpenStack Service Group configuration example**:

```
{
    "ServiceGroupName": "AppformixGlanceServiceGroup",
    "Protocol": "http_or_https",
    "Endpoints": [
        {
            "Url": "glance_url",
            "EndpointName": "glanceEndpoint",
            "Method": "GET",
            "Interval": 2
        }
    ],
    "ServiceGroupId": "GlanceServiceGroupId",
    "RefreshTokenData": {
        "RefreshToken": "False",
        "Username": "openstack_admin",
        "AuthType": "openstack",
        "Password": "openstack_password",
    }
}
```

**ICMP Service Group configuration example**:

```
{
    "ServiceGroupId": "ICMP_service_group_id",
    "Protocol": "icmp_ping",
    "ServiceGroupName": "ICMP_test",
    "Endpoints": [{
        "EndpointName": "icmp",
        "Url": "127.0.0.1",
        "Interval": 2
    }]
}
```

**Regular Service Group configuration example**:

```
{
    "ServiceGroupName": "ServiceGroup",
    "Protocol": "http_or_https",
    "Endpoints": [
        {
            "Url": "url",
            "EndpointName": "endpoint",
            "Method": "GET",
            "Interval": 2
        }
    ],
    "ServiceGroupId": "ServiceGroupId"
}
```

**Service Group Alarm configuration example**:

Service Group alarms have their own scope `service_group`.

```
{
    "Severity": "none",
    "IntervalDuration": "60s",
    "Module": "alarms",
    "ServiceGroupId": "Service_Group_Id",
    "ComputeMultipleBaselines": false,
    "IntervalCount": 1,
    "EventRuleType": "static",
    "IntervalsWithException": 1,
    "Name": "alarm_name",
    "LearningPeriodStart": 0,
    "ComparisonFunction": "above",
    "EventRuleScope": "service_group",
    "AggregationFunction": "max",
    "Sensitivity": "",
    "DisplayEvent": true,
    "MetricType": "service_group.heartbeat",
    "Threshold": 0,
    "Mode": "alert"
}
```

RELATED DOCUMENTATION

*OpenStack Services Monitoring Using Service Group Profiles*

*Service Monitoring from the UI*

# Service Monitoring from the UI

**IN THIS SECTION**

# Ceph Monitoring

IN THIS SECTION

Ceph is a unified, distributed storage system that provides object storage and block storage. Contrail Insights monitors Ceph performance, availability, and usage, with both charts and alarms.

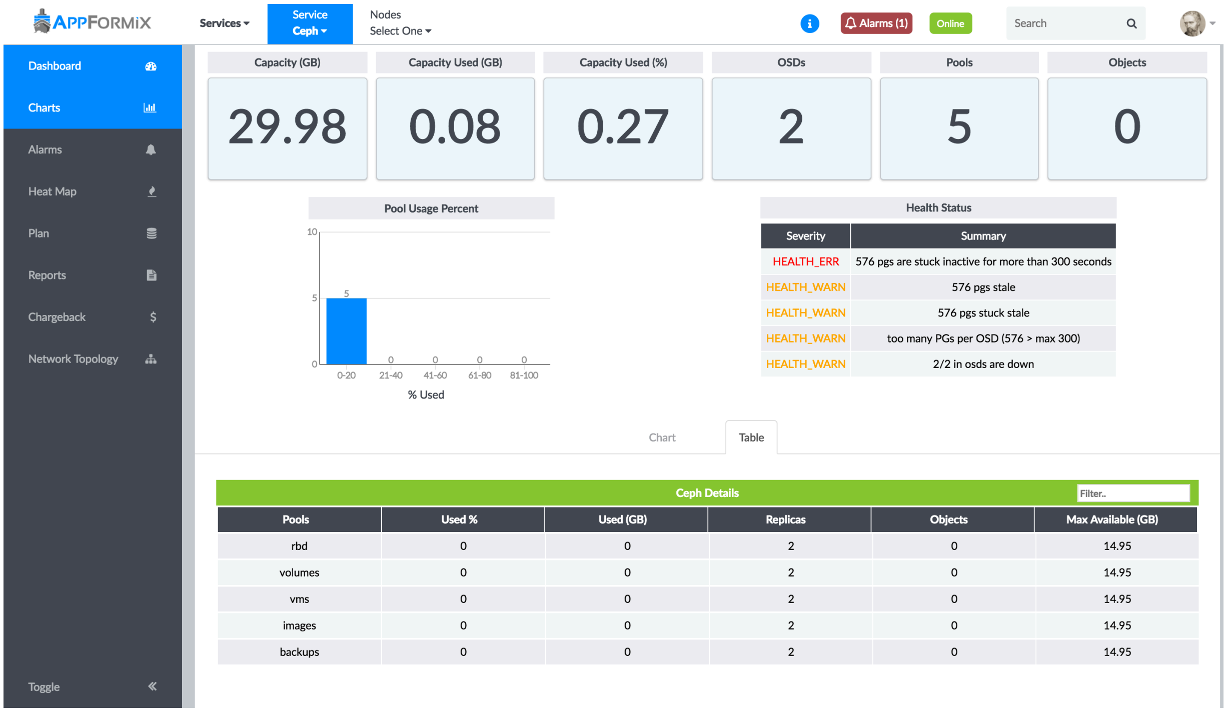
In addition, Contrail Insights Agent can be installed on the Ceph object storage daemon (OSD) and monitor hosts, for real-time health and performance monitoring of the storage hosts that power a Ceph storage cluster.

## Ceph Service Monitoring

From the context menu, select **Services > Ceph**. The Ceph service monitoring page displays a summary of the current usage of a Ceph cluster, including total cluster capacity, used capacity, and number of OSDs, pools, objects. The Health Status table displays errors and warnings of your Ceph cluster. Details about usage of each storage pool are shown in table and chart views.

Figure 100 on page 161 shows the Ceph service monitoring page and storage pool usage details in a table.

Figure 100: Ceph Service Summary of Current Usage of Ceph Cluster



| Capacity (GB) | Capacity Used (GB) | Capacity Used (%) | OSDs | Pools | Objects |
|---|---|---|---|---|---|
| 29.98 | 0.08 | 0.27 | 2 | 5 | 0 |

Health Status

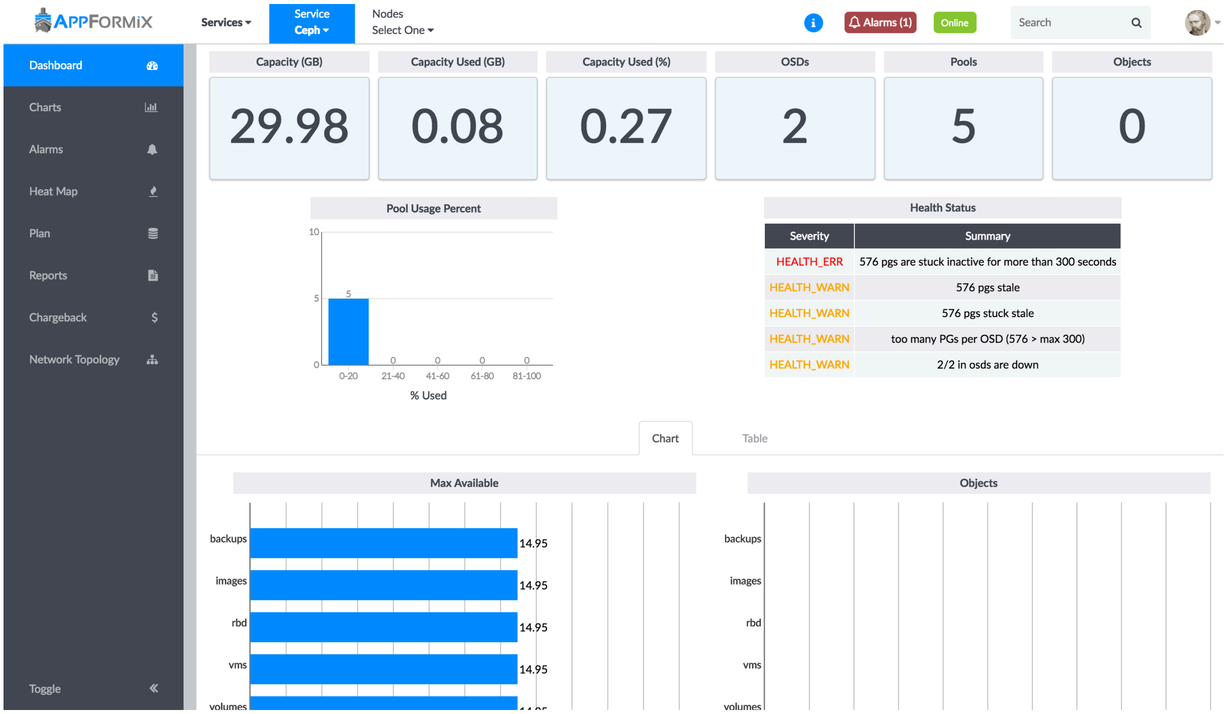
| Severity | Summary |
|---|---|
| HEALTH_ERR | 576 pgs are stuck inactive for more than 300 seconds |
| HEALTH_WARN | 576 pgs stale |
| HEALTH_WARN | 576 pgs stuck stale |
| HEALTH_WARN | too many PGs per OSD (576 > max 300) |
| HEALTH_WARN | 2/2 in osds are down |

Ceph Details

| Pools | Used % | Used (GB) | Replicas | Objects | Max Available (GB) |
|---|---|---|---|---|---|
| rbd | 0 | 0 | 2 | 0 | 14.95 |
| volumes | 0 | 0 | 2 | 0 | 14.95 |
| vms | 0 | 0 | 2 | 0 | 14.95 |
| images | 0 | 0 | 2 | 0 | 14.95 |
| backups | 0 | 0 | 2 | 0 | 14.95 |

Figure 101 on page 162 shows the Ceph service monitoring page and storage pool usage details in a chart.

Figure 101: Ceph Service Summary of Storage Pool Usage in Chart View



| Capacity (GB) | Capacity Used (GB) | Capacity Used (%) | OSDs | Pools | Objects |
|---|---|---|---|---|---|
| 29.98 | 0.08 | 0.27 | 2 | 5 | 0 |

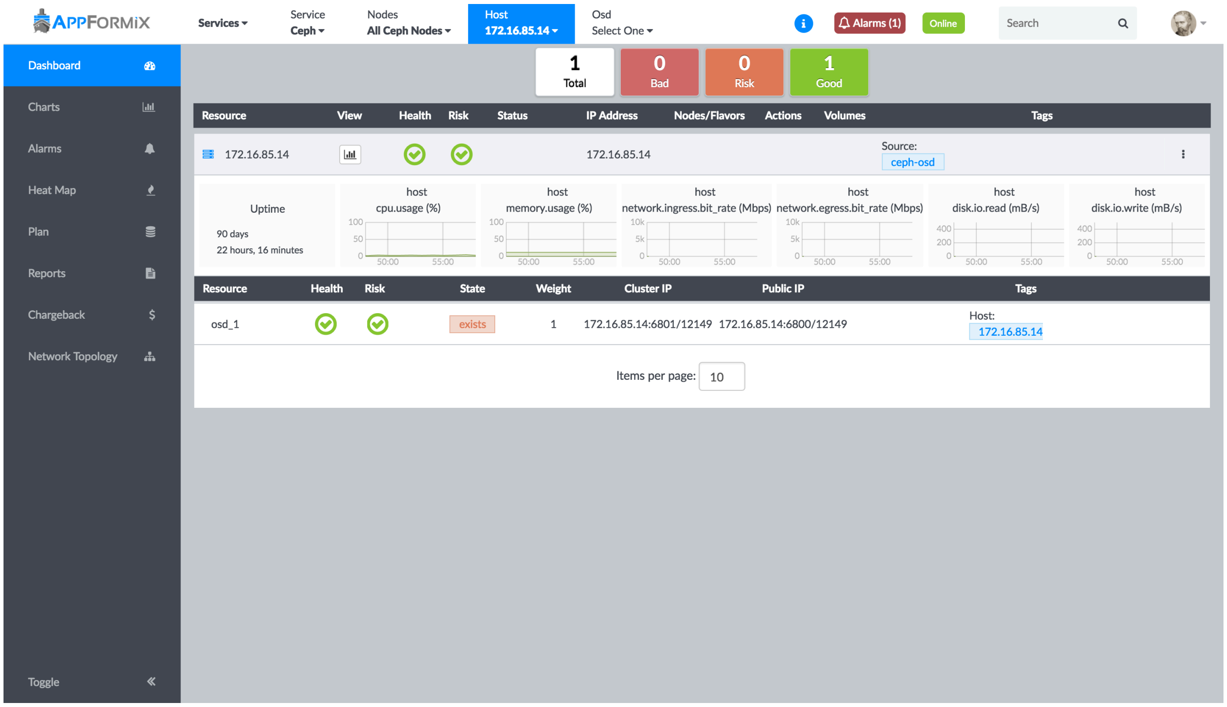
Health Status

| Severity | Summary |
|---|---|
| HEALTH_ERR | 576 pgs are stuck inactive for more than 300 seconds |
| HEALTH_WARN | 576 pgs stale |
| HEALTH_WARN | 576 pgs stuck stale |
| HEALTH_WARN | too many PGs per OSD (576 > max 300) |
| HEALTH_WARN | 2/2 in osds are down |

## Monitor Ceph OSD and Monitor Nodes

With Contrail Insights Agent installed on the Ceph storage hosts, details are available about each OSD and Monitor node in the cluster. Using the context menu, select **Services > Ceph > Nodes**. Each host in the list has a tag of `ceph-osd` or `ceph-monitor`. When a host with a `ceph-osd` tag is selected, a summary of host performance metrics are shown, as well as the health and status of each OSD on the host. See Figure 102 on page 163 for an example summary.

**Figure 102: Performance Metrics, Health, and Status for Each OSD on Host**



| Resource | View | Health | Risk | Status | IP Address | Nodes/Flavors | Actions | Volumes | Tags |
|---|---|---|---|---|---|---|---|---|---|
| 172.16.85.14 |  | ✓ | ✓ |  | 172.16.85.14 |  |  |  | Source:<br>ceph-osd |

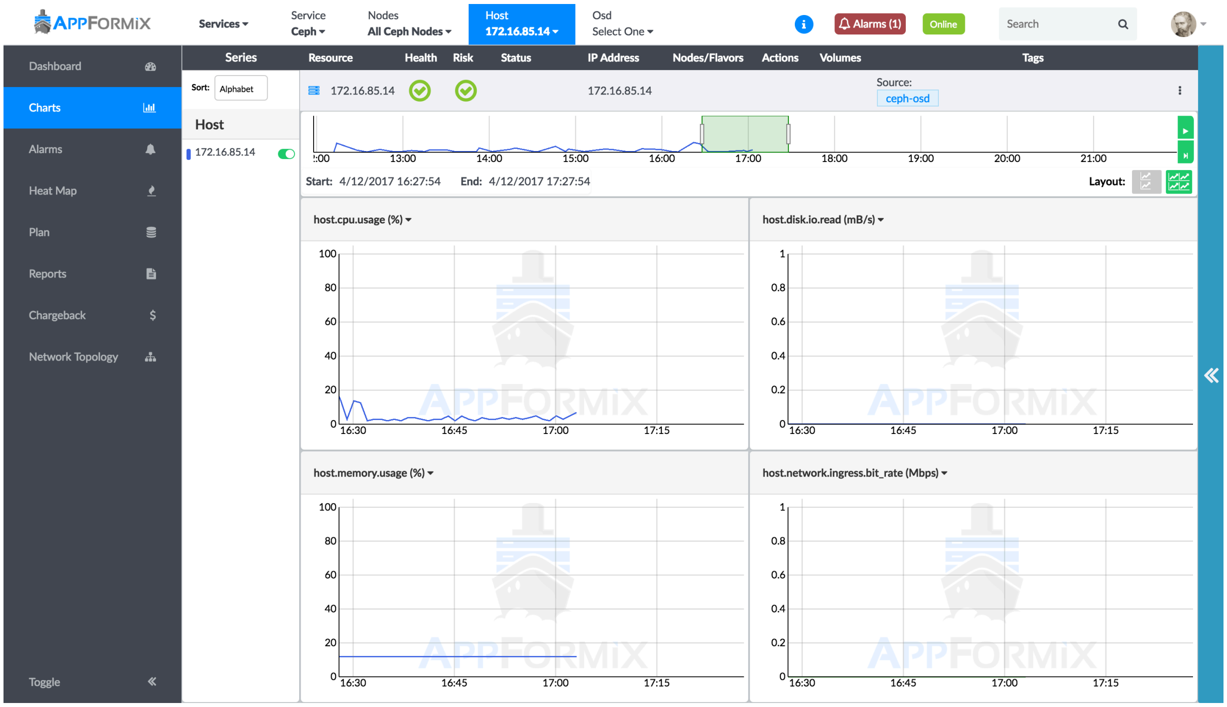
Uptime
90 days
22 hours, 16 minutes

| Resource | Health | Risk | State | Weight | Cluster IP | Public IP | Tags |
|---|---|---|---|---|---|---|---|
| osd_1 | ✓ | ✓ | exists | 1 | 172.16.85.14:6801/12149 | 172.16.85.14:6800/12149 | Host:<br>172.16.85.14 |

Items per page: 10

All of the Contrail Insights host monitoring functionality is available for the storage host, including *Charts* and *Alarms*. Navigate to Charts and Alarms in the left menu.

**Figure 103: Navigating to Host Chart View from Monitoring Nodes**



Sort: Alphabet

Start: 4/12/2017 16:27:54 End: 4/12/2017 17:27:54

## Service Alarms

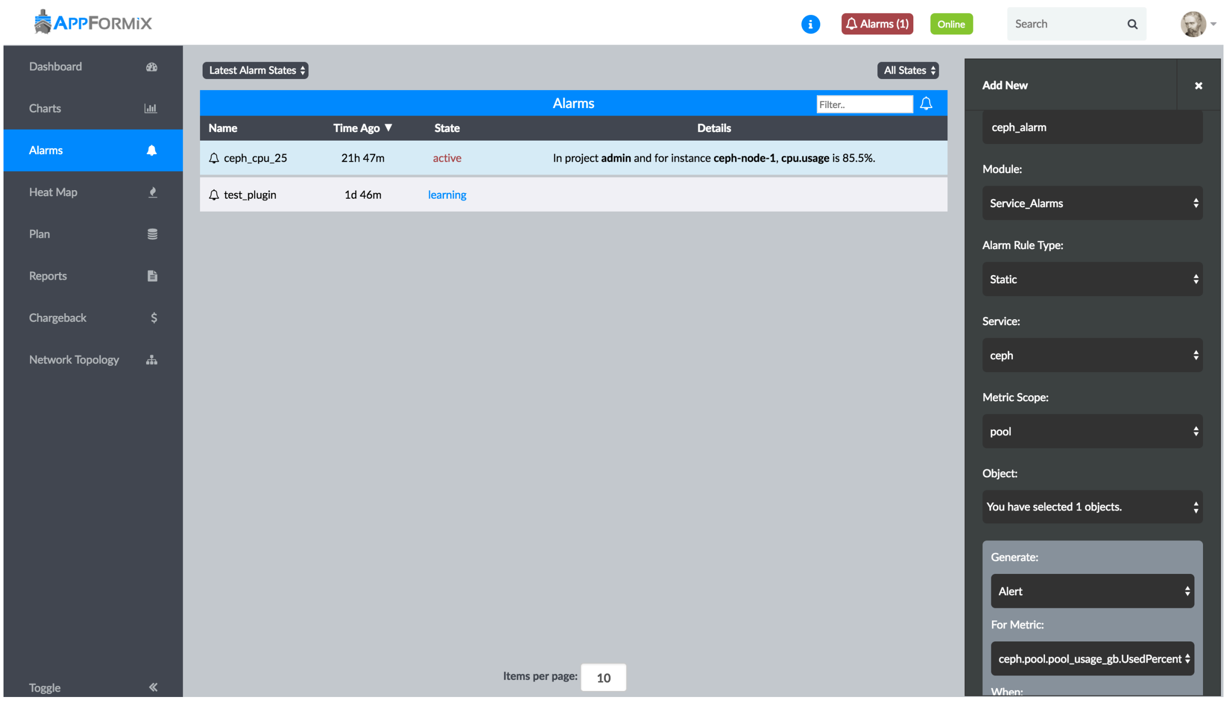
Alarms can be configured to monitor the Ceph cluster metrics at the cluster, pool, or host level.

To configure an alarm for cluster-wide and per-pool metrics, select **Alarms** in the left menu. Choose the **Service Alarms** module, and select **ceph** from the Service drop-down list. Ceph service alarms can be created to monitor a *cluster* or a *pool*. With cluster scope, an alarm can be configured for cluster-wide metrics, such as the cluster storage usage. With pool scope, an alarm can be configured to monitor per-pool metrics for one or multiple pools.

To configure an alarm for a Ceph storage host, select the **Alarms** module in the Alarms pane. An alarm can be configured for one or multiple Ceph storage hosts. See Configuring Alarms in *Alarms* for details.

As with all alarms in Contrail Insights, *Notifications* can be configured for Ceph alarms. Figure 104 on page 165 shows the alarm state for the Ceph cluster metrics.

**Figure 104: Alarm State for Ceph Cluster Metrics**



Alarms

| Name | Time Ago | State | Details |
|---|---|---|---|
| ceph_cpu_25 | 21h 47m | active | In project **admin** and for instance **ceph-node-1**, **cpu.usage** is 85.5%. |
| test_plugin | 1d 46m | learning | |

ceph_alarm

Module: Service_Alarms

Alarm Rule Type: Static

Service: ceph

Metric Scope: pool

Object: You have selected 1 objects.

Generate: Alert

For Metric: ceph.pool.pool_usage_gb.UsedPercent

When:

## Configuration

See *Service Monitoring Ansible Variables* for steps to configure Contrail Insights using Ansible to monitor a Ceph cluster.

# Contrail Monitoring

IN THIS SECTION

Contrail Networking is a software-defined networking (SDN) platform based on the open-source network virtualization project, OpenContrail. The Contrail Networking platform automates and orchestrates the creation of highly scalable virtual networks.

Contrail Insights provides monitoring and orchestration for the Contrail Service. See the *Service Monitoring Ansible Variables* instructions for how to configure Contrail monitoring.

## Service Monitoring Dashboard

Contrail Insights service monitoring Dashboard for a Contrail cluster displays the overall state of the cluster and its components.

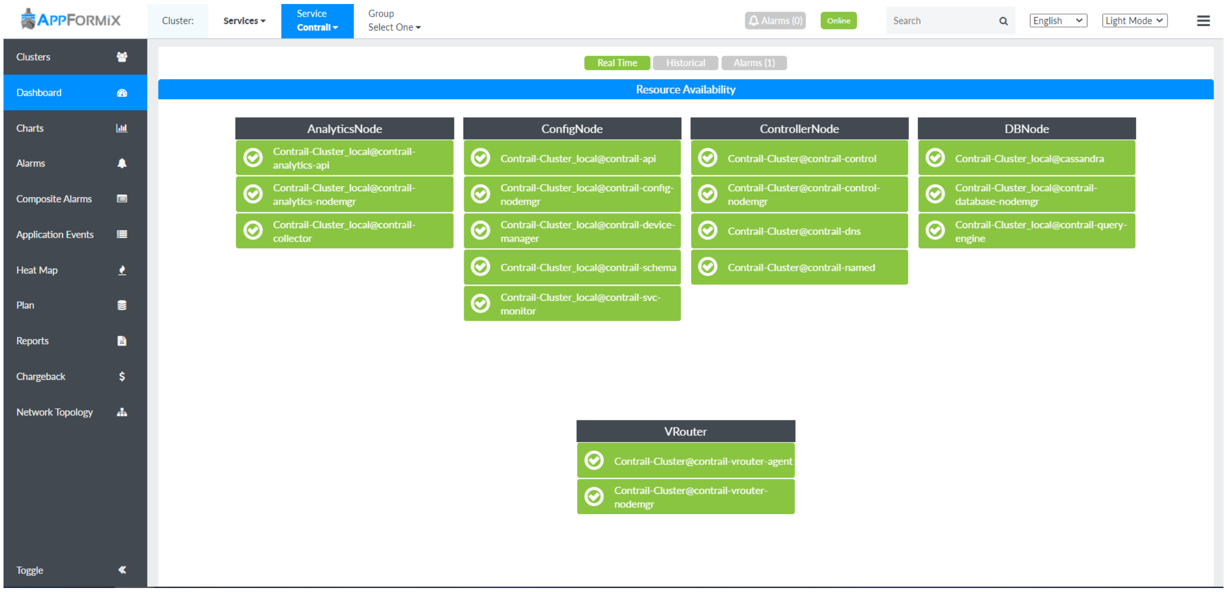
Contrail Insights provides real-time liveness for following five Contrail service groups.

- Analytics Nodes
- Config Nodes
- Controller Nodes
- DB Nodes
- vRouter

Figure 105 on page 167 shows real-time liveness for each Contrail service.

Starting with Contrail Insights Release 3.3.0, vRouter Contrail service group is also supported. These service groups run on all hosts that are configured during the Contrail Networking installation.

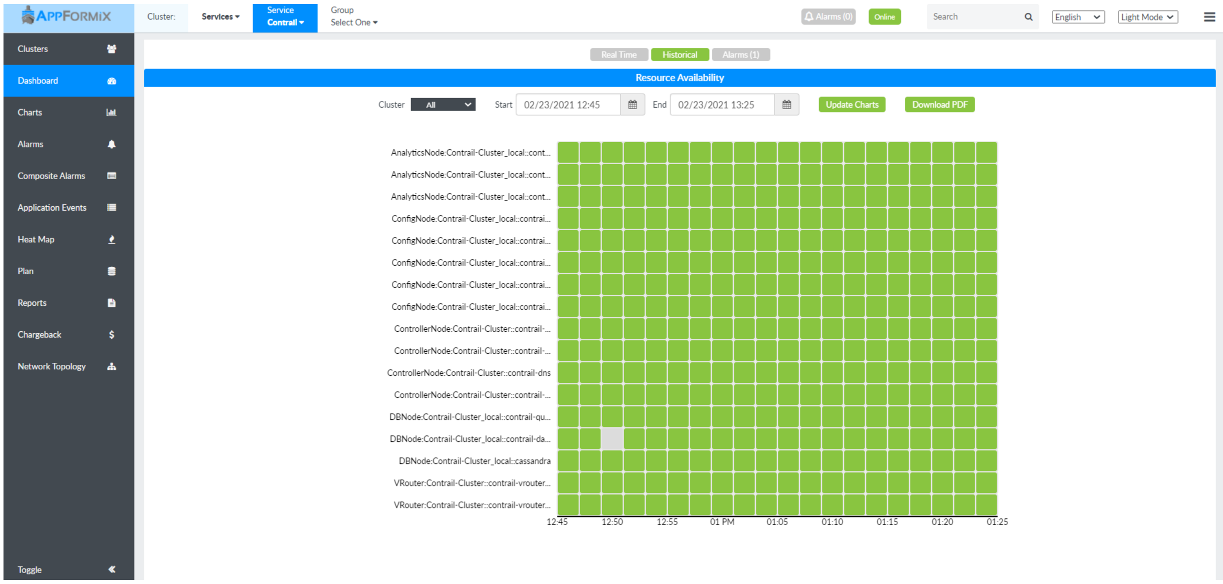
### Figure 105: Contrail Real-Time Liveness


APPFORMIX
Cluster:
Services
Service Contrail
Group Select One
Alarms (0)
Online
Search
English
Light Mode
Clusters
Dashboard
Charts
Alarms
Composite Alarms
Application Events
Heat Map
Plan
Reports
Chargeback
Network Topology
Toggle
Real Time
Historical
Alarms (1)
Resource Availability
AnalyticsNode
Contrail-Cluster_local@contrail-analytics-api
Contrail-Cluster_local@contrail-analytics-nodemgr
Contrail-Cluster_local@contrail-collector
ConfigNode
Contrail-Cluster_local@contrail-api
Contrail-Cluster_local@contrail-config-nodemgr
Contrail-Cluster_local@contrail-device-manager
Contrail-Cluster_local@contrail-schema
Contrail-Cluster_local@contrail-svc-monitor
ControllerNode
Contrail-Cluster@contrail-control
Contrail-Cluster@contrail-control-nodemgr
Contrail-Cluster@contrail-dns
Contrail-Cluster@contrail-named
DBNode
Contrail-Cluster_local@cassandra
Contrail-Cluster_local@contrail-database-nodemgr
Contrail-Cluster_local@contrail-query-engine
VRouter
Contrail-Cluster@contrail-vrouter-agent
Contrail-Cluster@contrail-vrouter-nodemgr

Contrail Insights also provides a historical liveness view of each Contrail service.

Figure 106 on page 167 show a historical liveness view.

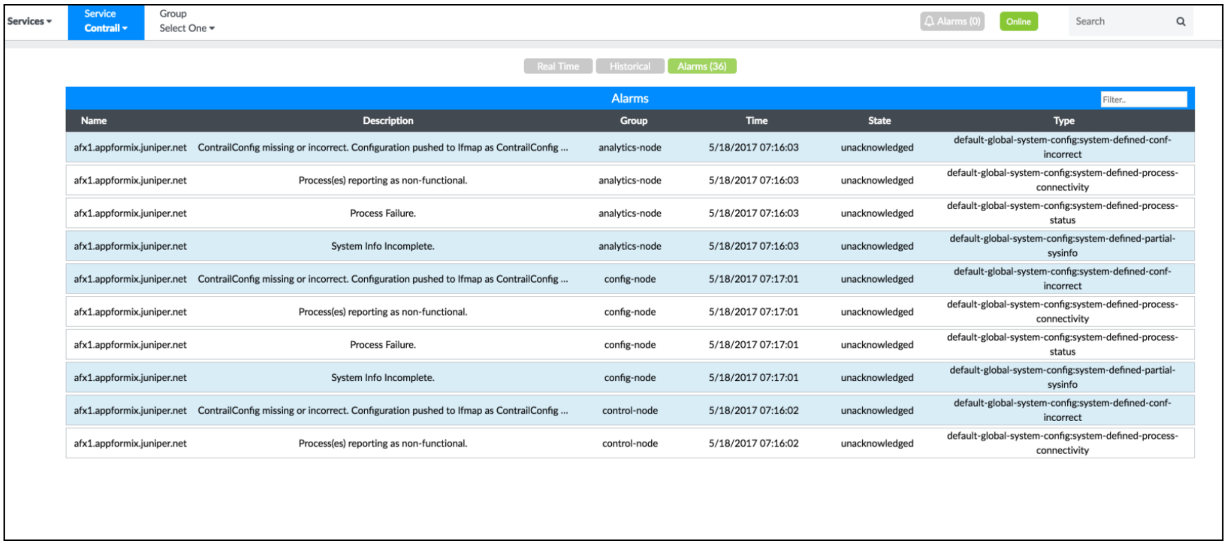
### Figure 106: Contrail Historical Liveness


APPFORMIX
Cluster:
Services
Service Contrail
Group Select One
Alarms (0)
Online
Search
English
Light Mode
Clusters
Dashboard
Charts
Alarms
Composite Alarms
Application Events
Heat Map
Plan
Reports
Chargeback
Network Topology
Toggle
Real Time
Historical
Alarms (1)
Resource Availability
Cluster
All
Start
02/23/2021 12:45
End
02/23/2021 13:25
Update Charts
Download PDF
AnalyticsNode:Contrail-Cluster_local::cont...
AnalyticsNode:Contrail-Cluster_local::cont...
AnalyticsNode:Contrail-Cluster_local::cont...
ConfigNode:Contrail-Cluster_local::contrai...
ConfigNode:Contrail-Cluster_local::contrai...
ConfigNode:Contrail-Cluster_local::contrai...
ConfigNode:Contrail-Cluster_local::contrai...
ConfigNode:Contrail-Cluster_local::contrai...
ControllerNode:Contrail-Cluster::contrail-...
ControllerNode:Contrail-Cluster::contrail-...
ControllerNode:Contrail-Cluster::contrail-dns
ControllerNode:Contrail-Cluster::contrail-...
DBNode:Contrail-Cluster_local::contrail-qu...
DBNode:Contrail-Cluster_local::contrail-da...
DBNode:Contrail-Cluster_local::cassandra
VRouter:Contrail-Cluster::contrail-vrouter...
VRouter:Contrail-Cluster::contrail-vrouter...
12:45
12:50
12:55
01 PM
01:05
01:10
01:15
01:20
01:25

In addition, any alarm generated by the Contrail Service can also be accessed from the Contrail Insights Dashboard.

Figure 107 on page 168 shows examples of Contrail service alarms.

### Figure 107: Contrail Service Alarms

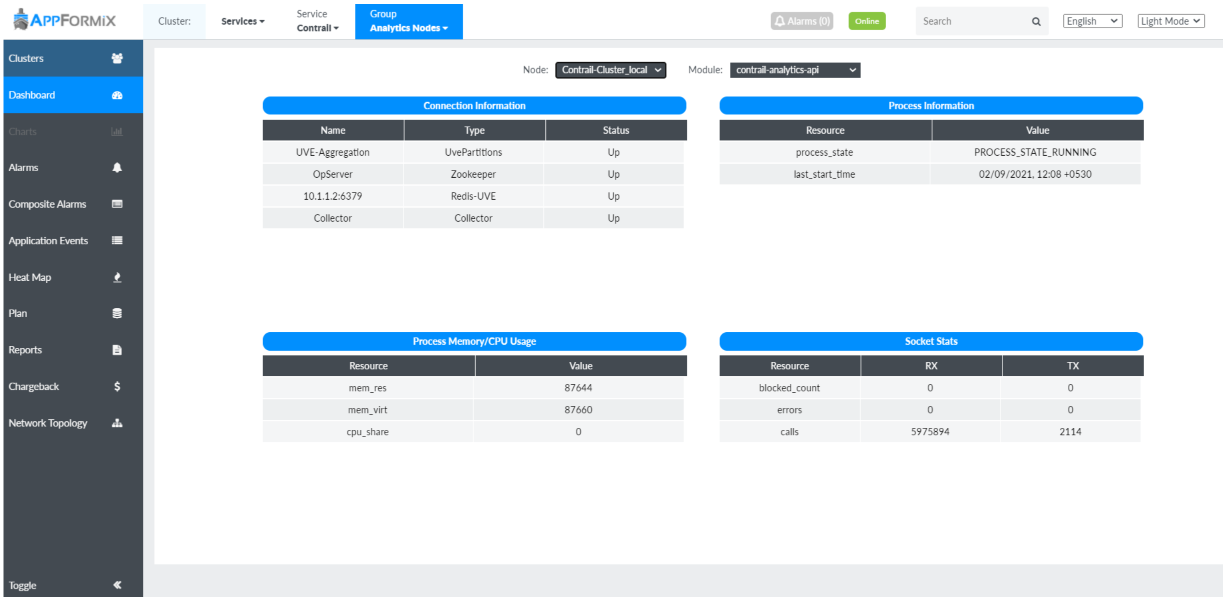


Alarms

| Name | Description | Group | Time | State | Type |
|---|---|---|---|---|---|
| afx1.appformix.juniper.net | ContrailConfig missing or incorrect. Configuration pushed to Ifmap as ContrailConfig ... | analytics-node | 5/18/2017 07:16:03 | unacknowledged | default-global-system-config:system-defined-conf-incorrect |
| afx1.appformix.juniper.net | Process(es) reporting as non-functional. | analytics-node | 5/18/2017 07:16:03 | unacknowledged | default-global-system-config:system-defined-process-connectivity |
| afx1.appformix.juniper.net | Process Failure. | analytics-node | 5/18/2017 07:16:03 | unacknowledged | default-global-system-config:system-defined-process-status |
| afx1.appformix.juniper.net | System Info Incomplete. | analytics-node | 5/18/2017 07:16:03 | unacknowledged | default-global-system-config:system-defined-partial-sysinfo |
| afx1.appformix.juniper.net | ContrailConfig missing or incorrect. Configuration pushed to Ifmap as ContrailConfig ... | config-node | 5/18/2017 07:17:01 | unacknowledged | default-global-system-config:system-defined-conf-incorrect |
| afx1.appformix.juniper.net | Process(es) reporting as non-functional. | config-node | 5/18/2017 07:17:01 | unacknowledged | default-global-system-config:system-defined-process-connectivity |
| afx1.appformix.juniper.net | Process Failure. | config-node | 5/18/2017 07:17:01 | unacknowledged | default-global-system-config:system-defined-process-status |
| afx1.appformix.juniper.net | System Info Incomplete. | config-node | 5/18/2017 07:17:01 | unacknowledged | default-global-system-config:system-defined-partial-sysinfo |
| afx1.appformix.juniper.net | ContrailConfig missing or incorrect. Configuration pushed to Ifmap as ContrailConfig ... | control-node | 5/18/2017 07:16:02 | unacknowledged | default-global-system-config:system-defined-conf-incorrect |
| afx1.appformix.juniper.net | Process(es) reporting as non-functional. | control-node | 5/18/2017 07:16:02 | unacknowledged | default-global-system-config:system-defined-process-connectivity |

Contrail Insights monitors the real-time status of every element of the Contrail cluster. You can select an element from the **Group** list for the Contrail service. For example, if you select **Analytics Nodes** service group, the Dashboard displays each service on every host that is configured for that particular service group. Liveness statistics and basic metrics are also available for each service in this view. Figure 108 on page 168 shows statistics and metrics for the Contrail analytics nodes.

### Figure 108: Contrail Service Analytics Nodes Statistics



Node: Contrail-Cluster_local
Module: contrail-analytics-api

Connection Information

| Name | Type | Status |
|---|---|---|
| UVE-Aggregation | UvePartitions | Up |
| OpServer | Zookeeper | Up |
| 10.1.1.2:6379 | Redis-UVE | Up |
| Collector | Collector | Up |

Process Information

| Resource | Value |
|---|---|
| process_state | PROCESS_STATE_RUNNING |
| last_start_time | 02/09/2021, 12:08 +0530 |

Process Memory/CPU Usage

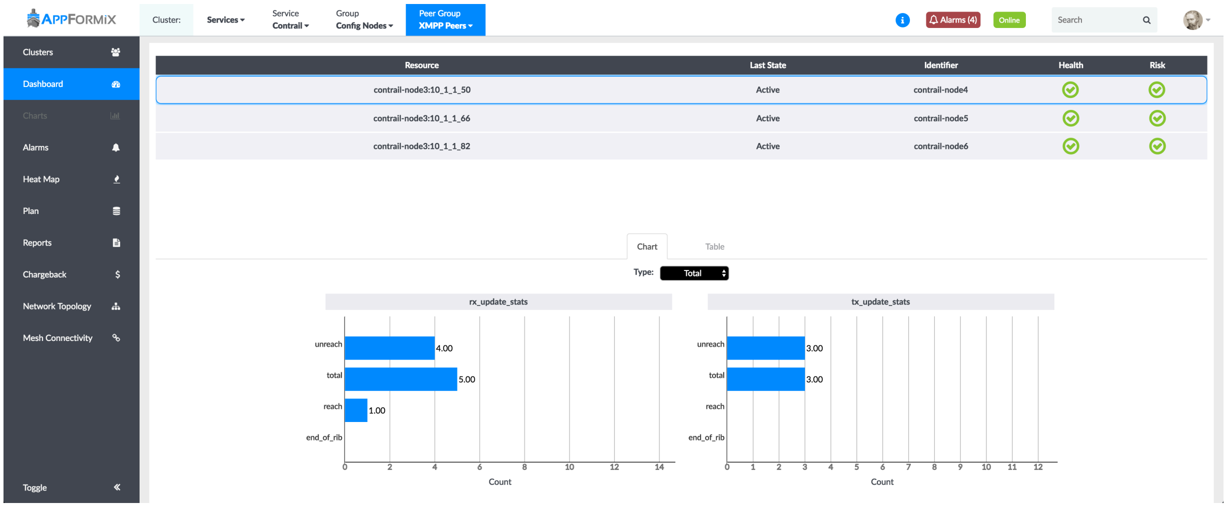
| Resource | Value |
|---|---|
| mem_res | 87644 |
| mem_virt | 87660 |
| cpu_share | 0 |

Socket Stats

| Resource | RX | TX |
|---|---|---|
| blocked_count | 0 | 0 |
| errors | 0 | 0 |
| calls | 5975894 | 2114 |

For Contrail **Config Nodes**, Contrail Insights enables a **Peer** view for **XMPP** and **BGP** peers. The information provides some *rx* and *tx* reachability statistics, as shown in Figure 109 on page 169.

Figure 109: Contrail Service XMPP Peers



| Resource | Last State | Identifier | Health | Risk |
|---|---|---|---|---|
| contrail-node3:10_1_1_50 | Active | contrail-node4 | ✓ | ✓ |
| contrail-node3:10_1_1_66 | Active | contrail-node5 | ✓ | ✓ |
| contrail-node3:10_1_1_82 | Active | contrail-node6 | ✓ | ✓ |

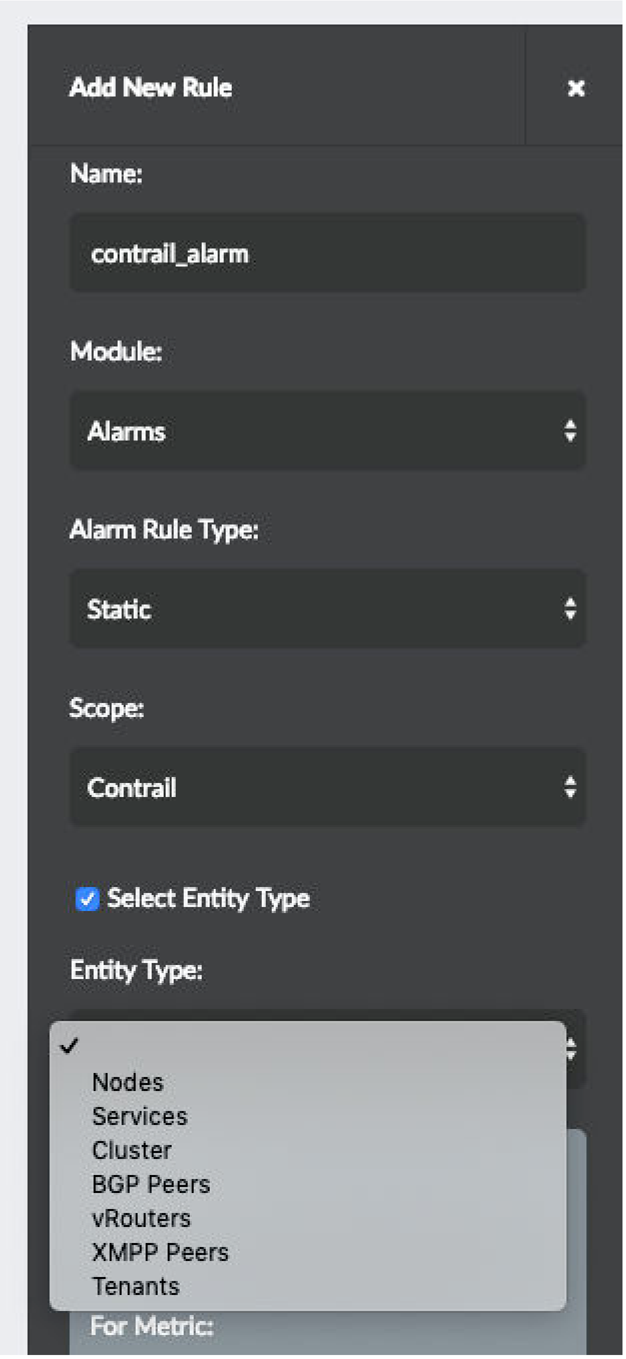
Type: Total

## Configuring Alarms

An alarm can be configured for any of the Contrail metrics collected. In the Alarm panel, select the **Alarms** module. Then select **Contrail** from the **Scope** list. Additionally, notifications can also be configured for Contrail alarms. Figure 110 on page 170 shows the Alarm pane for configuring Contrail alarms. For more information, see *Alarms* and *Notifications*.

**NOTE**: **Entity Type** and **Entity Names** are mandatory fields.

**Figure 110: Alarm Pane for Configuring Contrail Service Alarms**



Add New Rule

Name: contrail_alarm

Module: Alarms

Alarm Rule Type: Static

Scope: Contrail

☑ Select Entity Type

Entity Type:

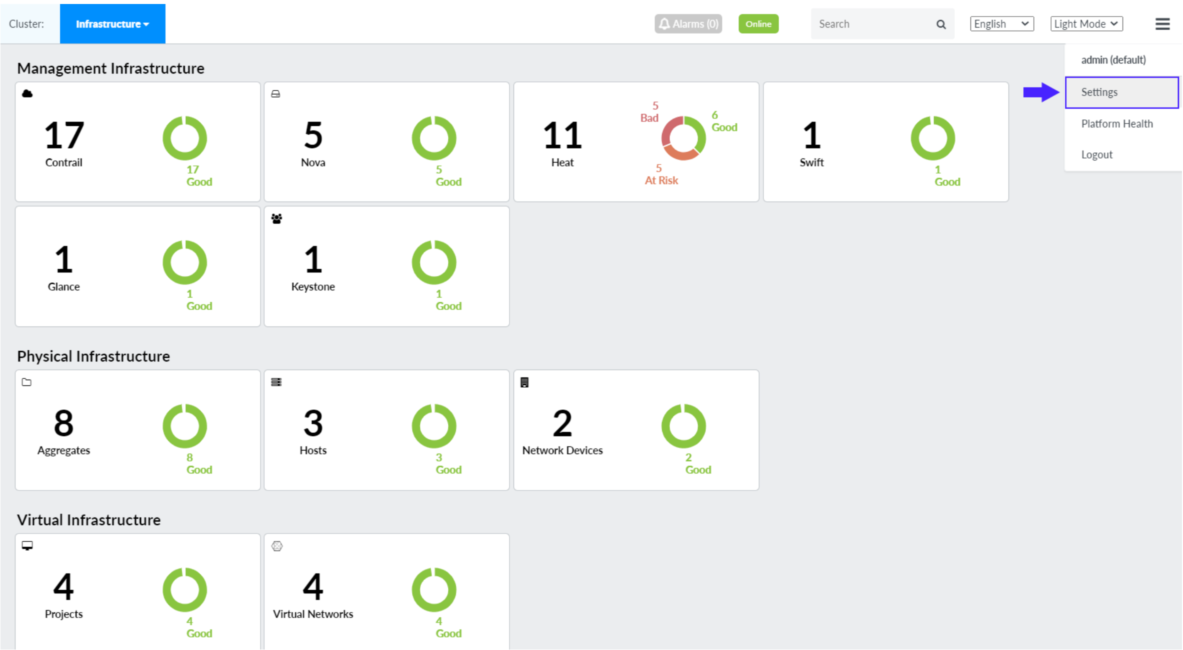
## Setting Health and Risk Rules for Contrail BGP Peers and XMPP Peers

In addition to Health and Risk rules that are preconfigured by Contrail Insights, you can set Health and Risk rules for two additional modules by following these steps:

1. Select **Settings** from the Dashboard as shown in Figure 111 on page 171.

   The AppFormix Settings page is displayed.

   **Figure 111: Select Settings from the Dashboard**

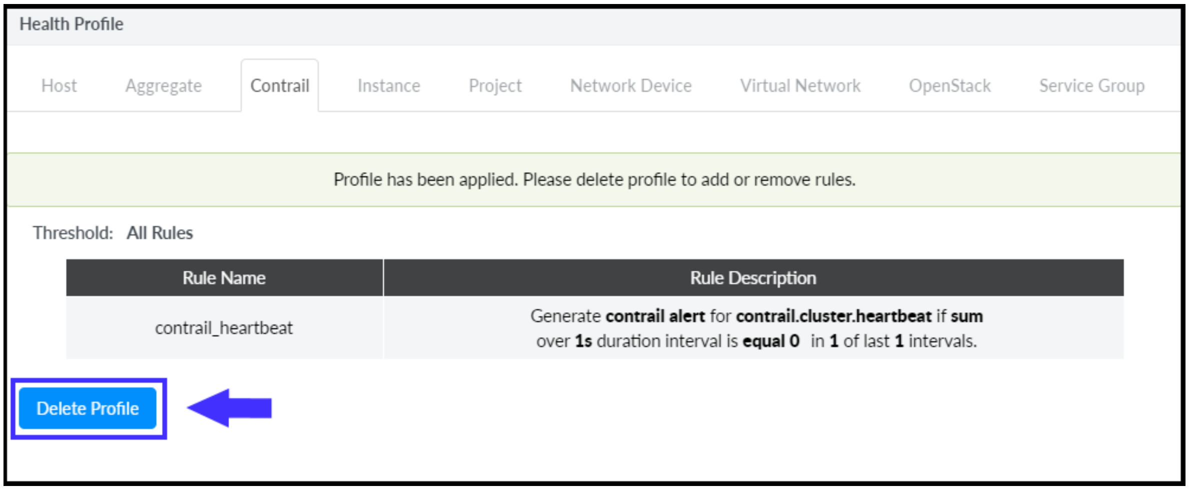
   

2. Click **SLA Settings** and then click **Health** or **Risk**.

   Existing rules if any are displayed in the Contrail tab.

3. To apply a new rule, delete the existing rule by clicking **Delete Profile** as shown in Figure 112 on page 172.

**Figure 112: Delete Existing Rule**



Profile has been applied. Please delete profile to add or remove rules.

Threshold: All Rules

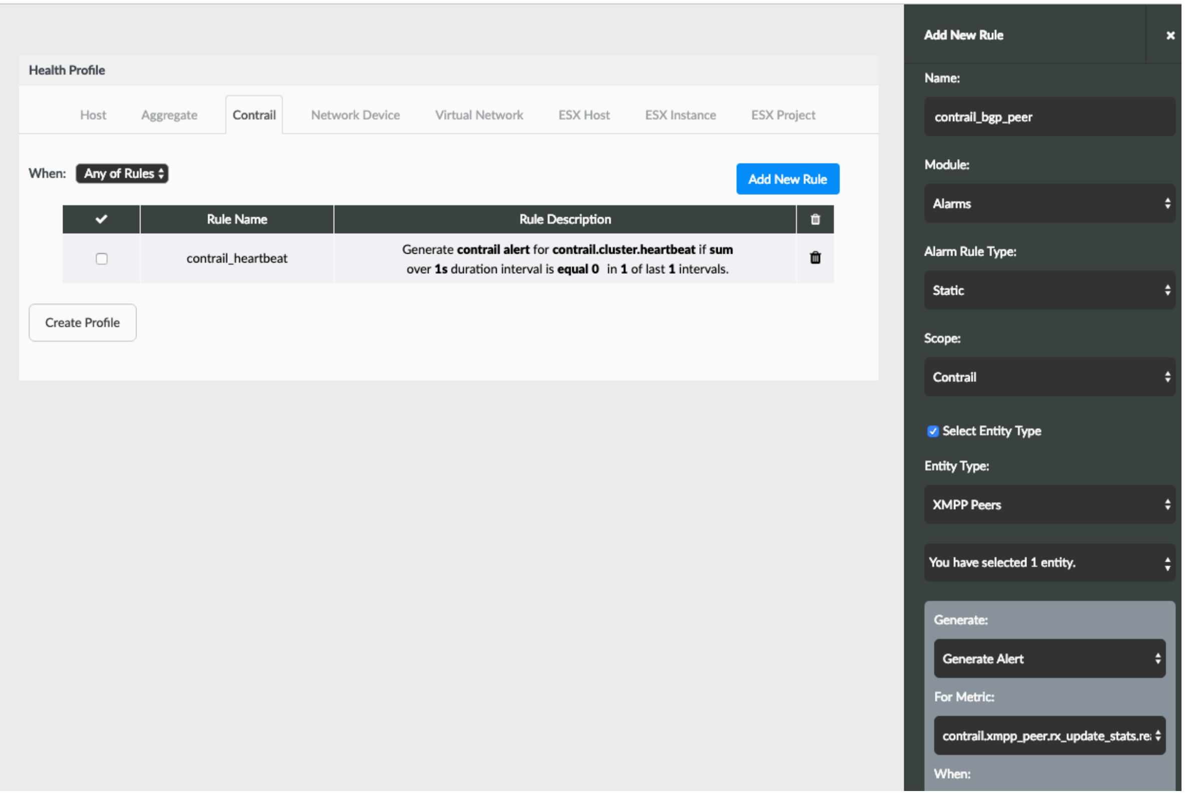
| Rule Name | Rule Description |
|---|---|
| contrail_heartbeat | Generate **contrail alert** for **contrail.cluster.heartbeat** if **sum** over **1s** duration interval is **equal 0** in **1** of last **1** intervals. |

4. After you have deleted the existing rule, click **Add New Rule**. See Figure 113 on page 173.

   The Add New Rule pane is displayed.

5. From the Entity Type list in the Add New Rule pane, select **BGP Peers** or **XMPP Peers**. See Figure 113 on page 173.

Figure 113: Setting Health or Risk Rules for Contrail Services



When: Any of Rules

| ✓ | Rule Name | Rule Description | 🗑 |
|---|---|---|---|
| □ | contrail_heartbeat | Generate **contrail alert** for **contrail.cluster.heartbeat** if **sum** over **1s** duration interval is **equal 0** in **1** of last **1** intervals. | 🗑 |

Name: contrail_bgp_peer

Module: Alarms

Alarm Rule Type: Static

Scope: Contrail

☑ Select Entity Type

Entity Type: XMPP Peers

You have selected 1 entity.

Generate: Generate Alert

For Metric: contrail.xmpp_peer.rx_update_stats.re:

When:

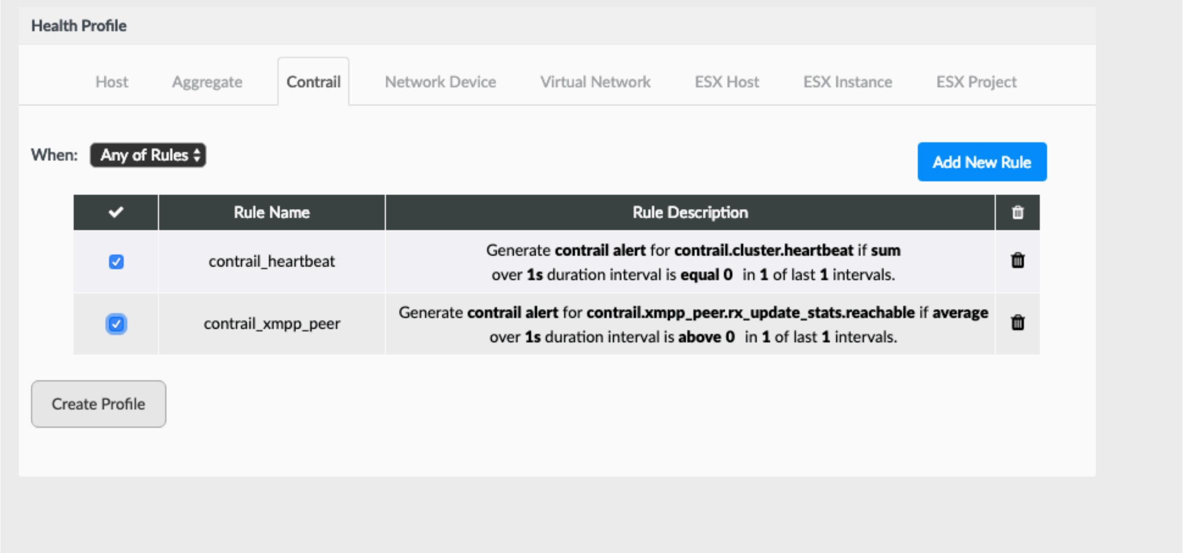
6. Click **Save**.

   The new rule is also added to the table as shown in Figure 114 on page 174.

7. Then select **SLA Settings** > **Health** or **Risk** > **Contrail** tab.

   Select both rules by selecting the check box next to the Rule Name as shown in Figure 114 on page 174, and then click **Create Profile**.

**Figure 114: Creating Health Profile for Contrail XMPP Peer**



| ✓ | Rule Name | Rule Description | 🗑 |
|---|---|---|---|
| ☑ | contrail_heartbeat | Generate **contrail alert** for **contrail.cluster.heartbeat** if **sum** over **1s** duration interval is **equal 0** in **1** of last **1** intervals. | 🗑 |
| ☑ | contrail_xmpp_peer | Generate **contrail alert** for **contrail.xmpp_peer.rx_update_stats.reachable** if **average** over **1s** duration interval is **above 0** in **1** of last **1** intervals. | 🗑 |

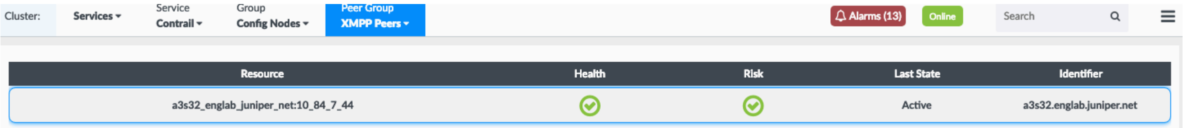
8. (Optional) You can also view XMPP and BGP peer resource and health information from the Contrail Insights Dashboard.

   For example, to view XMPP Peer resource and health information, click **Dashboard** and select **Services** from the context menu.

   From the Service list, select **Contrail**, and select **Config Nodes** from the Group list.

   Finally, from the Peer Group list, select **XMPP Peers** to view XMPP peer resource and health information. See Figure 115 on page 174.

**Figure 115: Viewing XMPP Peer Resource and Health**



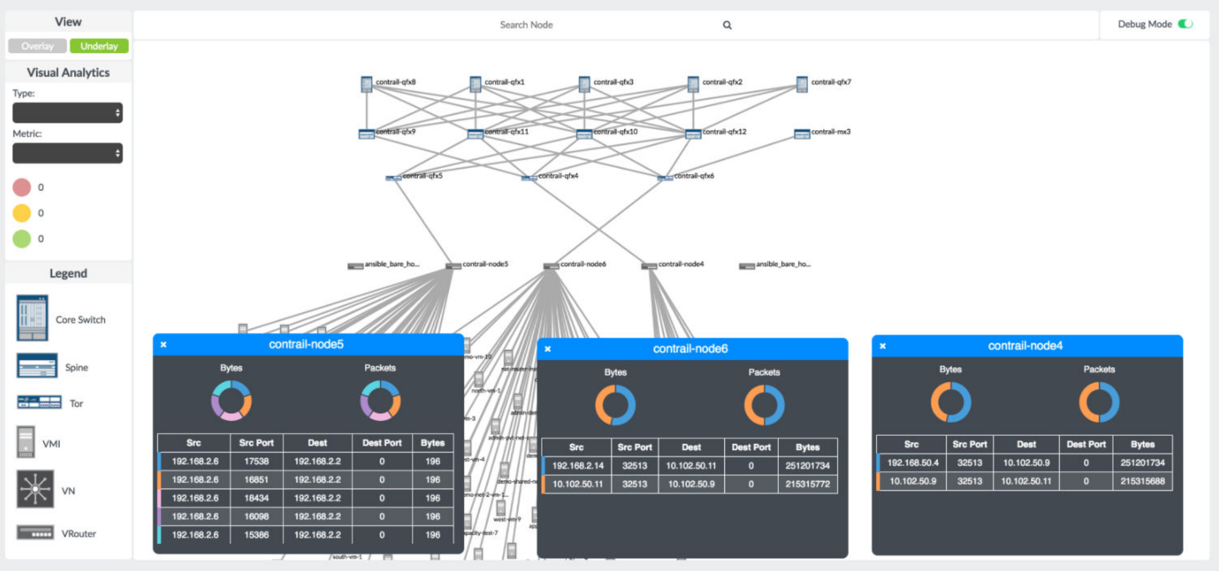
| Resource | Health | Risk | Last State | Identifier |
|---|---|---|---|---|
| a3s32_englab_juniper_net:10_84_7_44 | ✓ | ✓ | Active | a3s32.englab.juniper.net |

## Flow Monitoring with Contrail vRouter

When the Contrail vRouter is installed on a compute node, Contrail Insights provides debug mode functionality in the Network Topology panel.

In this mode, the top flows on each compute node are available for visualization with details on flow tuples, packets, and bytes. Figure 116 on page 175 shows the flow monitoring details and visualization.

**Figure 116: Flow Monitoring with Contrail vRouter**



contrail-node5

| Src | Src Port | Dest | Dest Port | Bytes |
|---|---|---|---|---|
| 192.168.2.6 | 17538 | 192.168.2.2 | 0 | 196 |
| 192.168.2.6 | 16851 | 192.168.2.2 | 0 | 196 |
| 192.168.2.6 | 18434 | 192.168.2.2 | 0 | 196 |
| 192.168.2.6 | 16098 | 192.168.2.2 | 0 | 196 |
| 192.168.2.6 | 15386 | 192.168.2.2 | 0 | 196 |

contrail-node6

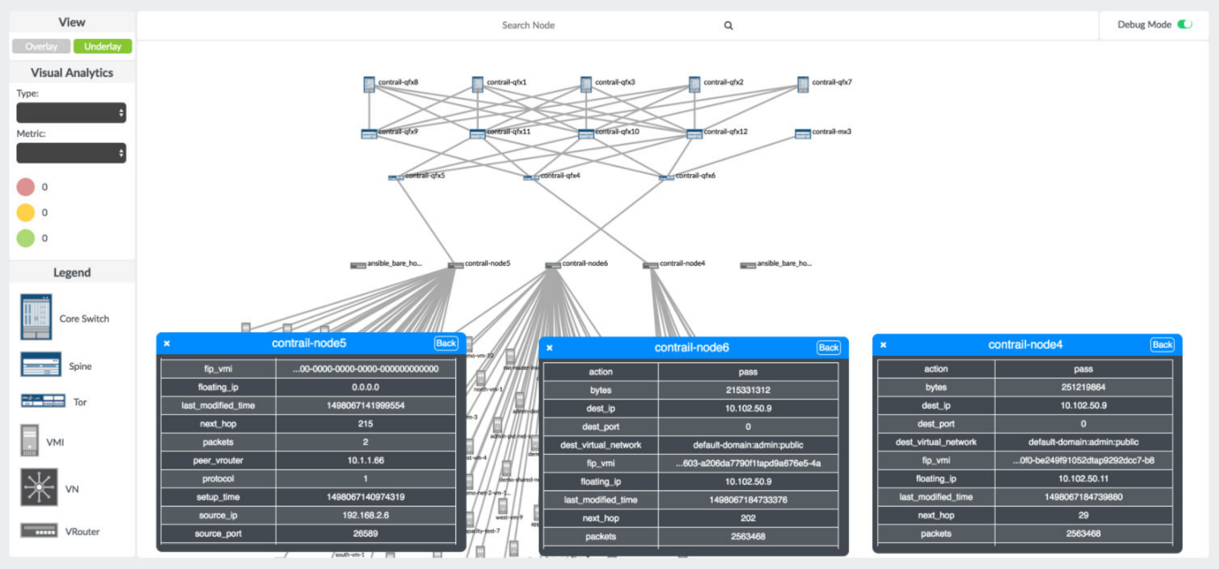
| Src | Src Port | Dest | Dest Port | Bytes |
|---|---|---|---|---|
| 192.168.2.14 | 32513 | 10.102.50.11 | 0 | 251201734 |
| 10.102.50.11 | 32513 | 10.102.50.9 | 0 | 215315772 |

contrail-node4

| Src | Src Port | Dest | Dest Port | Bytes |
|---|---|---|---|---|
| 192.168.50.4 | 32513 | 10.102.50.9 | 0 | 251201734 |
| 10.102.50.9 | 32513 | 10.102.50.11 | 0 | 215315688 |

In debug mode, you can analyze details on the *top-n* flows on any compute part of the Network Topology view. Figure 117 on page 175 shows the Contrail flow details.

**Figure 117: Contrail Flow Monitoring Details**



contrail-node5

| | |
|---|---|
| fip_vmi | ...00-0000-0000-0000-000000000000 |
| floating_ip | 0.0.0.0 |
| last_modified_time | 1498067141999554 |
| next_hop | 215 |
| packets | 2 |
| peer_vrouter | 10.1.1.66 |
| protocol | 1 |
| setup_time | 1498067140974319 |
| source_ip | 192.168.2.6 |
| source_port | 26589 |

contrail-node6

| | |
|---|---|
| action | pass |
| bytes | 215331312 |
| dest_ip | 10.102.50.9 |
| dest_port | 0 |
| dest_virtual_network | default-domain:admin:public |
| fip_vmi | ...603-a206da7790f1tapd9a676e5-4a |
| floating_ip | 10.102.50.9 |
| last_modified_time | 1498067184733376 |
| next_hop | 202 |
| packets | 2563468 |

contrail-node4

| | |
|---|---|
| action | pass |
| bytes | 251219864 |
| dest_ip | 10.102.50.9 |
| dest_port | 0 |
| dest_virtual_network | default-domain:admin:public |
| fip_vmi | ...0f0-be249f91052dtap9292dcc7-b8 |
| floating_ip | 10.102.50.11 |
| last_modified_time | 1498067184739880 |
| next_hop | 29 |
| packets | 2563468 |

## Configuring Contrail Cluster Connection Details

Contrail service monitoring is supported by the following Contrail Insights adapters:

- OpenStack
- Kubernetes
- Network Device Adapter

> **NOTE**: Network Device Adapter for monitoring Contrail service can only be used when Contrail Analytics endpoints are not authenticated.

- If more than one adapters are deployed, there is internal precedence to decide which adapter should monitor Contrail. Precedence ranking is as follows: Openstack, Kubernetes, Network Device Adapter.

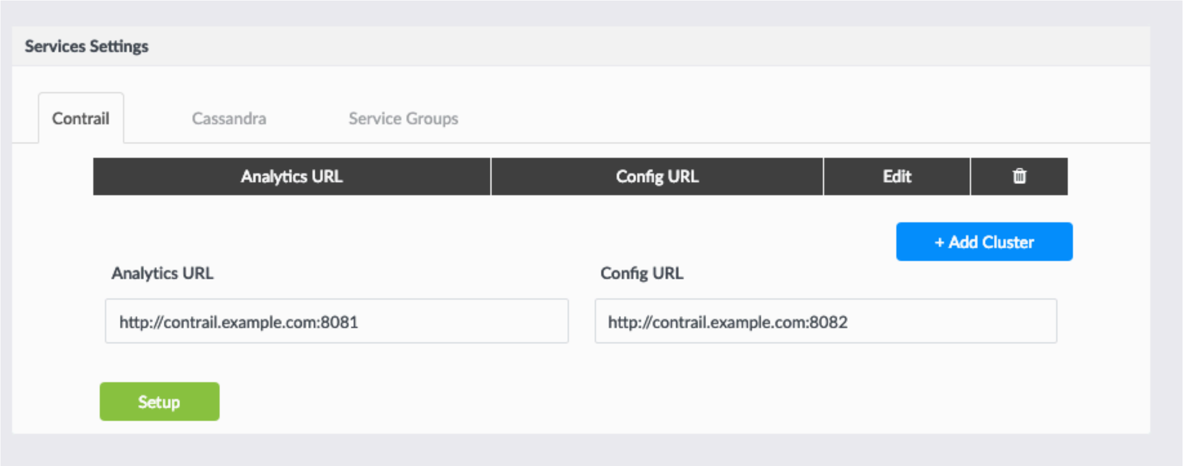
In order for Contrail Insights to monitor Contrail metrics, the Contrail Insights Platform host must be able to open connections to the Analytics API and Config API. For example, ports `8081` and `8082` on the Contrail controller.

Contrail cluster connection details can be configured in Contrail Insights Dashboard or Ansible playbooks.

To configure Contrail cluster connection details from the Dashboard:

1. Select **Settings > Service Settings**. Then select the **Contrail** tab, as shown in Figure 118 on page 176.

**Figure 118: Configure Contrail Cluster Connection Details**



Analytics URL: http://contrail.example.com:8081

Config URL: http://contrail.example.com:8082

2. Click **Add Cluster**.

   Enter the cluster name, analytics URL, and configuration URL. The URLs should specify only the protocol, address, and optional port.

For example, `http://contrail.example.com:8081` for the analytics URL and `http://contrail.example.com:8082` for the configuration URL.

3. Click **Setup**. On success, a **Submission Successful** message appears in the Dashboard.

Contrail service monitoring is configured by the Ansible role `appformix_contrail_config`. This Ansible role is applied to the `appformix_controller` group of hosts. Ansible performs the configuration if the variables are set as extra vars, group vars, etc.

For configuration using Ansible playbooks, see *Service Monitoring Ansible Variables* for steps to configure Contrail Insights to monitor a Contrail cluster.

Configuration of Contrail uses the same OpenStack credentials as provided for Contrail Insights to access OpenStack services. The Ansible role reads the credentials from environment variables (for example, `OS_USERNAME`, `OS_PASSWORD`). Administrator credentials to the OpenStack cluster are also needed. Contrail Insights connects to the analytics and configuration nodes of Contrail.

### Contrail Configuration Starting with Contrail Insights Release 2.15

Starting with Contrail Insights Release 2.15, connections to Contrail are configured by providing complete URLs to access the analytics and configuration API services.

- The URL for Contrail analytics API (`contrail_analytics_url`) should specify protocol, address, and port.

  For example, *http://contrail.example.com:8081*.

- The URL for Contrail configuration API (`contrail_config_url`) should specify protocol, address, and port.

  For example, *http://contrail.example.com:8082*.

- In certain cases, optional variables can be specified as well.

  For example, in the Dashboard, when the Contrail cluster name (`contrail_cluster_name`) is not provided, a default variable value (*default_contrail_cluster*) is set.

### Contrail Configuration Prior to Contrail Insights Release 2.15

For releases prior to Contrail Insights Release 2.15, the configuration is specified as a single hostname by which both the analytics and configuration APIs are accessed.

Contrail Insights connects to port 8081 for the analytics API and port 8082 for the configuration API.

Hostname (`contrail_host`) is the IP address or hostname of the Contrail API server.

## Configuring Dynamic Alarms Data Purge Rate

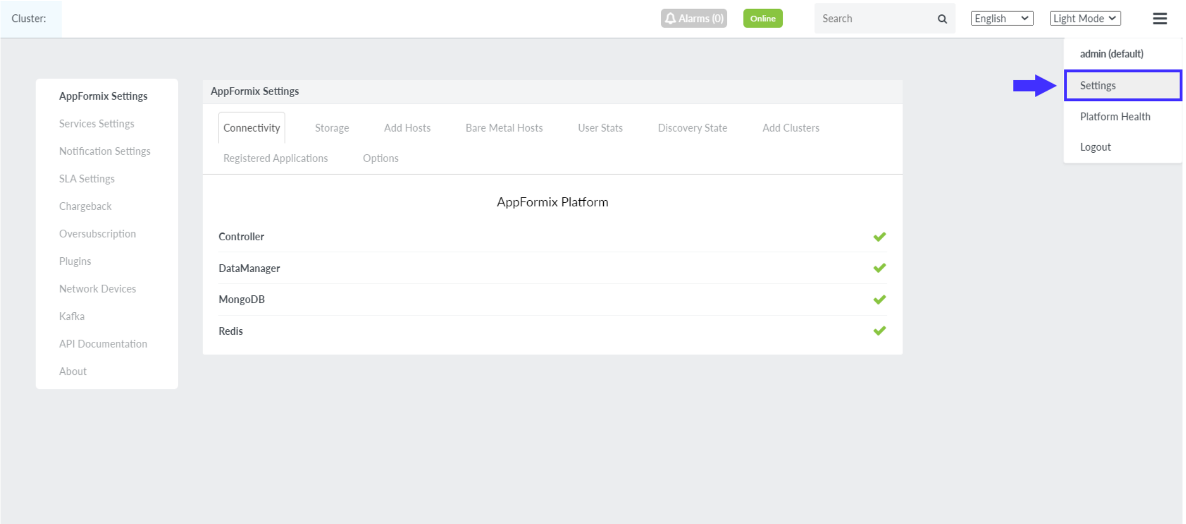
You can configure dynamic alarms data purge rate by using the Contrail Insights user interface (UI).

Follow these steps to configure dynamic alarms data purge rate.

1. Click **Settings** as shown in Figure 119 on page 178.

   The Connectivity tab of the AppFormix Settings page is displayed.

   **Figure 119: Click Settings Button**

   AppFormix Platform

   | | |
   |---|---|
   | Controller | ✓ |
   | DataManager | ✓ |
   | MongoDB | ✓ |
   | Redis | ✓ |

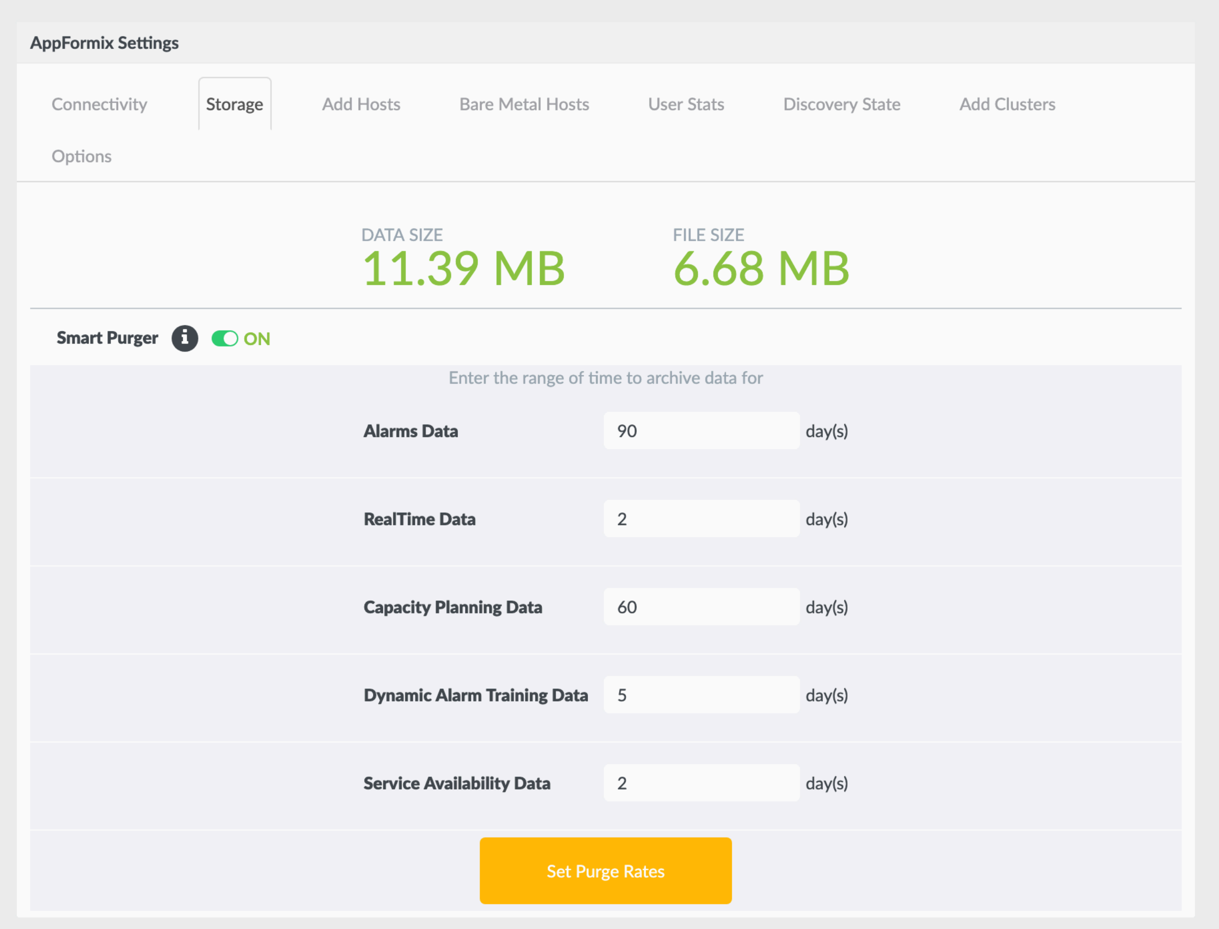
2. Click the **Storage** tab as shown in Figure 120 on page 179.

3. Enter the required values as shown in Figure 120 on page 179.

   **NOTE**: Ensure that the values entered in the **Dynamic Alarm Training Data** and **Service Availability Data** fields are not zero.

4. Click **Set Purge Rates** to confirm.

**Figure 120: Configure Dynamic Alarms Data Purge Rate**



DATA SIZE 11.39 MB

FILE SIZE 6.68 MB

Smart Purger ON

Enter the range of time to archive data for

Alarms Data 90 day(s)

RealTime Data 2 day(s)

Capacity Planning Data 60 day(s)

Dynamic Alarm Training Data 5 day(s)

Service Availability Data 2 day(s)

## Cassandra Monitoring

Contrail Insights supports Cassandra monitoring.

The Cassandra configuration can be specified by using Ansible or by using the Contrail Insights UI.

However, before you begin, ensure that you specify the Cassandra plug-in in the `group_vars/all` file.

```
appformix_plugins:
  - { plugin_info: certified_plugins/cassandra_node_usage.json }
```

1. **Configuring Cassandra by using Ansible**

Specify these variables in the `group_vars/all` file.

```
cassandra_username: Cassandra username to access API
cassandra_password: Cassandra password to access API
cassandra_host: Hostname or IP address of Cassandra API server
cassandra_cluster_name: A name by which the Cassandra instance will be displayed in the
Dashboard. If not
                        specified, this variable has a default value of
default_cassandra_cluster.
cassandra_cluster_port: (Optional) Port used by the Cassandra service. Default port is 9042.
appformix_cassandra_ssl_ca: (Optional) /path/to/ca_cert when SSL is enabled on cassandra
cluster.
```

When SSL is enabled on the Cassandra cluster, ensure that the Certificate Authority (CA) used for the certificates for the Cassandra nodes are trusted across all Contrail Insights platform nodes. In order for Contrail Insights containers to communicate with Cassandra, the CA file must be set as a group_vars/all variable (`appformix_cassandra_ssl_ca`) during installation. For SSL, Cassandra cluster must be added by using Ansible and not by using the UI.

2. **Configuring Cassandra by using the Contrail Insights UI**

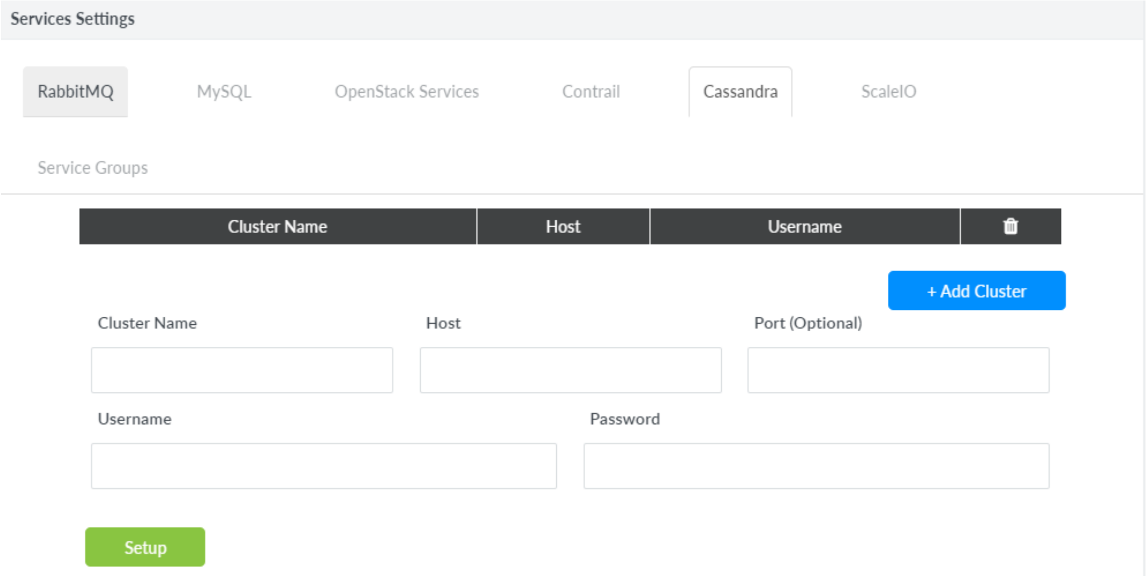
   Follow these steps to configure Cassandra by using the Contrail Insights UI.

   a. Navigate to the **Settings**>**Service Settings** page.

   b. Click the **Cassandra** tab and then click **+Add Cluster**.

      Fields related to configuring Cassandra cluster are displayed as shown in Figure 121 on page 181.

Figure 121: Cassandra Service Settings



c. Enter the information as provided in Table 20 on page 181.

Table 20: Configuring Cassandra Cluster

| Field | Action/Description |
|---|---|
| Cluster Name | Enter a name for the Cassandra cluster. |
| Host | Enter the Cassandra cluster host IP address. |
| Port (Optional) | Enter the port number.<br>This field is optional. |
| Username | Enter a user name for the Cassandra cluster. |
| Password | Enter a password for the Cassandra cluster. |

d. Click **Setup** to save configuration.

# MySQL Monitoring

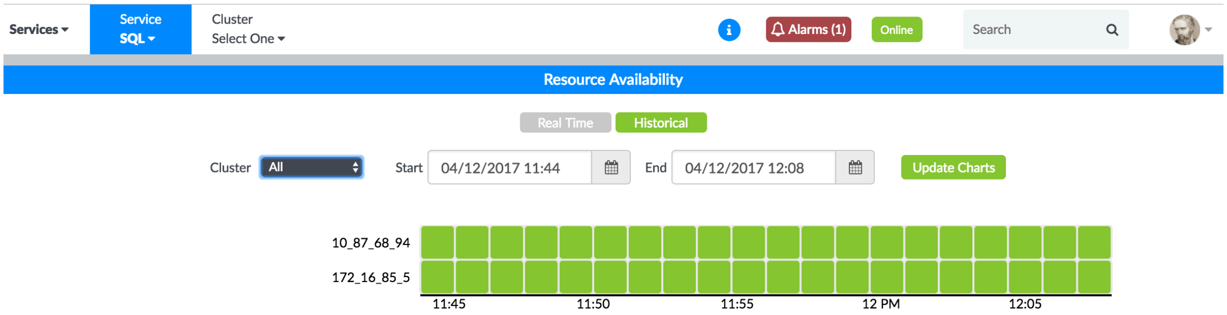
IN THIS SECTION

A MySQL database is integral to the operation of OpenStack infrastructure services. Metrics for MySQL performance are available in real-time charts and alarms. Mulitple MySQL clusters can be configured to be monitored.

## Resource Availability

The availability of MySQL nodes for each of the configured MySQL clusters is recorded periodically. You can view both the current status, as well as the historical status over a specified period of time by selecting **All Services > MySQL** from the context menu at the top and, then select **Dashboard** from the left pane. Figure 122 on page 182 shows the historical resource availability for the MySQL nodes.

**Figure 122: MySQL Nodes Historical Availability**



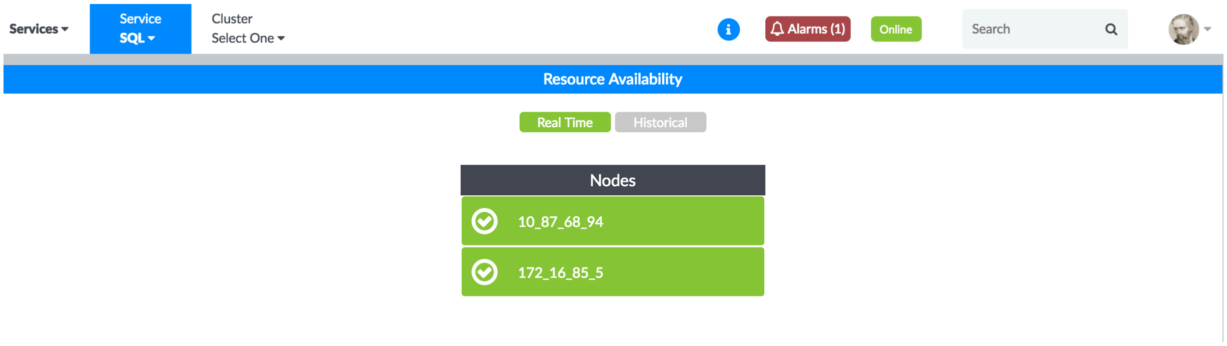
Cluster All

Start 04/12/2017 11:44

End 04/12/2017 12:08

Figure 123 on page 183 shows the real-time resource availability for the MySQL nodes.

### Figure 123: MySQL Nodes Real-Time Availability

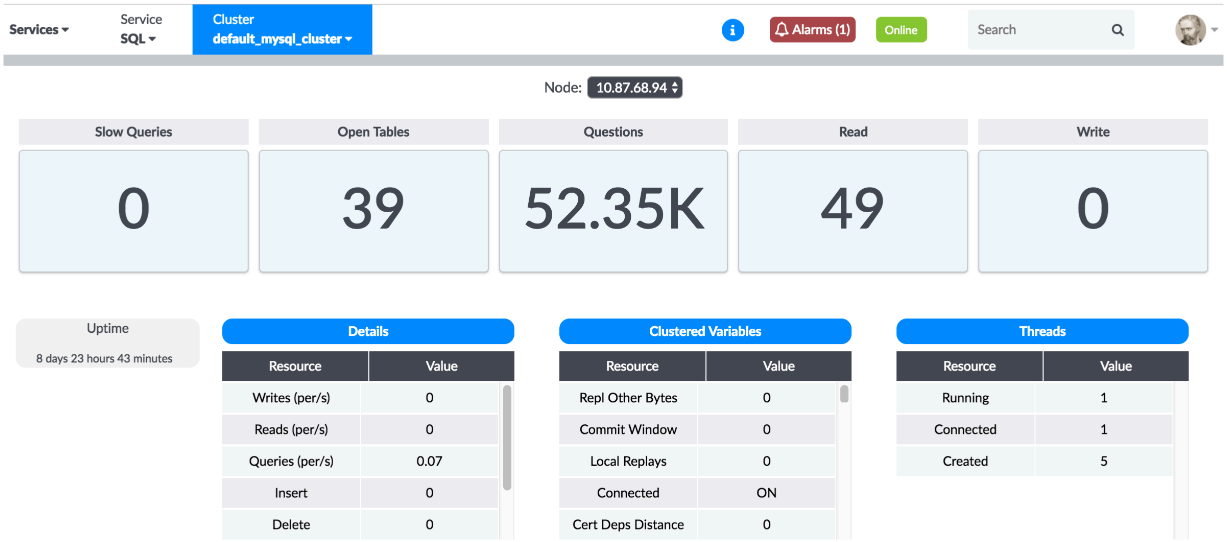


| Nodes |
|---|
| ✓ 10_87_68_94 |
| ✓ 172_16_85_5 |

## Dashboard

Each MySQL cluster has a dashboard displaying real-time usage metrics for each of its nodes, as shown in Figure 124 on page 183.

### Figure 124: Real-Time Usage Metrics for Cluster Nodes



Node: 10.87.68.94

| Slow Queries | Open Tables | Questions | Read | Write |
|---|---|---|---|---|
| 0 | 39 | 52.35K | 49 | 0 |

Uptime

8 days 23 hours 43 minutes

Details

| Resource | Value |
|---|---|
| Writes (per/s) | 0 |
| Reads (per/s) | 0 |
| Queries (per/s) | 0.07 |
| Insert | 0 |
| Delete | 0 |

Clustered Variables

| Resource | Value |
|---|---|
| Repl Other Bytes | 0 |
| Commit Window | 0 |
| Local Replays | 0 |
| Connected | ON |
| Cert Deps Distance | 0 |

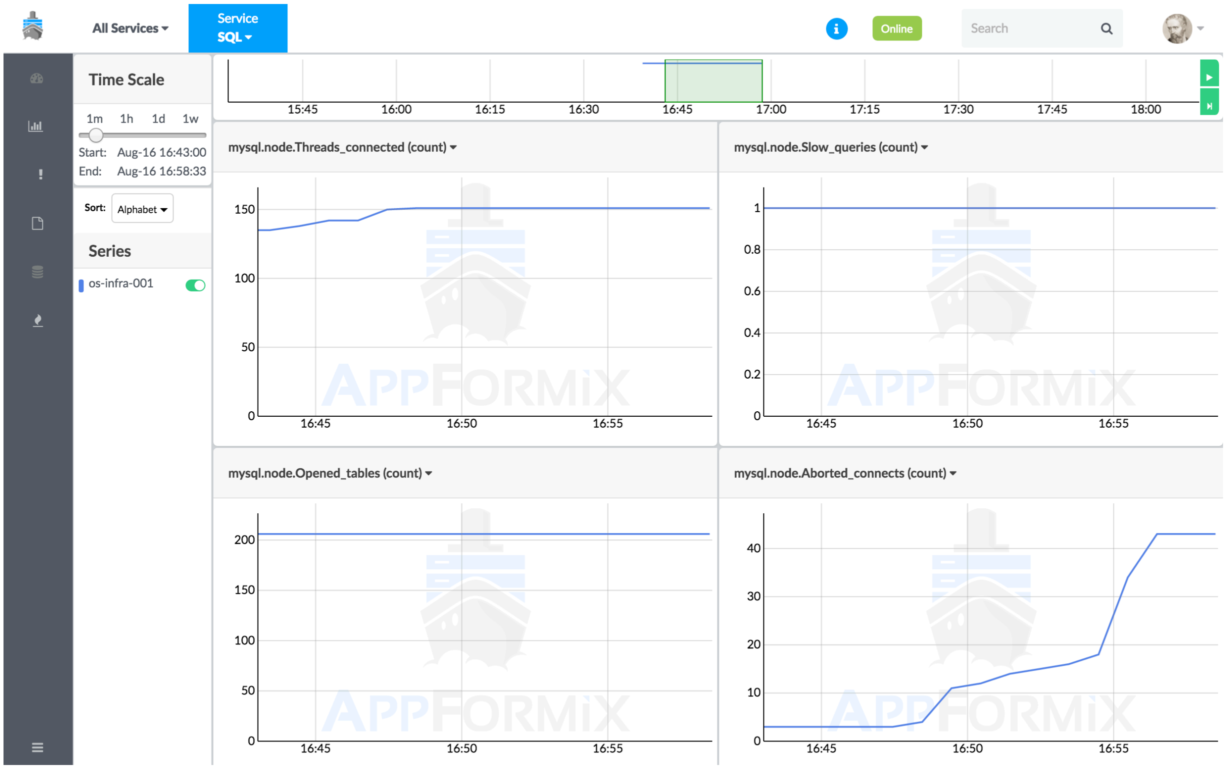
Threads

| Resource | Value |
|---|---|
| Running | 1 |
| Connected | 1 |
| Created | 5 |

## Real-Time Charts

From the context menu, select **All Services > MySQL**. Click the Charts icon from the left navigation pane. Figure 125 on page 184 shows MySQL performance metric charts.

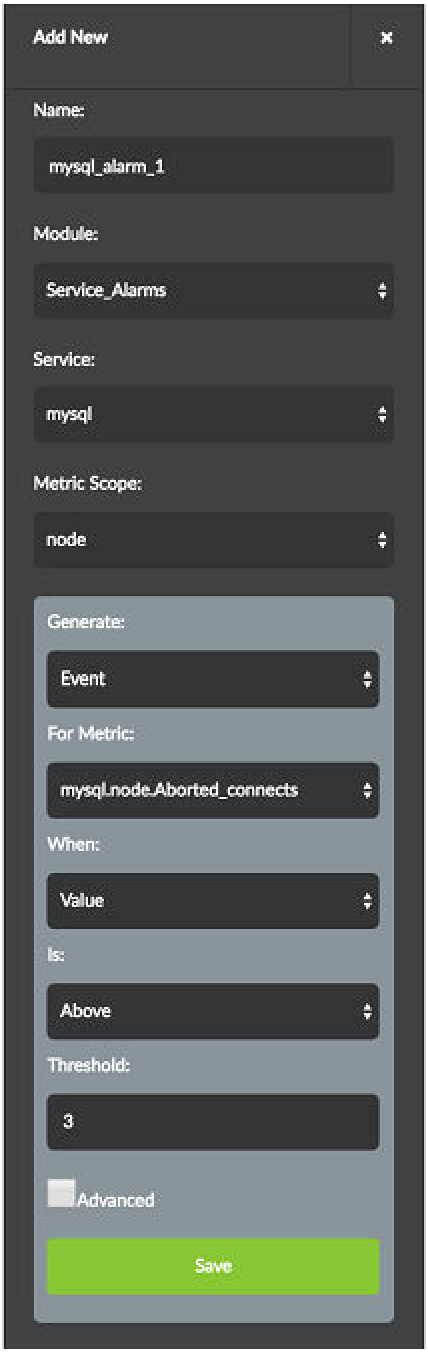
Figure 125: MySQL Performance Metric Charts



## Service Alarms

An alarm can be configured for any of the MySQL metrics collected. In the Alarm pane, select the **Service Alarms** module. Then select **mysql** from the Service drop-down list. MySQL alarms can be created for one or more MySQL nodes. Additionally, *Notifications* can also be configured for MySQL *Alarms*. Figure 126 on page 185 shows the Alarm Input pane for MySQL alarm configuration.

**Figure 126: Alarm Input Pane for MySQL**



Add New ×

Name: mysql_alarm_1

Module: Service_Alarms

Service: mysql

Metric Scope: node

Generate: Event

For Metric: mysql.node.Aborted_connects

When: Value

Is: Above

Threshold: 3

□ Advanced

## Configuration

For Contrail Insights to monitor MySQL metrics, there must exist a MySQL user with remote, read-permission. In this topic, we create a new user with read-only access to the database. Alternately, an existing user account can be used.

To configure MySQL monitoring:

1. Create a read-only user account 'appformix' that can access the MySQL database from any host:

```
$ mysql -u root -p
mysql> grant SELECT on *.* to ''appformix’'@’'%’'' identified by 'mypassword';
mysql> flush privileges;
```

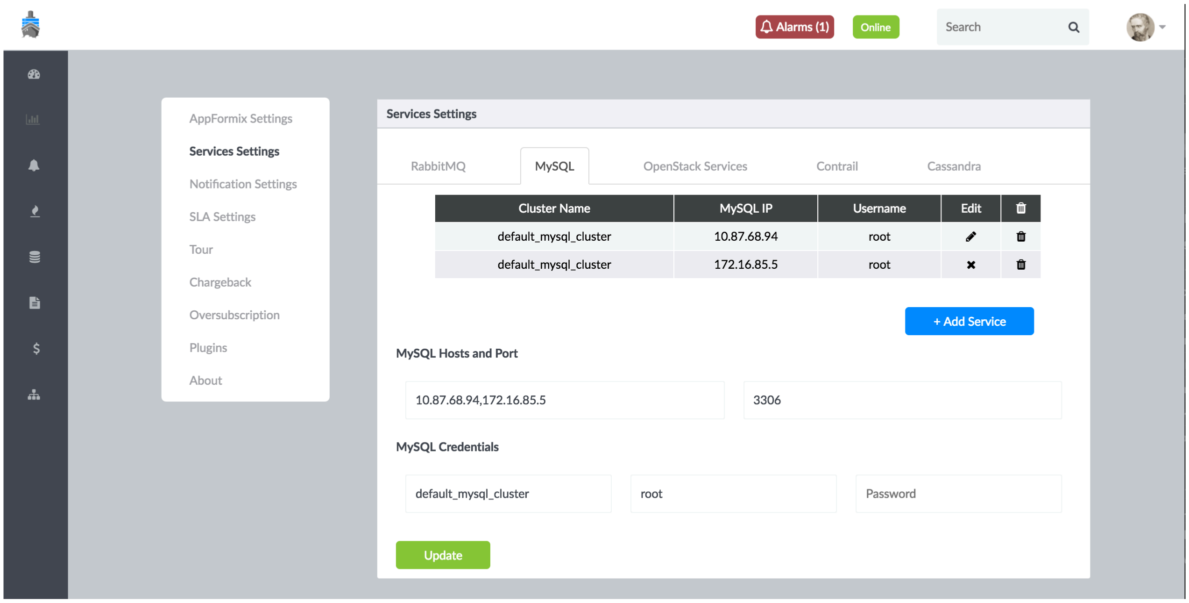
   Change 'mypassword' to a strong password. Optionally, you may restrict the 'appformix' account to only connect from a specific IP address or hostname by replacing '%' with the host on which Contrail Insights Platform runs.

2. Next, configure the MySQL connection details in Contrail Insights. From the Settings menu, select **Service Settings**. Then, select the **MySQL** tab.

3. Enter the host and port on which MySQL runs. The default port for MySQL is **3306**.

4. Enter the username and password from Step 1. Finally, click the **Setup** button. On success, the button changes to Submitted. Figure 127 on page 186 shows MySQL connection and credential settings.

**Figure 127: My SQL Connection and Credential Settings**



| Cluster Name | MySQL IP | Username | Edit | 🗑 |
|---|---|---|---|---|
| default_mysql_cluster | 10.87.68.94 | root | ✎ | 🗑 |
| default_mysql_cluster | 172.16.85.5 | root | ✖ | 🗑 |

MySQL Hosts and Port

10.87.68.94,172.16.85.5 | 3306

MySQL Credentials

default_mysql_cluster | root | Password

# OpenStack Services Monitoring

**IN THIS SECTION**

Contrail Insights monitors Keystone, Nova, and Neutron services that power the OpenStack cloud management system. Starting with Release 3.3.4, Contrail Insights also monitors Octavia that provides load balancing services.

Contrail Insights performs status checks for processes that implement the services on both controller and compute hosts. The overall connectivity to each API and the status of components that comprise of these services, are also monitored.

Overall connectivity is monitored in the following ways:

- Component service list in the case of Nova and Keystone.
- API call for listing all load balancers in the case of Octavia.
- Agent list in the case of Neutron.

For example, if the nova-api sub-service is up and responds to the API call successfully, then the Health of the `default_openstack_cluster_status` for Nova will be `Good` - even if an individual sub-service of Nova has failed.

As an alternative example, consider that the nova-scheduler is not running. In such a scenario, if the API call to list the status of Nova sub-services succeeds, then the `default_openstack_cluster_status` will be `Good`. However, health of the `nova-scheduler` will be `Bad`.

## Using Dashboard to View Current and Historical Status

You can view the current status and the historical status of a service over a specified period of time in the Dashboard.

To view the current status or historical status of a service, select **Dashboard** from the left pane and select the name of a service from **Services** list.

To view the current status or historical status of a service,

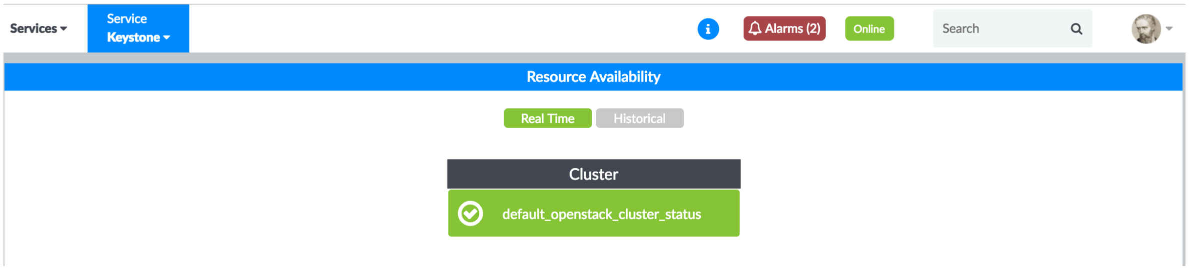
1. Select **Dashboard** from the left pane and then select **Services** from the context menu at the top.

   The Service list appears next to Services.

2. Select **Keystone** from the Service list.

   The Resource Availability pane is displayed showing the OpenStack Keystone nodes in real-time (current status) availability. See Figure 128 on page 188.

**Figure 128: OpenStack Keystone Nodes Real-Time Availability**

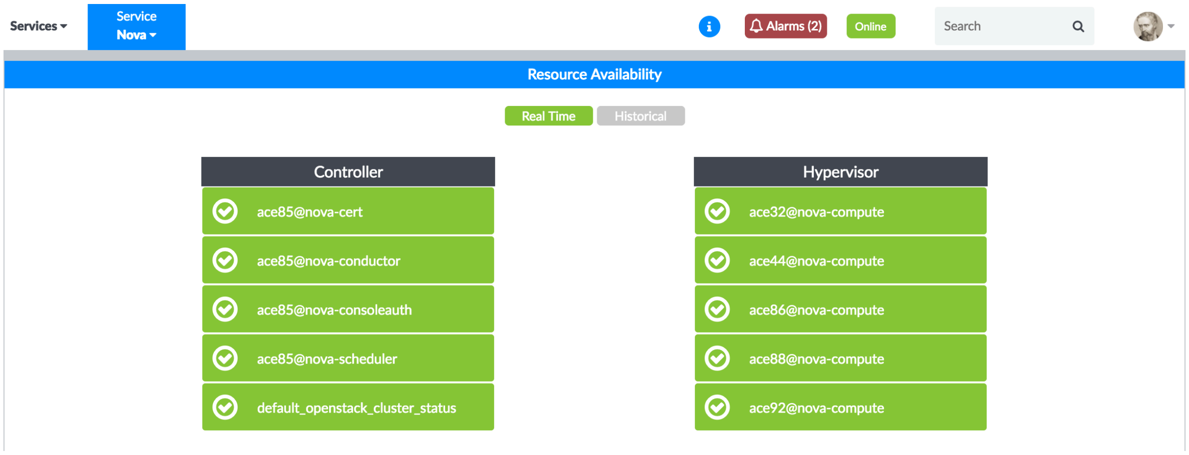


Cluster

default_openstack_cluster_status

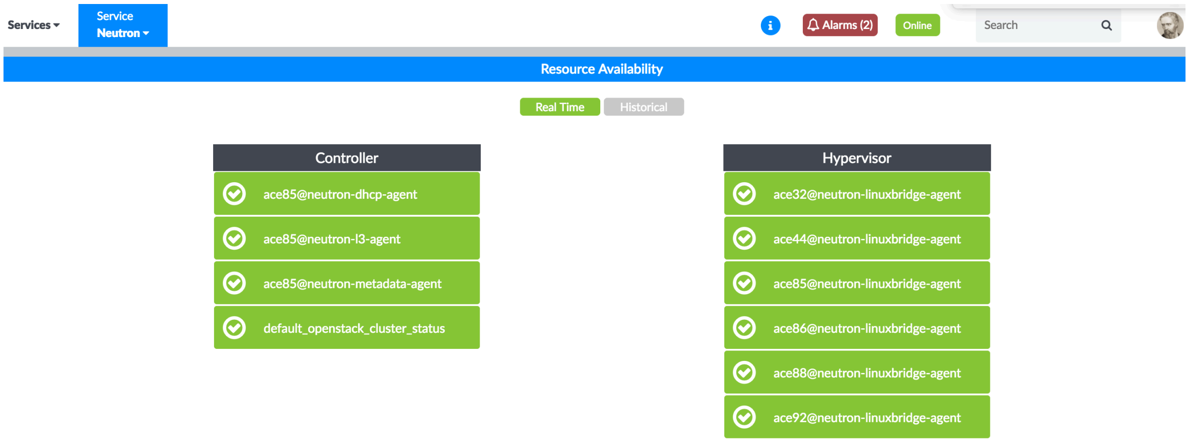
Figure 129 on page 188 and Figure 136 on page 194 are examples of real-time availability of OpenStack Nova and OpenStack Neutron nodes.

**Figure 129: OpenStack Nova Nodes Real-Time Availability**



| Controller | Hypervisor |
|---|---|
| ace85@nova-cert | ace32@nova-compute |
| ace85@nova-conductor | ace44@nova-compute |
| ace85@nova-consoleauth | ace86@nova-compute |
| ace85@nova-scheduler | ace88@nova-compute |
| default_openstack_cluster_status | ace92@nova-compute |

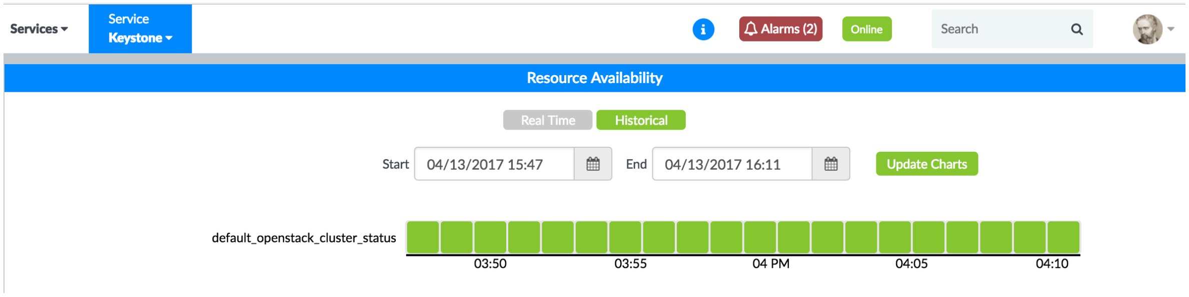
### Figure 130: OpenStack Neutron Nodes Real-Time Availability


Services
Service
Neutron
Alarms (2)
Online
Search
Resource Availability
Real Time
Historical
Controller
ace85@neutron-dhcp-agent
ace85@neutron-l3-agent
ace85@neutron-metadata-agent
default_openstack_cluster_status
Hypervisor
ace32@neutron-linuxbridge-agent
ace44@neutron-linuxbridge-agent
ace85@neutron-linuxbridge-agent
ace86@neutron-linuxbridge-agent
ace88@neutron-linuxbridge-agent
ace92@neutron-linuxbridge-agent

3. To view historical availability of a service, click **Historical** in the Resource Availability pane.

   For example, Figure 131 on page 189 shows the historical availability of the OpenStack Keystone nodes.

### Figure 131: OpenStack Keystone Nodes Historical Availability

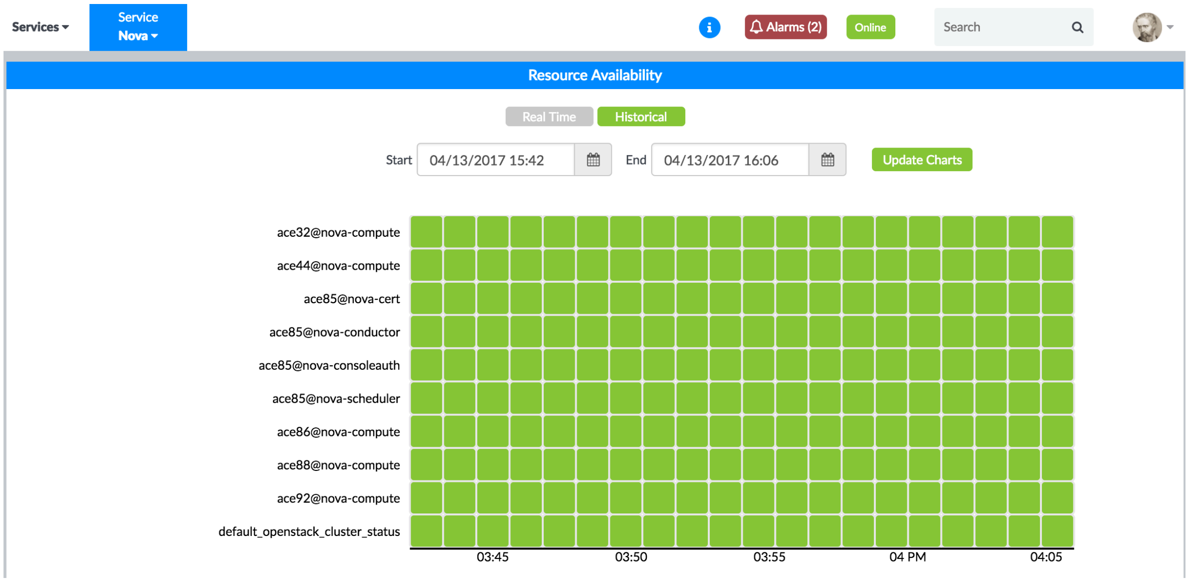


4. You can also view historical availability of a service in a particular period.

   To view historical availability of a service for a particular period, select start date and time from the **Start** list and end date and time from the **End** list, and click **Update Charts**.

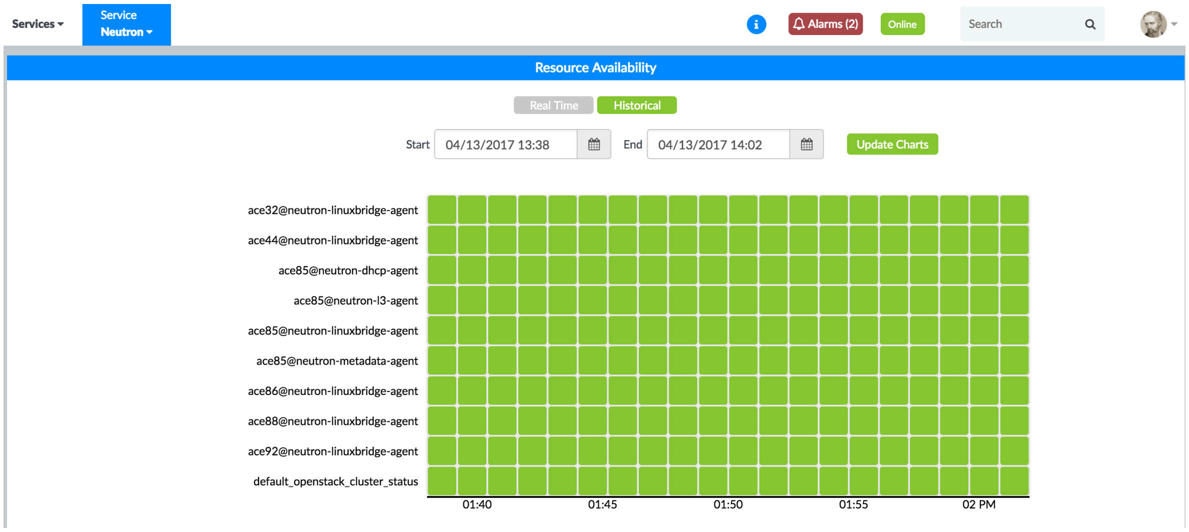
   Figure 132 on page 190 and Figure 133 on page 190 are examples of historical availability of OpenStack Nova nodes and OpenStack Neutron nodes.

Figure 132: OpenStack Nova Nodes Historical Availability



Start 04/13/2017 15:42 End 04/13/2017 16:06

Figure 133: OpenStack Neutron Nodes Historical Availability


Services
Service
Neutron
Alarms (2)
Online
Search
Resource Availability
Real Time
Historical
Update Charts
ace32@neutron-linuxbridge-agent
ace44@neutron-linuxbridge-agent
ace85@neutron-dhcp-agent
ace85@neutron-l3-agent
ace85@neutron-linuxbridge-agent
ace85@neutron-metadata-agent
ace86@neutron-linuxbridge-agent
ace88@neutron-linuxbridge-agent
ace92@neutron-linuxbridge-agent
default_openstack_cluster_status
01:40
01:45
01:50
01:55
02 PM

Start 04/13/2017 13:38 End 04/13/2017 14:02

## Configuring Service Alarms

An alarm can be configured for any OpenStack services.

To configure an alarm, select the **Service Alarms** module from the Alarm pane. Then select **openstack** from the Service drop-down list.

The metrics for which alarms can be configured are broadly categorized into three scopes:

**Cluster** Heartbeat metrics, such as liveness checks for Nova, Neutron, Octavia, and Keystone APIs.

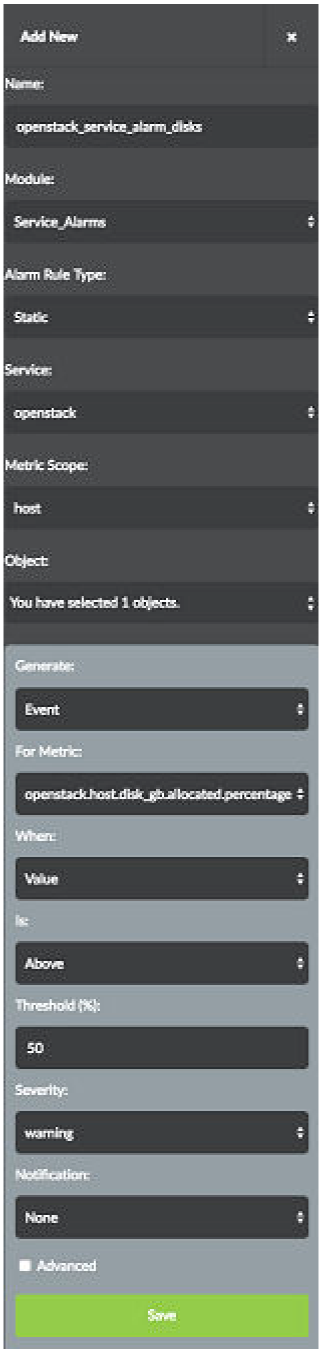
**Host** Allocation of resources on compute hosts. Alarms can be configured for absolute count or as a percentage of host capacity. Metrics include virtual CPU (vCPU), memory, and local storage.

**Project** Allocation of resources by a project. Alarms can be configured for absolute count or as a percentage of project quota. Resource metrics include instances, vCPU, memory, storage, floating IP addresses, and security groups.

As with other alarms, notifications can also be configured for any OpenStack service alarm, as shown in Figure 134 on page 192.

SLA profiles can be configured for Nova, Neutron, Octavia, and Keystone by navigating to the **Settings > SLA Settings** page. You can then select the appropriate tab for the service. A list of rules can be defined for both Health and Risk.

Figure 134: Alarm Input Pane for OpenStack Services



Name: openstack_service_alarm_disks

Module: Service_Alarms

Alarm Rule Type: Static

Service: openstack

Metric Scope: host

Object: You have selected 1 objects.

Generate: Event

For Metric: openstack.host.disk_gb.allocated.percentage

When: Value

Is: Above

Threshold (%): 50

Severity: warning

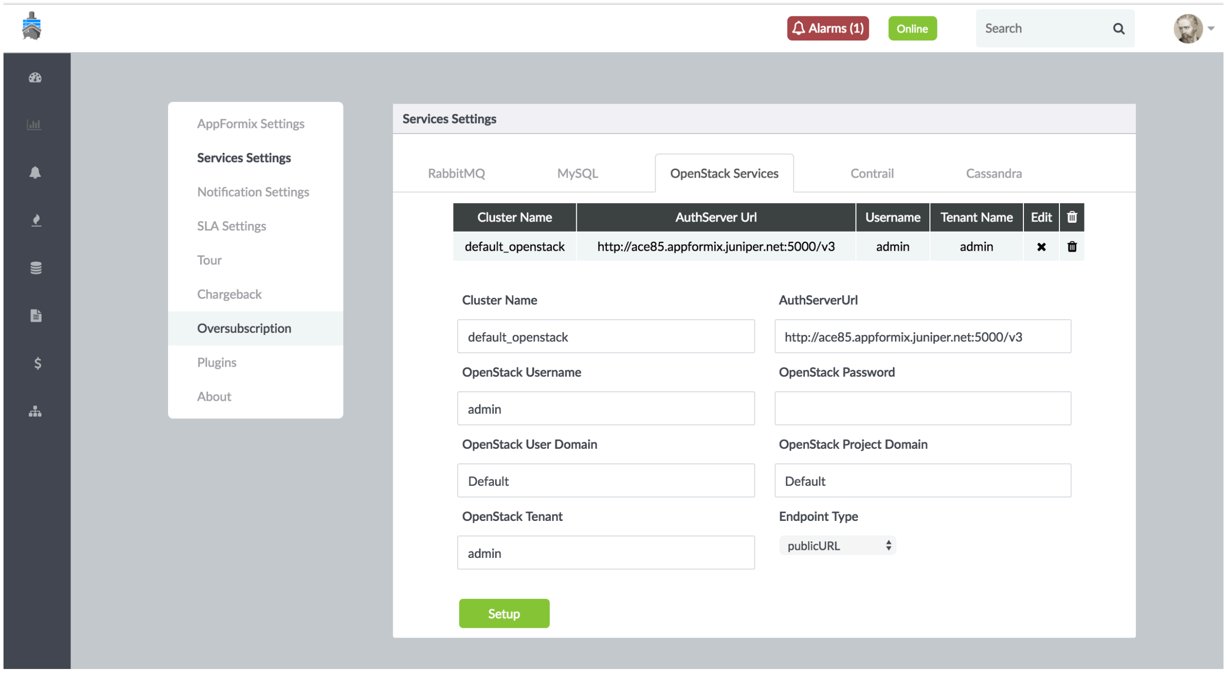
Notification: None

☐ Advanced

## OpenStack Configuration Parameters

The OpenStack configuration parameters provided during Contrail Insights installation are sufficient for monitoring OpenStack services. No additional configuration is required. To modify the current values, from the Settings menu, select **Service Settings**. Then select the **OpenStack Services** tab. Figure 135 on page 193 shows the OpenStack services settings and configuration parameters.

**Figure 135: OpenStack Services Settings and Configuration Parameters**



Services Settings

| Cluster Name | AuthServer Url | Username | Tenant Name | Edit | 🗑 |
|---|---|---|---|---|---|
| default_openstack | http://ace85.appformix.juniper.net:5000/v3 | admin | admin | ✖ | 🗑 |

Cluster Name: default_openstack

AuthServerUrl: http://ace85.appformix.juniper.net:5000/v3

OpenStack Username: admin

OpenStack Password:

OpenStack User Domain: Default

OpenStack Project Domain: Default

OpenStack Tenant: admin

Endpoint Type: publicURL

## RabbitMQ Monitoring

**IN THIS SECTION**

OpenStack depends on RabbitMQ to deliver messages between services. Contrail Insights Service Monitoring can be used to monitor RabbitMQ metrics through real-time charts. Service alarms can also be configured for these metrics.

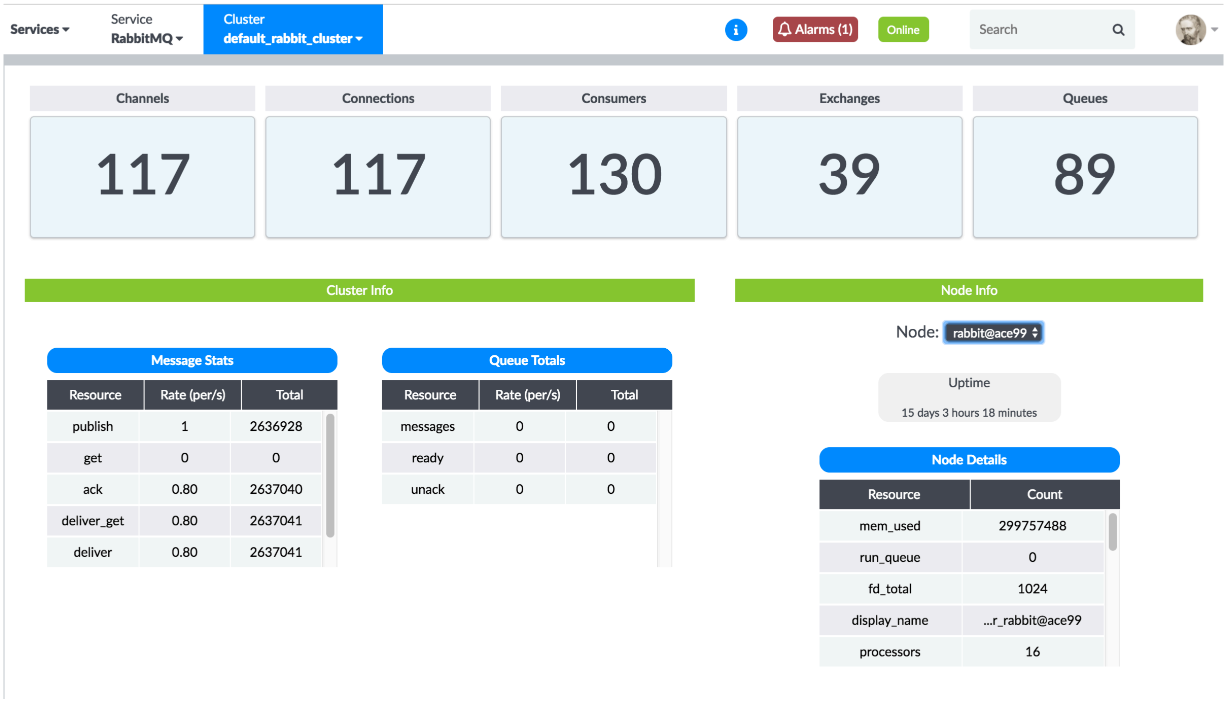
## Resource Availability

The connectivity of nodes for each of the configured Rabbit clusters is recorded periodically. You can view both the current status, as well as the historical status over a specified period of time by selecting **Services > RabbitMQ** from the context menu at the top, and selecting **Dashboard** in the left pane.

## Dashboard

The Dashboard also provides detailed metrics for a single RabbitMQ cluster, as shown in Figure 136 on page 194. Select **Dashboard** in the left pane, then **Services > RabbitMQ** in the top context menu, and then select a Rabbit Cluster by name.

### Figure 136: Real-Time Usage Metrics for RabbitMQ Cluster



| Channels | Connections | Consumers | Exchanges | Queues |
|---|---|---|---|---|
| 117 | 117 | 130 | 39 | 89 |

Message Stats

| Resource | Rate (per/s) | Total |
|---|---|---|
| publish | 1 | 2636928 |
| get | 0 | 0 |
| ack | 0.80 | 2637040 |
| deliver_get | 0.80 | 2637041 |
| deliver | 0.80 | 2637041 |

Queue Totals

| Resource | Rate (per/s) | Total |
|---|---|---|
| messages | 0 | 0 |
| ready | 0 | 0 |
| unack | 0 | 0 |

Node: rabbit@ace99

Uptime

15 days 3 hours 18 minutes

Node Details

| Resource | Count |
|---|---|
| mem_used | 299757488 |
| run_queue | 0 |
| fd_total | 1024 |
| display_name | ...r_rabbit@ace99 |
| processors | 16 |

The counters in the top pane display the number of active channels, connections, consumers, exchanges, and queues. Below, tables display statistics about message rates across the cluster, and per-node resource consumption.

## Real-Time Charts

Contrail Insights UI provides a real-time view of RabbitMQ metrics.

Follow these steps to view RabbitMQ metrics in real time.

1. Select **Dashboard** from the left-nav pane.

   The Contrail Insights dashboard is displayed.

2. When in the Dashboard view, select **Services** from the context menu.

   The Service drop-down list is displayed.

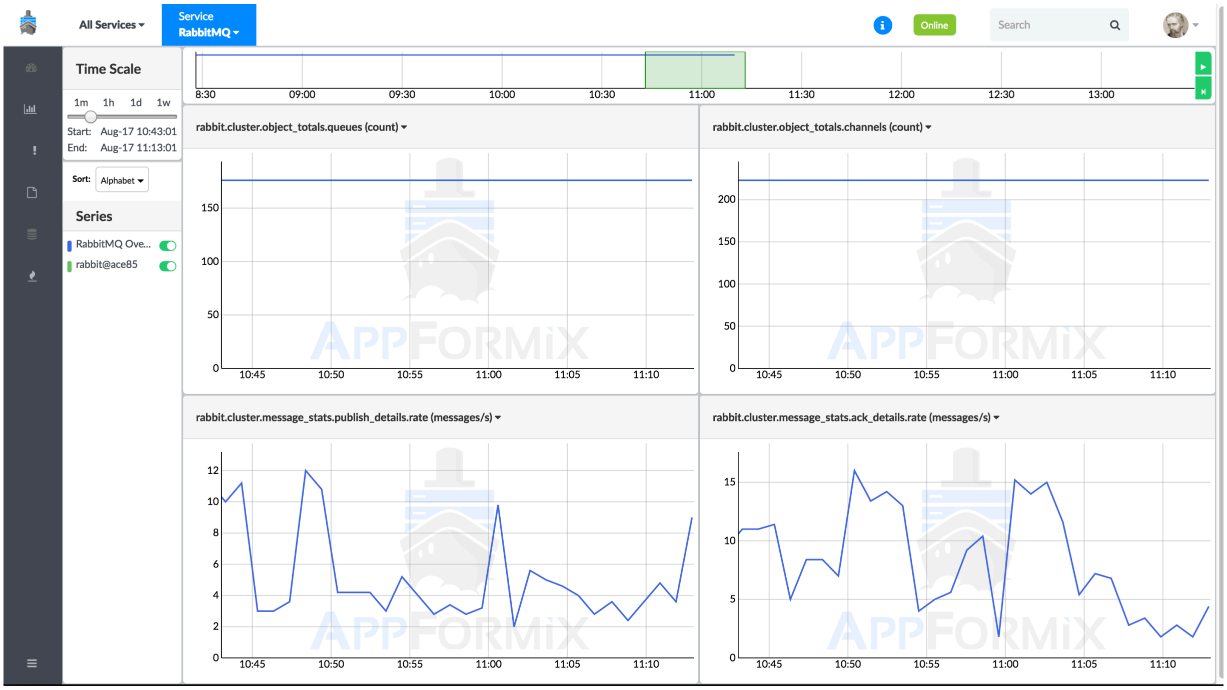
3. Select **RabbitMQ** from the service drop-down list.

   The Resource Availability page is displayed.

4. Click the **Charts** icon in the left-nav pane to view real-time metric charts.

   Figure 137 on page 195 shows RabbitMQ real-time metric charts.

### Figure 137: RabbitMQ Real-Time Metric Charts



Start: Aug-17 10:43:01
End: Aug-17 11:13:01

Sort: Alphabet

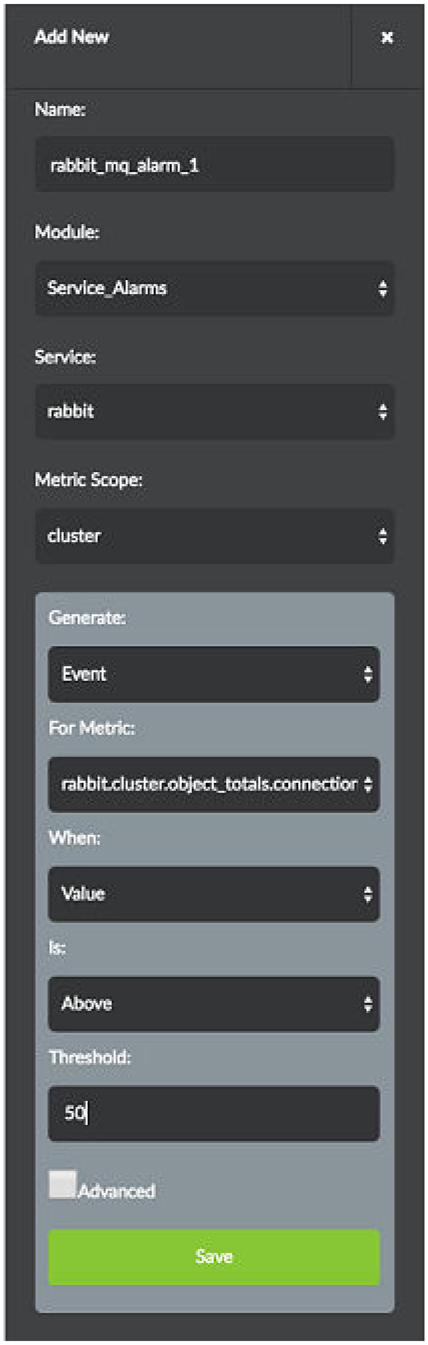
All Services
Service
RabbitMQ
Online
Search
Time Scale
1m
1h
1d
1w
Series
RabbitMQ Ove...
rabbit@ace85
8:30
09:00
09:30
10:00
10:30
11:00
11:30
12:00
12:30
13:00
rabbit.cluster.object_totals.queues (count)
rabbit.cluster.object_totals.channels (count)
rabbit.cluster.message_stats.publish_details.rate (messages/s)
rabbit.cluster.message_stats.ack_details.rate (messages/s)
0
50
100
150
200
2
4
6
8
10
12
5
15
10:45
10:50
10:55
11:00
11:05
11:10
APPFORMIX

## Service Alarms

In releases prior to Contrail Insights Release 3.3.0, you can configure a service alarm to monitor RabbitMQ metrics by selecting **Alarms** from the left-nav pane. For more information on alarms, see *Alarms*.

Ensure that you select **Service_Alarms** for the module, and **rabbit** for the service. An alarm can be configured for a metric on a per-cluster, per-node, or per-queue basis. After you have selected the appropriate metric scope, you then choose a metric to monitor. As with other alarms, you can optionally configure *Notifications* in the Advanced settings. Figure 138 on page 197 shows the RabbitMQ alarm configuration pane.

**Figure 138: RabbitMQ Alarm Configuration**



Add New

Name: rabbit_mq_alarm_1

Module: Service_Alarms

Service: rabbit

Metric Scope: cluster

Generate: Event

For Metric: rabbit.cluster.object_totals.connectior

When: Value

Is: Above

Threshold: 50

☐ Advanced

## Alarms

Starting with Contrail Insights Release 3.3.0, you can configure alarms to monitor RabbitMQ metrics.

Follow these steps to configure alarms to monitor RabbitMQ metrics from the Contrail Insights UI.

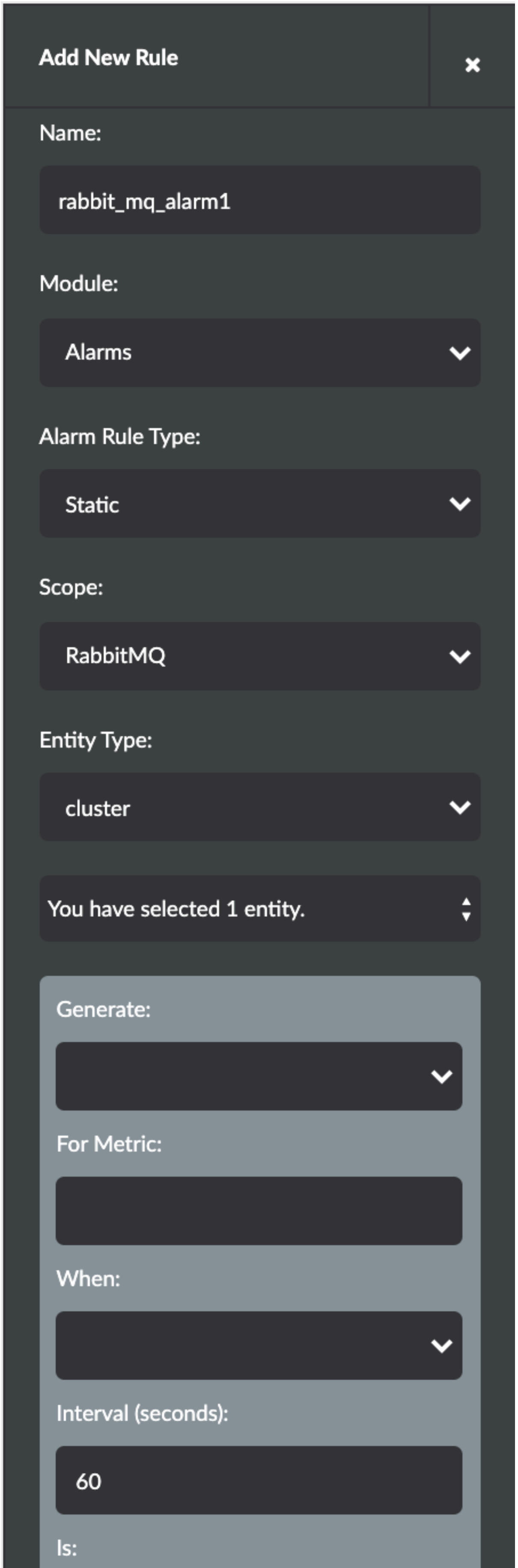
1. Select **Alarms** from the left-nave pane.

   The Alarms page and the Alarms pane is displayed.

2. Click **Add Rule** in the Alarms pane on the left.

   The Add New Rule pane is displayed. See Figure 139 on page 199.

Figure 139: Configure Alarm - Add New Rule



Add New Rule

Name: rabbit_mq_alarm1

Module: Alarms

Alarm Rule Type: Static

Scope: RabbitMQ

Entity Type: cluster

You have selected 1 entity.

Generate:

For Metric:

When:

Interval (seconds): 60

Is:

3. Enter the following information as given in Table 21 on page 200.

**Table 21: Configure New Alarm**

| Field | Action/Description |
|---|---|
| Name | Enter a name for the alarm. |
| Module | Select **Alarms** from the module drop-down list. |
| Alarm Rule Type | Select Static or Dynamic from the drop-down list. |
| Scope | Select **RabbitMQ** as the alarm scope from the drop-down list. |
| Entity Type | Select **cluster** from the drop-down list. |

**NOTE**: An alarm can be configured for a metric on a per-cluster, per-node, or per-queue basis.

4. In the Generate section,

   a. Select a generate option from the Generate drop-down list.

      You can either select Generate Event or Generate Alert.

   b. Select a metric to monitor from the For Metric drop-down list.

   c. Select from when you want to monitor the metric from the When drop-down list.

   d. Enter the interval (in seconds) in the Interval (seconds) field.

      60 seconds is the default value.

   e. Select the Is parameter from the Is drop-down list.

   f. Enter the threshold value in the Threshold (Number of connections in blocked or blocking state) field.

   g. Select the level of severity from Severity drop-down list.

   h. Select notification type from the Notification drop-down list.

   i. (Optional) Select the Advanced check box to configure the following advanced interval settings.

- Intervals with Exception

  1 is the default value.

- Of Last Intervals

  1 is the default value.

- Status

  Options: Enable, Disable

5. Click **Save** to save configuration for this alarm.

## Configuration

For Contrail Insights to be able to collect metrics from RabbitMQ, the RabbitMQ management plug-in must be enabled, and Contrail Insights must be configured with user credentials to collect RabbitMQ metrics.

To configure RabbitMQ monitoring:

1. Enable the RabbitMQ plug-in by issuing the following commands on the host that runs RabbitMQ:

   ```
   $ rabbitmq-plugins enable rabbitmq_management
   $ service rabbitmq-server restart
   ```

2. Contrail Insights requires RabbitMQ user credentials with privileges to read the metrics. You can use an existing RabbitMQ user with an *administrator* or *monitoring* role, or create a new user account. To create a user account with “monitoring” privileges, issue the following commands on the host that run RabbitMQ:"" "" ".*"

   ```
   $ rabbitmqctl add_user appformix mypassword
   $ rabbitmqctl set_user_tags appformix monitoring
   $ rabbitmqctl set_permissions -p / appformix "" "" ".*"
   ```

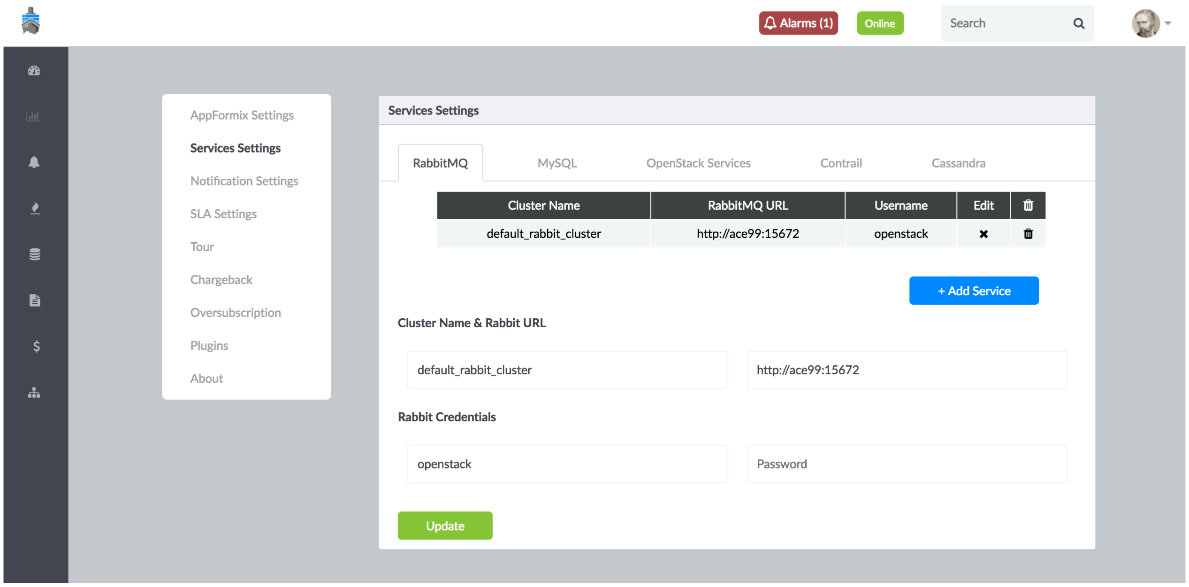
   Replace the sample `mypassword` with a strong password.

3. Verify the settings by opening http://<rabbit-host>:15672/ in a Web browser, and log in with the RabbitMQ user credentials.

4. Configure Contrail Insights with the details of the RabbitMQ cluster. Click **Settings** from the Dashboard. In the Services Settings page, select the **RabbitMQ** tab.

   Enter the Rabbit Cluster URL from Step 1. Enter the username and password from Step 2. Click **Setup**. On success, the button changes to *Submitted*. Figure 140 on page 202 shows the RabbitMQ URL and credential settings.

**Figure 140: RabbitMQ URL and Credential Settings**



| Cluster Name | RabbitMQ URL | Username | Edit | 🗑 |
|---|---|---|---|---|
| default_rabbit_cluster | http://ace99:15672 | openstack | ✖ | 🗑 |

Cluster Name & Rabbit URL

default_rabbit_cluster

http://ace99:15672

Rabbit Credentials

openstack

Password

## ScaleIO Monitoring

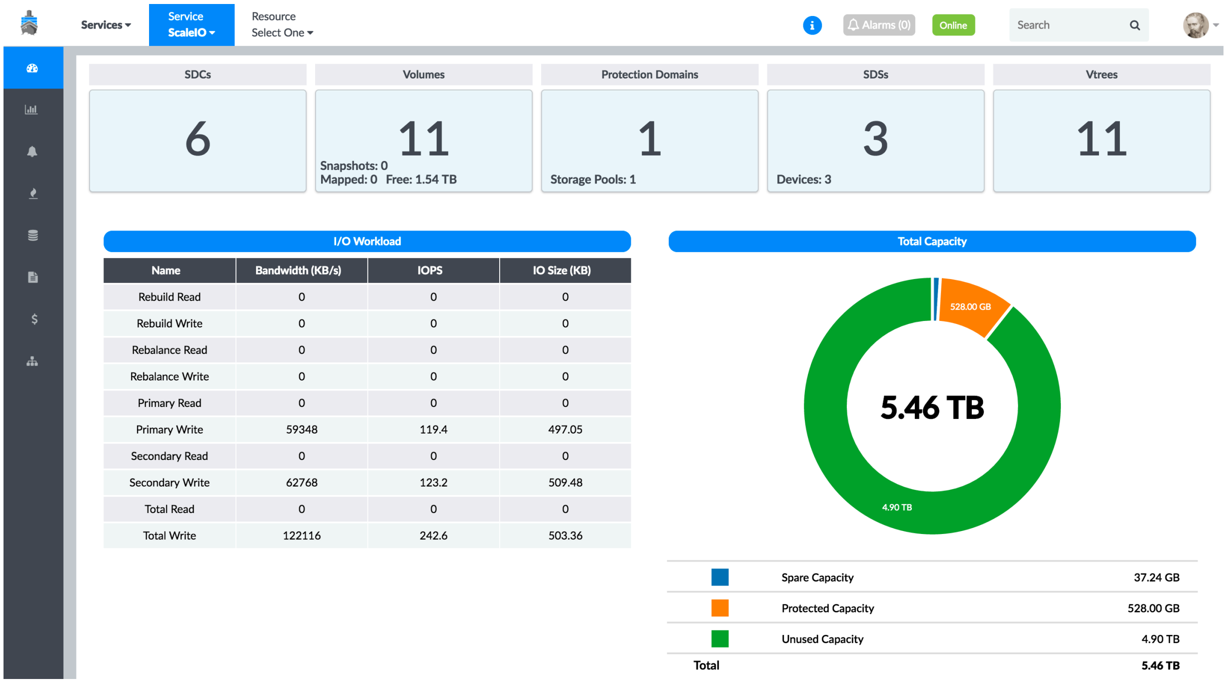
IN THIS SECTION

ScaleIO provides software-defined block storage. Contrail Insights metrics for ScaleIO performance and availability are available in real-time charts and alarms.

## Dashboard

The Contrail Insights service monitoring dashboard for a ScaleIO cluster displays the overall state of the cluster and its components. It also displays real-time storage capacity and read/write bandwidths of the cluster, as shown in Figure 141 on page 203.

### Figure 141: Real-Time Usage Metrics for ScaleIO Cluster



| SDCs | Volumes | Protection Domains | SDSs | Vtrees |
|---|---|---|---|---|
| 6 | 11<br>Snapshots: 0<br>Mapped: 0 Free: 1.54 TB | 1<br>Storage Pools: 1 | 3<br>Devices: 3 | 11 |

I/O Workload

| Name | Bandwidth (KB/s) | IOPS | IO Size (KB) |
|---|---|---|---|
| Rebuild Read | 0 | 0 | 0 |
| Rebuild Write | 0 | 0 | 0 |
| Rebalance Read | 0 | 0 | 0 |
| Rebalance Write | 0 | 0 | 0 |
| Primary Read | 0 | 0 | 0 |
| Primary Write | 59348 | 119.4 | 497.05 |
| Secondary Read | 0 | 0 | 0 |
| Secondary Write | 62768 | 123.2 | 509.48 |
| Total Read | 0 | 0 | 0 |
| Total Write | 122116 | 242.6 | 503.36 |

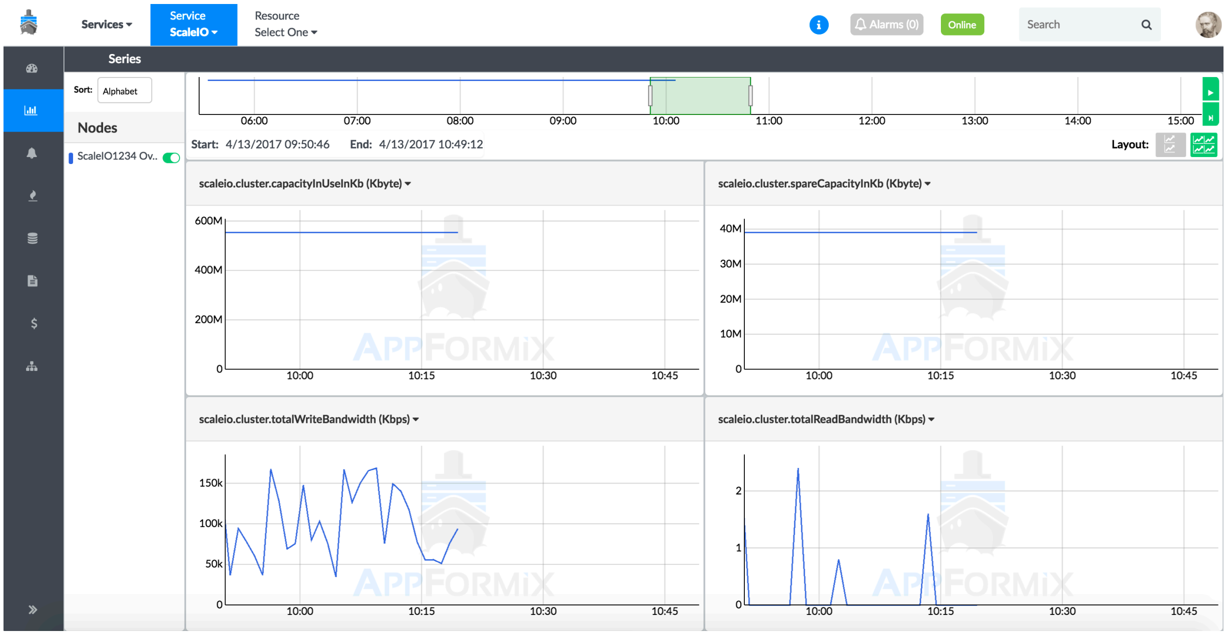
Total Capacity

| | | |
|---|---|---|
| 🟦 | Spare Capacity | 37.24 GB |
| 🟧 | Protected Capacity | 528.00 GB |
| 🟩 | Unused Capacity | 4.90 TB |
| Total | | 5.46 TB |

## Real-Time Charts

To view cluster-wide metrics in the charts, select **Services > ScaleIO** from the top context menu. Select the Charts icon from the left pane. Figure 142 on page 204 shows the ScaleIO service summary of cluster metrics in a chart view.

Figure 142: ScaleIO Service Summary of Cluster Metrics in Chart View



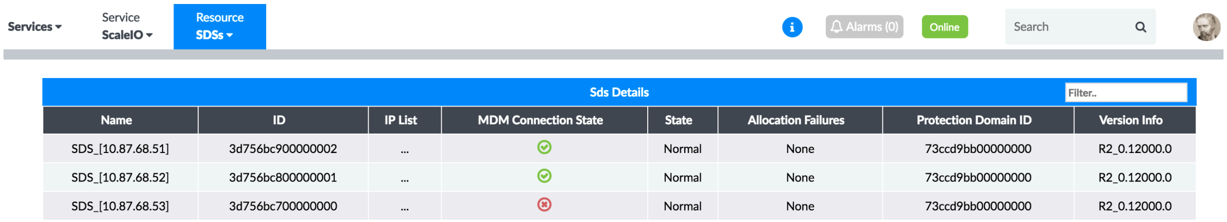
## Real-Time Status of ScaleIO Components

Contrail Insights monitors the real-time status of every element of the ScaleIO cluster. You can select an element from the **Resource** drop-down list.

SDS

Figure 143 on page 204 shows the real-time status of SDS elements of the ScaleIO cluster.

Figure 143: Real-Time Status of SDSs of the ScaleIO Cluster



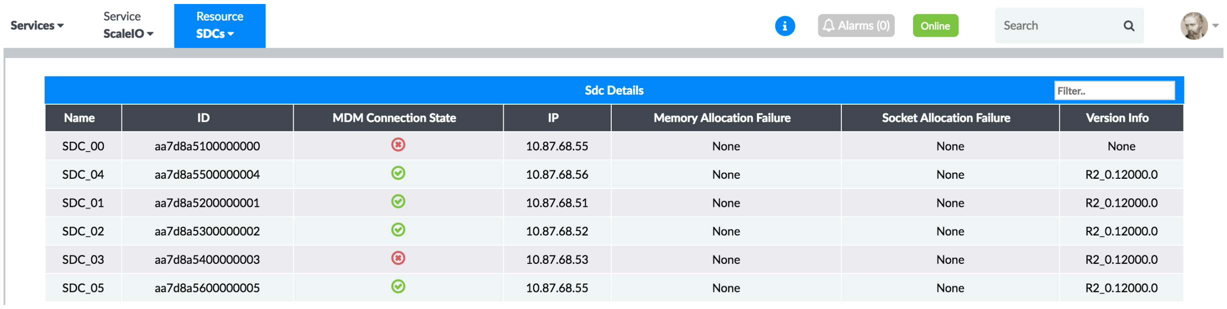
Sds Details

| Name | ID | IP List | MDM Connection State | State | Allocation Failures | Protection Domain ID | Version Info |
|---|---|---|---|---|---|---|---|
| SDS_[10.87.68.51] | 3d756bc900000002 | ... | ✓ | Normal | None | 73ccd9bb00000000 | R2_0.12000.0 |
| SDS_[10.87.68.52] | 3d756bc800000001 | ... | ✓ | Normal | None | 73ccd9bb00000000 | R2_0.12000.0 |
| SDS_[10.87.68.53] | 3d756bc700000000 | ... | ✗ | Normal | None | 73ccd9bb00000000 | R2_0.12000.0 |

SDC

Figure 144 on page 205 shows the real-time status of SDC elements of the ScaleIO cluster.

### Figure 144: Real-Time Status of SDCs of the ScaleIO Cluster



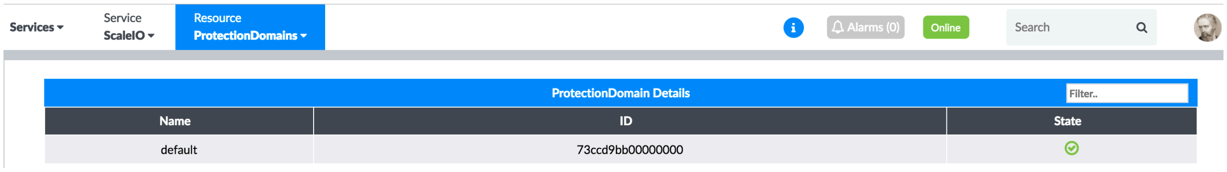
Sdc Details

| Name | ID | MDM Connection State | IP | Memory Allocation Failure | Socket Allocation Failure | Version Info |
|---|---|---|---|---|---|---|
| SDC_00 | aa7d8a5100000000 | ⊗ | 10.87.68.55 | None | None | None |
| SDC_04 | aa7d8a5500000004 | ✓ | 10.87.68.56 | None | None | R2_0.12000.0 |
| SDC_01 | aa7d8a5200000001 | ✓ | 10.87.68.51 | None | None | R2_0.12000.0 |
| SDC_02 | aa7d8a5300000002 | ✓ | 10.87.68.52 | None | None | R2_0.12000.0 |
| SDC_03 | aa7d8a5400000003 | ⊗ | 10.87.68.53 | None | None | R2_0.12000.0 |
| SDC_05 | aa7d8a5600000005 | ✓ | 10.87.68.55 | None | None | R2_0.12000.0 |

## Protection Domain

Figure 145 on page 205 shows the real-time status of the protection domains of the ScaleIO cluster.

### Figure 145: Real-Time Status of Protection Domains of the ScaleIO Cluster



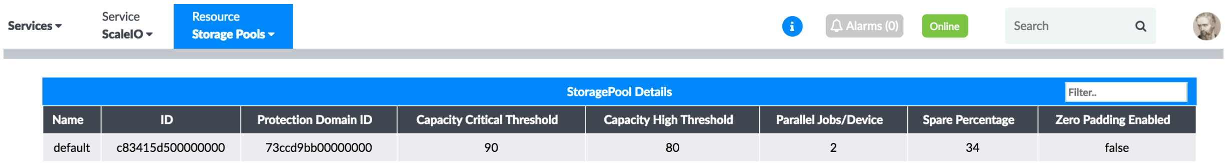
ProtectionDomain Details

| Name | ID | State |
|---|---|---|
| default | 73ccd9bb00000000 | ✓ |

## Storage Pools

Figure 146 on page 205 shows the real-time status of the storage pools of the ScaleIO cluster.

### Figure 146: Real-Time Status of Storage Pools of the ScaleIO Cluster



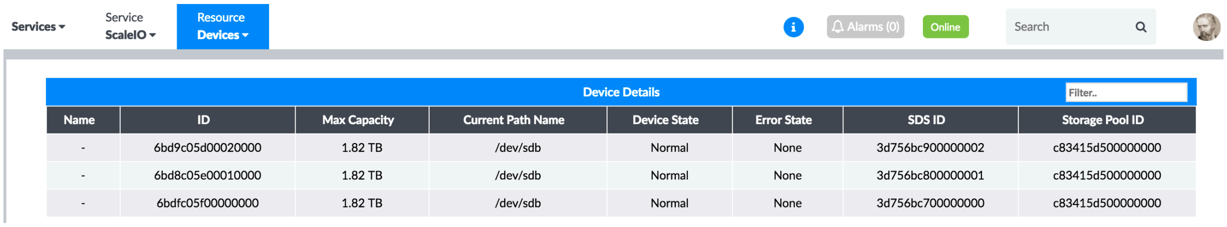
StoragePool Details

| Name | ID | Protection Domain ID | Capacity Critical Threshold | Capacity High Threshold | Parallel Jobs/Device | Spare Percentage | Zero Padding Enabled |
|---|---|---|---|---|---|---|---|
| default | c83415d500000000 | 73ccd9bb00000000 | 90 | 80 | 2 | 34 | false |

## Devices

Figure 147 on page 206 shows the real-time status of the devices of the ScaleIO cluster.

### Figure 147: Real-Time Status of Devices of the ScaleIO Cluster



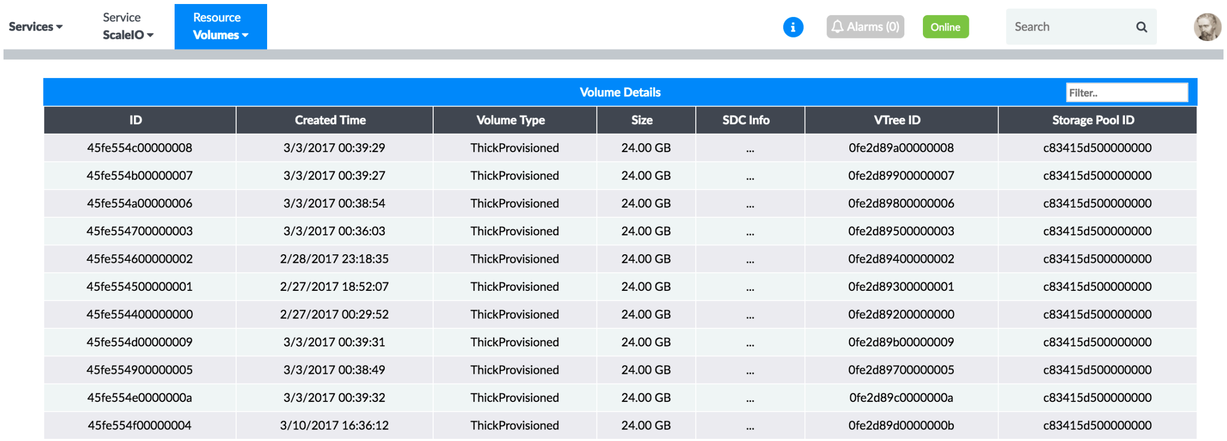
Device Details

| Name | ID | Max Capacity | Current Path Name | Device State | Error State | SDS ID | Storage Pool ID |
|---|---|---|---|---|---|---|---|
| - | 6bd9c05d00020000 | 1.82 TB | /dev/sdb | Normal | None | 3d756bc900000002 | c83415d500000000 |
| - | 6bd8c05e00010000 | 1.82 TB | /dev/sdb | Normal | None | 3d756bc800000001 | c83415d500000000 |
| - | 6bdfc05f00000000 | 1.82 TB | /dev/sdb | Normal | None | 3d756bc700000000 | c83415d500000000 |

## Volumes

Figure 148 on page 206 shows the real-time status of the volumes of the ScaleIO cluster.

### Figure 148: Real-Time Status of Volumes of the ScaleIO Cluster



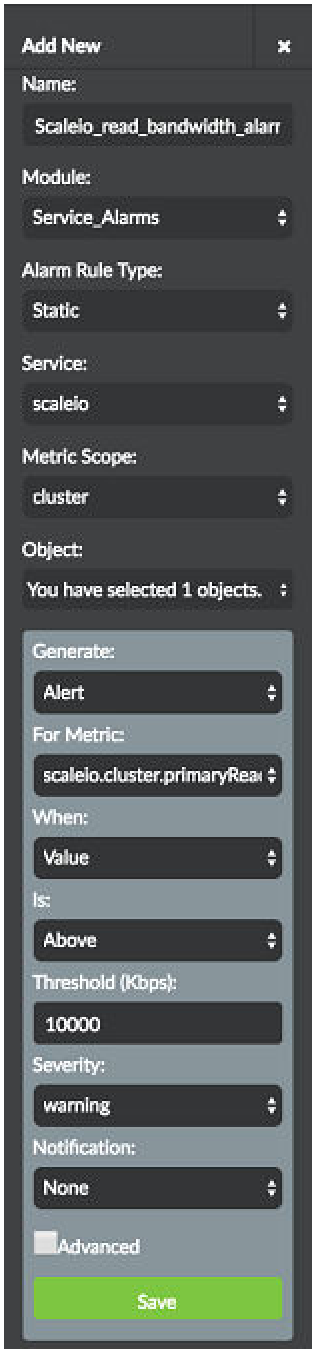
Volume Details

| ID | Created Time | Volume Type | Size | SDC Info | VTree ID | Storage Pool ID |
|---|---|---|---|---|---|---|
| 45fe554c00000008 | 3/3/2017 00:39:29 | ThickProvisioned | 24.00 GB | ... | 0fe2d89a00000008 | c83415d500000000 |
| 45fe554b00000007 | 3/3/2017 00:39:27 | ThickProvisioned | 24.00 GB | ... | 0fe2d89900000007 | c83415d500000000 |
| 45fe554a00000006 | 3/3/2017 00:38:54 | ThickProvisioned | 24.00 GB | ... | 0fe2d89800000006 | c83415d500000000 |
| 45fe554700000003 | 3/3/2017 00:36:03 | ThickProvisioned | 24.00 GB | ... | 0fe2d89500000003 | c83415d500000000 |
| 45fe554600000002 | 2/28/2017 23:18:35 | ThickProvisioned | 24.00 GB | ... | 0fe2d89400000002 | c83415d500000000 |
| 45fe554500000001 | 2/27/2017 18:52:07 | ThickProvisioned | 24.00 GB | ... | 0fe2d89300000001 | c83415d500000000 |
| 45fe554400000000 | 2/27/2017 00:29:52 | ThickProvisioned | 24.00 GB | ... | 0fe2d89200000000 | c83415d500000000 |
| 45fe554d00000009 | 3/3/2017 00:39:31 | ThickProvisioned | 24.00 GB | ... | 0fe2d89b00000009 | c83415d500000000 |
| 45fe554900000005 | 3/3/2017 00:38:49 | ThickProvisioned | 24.00 GB | ... | 0fe2d89700000005 | c83415d500000000 |
| 45fe554e0000000a | 3/3/2017 00:39:32 | ThickProvisioned | 24.00 GB | ... | 0fe2d89c0000000a | c83415d500000000 |
| 45fe554f00000004 | 3/10/2017 16:36:12 | ThickProvisioned | 24.00 GB | ... | 0fe2d89d0000000b | c83415d500000000 |

## Service Alarms

An alarm can be configured for any of the ScaleIO metrics collected. In the Alarm pane, select the **Service Alarms** module. Then select **scaleio** from the Service drop-down list. Additionally, notifications can also be configured for ScaleIO alarms, as shown in Figure 149 on page 207.

Figure 149: Alarm Input Pane for ScaleIO


Add New

Name: Scaleio_read_bandwidth_alarr

Module: Service_Alarms

Alarm Rule Type: Static

Service: scaleio

Metric Scope: cluster

Object: You have selected 1 objects.

Generate: Alert

For Metric: scaleio.cluster.primaryRea

When: Value

Is: Above

Threshold (Kbps): 10000

Severity: warning

Notification: None

□ Advanced

## Per-Instance Storage Volume Metrics

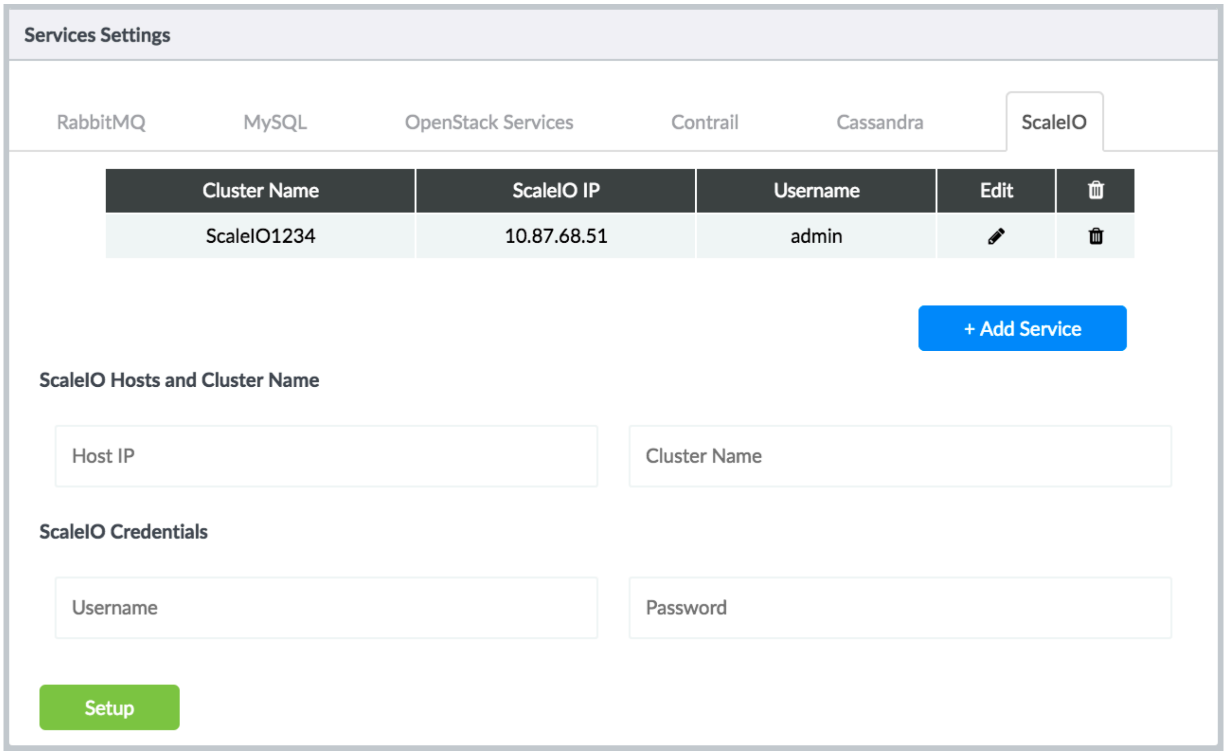
When a virtual machine mounts a storage volume, Contrail Insights Agent monitors the disk latency and throughput to the network attached storage volume. Instance metrics for storage I/O and latency (such as disk.* metrics) are available on a per-volume basis in the charts. An alarm on such a metric will indicate the volume for which the alarm triggered.

## Configuration

For Contrail Insights to monitor ScaleIO metrics, there must exist a ScaleIO user with admin authorization of the cluster. ScaleIO cluster connection details can be configured in Contrail Insights. From the Settings menu, select **Service Settings**. Then, select the **ScaleIO** tab.

Enter the cluster name and host on which ScaleIO runs. Enter the username and password, then click **Setup**. On success, the button changes to Submitted. Figure 150 on page 208 shows the ScaleIO services and credentials settings.

**Figure 150: ScaleIO Services and Credentials Settings**



| Cluster Name | ScaleIO IP | Username | Edit | 🗑 |
|---|---|---|---|---|
| ScaleIO1234 | 10.87.68.51 | admin | ✎ | 🗑 |

# Swift Service Monitoring

IN THIS SECTION

The OpenStack Object Store project, known as Swift, offers cloud storage software so that you can store and retrieve lots of data with a simple API. It's built for scale and optimized for durability, availability, and concurrency across the entire data set. Swift is ideal for storing unstructured data that can grow without bound.

## OpenStack Swift Service Hierarchy

The Object Storage system organizes data in a hierarchy, as follows:

**Account** Represents the top-level of the hierarchy.

**Container** Defines a namespace for objects. An object with the same name in two different containers represents two different objects. You can create any number of containers within an account.

**Object** Stores data content, such as documents, images, and so on. You can also store custom metadata with an object.

## Dashboard

Contrail Insights provides an easy way for you to examine the object storage usage of your OpenStack cluster. Contrail Insights automatically discovers all of the Swift Containers in your OpenStack cluster and shows you the details of these discovered Swift Containers. Contrail Insights syncs with OpenStack every minute and updates the Swift Containers information.

Select **Dashboard** > **Services** > **Swift** to view all of the Swift Containers in your OpenStack cluster in the Contrail Insights Dashboard, as shown in Figure 151 on page 210.

### Figure 151: Swift Containers in OpenStack Cluster

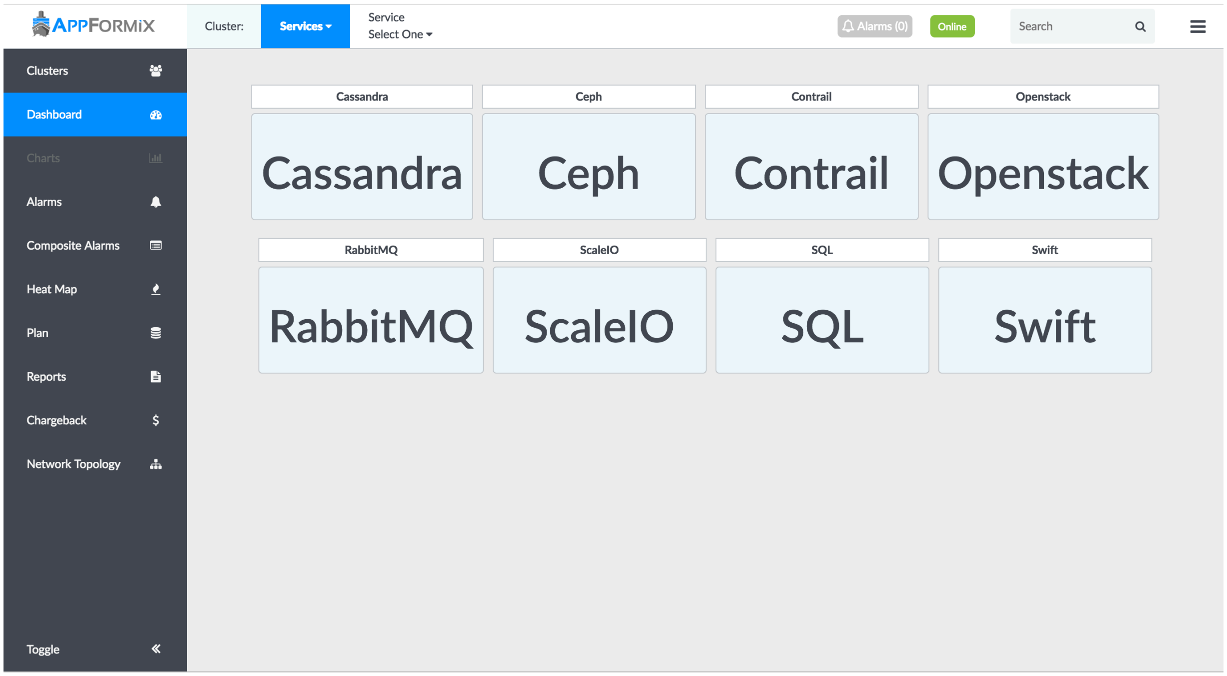
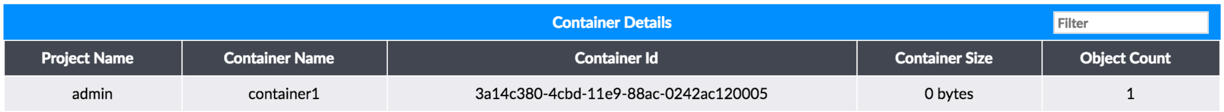


Figure 152 on page 210 shows an example of a Swift Container displaying in the Contrail Insights Dashboard.

### Figure 152: Swift Container Details


Container Details
Filter

| Project Name | Container Name | Container Id | Container Size | Object Count |
|---|---|---|---|---|
| admin | container1 | 3a14c380-4cbd-11e9-88ac-0242ac120005 | 0 bytes | 1 |

Contrail Insights provides the following information for a Swift Container: Project Name, Container Name, Container Id, Container Size, and Object Count.

### Release History Table

| Release | Description |
|---|---|
| 3.3.4 | Starting with Release 3.3.4, Contrail Insights also monitors Octavia that provides load balancing services. |
| 3.3.0 | Starting with Contrail Insights Release 3.3.0, vRouter Contrail service group is also supported. |

RELATED DOCUMENTATION

*Alarms*

*Contrail Insights JTI (UDP) Monitoring*

*Contrail Insights JTI (gRPC) Monitoring*

*Contrail Insights SNMP Monitoring*

*Contrail Insights User-Defined Plug-Ins*

# Contrail Insights VNF Monitoring

IN THIS SECTION

## Contrail Insights VNF Configuration

Contrail Insights will identify all the instances on the hosts/devices where Contrail Insights Agent is installed. You need to specify the following in your `group_vars/all`:

```
appformix_kvm_instance_discovery: true
```

After you install Agent on hosts/devices, Contrail Insights identifies all the instances running on those hosts/devices.

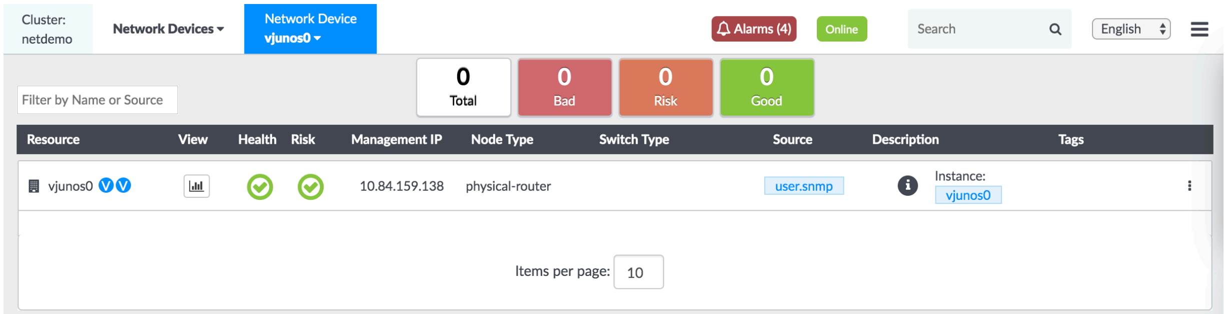
In addition to posting those instances, Contrail Insights will also post instances with name prefixed as `vsrx` and `vjunos` network devices. You can go to the Contrail Insights Dashboard **Settings -> Network Devices** and input essential information for those devices.

If your VNF has a name prefix other than `vsrx` and `vjunos`, you can manually add those network devices from the Settings page. Select **Virtual** as your Chassis Type and input this VNF's Instance ID.

## Contrail Insights VNF Monitoring

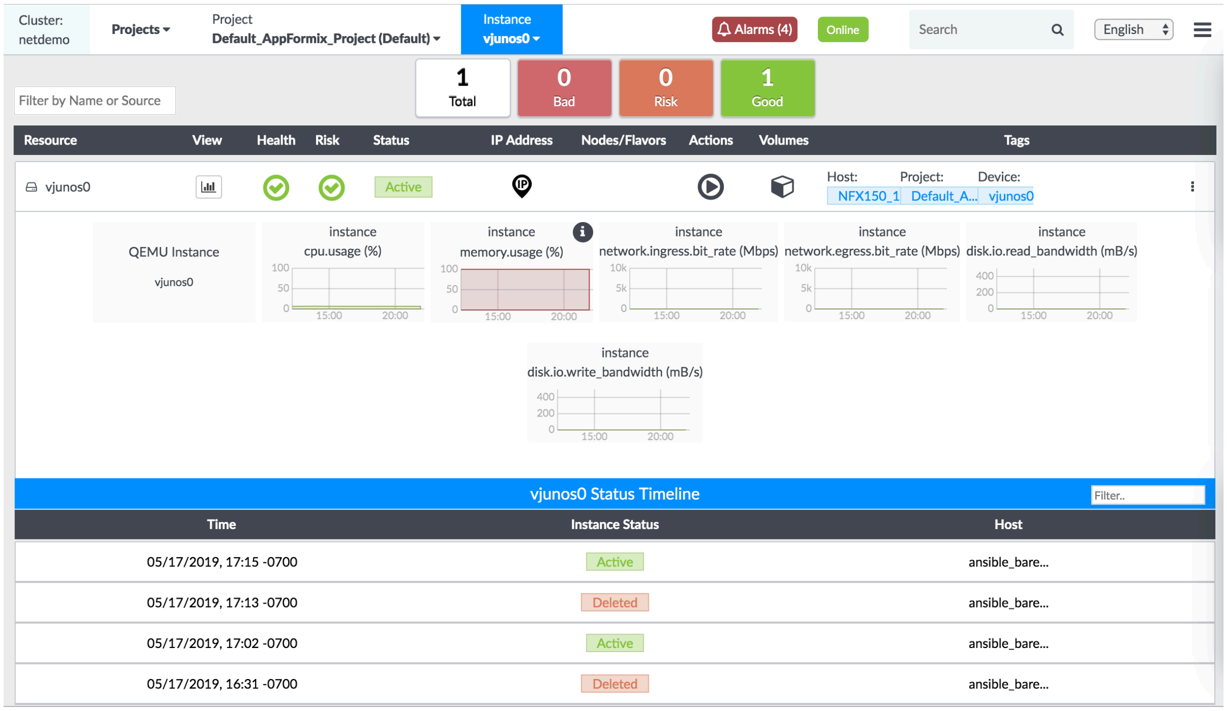
Contrail Insights supplies VNF monitoring by using both instance metrics (instance CPU, memory, disk usage, and so on.) and SNMP Network Device metrics (interface, TCP states, routes metrics, and so on.). You can travel between instance and network device of this VNF easily. As you can see in Figure 153 on page 212 and Figure 154 on page 212, you can find the `instance` tag in device view and `device` tag in instance view.

### Figure 153: Instance Tag in Network Device View



| Resource | View | Health | Risk | Management IP | Node Type | Switch Type | Source | Description | Tags |
|---|---|---|---|---|---|---|---|---|---|
| vjunos0 | | ✓ | ✓ | 10.84.159.138 | physical-router | | user.snmp | | Instance: vjunos0 |

### Figure 154: Device Tag in Instance View

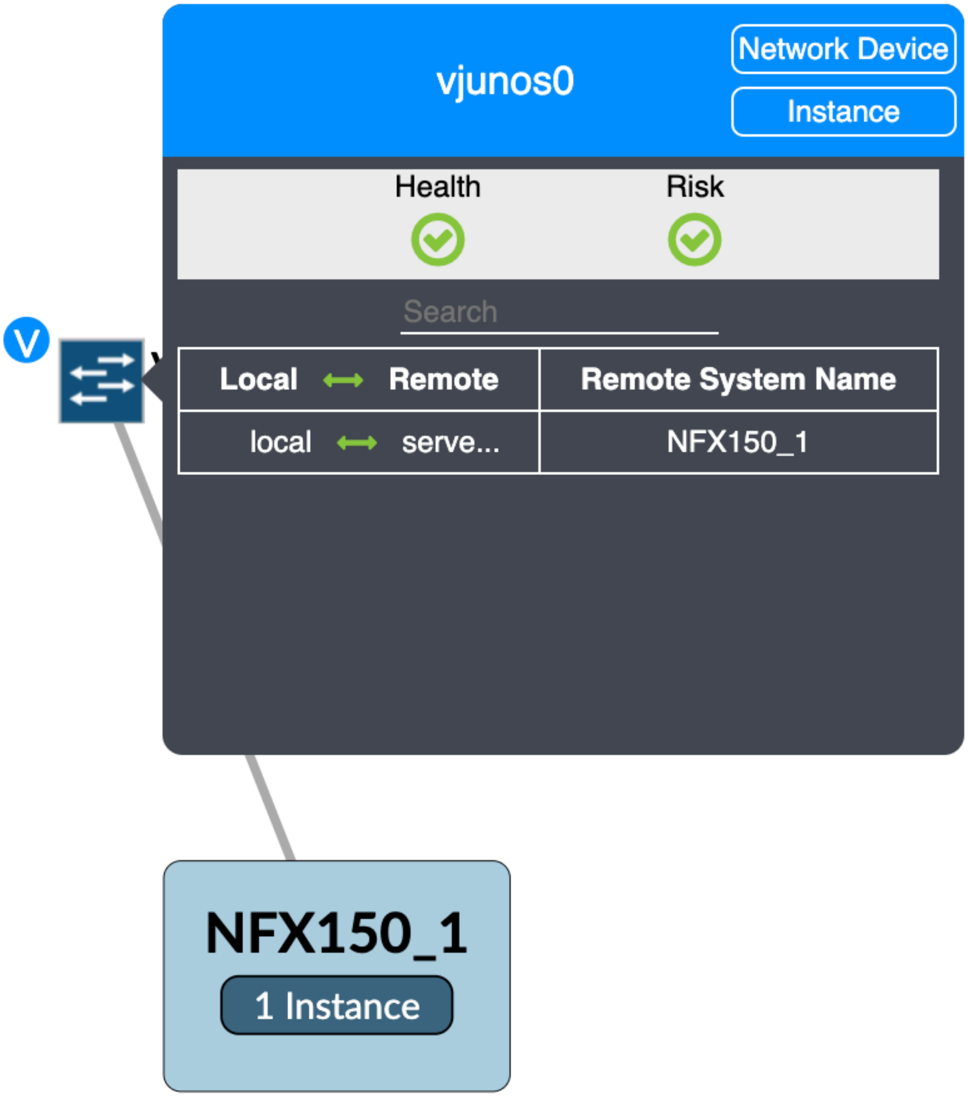


| Resource | View | Health | Risk | Status | IP Address | Nodes/Flavors | Actions | Volumes | Tags |
|---|---|---|---|---|---|---|---|---|---|
| vjunos0 | | ✓ | ✓ | Active | | | | | Host: NFX150_1 Project: Default_A... Device: vjunos0 |

| Time | Instance Status | Host |
|---|---|---|
| 05/17/2019, 17:15 -0700 | Active | ansible_bare... |
| 05/17/2019, 17:13 -0700 | Deleted | ansible_bare... |
| 05/17/2019, 17:02 -0700 | Active | ansible_bare... |
| 05/17/2019, 16:31 -0700 | Deleted | ansible_bare... |

For the Network Topology page, click the VNF object to show the connection between this VNF and its host. Also, the pop up of this object is linked to both the Network Device and Instance view.

**Figure 155: Connection Between VNF and Host**



| Local ↔ Remote | Remote System Name |
|---|---|
| local ↔ serve... | NFX150_1 |

# Contrail Insights JTI (UDP) Monitoring

IN THIS SECTION

## Configure JTI Device

Contrail Insights supports UDP-based Junos Telemetry Interface (JTI) from network devices. With network devices supporting UDP-based JTI, Contrail Insights is able to stream data from the devices.

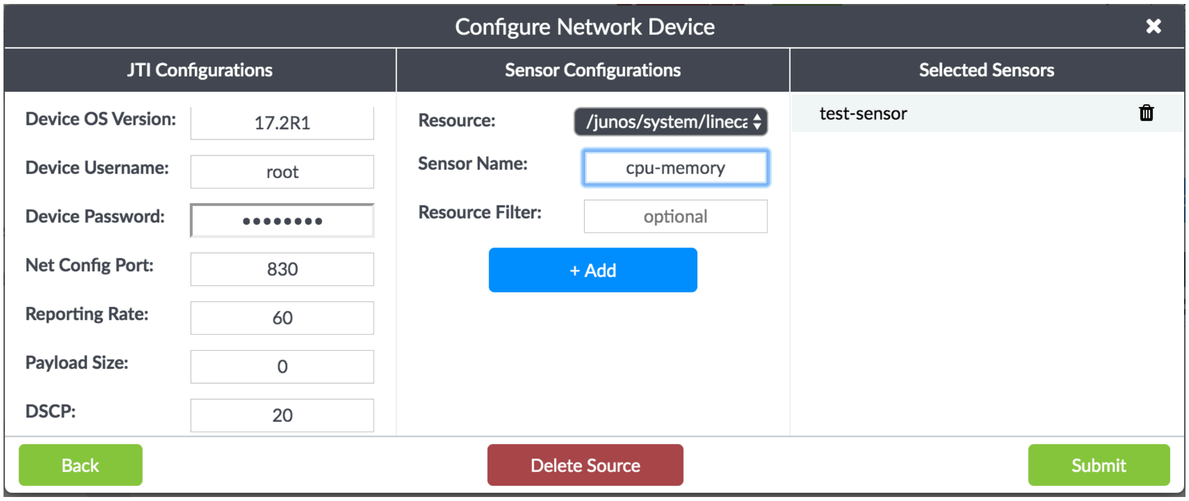
When configuring JTI devices, you can select all the sensors that need to be monitored. Using the required and optional configuration parameters that you input in the Configure Network Device page, Contrail Insights will push the configuration to the device and enable the device to stream data to collectors.

To configure a JTI device:

1. Select **Settings** in the top right of the Dashboard, then select **Network Devices**.

2. Click **+Add Device** and complete the configuration parameter fields. See Figure 156 on page 215.

**Figure 156: JTI Configuration Parameters in Configure Network Device Page**



Device OS Version: 17.2R1

Device Username: root

Device Password: ••••••••

Net Config Port: 830

Reporting Rate: 60

Payload Size: 0

DSCP: 20

Resource: /junos/system/lineca

Sensor Name: cpu-memory

Resource Filter: optional

test-sensor

3. To allow Contrail Insights to configure the network device, have the following settings on your device and supply the device `username` and `password`:

```
set system services netconf ssh
```

Following is an example configuration that Contrail Insights adds on the device:

```
streaming-server appformix-telemetry {
    remote-address x.x.x.x; # collector ip, Contrail Insights will automatically assign the
collector
    remote-port 42596;
}
export-profile appformix {
    local-address y.y.y.y; # Device local ip to send out data, need to be a revenue port
    local-port 21112;
    dscp 20;
    reporting-rate 60;
    format gpb;
    transport udp;
}
sensor test-sensor {
    server-name appformix-telemetry;
    export-name appformix;
    resource /junos/system/linecard/interface/;
}
```

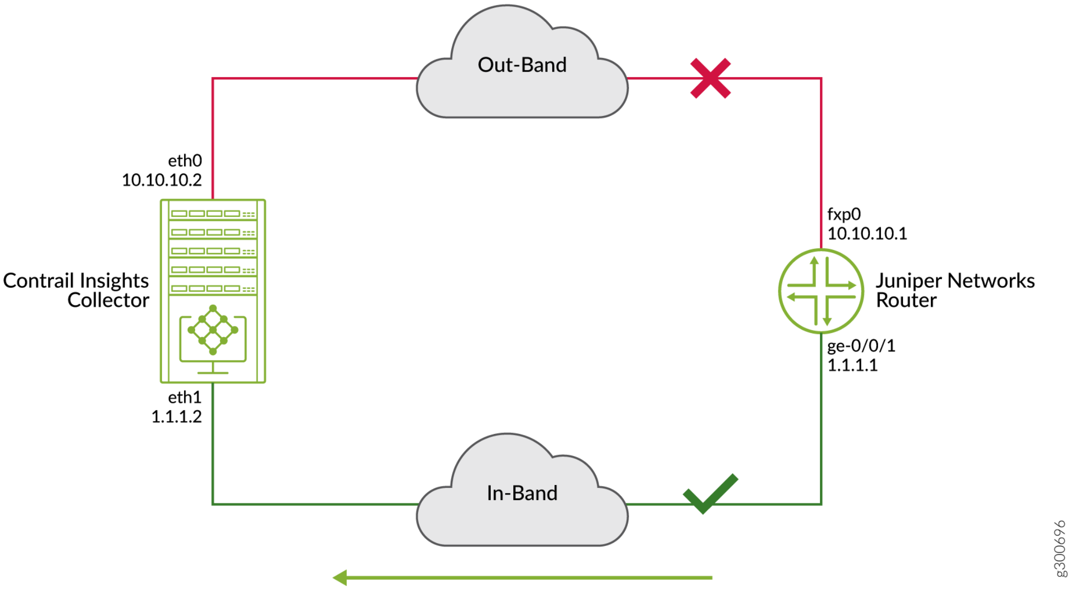
4. In addition, you need to enable JTI plug-ins in your `group_vars/all` to enable JTI monitoring in Contrail Insights and define `appformix_install_jti_dependencies`:

```
appformix_plugins:
  - { plugin_info: 'certified_plugins/jti_config_all_sensors.json' }
appformix_install_jti_dependencies: true
```

## JTI Monitoring Special Requirements

Traffic from JTI sensors is injected into the forwarding path, so the collector must be reachable by means of in-band connectivity. JTI sensor traffic does not get forwarded through the router's management interface (for example, fxp0). Contrail Insights Collector in Figure 157 on page 216 includes Contrail Insights Agent and network devices.

**Figure 157: Traffic from JTI Sensors through In-Band Connectivity**



In Contrail Insights, you can edit `ManagementIp` and `MetaData.JtiConfig.LocalAddress` in the device JSON file. If `MetaData.JtiConfig.LocalAddress` is not specified, Contrail Insights uses the `ManagementIp` as the device in-band IP setting in device. In addition, Contrail Insights configures the device so that it streams its JTI data to one of the `appformix_network_agents` nodes.

You can specify jti_inband_ip in the Ansible inventory files to specify the in-band IP address of the collector (server). See Figure 157 on page 216.

```
[appformix_network_agents]
10.10.10.2 ansible_ssh_user='user' ansible_ssh_pass='pwd' jti_inband_ip='1.1.1.2'
```

**NOTE**: If the jti_inband_ip is not specified in the Ansible inventory file, Contrail Insights uses the hostname of the appformix_network_agents node.

## JTI Out of Band Configuration

Contrail Insights configures the devices properly based on user input including sensor name, sensor path, collector IP address, and device source IP address.

In some scenarios, user does not want to share credentials with Contrail Insights. As a result, Contrail Insights does not have the device credentials to configure the devices. Alternatively, you can use out of band JTI configuration scripts in SDK instead. Contrail Insights will discover all JTI network devices in your environment and push configurations to your devices using the script. This script only works when you have only one JTI collector in your setup.

Example out_of_band_jti_configuration.py script:

```
from jnpr.junos import Device
from jnpr.junos.utils.config import Config
import sys
import rest
import json
import os

# 1) This script runs inside appformix-controller container.
# 2) It assumes that appformix_token.rst file is present in the current directory
# 3) It assumes that NETCONF user and password is supplied as arg1, arg2 for
# the script and netconf ssh port as arg4
# 4) It takes collector inband ip as a argument as arg3. It assumes that there
# is only one collector for JTI.
# TODO: Read JTI distribution map from plugin definition, read jti_inband_ip
# from server definition and assign the devices to its correct collector. The
# blocking item here is we don't have v2 API for plugin definition
```

```
with open('appformix_token.rst') as json_file:
    data = json.load(json_file)
APPFORMIX_MASTER_TOKEN = data['Token']['TokenId']
DEVICE_NETCONF_USERNAME = sys.argv[1]
DEVICE_NETCONF_PASSWORD = sys.argv[2]
# jti_inband_ip of appformix_platform
APPFORMIX_CONFIG_COLLECTOR_DATA_IP = sys.argv[3]
NETCONF_PORT = sys.argv[4]
# You can change the following parameters based on requirement
LOCAL_PORT = '21112'
PAYLOAD_SIZE = '5000'
APPFORMIX_JTI_LISTEN_PORT = '42596'

HEADERS = {'content-type': 'application/json',
           'X-Auth-Type': 'appformix',
           'X-Auth-Token': APPFORMIX_MASTER_TOKEN}
url = 'http://localhost:80/appformix/controller/v2.0/network_devices'

resp = rest.get(url=url, headers=HEADERS)
result = json.loads(resp.text)
devices = []

for entry in result['NetworkDeviceProfile']:
    if 'user.jti' in entry['NetworkDevice']['Source']:
        device_config = {'ip': entry['NetworkDevice']['ManagementIp'],
                         'sensor_list':
                         entry['NetworkDevice']['MetaData']['JtiConfig']['SensorList'],
                         'device_data_ip':
                         entry['NetworkDevice']['MetaData']['JtiConfig']['LocalAddress'],
                         'report_rate':
                         entry['NetworkDevice']['MetaData']['JtiConfig']['ReportRate']}
        devices.append(device_config)


for entry in devices:
    # Create a Device Object
    print "Connecting to device {}".format(entry['ip'])
    dev = Device(host=entry['ip'],
                 user=DEVICE_NETCONF_USERNAME,
                 password=DEVICE_NETCONF_PASSWORD,
                 port=NETCONF_PORT)
    try:
        dev.open()
        cu = Config(dev)
    except Exception as e:
        print "Fail to connect to device {}: {}".format(
            entry['ip'], e)
        continue

    print "Configuring the streaming-server in device"
    # Update the streaming-server, update the collector' in_band ip
    msg = ("set services analytics streaming-server " +
           "appformix-telemetry remote-address {} remote-port {}").format(
                APPFORMIX_CONFIG_COLLECTOR_DATA_IP, APPFORMIX_JTI_LISTEN_PORT)
    cu.load(msg, format='set')

    print "Configuring the export-profile in device"
    # Update the analytics export-profile, update the device's in_band ip
    msg = ("set services analytics export-profile appformix " +
           "local-address {}").format(entry['device_data_ip'])
    cu.load(msg, format='set')
    msg = ("set services analytics export-profile appformix " +
           "transport udp format gpb reporting-rate {} " +
           "local-port {} payload-size {}")
    msg = msg.format(entry['report_rate'], LOCAL_PORT, PAYLOAD_SIZE)
    cu.load(msg, format='set')

    # Commit the change to device, rollback if commit fail
    try:
        cu.commit()
    except Exception as e:
        print "Fail to configure device {}".format(e)
        cu.rollback()
        continue

    # Add sensor to the device
    for sensor in entry['sensor_list']:
        print "Configuring the sensor {} in device".format(sensor['Resource'])
        msg = ("set services analytics sensor {} resource {} " +
               "export-name appformix server-name appformix-telemetry")
        msg = msg.format(sensor['Name'], sensor['Resource'])
        cu.load(msg, format='set')
    try:
        cu.commit()
    except Exception as e:
        print "Fail to configure device sensor {}".format(e)
```

```
        cu.rollback()
    dev.close()
    print "Closing connection to device {}".format(entry['ip'])
```

# Troubleshooting

1. On the Contrail Insights Platform host, check if the Agent is listening on UDP port 42596 by running the following command.

   ```
   netstat -lanp | grep 42596
   ```

   If not, check if plug-in is posted. Check the `jti_network_device` plug-in from `plugin_definition` endpoint in the Contrail Insights Platform API to see if the distribution_map in **Config > ObjectList** is correct.

2. Check the network device configuration. On the device, from the CLI Configuration mode, running `show service analytics` should have:

   - A streaming server named "appformix-telemetry”
   - An export profile named “appformix”
   - And a sensor named “Interface_Sensor”

   If any of these items are missing, look at the following file and check the log for authentication failures.

   ```
   /var/log/appformix/controller/appformix/appformix_celery_queue_server_worker_celery.log
   ```

3. Check if data is being received at Contrail Insights Platform host. Run `tcpdump` to check if data is received by the Contrail Insights Platform host on UDP port 42596. If data is not being received from the network device on UDP port 42596, then it is likely that the in-band connectivity is not working. The `local-address` configured in streaming server "appformix-telemetry" must be able to reach the Contrail Insights Platform host address configured in the export profile.

4. Check if data is being dropped by kernel. Following is an example output of `tcpdump`:

   ```
    root@ubuntu:/home/acelio# tcpdump -nli p1p1 port 42596
    tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
   ```

```
 listening on p1p1, link-type EN10MB (Ethernet), capture size 65535 bytes
 14:18:32.373370 IP 10.87.68.120.21112 > 10.87.68.13.42596: UDP, length 2320
```

If your output is similar to the following example, it indicates AppFormix-VM is dropping packets coming from the device, which can be a maximum transmission unit (MTU) issue:

```
root@ubuntu:~# tcpdump -nli eth0 port 42596
 tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
 listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
 16:28:25.165580 IP 10.27.73.254.21112 > 10.27.73.155.42596: UDP, bad length 3245 > 1472
```

5. If you are using CentOS or Red Hat software, check your IPtables rules if they block the traffic. You can run the following commands to remove IPtables rules in your AppFormix-VM:

```
iptables --flush
 iptables -P FORWARD ACCEPT
 iptables -P OUTPUT ACCEPT
 iptables -P INPUT ACCEPT
 service iptables save
```

   These commands will remove all IPtables rules blocking the traffic and add rules accepting traffic.

6. You might also need to disable `rp_filter` on the collector side:

```
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
 echo 0 > /proc/sys/net/ipv4/conf/{jti_interface_name}/rp_filter
```

7. Further debugging can be done using the following script bundled with Contrail Insights. This script should be run on the Contrail Insights Agent that is monitoring the affected network device:

```
cd /opt/appformix/manager/tailwind/manager/
 source ../ven/bin/activate
 python check_jti_device_test.py
```

   This script will print out data if Contrail Insights receives JTI messages from the socket. If you do see tcpdump in the port 42596 but no data from this script, it means message has been dropped by the kernel.

## Packages Needed for JTI Network Device Monitoring

Currently, you need to specify on which Agents JTI network devices should stream their metrics to. On those Contrail Insights Agents, you need to install the following three packages:

```
sudo apt-get install netcat
sudo apt-get install protobuf-compiler
sudo apt-get install libprotobuf-dev
```

These packages are needed for receiving and decoding JTI messages.

### RELATED DOCUMENTATION

*Configure Network Devices from the UI*

*Contrail Insights Auto Discovery of Network Devices from Contrail Networking*

*Contrail Insights Network Device Monitoring Common Issues*

*Contrail Insights JTI (gRPC) Monitoring*

*Contrail Insights SNMP Monitoring*

# Contrail Insights JTI (gRPC) Monitoring

IN THIS SECTION

## Set Up gRPC-based Streaming

Starting with Junos OS Release 16.1R3, you can stream telemetry data for various network elements through gRPC, an open source framework for handling remote procedure calls based on TCP. The Junos Telemetry Interface relies on a so-called push model to deliver data asynchronously, which eliminates polling.

The Junos Telemetry Interface and gRPC streaming are supported on QFX10000 and QFX5200 switches starting with Junos OS Release 17.2R1. The Junos Telemetry Interface and gRPC streaming is supported on QFX5110 switch starting with Junos OS Release 17.3R1. For more information on supported devices, and to configure gRPC for Junos Telemetry Interface, see gRPC Services for Junos Telemetry Interface.

For all Juniper devices that run a version of Junos OS with upgraded FreeBSD kernel, you must install the Junos Network Agent software package, which provides the interfaces to manage gRPC subscriptions. For Juniper Network devices that run other all other versions of the Junos OS, this functionality is embedded in the Junos OS software. For more information, see Installing the Network Agent Package. You must also install the OpenConfig for Junos OS module and the YANG models. For more details, see Understanding OpenConfig and gRPC on Junos Telemetry Interface.

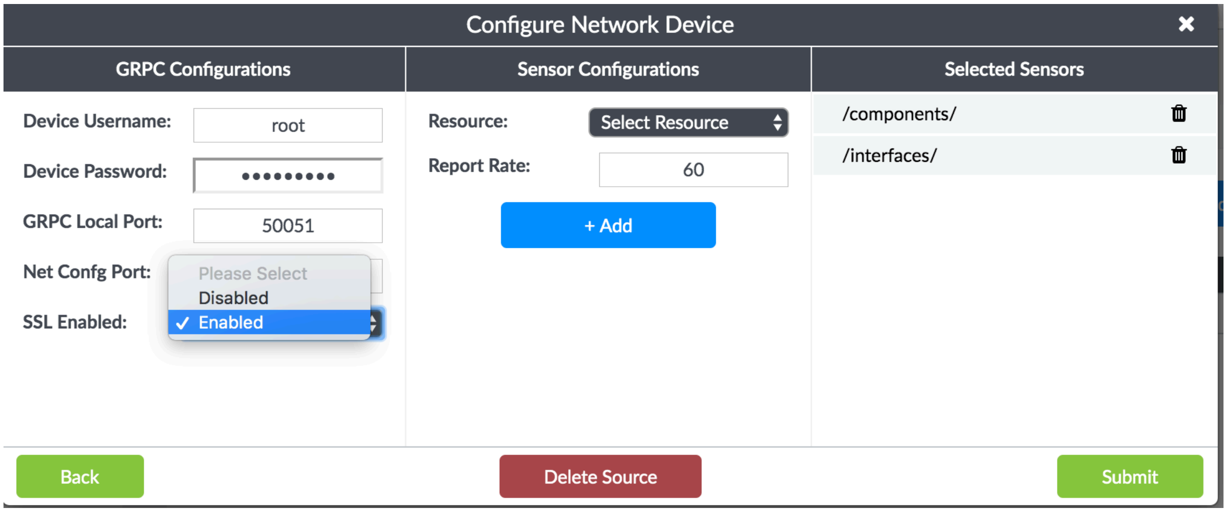
Contrail Insights automatically configures the network device based on the JSON file you provide. Contrail Insights will stream the gRPC metrics with a default interval of `60` seconds.

After completing the above steps, verify the following configuration on the network device:

```
root@B5R4-QFX5K-1> show configuration | display set | grep exten
set groups appformix-grpc system services extension-service request-response grpc clear-text
port 50051
set groups appformix-grpc system services extension-service request-response grpc skip-
authentication
set groups appformix-grpc system services extension-service notification allow-clients address
0.0.0.0/0

{master:0}
root@B5R4-QFX5K-1> show system processes | grep na-
7563  -  S        2:29.58 /usr/sbin/na-mqttd -c /opt/telemetry/na-mqttd/na-mqt
7572  -  I        0:44.58 /usr/sbin/na-grpcd -c /opt/telemetry/na-grpcd/na-grp

{master:0}
root@B5R4-QFX5K-1> show system software | grep open
junos-openconfig-x86-32-0.0.0I20180615_1002_rbu-builder  --  junos openconfig
```

While configuring gRPC devices, you can select to enable SSL on the gRPC subscription. Select **Settings** in the top right of the Dashboard, **Network Devices** > **+Add Device**. Also, see section “Secure Socket Layer (SSL) gRPC Configuration.”

**Figure 158: Configure gRPC Network Device Telemetry and Enable SSL**



Device Username: root

Device Password: •••••••••

GRPC Local Port: 50051

Net Confg Port:

SSL Enabled: Please Select / Disabled / ✓ Enabled

Resource: Select Resource

Report Rate: 60

/components/ 🗑

/interfaces/ 🗑

In addition, you need to enable gRPC plug-in in your `group_vars/all` file to enable gRPC monitoring in Contrail Insights:

```
appformix_plugins:
  - { plugin_info: 'certified_plugins/grpc_config_all_sensors.json' }
```

To allow Contrail Insights to configure the network device, have the following settings on your device and supply the device `username` and `password`:

```
set system services netconf ssh
```

## Unsecured gRPC Configuration

Following is the configuration Contrail Insights adds on the device when you select `SSLEnabled = False` when configuring the device.

```
root@5b9-qfx2# show groups
appformix-grpc {
    system {
        services {
            extension-service {
                request-response {
                    grpc {
                        clear-text {
                            port 50051;
                        }
                        skip-authentication;
                    }
                }
                notification {
                    allow-clients {
                        address 0.0.0.0/0;
                    }
                }
            }
        }
    }
}
root@5b9-qfx2# show apply-groups
apply-groups appformix-grpc;
```

## Secure Socket Layer (SSL) gRPC Configuration

In order for Contrail Insights to subscribe to devices over SSL technology, complete the following steps in advance of enabling SSL.

1. Certificates for all devices need to be signed by one single certificate authority (CA).
2. Common Name (CN) value specified for the certificate used by a particular device, should be that device's Domain Name System (DNS) name.

3. Certificates need to be preloaded on the device as name `appformix` by running the following command:

   ```
   set security certificates local appformix <path_to_certificate>
   ```

4. When configuring the devices in Contrail Insights, enter the device DNS name or IP address in the `ManagementIp` field. The `ManagementIp` should be able to resolve (translate) the device DNS name from the Contrail Insights Platform node.

   Example configuration Contrail Insights puts on the device:

   ```
   root@5b9-qfx2# show groups
   appformix-grpc {
       system {
           services {
               extension-service {
                   request-response {
                       grpc {
                           ssl {
                               port 50051;
                               local-certificate appformix;
                           }
                           skip-authentication;
                       }
                   }
                   notification {
                       allow-clients {
                           address 0.0.0.0/0;
                       }
                   }
               }
           }
       }
   }
   root@5b9-qfx2# show apply-groups
   apply-groups appformix-grpc;
   ```

## Distribute gRPC Network Device CA Using Ansible

In order for Contrail Insights to have secure connections between collectors (Contrail Insights Agent and devices), the collector needs to have the CA, which signed all of the devices' certificates, in `/opt/appformix/etc/cert/`.

Then use Ansible to distribute the CA to all Contrail Insights Agents. Add the following in your `group_vars/all` file and then run the playbook.

```
appformix_grpc_ssl_ca: <path to your certificate file>
```

### RELATED DOCUMENTATION

gRPC Services for Junos Telemetry Interface

Understanding OpenConfig and gRPC on Junos Telemetry Interface

# Contrail Insights SNMP Monitoring

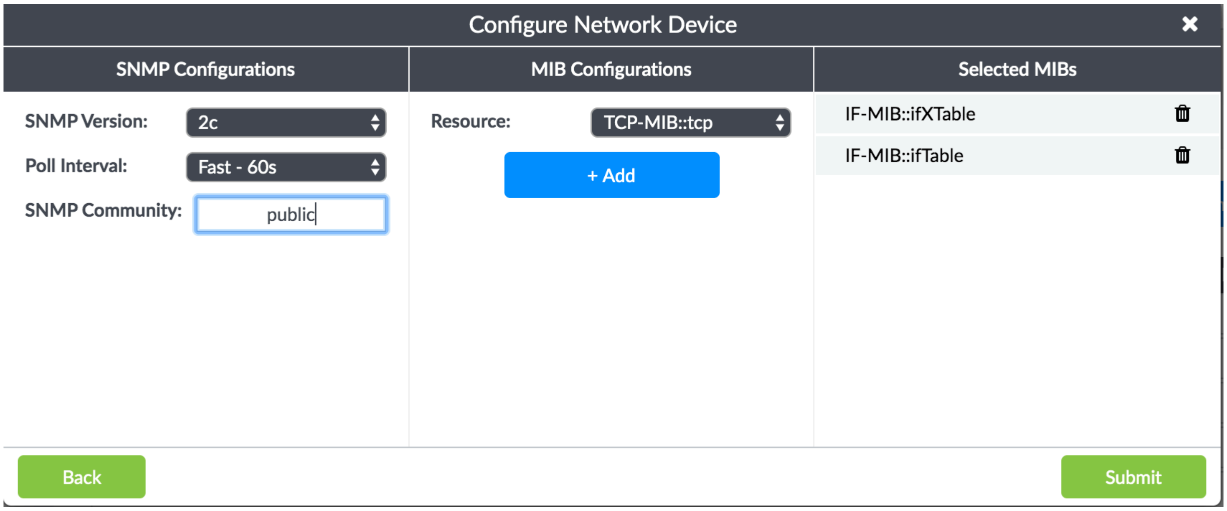
IN THIS SECTION

## Configure SNMP Device

Contrail Insights supports SNMPv2c and SNMPv3 monitoring for Network Devices as well as SNMP trap (v2c/v3) monitoring.

While adding SNMP devices, you can select all of the MIBs that need to be monitored. With the required and optional configuration parameters that you input in the Configure Network Device page, Contrail Insights will start to poll SNMP data from this device and display all the SNMP traps received from this device.

Figure 159: SNMP Configuration Parameters in Configure Network Device Page



SNMP Version: 2c

Poll Interval: Fast - 60s

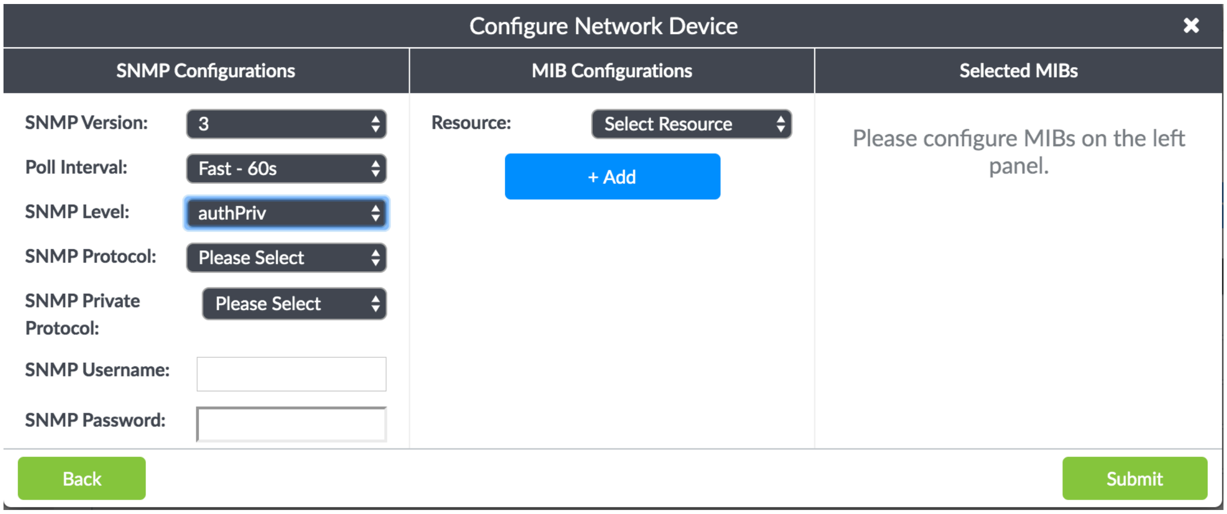
SNMP Community: public

Resource: TCP-MIB::tcp

IF-MIB::ifXTable 🗑

IF-MIB::ifTable 🗑

Figure 160: SNMP Versions Require Different Configuration Parameters



SNMP Version: 3

Poll Interval: Fast - 60s

SNMP Level: authPriv

SNMP Protocol: Please Select

SNMP Private Protocol: Please Select

SNMP Username:

SNMP Password:

Resource: Select Resource

Please configure MIBs on the left panel.

Note that different SNMP versions need different parameters. For example:Parameter Poll Interval determines the period Contrail Insights polls SNMP data from the device, SNMP Engine ID is required when you want to enable SNMP trap monitoring for this device (with SNMP version 3).

**Parameter Poll Interval** Determines the period Contrail Insights polls SNMP data from the device.

| | |
|---|---|
| **SNMP Engine ID** | Required when you want to enable SNMP trap monitoring for the device (with SNMP version 3). |

In addition, you need to enable SNMP plug-ins in your `group_vars/all` file to enable SNMP monitoring in Contrail Insights and define `appformix_install_snmp_dependencies`, for example:

```
appformix_plugins:
   - { plugin_info: 'certified_plugins/snmp_network_device_usage.json' }
   - { plugin_info: 'certified_plugins/snmp_config_ifxtable_mib.json' }
appformix_install_snmp_dependencies: true
```

All of the available SNMP plug-ins are located in `certified_plugins` in your Ansible installation folder. Or you can also enable the plug-in `appformix_network_device_factory_plugins` for all SNMP network device plug-ins.

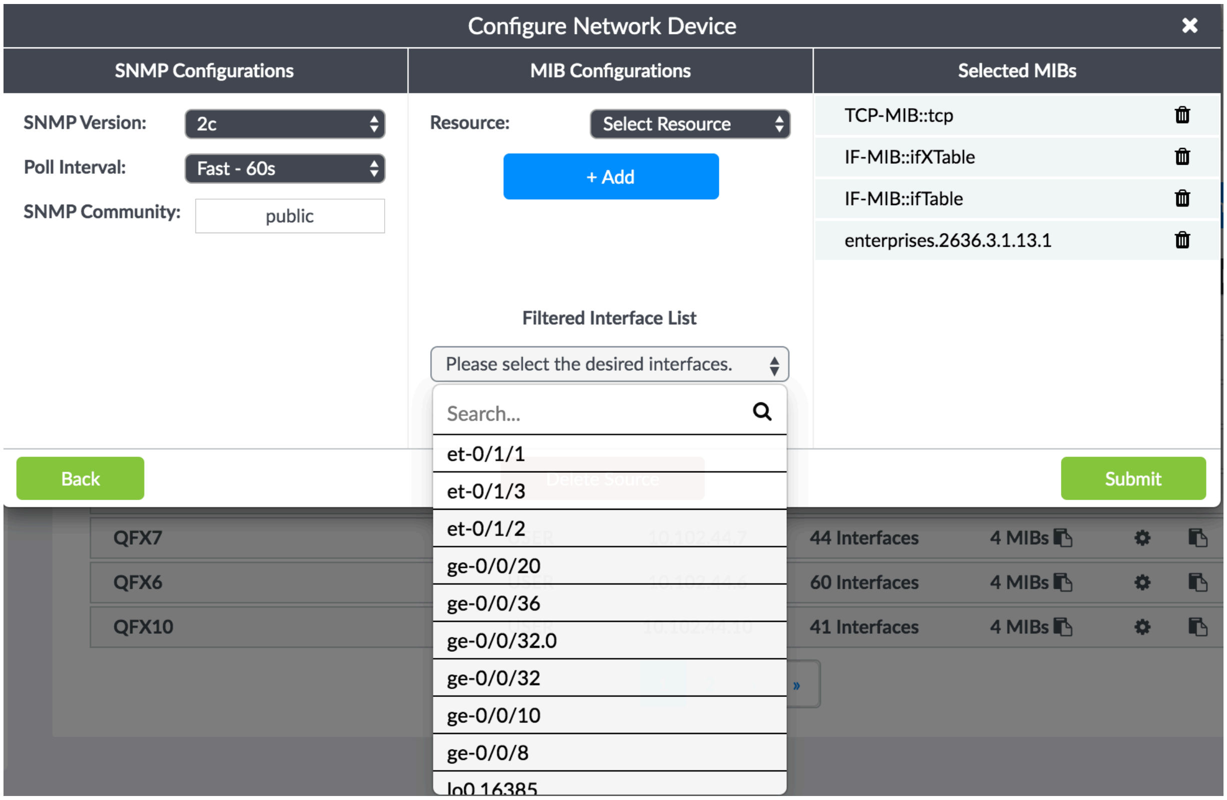
For SNMP trap information, see "SNMP Traps in Contrail Insights" on page 71.

## SNMP Filter Interface List

Generally, Snmpwalk has high device resource usage. To run Snmpwalk against only some of the interfaces, you can provide Contrail Insights with a list, and Contrail Insights will run Snmpwalk (ifTable, ifXtable MIB) only on those specified interfaces.

The filtered interface list can only be configured after the device is added to Contrail Insights. Contrail Insights discovers the device interfaces after device is added to Contrail Insights. Then you can edit the device from UI and select the specific interfaces to monitor.

Figure 161: SNMP Filter Interface List



Configure Network Device

SNMP Version: 2c

Poll Interval: Fast - 60s

SNMP Community: public

Resource: Select Resource

Selected MIBs:

- TCP-MIB::tcp
- IF-MIB::ifXTable
- IF-MIB::ifTable
- enterprises.2636.3.1.13.1

Filtered Interface List

Please select the desired interfaces.

Search...

- et-0/1/1
- et-0/1/3
- et-0/1/2
- ge-0/0/20
- ge-0/0/36
- ge-0/0/32.0
- ge-0/0/32
- ge-0/0/10
- ge-0/0/8
- lo0.16385

| | | | |
|---|---|---|---|
| QFX7 | 44 Interfaces | 4 MIBs | |
| QFX6 | 60 Interfaces | 4 MIBs | |
| QFX10 | 41 Interfaces | 4 MIBs | |

## RELATED DOCUMENTATION

SNMP Traps in Contrail Insights | 71

Contrail Insights JTI (UDP) Monitoring | 214

Contrail Insights JTI (gRPC) Monitoring | 222

# Contrail Insights NETCONF CLI Monitoring

## IN THIS SECTION

# Add Network Device in Contrail Insights to Retrieve Data from NETCONF CLI

To enable NETCONF CLI monitoring, you need to post the NETCONF plug-in when running Ansible.

## Enable NETCONF CLI Monitoring

To enable NETCONF CLI monitoring:

1. Post the NETCONF plug-in when running Ansible and include the following lines in the `group_vars/all` file:

```
appformix_plugins:
  - { plugin_info: 'certified_plugins/netconf_commandline.json' }
```

2. Have the following setting on your device and supply the device `username` and `password` correctly:

```
set system services netconf ssh
```

## Add Network Devices to Retrieve Data from NETCONF CLI and Push to Kafka

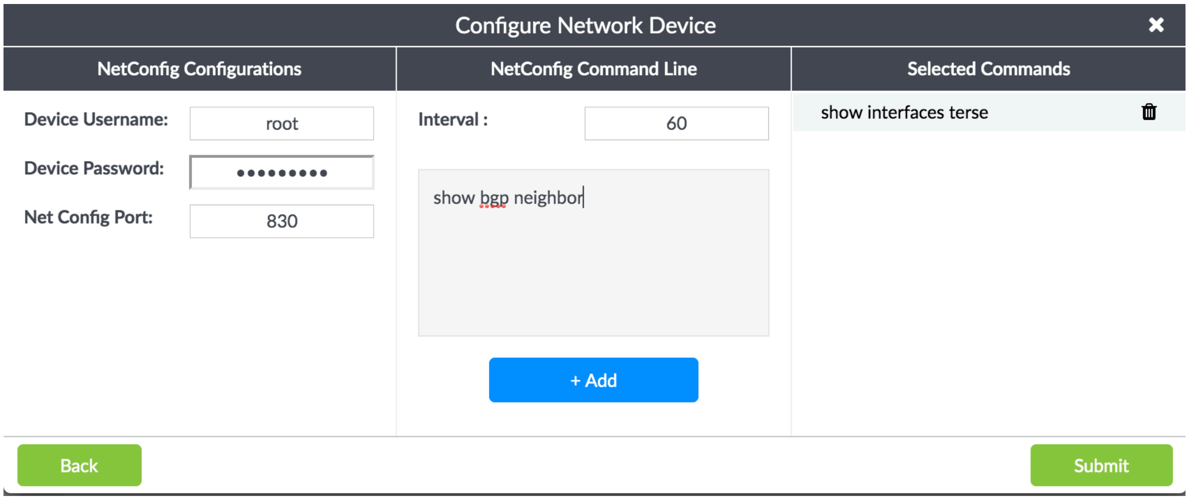
You can add network devices in Contrail Insights and configure the device to retrieve data from NETCONF CLI periodically and push data to Kafka.

To add network devices to retrieve data from NETCONF CLI and push to Kafla:

1. Select **Settings** in the upper right corner, then select **AppFormix Settings** > **Network Devices**.

2. Enter the command you want to run in the device and the interval that you want to run this command. The interval should be multiples of 60 seconds. Next, click **+ Add**.

Figure 162: Configure NETCONF Command and Interval for Network Devices



Device Username: root

Device Password: •••••••••

Net Config Port: 830

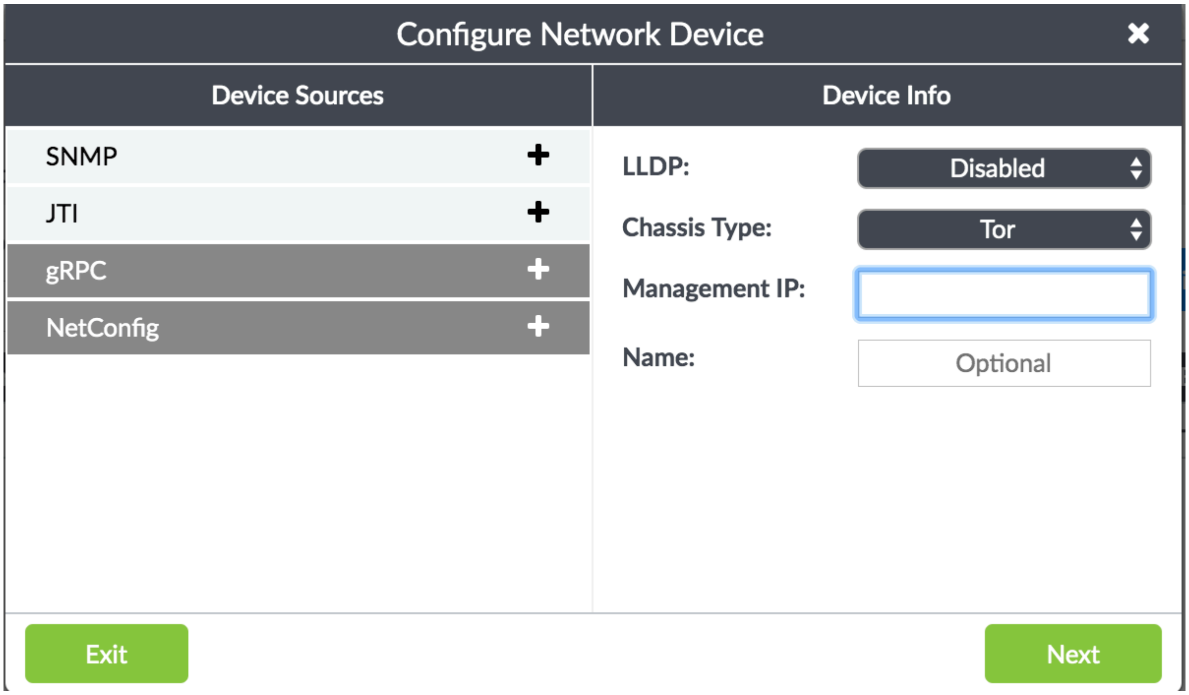
Interval : 60

show bgp neighbor

show interfaces terse

3. Select **NetConfig** in Device Sources and enter Management IP in Device Info

Figure 163: Add NETCONF Network Devices



LLDP: Disabled

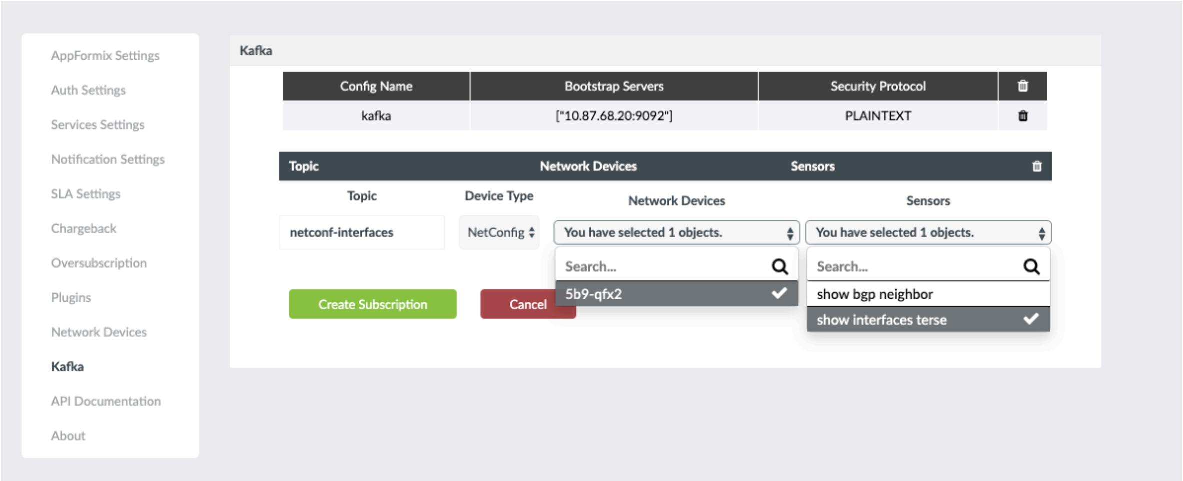
Chassis Type: Tor

Management IP:

Name: Optional

4. After adding NETCONF devices, select **Settings** in the upper right corner, then select **AppFormix Settings > Kafka.** to set up the Kafka listener and subscription. The Sensors drop-down shows the superset of NETCONF commands you added to devices in Contrail Insights.

Figure 164: Sensors and Superset of NETCONF Commands Added to Devices in Contrail Insights



Kafka

| Config Name | Bootstrap Servers | Security Protocol |
|---|---|---|
| kafka | ["10.87.68.20:9092"] | PLAINTEXT |

| Topic | Network Devices | Sensors |
|---|---|---|

Topic: netconf-interfaces

Device Type: NetConfig

Network Devices: You have selected 1 objects. — 5b9-qfx2 ✓

Sensors: You have selected 1 objects. — show bgp neighbor; show interfaces terse ✓

For more information about how to retrieve data from Kafka, see *Contrail Insights with Kafka*.

## RELATED DOCUMENTATION

*Custom Sensors for JTI, gRPC, and NETCONF*

# Contrail Insights Network Device Monitoring Common Issues

### IN THIS SECTION

## JTI Timestamp is Off in Contrail Insights Chart

There is an issue with the timestamp of JTI data not synchronizing with user's current timestamp. As a result, JTI data is shown as ahead or behind in the Contrail Insights charts.

To solve the JTI timestamp not synchronizing with user's timestamp:

1. Use Network Time Protocol (NTP) to sync Junos device time. Verify the result of `show system uptime` command is the same as the time of AppFormix-VM.

2. The JTI stream comes directy from the virtual Forwarding Engine (vFPC) with vFPC timestamp, and the vFPC/vCP has separate NTP service. You should force NTP sync between vFPC – vCP and remove local time failover.

   ```
   [root@vfpc]# vi /etc/ntp.conf
   server 128.0.0.1 iburst                   ####### 128.0.0.1 is vCP internal IP
   #server 127.127.0.1                         #### comment out LOCAL HARDWARE CLOCK
   ```

   Then run:

   ```
   [root@vfpc]# service ntpd stop && service ntpd start
   ```

3. Run the following command to check that offset is back to normal:

   ```
   run ntpq -p
   ```

## JTI Device Not Showing Data in the Chart

For troubleshooting information about JTI device not showing data in chart, see *Contrail Insights JTI (UDP) Monitoring*.

## SNMP Device Not Reporting Data

There are several reasons why SNMP devices are not reporting data including:

- Device reachability.

- MIBs not getting installed.
- Contrail Insights plug-ins not distributing the device data to the correct Contrail Insights collector.

To correct device reachability or MIBs not getting installed:

1. Log in to your `appformix_network_agents` nodes. If you have multiple hosts in this aggregate, verify in all these hosts.

2. Run `cd /opt/appformix/manager/tailwind_manager/`.

3. Run the plug-in files directly from this folder. If some specific MIBs are not working (for example, the `plugin_config_file` for that MIB is `config_file.py`), run following command:

   ```
   python check_snmp_network_device_template.py -d {ip} -f config_file -c {snmp_community} -v 2c
   ```

   The command can be changed due to different SNMP version.

4. Run the following command to check the possible variables in the script:

   ```
   python check_snmp_network_device_template.py -h
   ```

   To check the configuration file name of a plug-in, get information from the JSON file of that plug-in in the `certified_plugins` folder of the Ansible installer.

To correct Contrail Insights plug-ins not distributing the device data to the correct collector:

1. Use the Contrail Insights plug-in API to get the distribution map of any SNMP plug-ins. It is located in **Plugin** > **Config** > **ObjectList**. For more information, contact mailto:AppFormix-Support@juniper.net with your specific case. You can also view data from the Dashboard by selecting **Settings** > **Plugins**, then select a specific plug-in to view enabled metrics.

## gRPC Devices Not Reporting Data

There are several reasons why gRPC devices are not reporting data including:

1. Device is not installed correctly with the openconfig/network-agent package.

2. Device is not configured correctly.

3. `appformix_network_agents` cannot receive data from devices.

To correct device not installed correctly with openconfig package, network-agent package, or device not configured correctly:

1. Log in to your device to verify if it has the correct packages and configuration.

2. Run `show version` on the device to check the device module and Junos version.

3. Run `show system software | grep na` to check if the network_agent package is correctly installed on the device.

4. Run `show system software | grep open` to check if the openconfig package is correctly installed on the device.

5. Run `show system services extension-service` to check the gRPC configuration on the device. Following is an example of the desired output:

   ```
   request-response {
       grpc {
           clear-text {
               port 50051;
           }
           skip-authentication;
       }
   }
   notification {
       allow-clients {
           address 0.0.0.0/0;
       }
   }
   ```

To correct `appformix_network_agents` not receiving data from devices:

1. Verify that you do not have any firewall IPtables preventing the connections. Run the following commands to flush the IPtables rules:

   ```
   iptables -F
   iptables -P FORWARD ACCEPT
   iptables -P OUTPUT ACCEPT
   iptables -P INPUT ACCEPT
   ```

2. (Optional) Use the Contrail Insights plug-in API to get the distribution map of any SNMP plug-ins. It is located in **Plugin > Config > ObjectList**. For more information, contact mailto:AppFormix-Support@juniper.net with your specific case.

3. Run the gRPC test script from appformix_network_agents to check if Contrail Insights can get gRPC data from devices. Contrail Insights supplies a test script in the /opt/appformix/manager/tailwind_manager folder named check_grpc_device_test.py. Run the following commands to debug:

   ```
   cd /opt/appformix/manager/tailwind_manager
   source ../venv/bin/activate
   python check_grpc_device_test.py -ip {device_ip} -port {port} -sensor {sensor_path}
   ```

   If you can get data from check_grpc_device_test.py, you are able to get data from the Contrail Insights software.

   If you cannot get data from check_grpc_device_test.py, you can enable the gRPC logs on the device by running the following commands:

   ```
   set system services extension-service traceoptions file extension-service.log
   set system services extension-service traceoptions file size 5m
   set system services extension-service traceoptions file files 2
   set system services extension-service traceoptions flag all
   ```

4. To get the gRPC logs, run the command:

   ```
   show log extension-service.log
   ```

## JTI Data Not Delivered to Application Socket Due to rp_filter

In some cases, UDP packets from devices are received by interfaces (based on tcpdump output) but cannot be received to application socket. When you run socket.recvfrom in Python code, you cannot receive any data on port 42596.

To correct this issue, disable rp_filter on the eth1 interface (which is the interface device sends data to) by running the following commands:

```
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/eth1/rp_filter
```

Now you should see data in the Contrail Insights Dashboard.

## SNMP Traps Not Shown in Dashboard

For troubleshooting why SNMP traps are not showing in the Dashboard, perform the following steps to determine if anything is incorrect:

1. Check if port 42597 is open and listening in all `appformix_controller` nodes by running `netstat -plan|grep 42597`.

2. Confirm the `snmp_trap_network_device` plug-in is present in the cluster. Select **Settings** > **Plugins**.

3. Check if the alarm named `network_device_snmp_trap` is present in the cluster from the Dashboard Alarms page.

4. Verify the SNMP trap configurations on the devices are correct. See *SNMP Traps in Contrail Insights* for complete configuration details.

5. Check if all Contrail Insights Platform nodes are reporting data. You can confirm this if you see data in the host charts for the Platform nodes. Select **Dashboard** > **Hosts** tab, then select the host node to view more detail.

If you identify issues with any of the above, there are a few things to try. Check if the problem is fixed after each step since all steps might not be needed:

- Re-run the playbook to add the plug-in and the alarm again (Step 2 and Step 3).

- Verify and update the SNMP trap configuration on the devices (Step 4).

- Lastly, restart the Contrail Insights Agent on the Platform Nodes.

### RELATED DOCUMENTATION

*Configure Network Devices from the UI*

*Contrail Insights Auto Discovery of Network Devices from Contrail Networking*

*Contrail Insights Network Device Monitoring Common Issues*

*Contrail Insights JTI (UDP) Monitoring*

*Contrail Insights JTI (gRPC) Monitoring*

*Contrail Insights SNMP Monitoring*

4

CHAPTER

# Contrail Insights Alarms

# Alarms

IN THIS SECTION

With Contrail Insights Alarms, you can configure an alarm to be generated when a condition is met in the infrastructure. Contrail Insights performs distributed analysis of metrics at the point of collection for efficient and responsive detection of events that match an alarm. Contrail Insights has two types of alarms:

- Static—User-provided static threshold is used for comparison.
- Dynamic—Dynamically-learned adaptive threshold is used for comparison.

Sections in this topic include:

## Contrail Insights Alarms Overview

For both static and dynamic alarms, Contrail Insights Agent continuously collects measurements of metrics (see *Metrics Collected by Contrail Insights*) for different entities, such as hosts, instances, and network devices. Beyond simple collection, the agent also analyzes the stream of metrics at the time of collection to identify alarm rules that match. For a particular alarm, the agent aggregates the samples according to a user-specified function (average, standard deviation, min, max, sum) and produces a single measurement for each user-specified measurement interval. For a given measurement interval, the agent compares each measurement to a threshold. For an alarm with a static threshold, a measurement is compared to a fixed value using a user-specified comparison function (above, below, equal). For dynamic thresholds, a measurement is compared with a value learned by Contrail Insights over time.

You can further configure alarm parameters that require multiple intervals to match. This allows you to configure alarms to match sustained conditions, while also detecting performance over small time periods. Maximum values over a wide time range can be over-exaggerate conditions. Yet, averages can dilute the information. A balance is better achieved by measuring over small intervals and watching for repeated matches in multiple intervals. For example, to monitor CPU usage over a three-minute period, an alarm may be configured to compare average CPU utilization over fiveseconds intervals, yet only raise an alarm when 36 (or some subset of 36) intervals match the alarm condition. This provides better visibility into sustained performance conditions than a simple average or maximum over three minutes.

Dynamic thresholds enable outlier detection in resource consumption based on historical trends. Resource consumption may vary significantly at various hours of the day and days of the week. This makes it difficult to set a static threshold for a metric. For example, 70% CPU usage may be considered normal for Monday mornings between 10:00 AM and 12:00 PM, but the same amount of CPU usage may be considered abnormally high for Saturday nights between 9:00 PM and 10:00 PM.

With dynamic thresholds, Contrail Insights learns trends in metrics across all resources in scope to which an alarm applies. For example, if an alarm is configured for a host aggregate, Contrail Insights learns a baseline from metric values collected for hosts in that aggregate. Similarly, an alarm with a dynamic threshold configured for a project learns a baseline from metric values collected for instances in that project. Then, the agent generates an alarm when a measurement deviates from the baseline value learned for a particular time period.

When creating an alarm with a dynamic threshold, you select a metric, a period of time over which to establish a baseline, and the sensitivity to measurements that deviate from the baseline. The sensitivity can be configured as *high*, *medium*, or *low*. Higher sensitivity will report smaller deviations from the baseline and vice versa.

## Contrail Insights Alarms Operation

Contrail Insights Agent performs distributed, real-time statistical analysis on a time-series data stream. Agent analyzes metrics over multiple measurement intervals using a configurable sliding window mechanism. An alarm is generated when the Contrail Insights Agent determines that metric data matches the alarm criteria over a configurable number of measurement intervals. The type of sample aggregation and the threshold for an alarm is configurable. Two types of alarms are supported: static and dynamic. The difference is how the threshold is determined and used to compare measured metric data. The following sections describe the overall sliding window analysis, and explains the details of static thresholds and dynamic baselines used by the analysis.

### Sliding Window Analysis

Contrail Insights Agent evaluates alarms using sliding window analysis. The sliding window analysis compares a stream of metrics within a configurable measurement interval to a static threshold or dynamic baseline. The length of each measurement interval is configurable to one-second granularity. In each measurement interval, raw time-series data samples are combined using an aggregation function, such as *average*, *max*, and *min*. The aggregated value is compared against the static threshold or dynamic baseline using a configurable comparison function, such as above or below. Multiple measurement intervals comprise a sliding window. A configurable number of intervals in the sliding window must match the rule criteria for the agent to generate a notification for the alarm.

**Figure 165: Alarm Generation Mechanics**

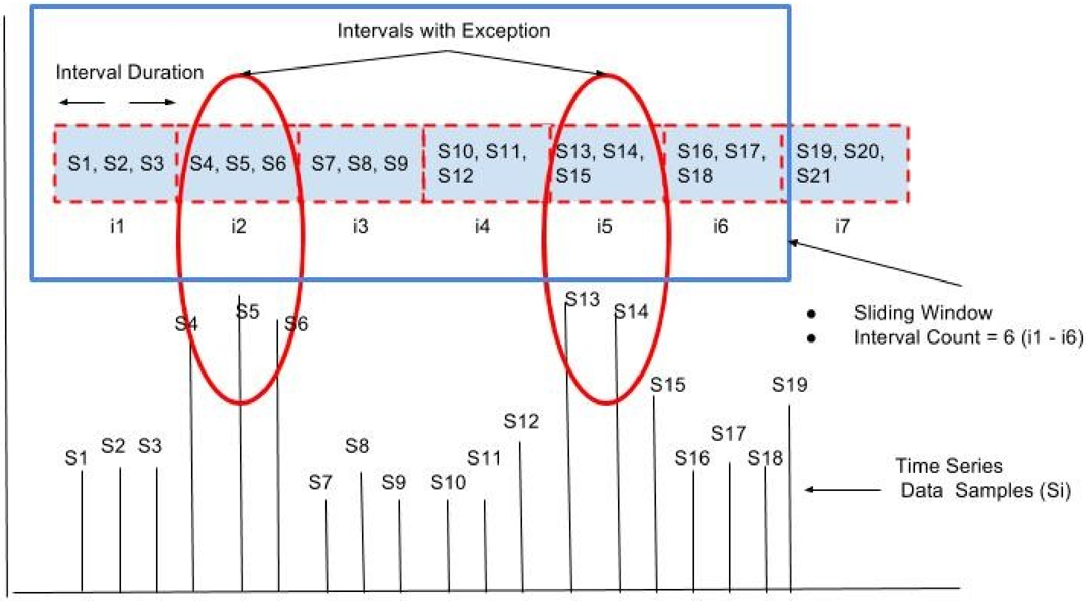


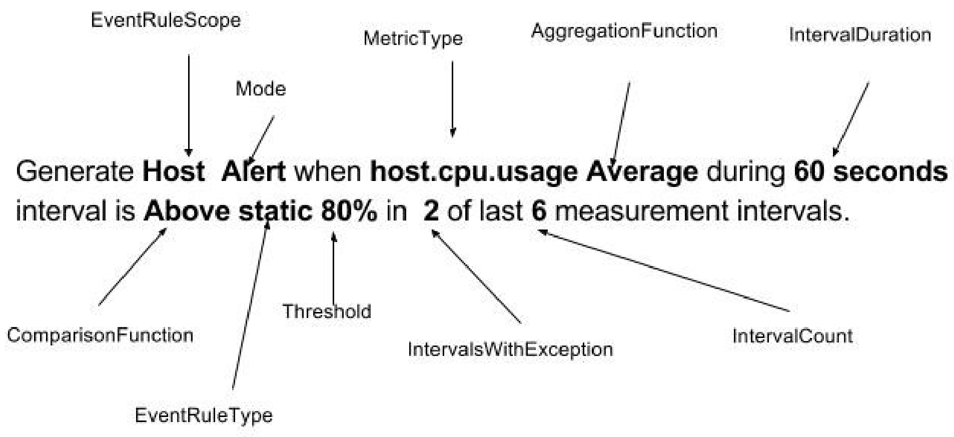
Figure 165 on page 242 shows an example in which the sliding window consists of six adjacent measurement intervals (i1 to i6), as specified by the Interval Count parameter. In measurement interval i1, the average of samples S1, S2, S3 is computed as $S_{avg}$. Depending on the alarm type *static* or *dynamic*, $S_{avg}$ is then compared with the configured static threshold or dynamically learned baseline using a user-specified comparison function such as *above* or *below*. The output of the comparison determines whether a specific measurement interval is marked as an *interval with exception*. This evaluation is repeated for each measurement interval within the sliding window (for example, i1 to i6).

In the example in Figure 165 on page 242, the agent determines that two intervals, i2 and i5, are *intervals with exception* by comparing the aggregate value for the measurement interval with a static threshold or dynamic baseline, depending on alarm type. Assuming interval i1 is the first interval for which the alarm is configured, the alarm becomes active at end of interval i6, when Contrail Insights Agent determines that at least two out of the most recent six measurement intervals are marked as exceptions. When an alarm is configured using the Dashboard, Interval Count, and Intervals with Exception are set to 1 by default. As a result, the agent can generate an alarm after processing data for one measurement interval.

## Static Alarm

A static alarm threshold is provided at the time of alarm definition. Figure 166 on page 243 depicts an example of a static alarm definition, followed by the equivalent JSON used for API configuration of an alarm. The condition defined in the example is to evaluate an average of `host.cpu.usage` samples over a 60 second measurement interval. The measured value is compared against a static threshold of 80% to determine if a given measurement interval matches the alarm rule. Figure 166 on page 243 identifies the components in a static alarm definition.

**Figure 166: Static Alarm Definition**


EventRuleScope
MetricType
AggregationFunction
IntervalDuration
Mode
Generate **Host** **Alert** when **host.cpu.usage Average** during **60 seconds**
interval is **Above static 80%** in **2** of last **6** measurement intervals.
Threshold
ComparisonFunction
IntervalsWithException
IntervalCount
EventRuleType

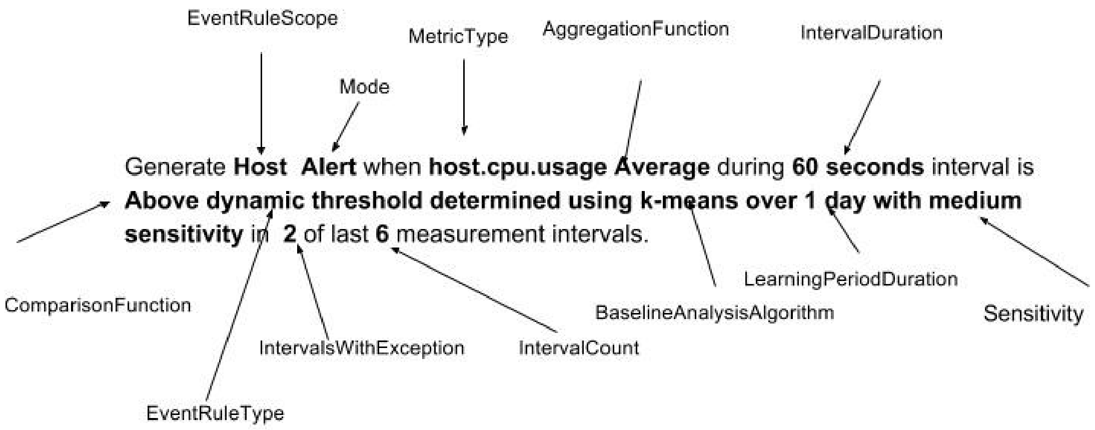
The following example shows the JSON equivalent to the static alarm definition shown in Figure 166 on page 243:

```
"EventRule": {
        "Name": "Host-CPU-usage",
        "EventRuleType": "static",
        "EventRuleScope": "host",
        "MetricType": "cpu.usage",
        "Mode": "alert”,
        "AggregationFunction": "average",
        "IntervalDuration": "60",
        "ComparisonFunction": "above",
        "Threshold": 80,
        "IntervalsWithException": 2,
        "IntervalCount": 6,
        "DisplayEvent": true,
        "Status": "enabled",
        "Module": "alarms",
        "Severity": "warning",
      }
```

## Dynamic Alarm

A dynamic alarm threshold is learned by Contrail Insights using historical data for the set of entities for which an alarm is configured. Figure 167 on page 244 shows an example of a dynamic alarm definition, followed by the equivalent JSON used for API configuration of an alarm. Figure 167 on page 244 identifies the components in a dynamic alarm definition.

**Figure 167: Dynamic Alarm Definition**


EventRuleScope
MetricType
AggregationFunction
IntervalDuration
Mode
Generate **Host** **Alert** when **host.cpu.usage** **Average** during **60 seconds** interval is
**Above dynamic threshold determined using k-means over 1 day with medium**
**sensitivity** in **2** of last **6** measurement intervals.
LearningPeriodDuration
ComparisonFunction
BaselineAnalysisAlgorithm
Sensitivity
IntervalsWithException
IntervalCount
EventRuleType

The following example shows the JSON equivalent to the static alarm definition shown in Figure 167 on page 244:

```
  "EventRule": {
         "Name": "Host-CPU-usage",
         "EventRuleType": "dynamic",
         "EventRuleScope": "host",
         "MetricType": "cpu.usage",
         "Mode": "alert”,
         "AggregationFunction": "average",
         "IntervalDuration": "60",
         "ComparisonFunction": "above",
         “BaselineAnalysisAlgorithm”: “k-means”,
         “LearningPeriodDuration”: “1d”,
         “Sensitivity”: “medium”,
         "IntervalsWithException": 2,
         "IntervalCount": 6,
         "DisplayEvent": true,
         "Status": "enabled",
         "Module": "alarms",
         "Severity": "warning",
       }
```

When using a dynamic threshold, you do not configure a static threshold value. Instead, you specify three parameters that control how the learning is performed. The learning algorithm produces a baseline across the entities. The baseline is comprised of a mean value and a standard deviation. The baseline is updated continuously as additional metric data is collected.

Following is a list of the three learning parameters and information about how they work:

**BaselineAnalysisAlgorithm**

Selects the machine learning algorithm used for determining the dynamic threshold. The following algorithms are available:

**k-means**

Contrail Insights employs a k-means algorithm to produce an expected operating range for a set of entities at a granularity of each hour of each day (up to one week). The learned baselines are computed using data from a configurable learning period duration. The baselines are updated continuously over time, based on the most recent data. The k-means Baseline Analysis Algorithm is useful for observing performance that is unexpected for a given time of day.

For example, a k-means algorithm may learn a dynamic baseline for 1:00 PM - 2:00 PM that may be 80% +/- 10%, whereas, the baseline between 3:00 AM - 4:00 AM may be 20% +/- 5%. An alarm is raised if the measured metric is 75% of the value between 3:00 AM - 4:00 AM, but the same measurement is acceptable during 1:00 PM - 2:00 PM time period.

**ewma**

The Exponentially Weighted Moving Average (EWMA) algorithm produces a single baseline that is updated hourly. The configurable Learning Period duration allows you to control the relative weight assigned to recent data versus older data. This algorithm is useful to create an alarm that can detect sudden changes in a metric.

For example, an EWMA algorithm can learn a dynamic baseline of 60% +/- 10% from data over the last 24 hours. This baseline is used for the next 1-hour interval to determine if real-time data deviates from the normal operating region. After every 1-hour interval, the EWMA baseline is updated and a new updated baseline is used for alarm generation in the future.

**LearningPeriodDuration**

A dynamic baseline is determined using the historical data. This parameter determines the length of time period from which most recent historical data is used to compute a dynamic baseline. For example, 1 hour, 1 day, or 1 week. At the time of rule configuration, Contrail Insights might not yet have enough historical data for a given entity. In this case, learning is performed as data becomes available. Alarm evaluation begins after one Learning Period of data is available and baselines are generated.

**Sensitivity**

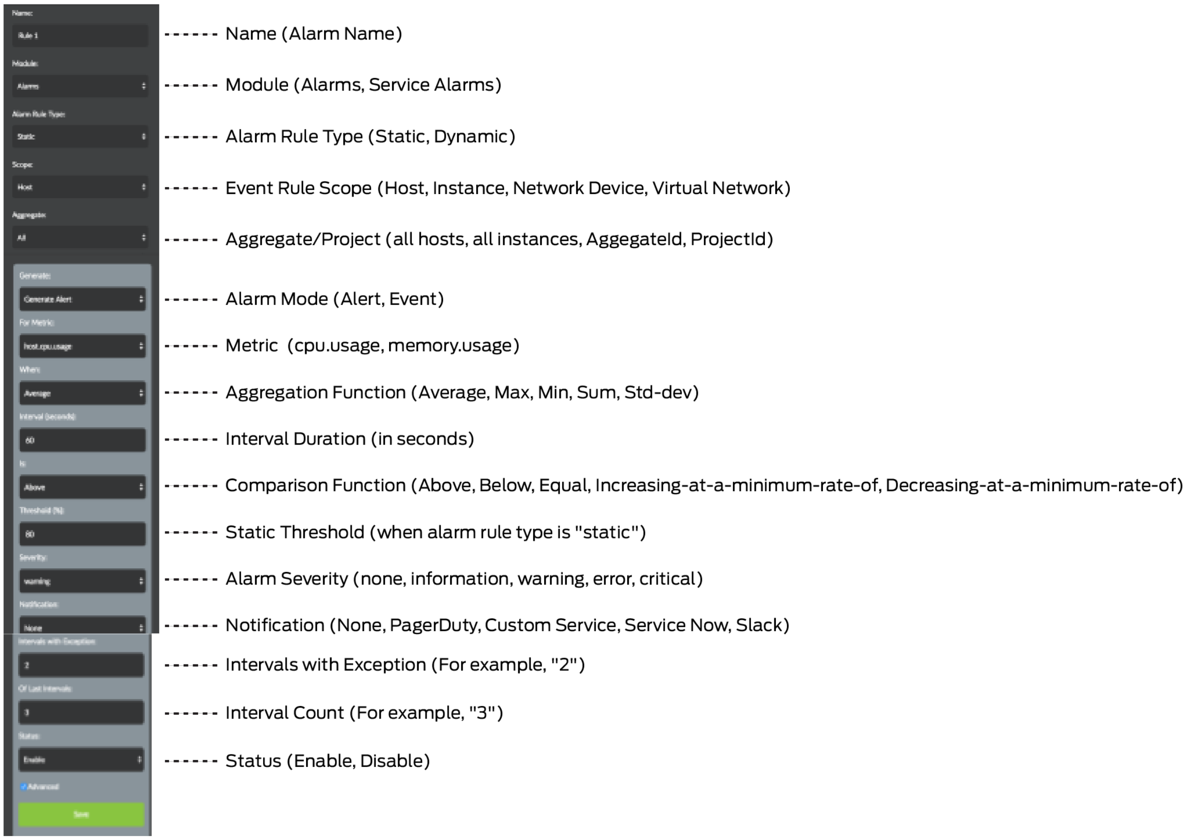
The sensitivity of a dynamic alarm controls the allowable magnitude of deviation from the learned mean. The sensitivity parameter controls a multiplier of the learned standard deviation. You can select *low*, *medium*, or *high* as sensitivity. Contrail Insights Agent compares real-time measurements to the range defined by:

```
mean - sensitivity * std_dev < x < mean + sensitivity * std_dev
```

## Alarm Definition

Figure 166 on page 243 shows an example of a static alarm definition and is followed by the JSON for the same rule. Every alarm definition has the following components shown in Figure 168 on page 247.

**Figure 168: Static Alarm Rule Configuration Example**



Name: Rule 1

Module: Alarms

Alarm Rule Type: Static

Scope: Host

Aggregate: All

Generate: Generate Alert

For Metric: host.cpu.usage

When: Average

Interval (seconds): 60

Is: Above

Threshold (%): 80

Severity: warning

Notification: None

Intervals with Exception: 2

Of Last Intervals: 3

Status: Enable

☑ Advanced

Name (Alarm Name)

Module (Alarms, Service Alarms)

Alarm Rule Type (Static, Dynamic)

Event Rule Scope (Host, Instance, Network Device, Virtual Network)

Aggregate/Project (all hosts, all instances, AggegateId, ProjectId)

Alarm Mode (Alert, Event)

Metric (cpu.usage, memory.usage)

Aggregation Function (Average, Max, Min, Sum, Std-dev)

Interval Duration (in seconds)

Comparison Function (Above, Below, Equal, Increasing-at-a-minimum-rate-of, Decreasing-at-a-minimum-rate-of)

Static Threshold (when alarm rule type is "static")

Alarm Severity (none, information, warning, error, critical)

Notification (None, PagerDuty, Custom Service, Service Now, Slack)

Intervals with Exception (For example, "2")

Interval Count (For example, "3")

Status (Enable, Disable)

The listed components for alarm definition are numbered and described in the following text:

**1. Name**

A name identifies the alarm. Name is displayed in the Dashboard and is the user-facing identifier for external notification systems.

**2. Module**

When **Alarms** is selected, you can configure alarms for entities such as hosts, instances, and network devices. When **Service Alarms** is selected, then you are able to configure alarms for services such as RabbitMQ, MySQL, ScaleIO, and OpenStack services.

**3. Alarm Rule Type**

This determines the type of threshold that alarm uses to determine if alarm should be generated or not. Following are the two types that are supported.

- Static—When an alarm is defined as static, the rule definition should include a predefined static threshold. For example, `cpu.usage` static threshold can be 80%.
- Dynamic—When an alarm is defined as dynamic, the baseline is learned using historical data. Additional parameters are required such as baseline analysis algorithm, learning period duration, and sensitivity.

**4. Event Rule Scope**

Type of entity such as host, instance, or network device to which the alarm applies. For example, if scope is selected as **Instance**, then you can further select to configure rule to all instances present in the infrastructure, or instances that are present in a specific project or an aggregate.

**5. Aggregate**

Select the set of entities an alarm will monitor. If Scope is **Instance**, then you can configure an alarm for the set of instances present in a specific project, aggregate, or all instances in the infrastructure. If Scope is **Host**, then you can configure an alarm for a set of hosts present in a specific aggregate or all hosts in the infrastructure.

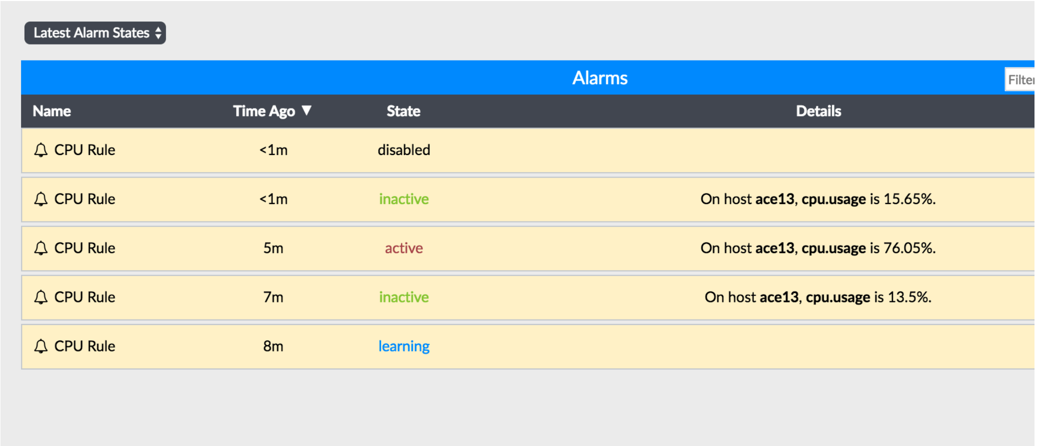
**6. Alarm Mode**

Mode can be configured as an alert or event.

- Alert—An alarm with the mode set to **Alert** has state. Events are generated and recorded only for changes in the state of the alarm. Table 22 on page 249 shows all possible states for an alarm with the mode configured as alert. Figure 169 on page 248 shows an example of different state transitions for an alarm for the cpu.usage metric with a static threshold of 50%.

- Event—An alarm with the mode set to **Event** is evaluated similar to an alarm with the mode set to **Alert**. The key difference is that an alarm with the mode set to **Event** keeps generating notifications with a state of *triggered* for each interval in which the condition for the alarm is satisfied. When the conditions for an alarm are not satisfied, then the agent stops generating notifications about the alarm. As shown in Figure 170 on page 249, an alarm with the mode set to **Event** generates significantly more notifications compared to an alarm with the mode set to **alert**.

**Figure 169: Alarm State Transition with Mode as Alert for Cpu.usage Static Threshold = 50%**

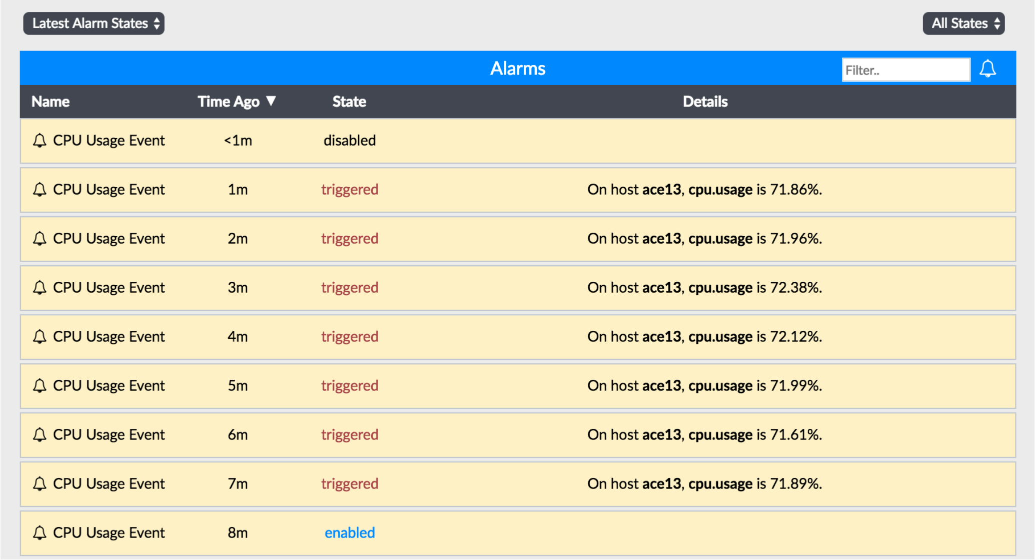


| Name | Time Ago ▼ | State | Details |
|---|---|---|---|
| CPU Rule | <1m | disabled | |
| CPU Rule | <1m | inactive | On host **ace13**, **cpu.usage** is 15.65%. |
| CPU Rule | 5m | active | On host **ace13**, **cpu.usage** is 76.05%. |
| CPU Rule | 7m | inactive | On host **ace13**, **cpu.usage** is 13.5%. |
| CPU Rule | 8m | learning | |

### Table 22: States for Alarm Mode Defined as Alert

| State | Description |
|---|---|
| Learning | This is the initial state of each alarm. In this state, the alarm is processing real-time data and alarm stays in this state until sufficient data has been processed to make the decision about if an alarm should be generated or not. The duration of the learning period depends on the sliding window parameters. Figure 169 on page 248 shows the learning state when rule is configured in the system. |
| Active | The condition specified by an alarm is met. Alarm will stay in this state as long as alarm conditions are satisfied. Figure 169 on page 248 shows the active state when CPU usage is detected as 76.05%. |
| Inactive | Condition specified by an alarm is not met. In Figure 169 on page 248, after the learning state, the alarm transitions to inactive state because CPU usage was 13.5% (below the 50% threshold). The alarm transitions from active state to inactive state when CPU usage drops to 15.65%. |
| Disabled | Agent is not actively analyzing data for this alarm. The alarm is either deleted or temporarily deactivated by the user. |

### Figure 170: Alarm State Transition with Mode as Event



| Name | Time Ago ▼ | State | Details |
|---|---|---|---|
| CPU Usage Event | <1m | disabled | |
| CPU Usage Event | 1m | triggered | On host **ace13**, **cpu.usage** is 71.86%. |
| CPU Usage Event | 2m | triggered | On host **ace13**, **cpu.usage** is 71.96%. |
| CPU Usage Event | 3m | triggered | On host **ace13**, **cpu.usage** is 72.38%. |
| CPU Usage Event | 4m | triggered | On host **ace13**, **cpu.usage** is 72.12%. |
| CPU Usage Event | 5m | triggered | On host **ace13**, **cpu.usage** is 71.99%. |
| CPU Usage Event | 6m | triggered | On host **ace13**, **cpu.usage** is 71.61%. |
| CPU Usage Event | 7m | triggered | On host **ace13**, **cpu.usage** is 71.89%. |
| CPU Usage Event | 8m | enabled | |

Table 23: States for Alarm Mode Defined as Event

| State | Description |
|---|---|
| Enabled | This is the initial state of the alarm with the mode set to **Event** when a rule is configured. It stays in this state until conditions are met to generate an alarm. Figure 170 on page 249 shows state *enabled* is logged when alarm with mode as event is configured. |
| Triggered | When conditions for alarm generation are satisfied, then an alarm is generated with a state of *triggered*. Alarm generation is logged at the end of each measurement interval as long conditions for alarms continue to be met. In Figure 170 on page 249, seven alarm events are generated for the duration when cpu.usage stays above 50%. |
| Disabled | Agent is not actively analyzing data for this alarm. The alarm is either deleted or has been temporarily deactivated by the user. |

**7. Metric Name**

*Metrics Collected by Contrail Insights* that will be monitored. For example, host.cpu.usage or instance.cpu.usage.

**8. Aggregation Function**

Determines how data samples received in one measurement interval are processed to generate an aggregated value for comparison. Agent collects multiple samples of a metric during a measurement interval. Agent combines the samples according to the aggregation function, in order to determine a single value for comparison with the threshold (static or dynamic) in a measurement interval. Table 24 on page 250 lists and describes the aggregation functions for alarm processing.

Table 24: Aggregation Functions for Alarm Processing

| Aggregation Function | Description |
|---|---|
| Average | Statistical average of all data samples received within one measurement interval.<br>Example: Generate **Host Alert** when **Cpu-Usage Average** during a **60 seconds** interval is **Above 80%** of **2** of the last **3** measurement intervals.<br>In this example, the measurement interval is 60 seconds. An alarm is generated if the average of the CPU usage samples exceeds 80% in any 2 measurement intervals out of 3 adjacent measurement intervals. |

Table 24: Aggregation Functions for Alarm Processing *(Continued)*

| Aggregation Function | Description |
|---|---|
| Sum | Sum of all data samples received within one measurement interval.<br>Example: Generate **Host Alert** when **Cpu-Usage Sum** during a **60 seconds** interval is **Above 250%** of **2** of the last **3** measurement intervals.<br>In this example, An alarm is generated if the CPU usage sum is above 250% in any 2 measurement intervals out of 3 adjacent measurement intervals, where each measurement interval is 60 seconds in duration. |
| Max | Maximum sample value observed within one measurement interval.<br>Example: Generate **Host Alert** when **Cpu-Usage Max** during a **60 seconds** interval is **Above 95%** of **2** of the last **3** measurement intervals.<br>In this example, the alarm is generated if the maximum CPU usage is above 95% in any 2 measurement intervals out of 3 adjacent measurement intervals, where each measurement interval is 60 seconds in duration. |
| Min | Minimum sample value observed within one measurement interval.<br>Example: Generate **Host Alert** when **Cpu-Usage Min** during a **60 seconds** interval is **Below 5%** of **2** of the last **3** measurement intervals.<br>In this example, the alarm is generated if the minimum CPU usage is below 5% in any 2 measurement intervals out of 3 adjacent measurement intervals, where each measurement interval is 60 seconds in duration. |
| Std-Dev | Standard Deviation of the time-series data is determined based on the samples received until current measurement interval.<br>Example: Generate **Host Alert** when **Cpu-Usage std-dev** during a **60 seconds interval** is **Above 2 sigma** of **2** of the last **3** measurement intervals.<br>In this example, the alarm is generated when the raw time series samples are above `mean + 2*sigma` in at least 2 measurement intervals out of the last 3 measurement intervals, where each measurement interval is a duration of 60 seconds. |

**9. Comparison Function**

Determines how to compare output of the Aggregation Function with the static or dynamic threshold. Table 25 on page 253 shows different comparison functions supported for Contrail Insights alarms. Figure 171 on page 252 and Figure 172 on page 253 show examples of the Comparison Function, showing both increases and decreases at a minimum rate.

Figure 171: Comparison Function Showing Increasing-at-a-minimum-rate-of

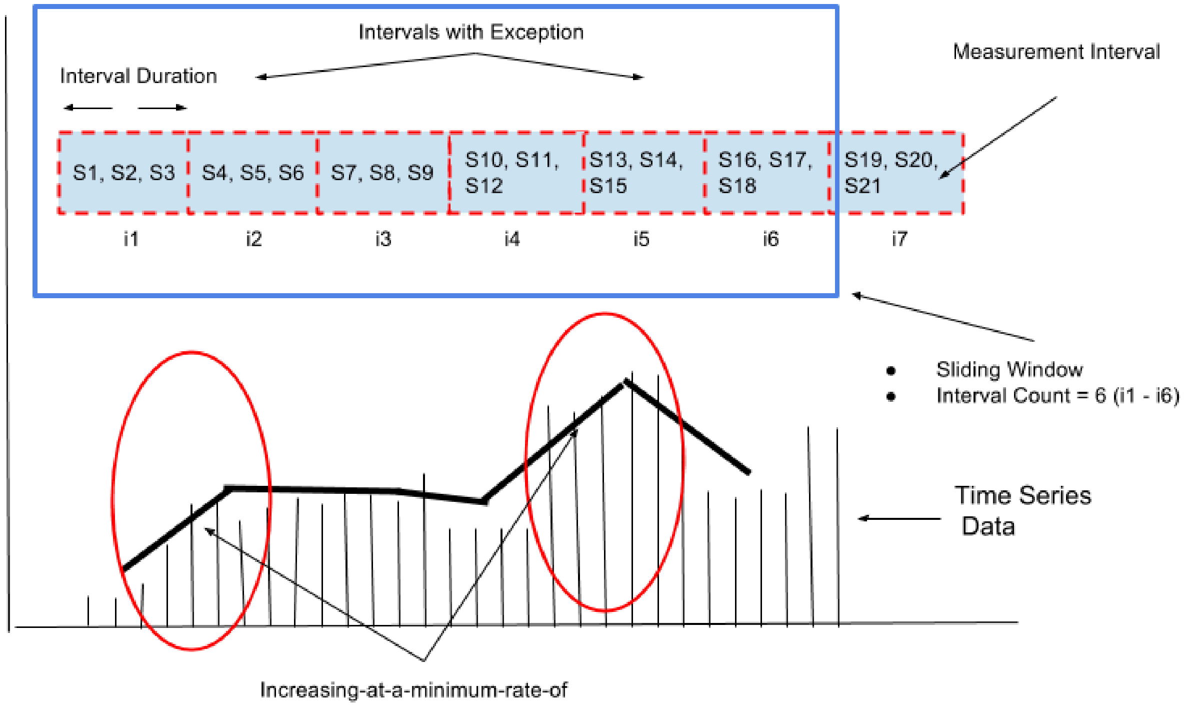


**Figure 172: Comparison Function Showing Decreasing-at-a-minimum-rate-of**

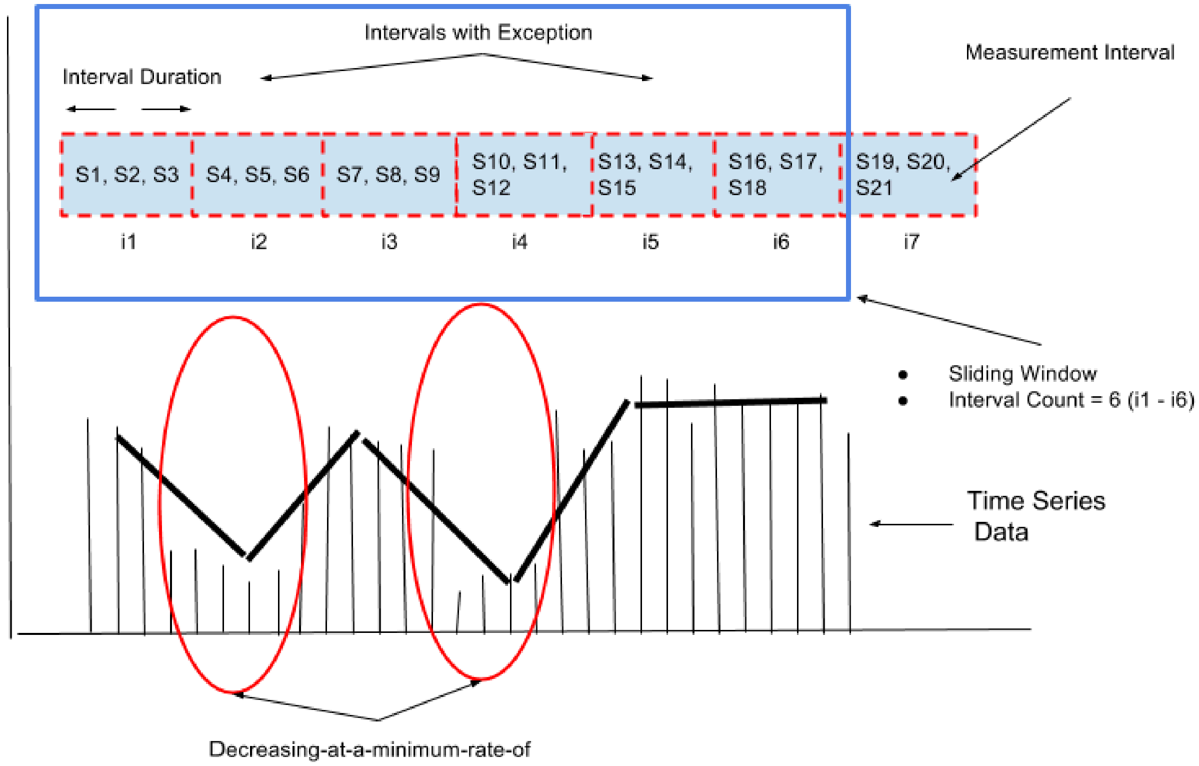


**Table 25: Comparison Functions for Alarm Processing**

| Comparison Operator | Description |
|---|---|
| Above | Determine if result of the aggregation function within a given measurement interval is *above* the threshold.<br>**NOTE**: For dynamic threshold *above*, Contrail Insights compares whether the result of the aggregation function is outside of the normal operating region (mean +/- sigma*sensitivity). |
| Below | Determine if result of the aggregation function determined for a given measurement interval is *below* the threshold.<br>**NOTE**: For dynamic threshold, *below* compares whether the result of aggregation function is within the normal operating region (mean +/- sigma*sensitivity). |
| Equal | Determine if result of the aggregation function is *equal* to the threshold. |

Table 25: Comparison Functions for Alarm Processing *(Continued)*

| Comparison Operator | Description |
|---|---|
| Increasing-at-a-minimum-rate-of | This comparison function is useful when you are interested in tracking a sudden increase in the value of a given metric instead of its absolute value. For example, if ingress or egress network bandwidth starts increasing within short intervals then you might want to raise an alarm. Figure 171 on page 252 shows sudden increase in metric average between measurement interval i1 and i2. Similarly, sudden increase is observed in metric average between measurement intervals i4 to i5.<br>Example: Generate **Host Alert** when the host.network.ingress.bit_rate **average** during a **60 seconds** interval is **increasing-at-a-minimum-rate-of 25%** of **2** of the last **3** measurement intervals.<br>In the example, if the mean ingress bit rate increases by at least 25% in 2 measurement intervals out of 3, then an alarm is raised. |
| Decreasing-at-a-minimum-rate-of | This comparison function is useful when you are interested in tracking sudden decrease in the value of a given metric instead of its absolute value. For example, egress network bandwidth starts decreasing within short intervals then you might want to raise an alarm to investigate the root cause. Figure 172 on page 253 shows sudden decrease in metric average between measurement interval i1 and i2. Similarly, sudden decrease is observed in metric average between measurement intervals i3 and i4.<br>Example: Generate **Host Alert** when the host.network.egress.bit_rate **average** during a **60 seconds** interval is **decreasing-at-a-minimum-rate-of 25%** of **2** of the last **3** measurement intervals.<br>In the example, if the mean egress bit rate decreases by at least 25% in 2 measurement intervals out of 3, then an alarm is raised. |

10. Threshold

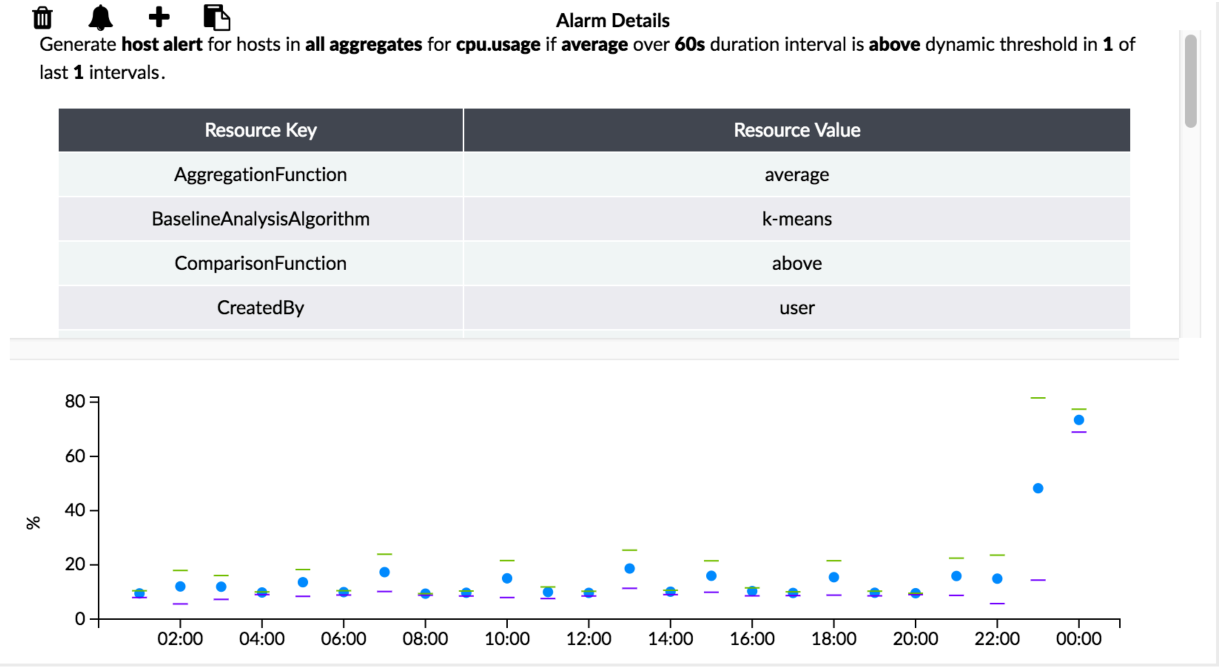
A numeric value to which measurements are compared. Contrail Insights supports two types of thresholds: static or dynamic.

- Static Threshold—A fixed value that is specified when an alarm is configured. For example **host.cpu.usage above 90%**, where 90% is the static threshold.

- Dynamic Threshold—The threshold is learned dynamically by the system. Unsupervised learning is used to learn about historical trends to determine the dynamic threshold. For example, if an event rule is defined for Host aggregate, then the dynamic baseline is determined for the aggregate by applying the baseline analysis algorithm to data received from all member hosts of the aggregate. Figure 173 on page 255 shows the dynamic baseline determined using the most recent 24-hour time frame of historical data and k-means clustering algorithm. This baseline is used for the next 24 hours for alarm generation while considering the hour of the day and its corresponding baseline mean and standard deviation. For example, on Tuesday 8:00 AM - 9:00 AM, a baseline computed for Monday 8:00 AM - 9:00 AM is used as a reference threshold for alarm generation.

Figure 173 on page 255 shows the dynamic baseline computed by 24 hours of data and the k-means clustering algorithm. For a given hour of the day, the blue dot is the `mean`; the green bar is the `mean + std-dev`; the purple bar is `mean - std-dev`.

**Figure 173: Dynamic Baseline Determined by Last 24 Hours of Data and K-Means Clustering Algorithm**



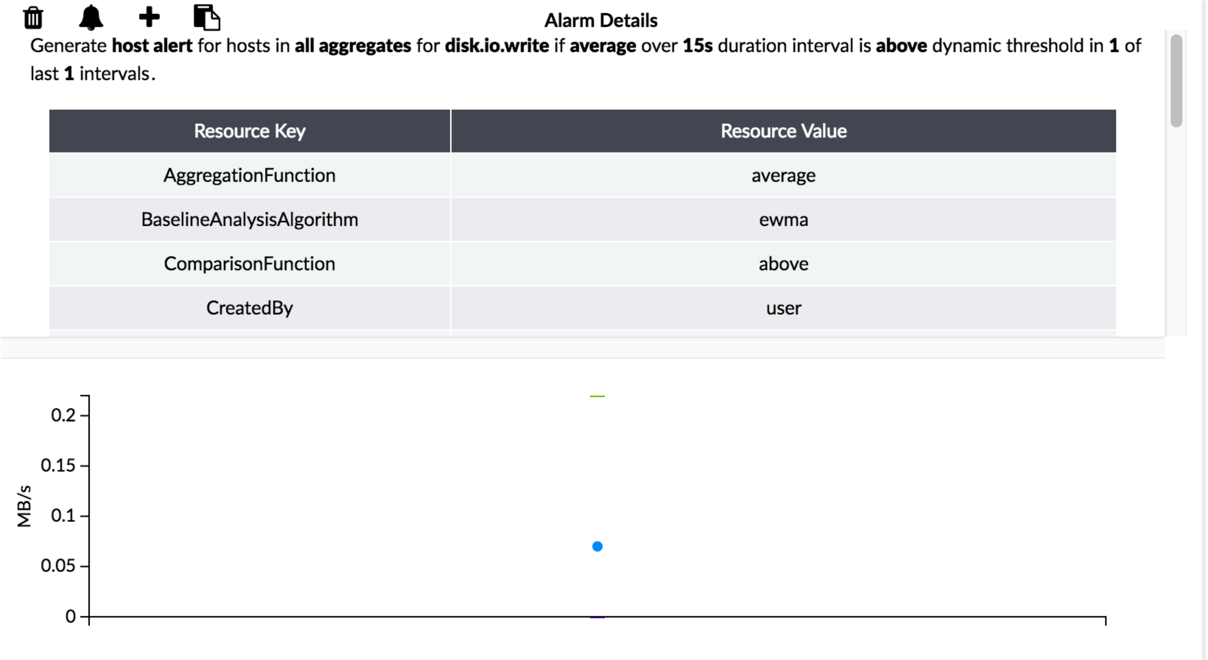
**Alarm Details**

Generate **host alert** for hosts in **all aggregates** for **cpu.usage** if **average** over **60s** duration interval is **above** dynamic threshold in **1** of last **1** intervals.

| Resource Key | Resource Value |
|---|---|
| AggregationFunction | average |
| BaselineAnalysisAlgorithm | k-means |
| ComparisonFunction | above |
| CreatedBy | user |

Figure 174 on page 256 shows the dynamic baseline computed by 24 hours of historical data using the EWMA algorithm. This baseline is used for the next 1 hour for alarm generation until it is updated again using the most recent 24 hours of data.

Figure 174: Dynamic Baseline Determined by Last 24 Hours of Historical Data Using EWMA



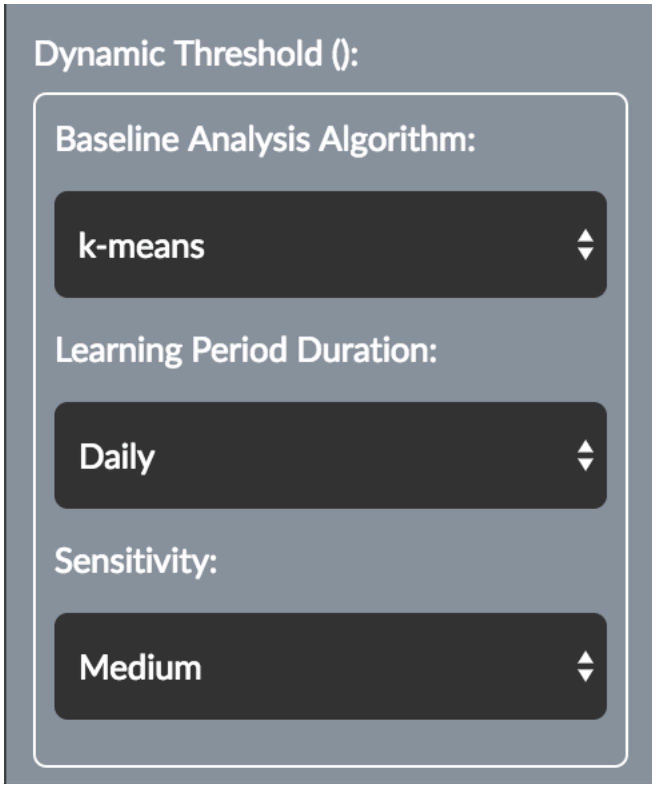
Alarm Details

Generate **host alert** for hosts in **all aggregates** for **disk.io.write** if **average** over **15s** duration interval is **above** dynamic threshold in **1** of last **1** intervals.

| Resource Key | Resource Value |
|---|---|
| AggregationFunction | average |
| BaselineAnalysisAlgorithm | ewma |
| ComparisonFunction | above |
| CreatedBy | user |

Figure 175 on page 257 shows the mandatory parameters that must be specified to configure a dynamic alarm.

Figure 175: Required Parameters for the Dynamic Threshold in the Alarm Definition



Dynamic Threshold ():

Baseline Analysis Algorithm: k-means

Learning Period Duration: Daily

Sensitivity: Medium

Table 26 on page 257 describes the required parameters for a dynamic alarm and the supported options.

Table 26: Required Parameters for Dynamic Alarm

| Required Parameters for Dynamic Threshold | Description | Supported Options |
|---|---|---|
| Baseline Analysis Algorithm | Baseline Analysis Algorithm is used to perform unsupervised learning on historical data. The baseline analysis is performed continuously as new data is received. | • K-Means clustering<br>• Exponential Weighted Mean Average (EWMA) |

Table 26: Required Parameters for Dynamic Alarm *(Continued)*

| Required Parameters for Dynamic Threshold | Description | Supported Options |
|---|---|---|
| Learning Period Duration | The Learning Period Duration specifies the amount of historical data used by the Baseline Analysis Algorithm to determine a baseline. The dynamic baseline is continuously updated using data from the most recent Learning Duration.<br>When a dynamic alarm is configured, baseline analysis is performed using data from the most recent Learning Duration, if available. If there is not sufficient data available, Contrail Insights Agent evaluates metrics as soon as enough data is present to learn the first set of baselines.<br>Example: When Learning Duration is 1 day, the agent compares metrics to per-hour baselines for the last 24 hours.<br>Example: When Learning Duration is 1 week, the agent compares metrics to per-hour baselines for the last 7 x 24 hours. | • 1 week—Baseline is determined for each hour of last 1 week of data. Next 1 week of baselines are determined based on data of the last week.<br>• 1 month—Baseline is determined based on last 4 weeks of data. Baselines are learned for each hour of each day of week (7 x 24 baselines). Next 1 week of baselines are determined based on data of the last 4 weeks. For example, a baseline on Monday at 2:00 PM - 3:00 PM is learned using metric data from the last 4 Mondays at 2:00 PM - 3:00 PM. |
| Sensitivity | The dynamic baseline provides a normal operating region of a given metric for a given scope. As seen in Figure 173 on page 255, the dynamic baseline is a tuple which has mean and std-dev applicable for a specific hour of the day.<br>The sensitivity factor determines what is the allowable band of operation. Measurements outside of the band of operation cause an interval with exception. For example, if the baseline mean is 20 and std-dev is 2, then normal operating region is between 18 and 22. When sensitivity is *low* then normal operating region is treated as 10 (mean - 5*std-dev) and 30 (mean + 5*std-dev). In this case, if the measured average of a metric is between 10 and 30, then no alarm is raised. In contrast, if the average is 5 or 35, then an alarm is raised. | • Low—Any data point beyond `5 * std-dev` from the baseline mean is outlier.<br>• Medium—Any data point beyond `3 * std-dev` from baseline mean is outlier.<br>• High—Any data point beyond `2 * std-dev` from baseline mean is outlier. |

**11. Alarm Severity** Indicates seriousness of the alarm. Critical indicates a major alarm. Information indicates a minor alarm.

**12. Notification** Methods of notification alerting you to conditions of operation.

**13. Interval Duration** The duration of one measurement interval in seconds. Depending on the sampling frequency of a metric under observation, one or more raw samples might be received within an interval duration. All raw samples received within Interval duration are processed using aggregation functions such as average, sum, max, min, and std-dev.

**14. Intervals with Exception** This is the minimum number of measurement intervals within the sliding window for which a condition for an alarm must be met to raise the alarm. In Figure 167 on page 244, there are two Intervals with Exception: i2 and i5. When configuring an alarm in the Dashboard, Intervals with Exception is set to 1 by default. The Interval with Exception can be specified in the Dashboard by selecting **Alarms > Add New Rule**. Then select **Advanced** to view the Advanced settings. Intervals with Exception can not be greater than the Interval Count.

**15. Interval Count** Maximum number of adjacent measurement intervals for which a statistical analysis is performed before deciding if an alarm is generated or not. In Figure 167 on page 244, there are 6 measurement Intervals (i1 to i6) in the sliding window. Each measurement interval has duration specified by the Interval Duration parameter. When configuring an alarm in Dashboard, Interval Count is set to 1 by default. The Interval Count can be specified in the Dashboard by selecting **Alarms > Add New Rule**. Then select **Advanced** to view the Advanced settings.

**16. Status** Used to set and also verify status of alarm rule. Set status as enabled or disabled.

## RELATED DOCUMENTATION

*Composite Alarms*

*Capacity Planning*

*Chargeback*

*Charts*

*Health Monitor*

*Metrics Collected by Contrail Insights*

*Notifications*

*Extensibility Using Plug-Ins*

*Reports*

*Service Monitoring from the UI*

# Composite Alarms

IN THIS SECTION

A composite alarm is comprised of multiple individual alarms. The state of a composite alarm is a combination of the states of the individual alarm rules. Each individual alarm rule in a composite alarm must have the same metric scope, but each alarm can analyze a different metric. For example, composite alarm C1 for a given metric scope, such as host, can be comprised of alarms R1, R2 that analyze two different metrics M1, M2, respectively. The rules of a composite alarm can be combined in one of three ways:

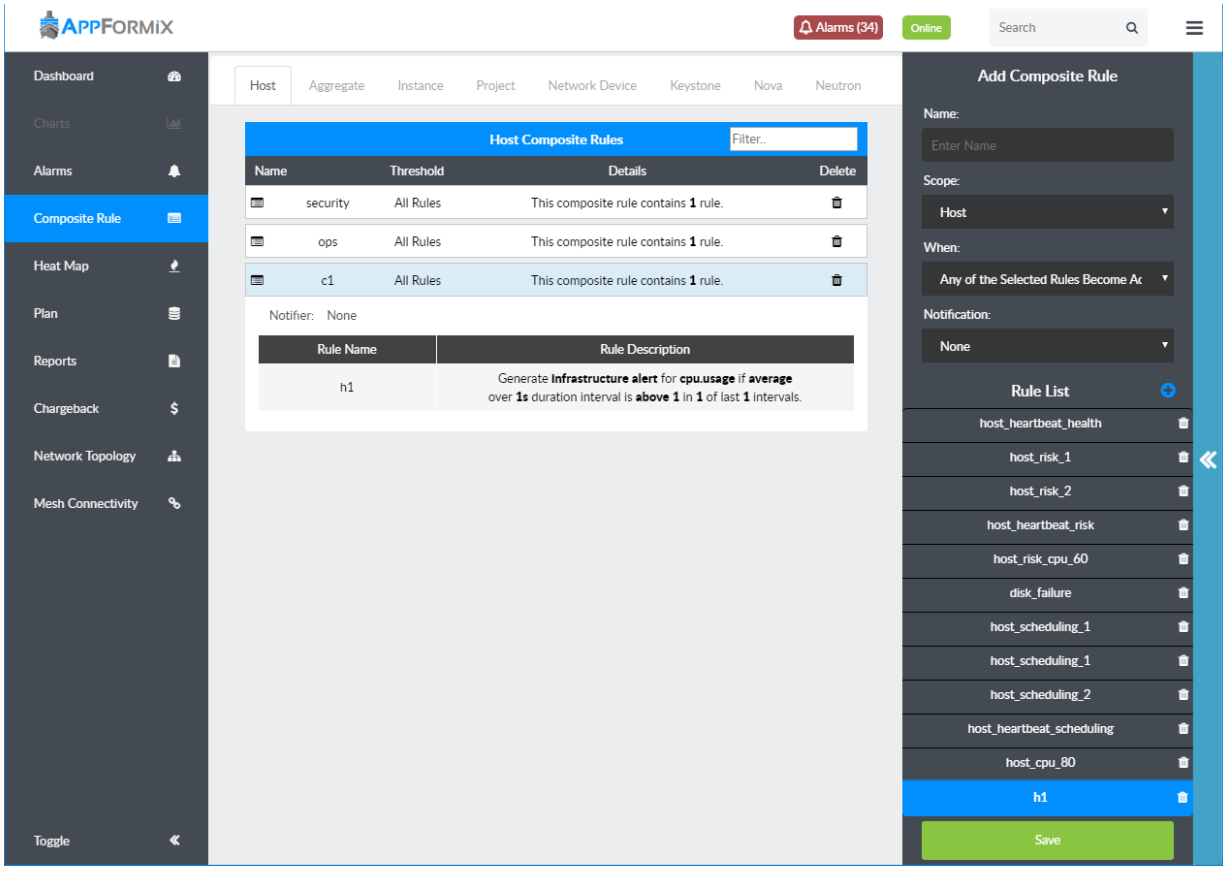
1. Active if any one of the rules is active.

2. Active if all of the rules are active.

3. Active if a weighted combination of rules is active. In this case, each rule is assigned a user-specified weight. The composite alarm is active when the sum of weights of active rules exceeds a user-specified threshold.

Figure 176 on page 261 shows an example of configured host compsite rules and the rule definition components for adding a composite rule.

Figure 176: Composite Rule Configuration



Host Composite Rules

| Name | Threshold | Details | Delete |
|---|---|---|---|
| security | All Rules | This composite rule contains **1** rule. | |
| ops | All Rules | This composite rule contains **1** rule. | |
| c1 | All Rules | This composite rule contains **1** rule. | |

Notifier: None

| Rule Name | Rule Description |
|---|---|
| h1 | Generate **Infrastructure alert** for **cpu.usage** if **average** over **1s** duration interval is **above** **1** in **1** of last **1** intervals. |

Add Composite Rule

Name: Enter Name

Scope: Host

When: Any of the Selected Rules Become Ac

Notification: None

Rule List

- host_heartbeat_health
- host_risk_1
- host_risk_2
- host_heartbeat_risk
- host_risk_cpu_60
- disk_failure
- host_scheduling_1
- host_scheduling_1
- host_scheduling_2
- host_heartbeat_scheduling
- host_cpu_80
- h1

## Add Composite Rule

To add a composite rule:

1. From the Contrail Insights Dashboard, select **Composite Rule**.
2. Select the tab for the entity that you want to configure the composite rule, such as host, aggregate, instance, project, and so on.
3. In the Add Composite Rule panel, add a name and the necessary parameters to create the composite rule.
4. Select **Save** to save your changes.

RELATED DOCUMENTATION

Alarms | 240

Notifications | 262

# Notifications

*Alarms* defines a policy that applies to a set of entities that are monitored, such as virtual machines in a project. A notification is generated when the condition of an alarm is observed for a given entity.

You can configure an alarm to post notifications to an external HTTP endpoint. Contrail Insights will post a JSON payload to the endpoint for each notification. The schema of the payload is as follows:

**NOTE**: The `string` and `0` are generic placeholders to indicate type of value; string and number, respectively.

```
{
    "apiVersion": "v1",
    "kind": "Alarm",
    "spec": {
        "name": "string",
        "eventRuleId": "string",
        "severity": "string",
        "metricType": "string",
        "mode": "string",
        "module": "string",
        "aggregationFunction": "string",
        "comparisonFunction": "string",
        "threshold": 0,
        "intervalDuration": 0,
        "intervalCount": 0,
        "intervalsWithException": 0
    },
    "status": {
        "timestamp": 0,
        "state": "string",
        "entityType": "string",
        "entityId": "string",
        "entityDetails": {}
    }
}
```

The `spec` object describes the alarm configuration for which this notification is generated. The `status` object describes the temporal event information for this particular notification, such as the time when the condition was observed and the entity on which the condition was observed. Table 27 on page 263 describes the object string values.

**Table 27: Object String Values**

| Value | Description |
|---|---|
| severity | Level of severity (critical, error, warning, information, none). |
| metricType | Measured value for hosts, instances and network devices. See *Metrics Collected by Contrail Insights*. |
| mode | One of two modes (alert, event). |
| module | The Analytics modules that generated the alarm (alarms, health/risk, service_alarms). |
| state | State of the alarm. For *alert* mode alarms, valid values are *active*, *inactive*, *learning*. For *event* mode alarms, the state is always *triggered*. |
| threshold | Units of threshold correspond to metricType. |
| entityType | One of instance, host, service, network device. |
| entityId | UUID of the entity. |

**Table 27: Object String Values *(Continued)***

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>entityDetails</td><td>Supplemental details about an entity. The contents of this object depend on the entityType. For a <i>host</i> or <i>service</i>, the object is empty. For an <i>instance</i>, the object contains hostId and projectId.
<pre>
{
    "entityDetails": {
        "hostId": "uuid",
        "projectId": "uuid"
    }
}
</pre></td></tr>
</table>

## RELATED DOCUMENTATION

- *Alarms*
- *Capacity Planning*
- *Chargeback*
- *Charts*
- *Health Monitor*
- *Heat Map*
- *Metrics Collected by Contrail Insights*
- *Extensibility Using Plug-Ins*
- *Reports*
- *Service Monitoring from the UI*

# Manage PagerDuty Notifications

SUMMARY

You can configure PagerDuty notification service, set up alarms, and verify that PagerDuty incidents are triggered from the Contrail Insights UI.

IN THIS SECTION

## Configure PagerDuty Notifications

Follow these steps to configure PagerDuty notifications:

1. Click the hamburger button and click **Settings**.

   The **Appformix Settings** page is displayed.

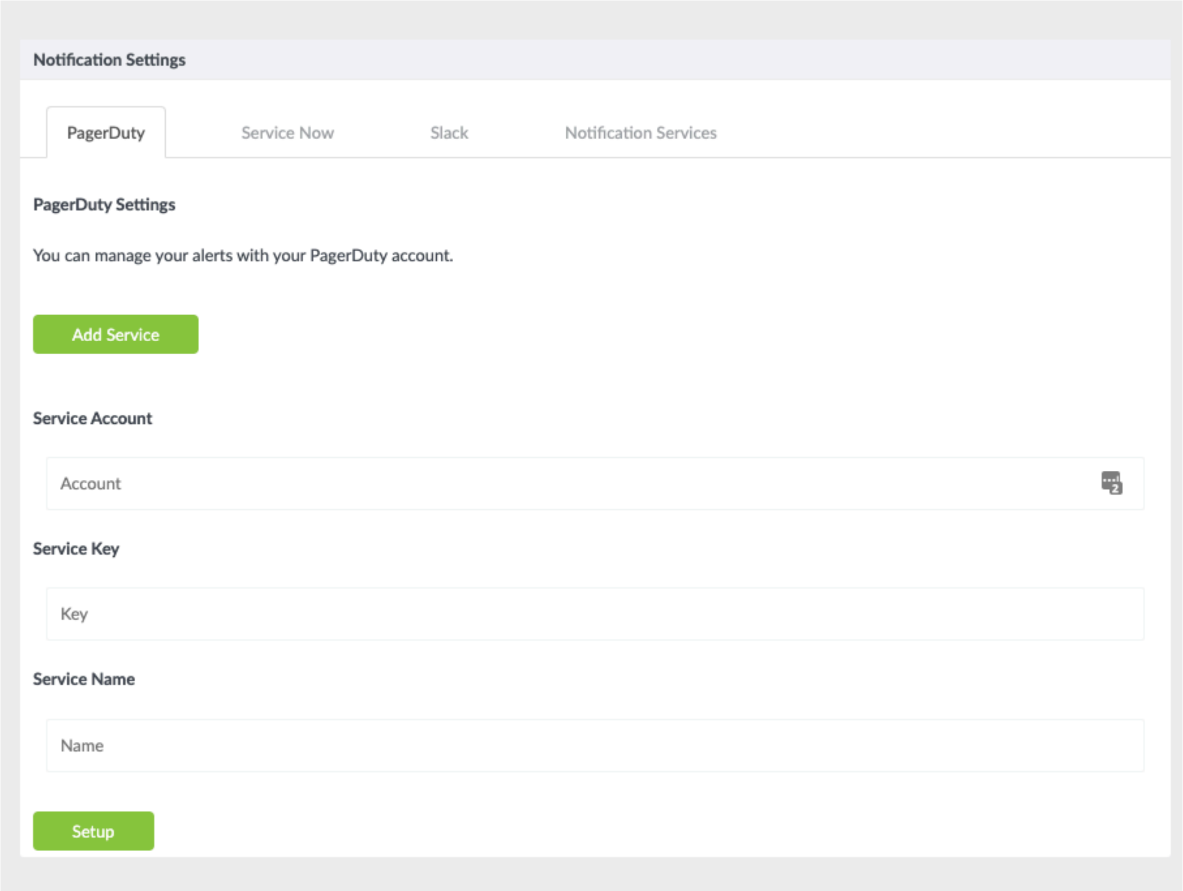
2. Click **Notification Settings** on the **Appformix Settings** page.

   The **PageDuty** tabbed page of the Notification Settings page is displayed.

3. Click **Add Service** on the PagerDuty tabbed page.

   The **Service Account**, **Service Key**, and **Service Name** text boxes are displayed.

**Figure 177: PagerDuty Settings**



Notification Settings

PagerDuty Settings

You can manage your alerts with your PagerDuty account.

4. Enter the following information:

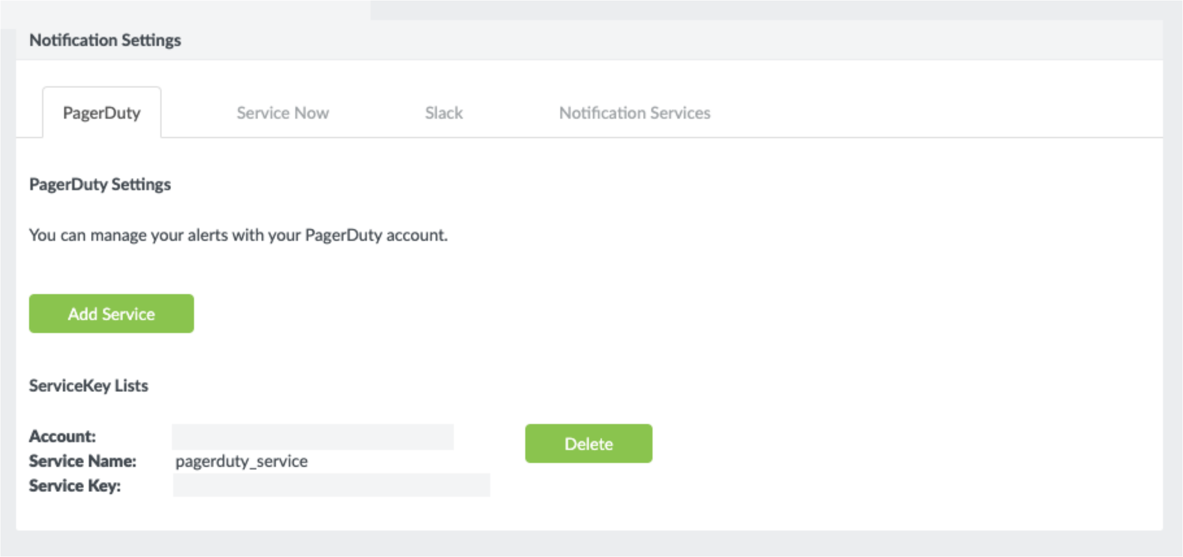
   a. Enter account information in the **Service Account** text box.

   b. Enter service key information in the **Service Key** text box.

   c. Enter a name for the service in the **Service Name** text box.

   Click **Setup** to add the new PagerDuty settings. The PageDuty settings that you configured is displayed in the **PagerDuty** tabbed page.

Figure 178: View PagerDuty Settings



PagerDuty Settings

You can manage your alerts with your PagerDuty account.

ServiceKey Lists

**Account:**
**Service Name:** pagerduty_service
**Service Key:**

## Set Up Alarms

Follow these steps to set up an alarm:

1. Click **Alarms** in the left-nav bar.

   The list available alarms are displayed.

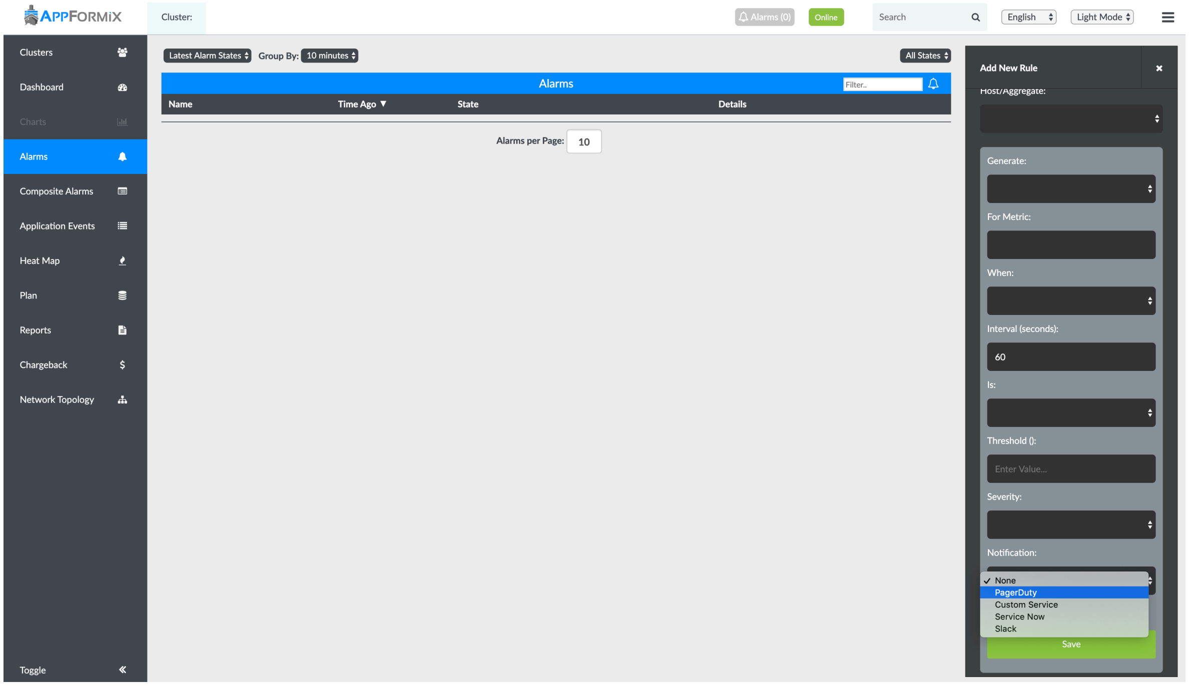
2. Click **Add Rule** in the **Alarm Rules** section.

   The **Add New Rule** section is displayed.

3. Enter the following information in the **Add New Rule** section to set up an alarm:

| Field | Description |
|---|---|
| Name | Enter a name for the alarm. |
| Module | Select **Alarm** from the options. |
| Alarm Rule Type | Select **Static** rule type. |
| Scope | Select **Host** as the scope. |
| Interval (seconds) | Enter **60** seconds as the interval. |
| Notification | Select **PagerDuty** from the drop-down list. The **Services** drop-down list is displayed. |
| Services | Select the services you want to apply to this alarm from the drop-down list. |

Figure 179: Select PagerDuty from Notifications drop-down list

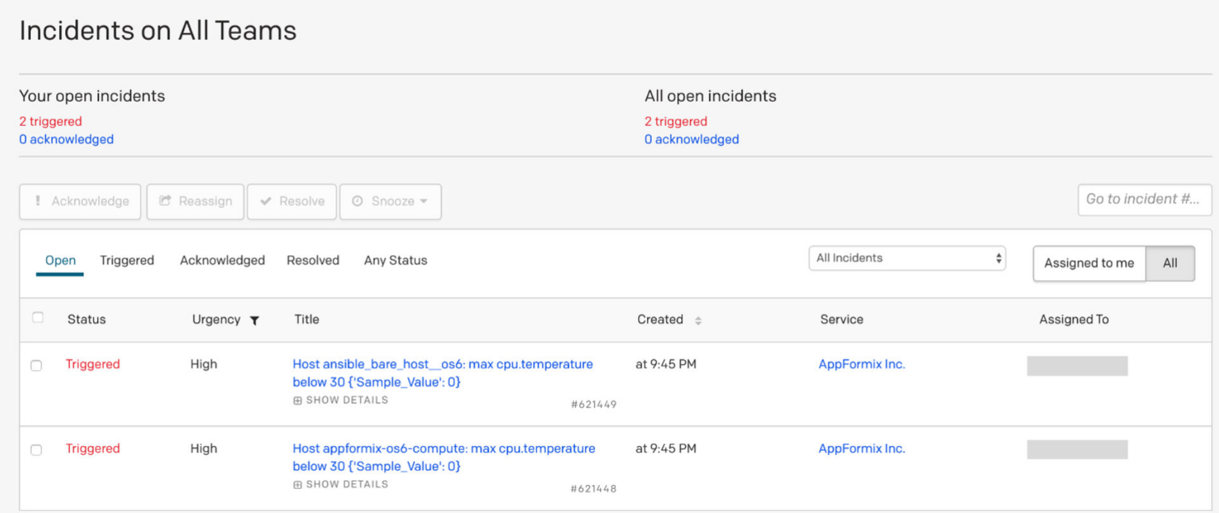


4. Click **Save** to save the alarm.

## Verification

After you have configured notifications and set up alarms, you can verify that PagerDuty incidents are being triggered from the PagerDuty UI. Navigate to the **Incidents** page to view triggered alarms.

### Figure 180: View Triggered Alarms



Incidents on All Teams

Your open incidents
2 triggered
0 acknowledged

All open incidents
2 triggered
0 acknowledged

| □ | Status | Urgency | Title | Created | Service | Assigned To |
|---|---|---|---|---|---|---|
| □ | Triggered | High | Host ansible_bare_host__os6: max cpu.temperature below 30 {'Sample_Value': 0}<br>⊞ SHOW DETAILS #621449 | at 9:45 PM | AppFormix Inc. | [redacted] |
| □ | Triggered | High | Host appformix-os6-compute: max cpu.temperature below 30 {'Sample_Value': 0}<br>⊞ SHOW DETAILS #621448 | at 9:45 PM | AppFormix Inc. | [redacted] |

5

CHAPTER

# Contrail Insights Chargeback

# Chargeback

IN THIS SECTION

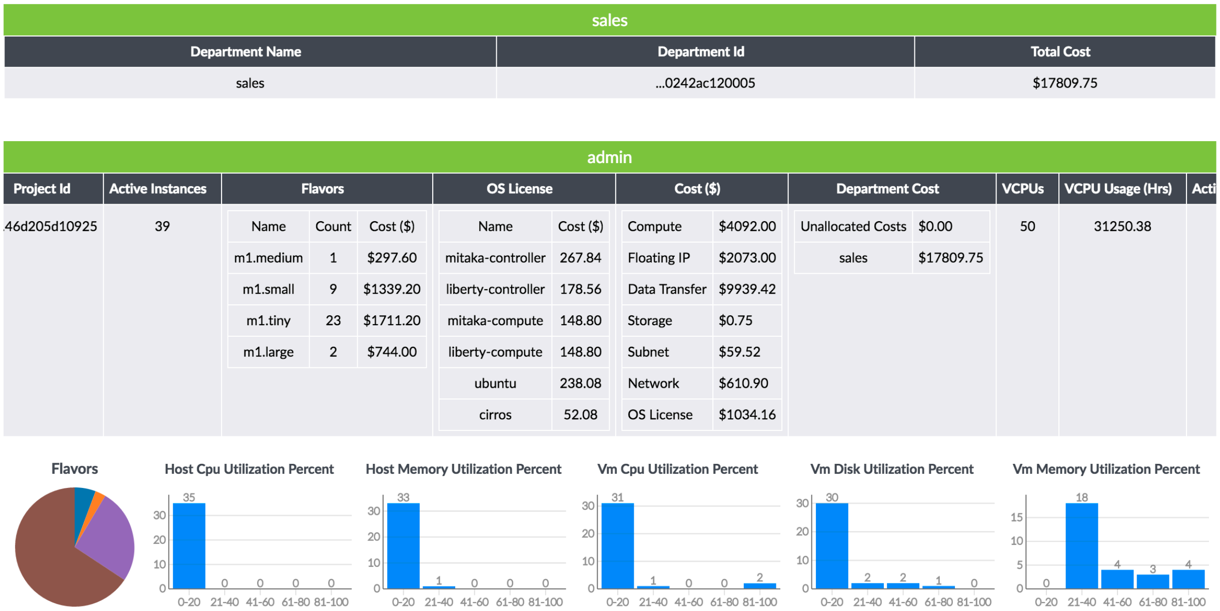
Contrail Insights Chargeback calculates a cost for use of compute, network, and storage resources. The price of each resource is configurable by an administrator. Chargeback relies on two concepts of organization: project and departments.

**Project** A project is a collection of instances. A project is a technical organizational unit, often defined by a cloud management system. For example, in OpenStack, a project (formerly called tenant) is the means by which users share a quota of resource allocation and a collection of virtual machines, virtual networks, and storage volumes.

**Department** A department is a business organizational unit defined in Contrail Insights because the technical organization provided by project may not map directly to business groups in an organization. An administrator can assign the cost accrued by a project to one or more departments, on a percentage basis.

Contrail Insights generates monthly invoices for each department. A monthly invoice shows total cost charged to a department. The total cost is the sum of a department's share of the cost of each project. In the Contrail Insights Dashboard, a user may view a monthly invoice that displays detailed breakdown of cost based on compute, network, storage, and other resources. Figure 181 on page 272 shows resource consumption by department.

### Figure 181: Resource by Consumption Department Report

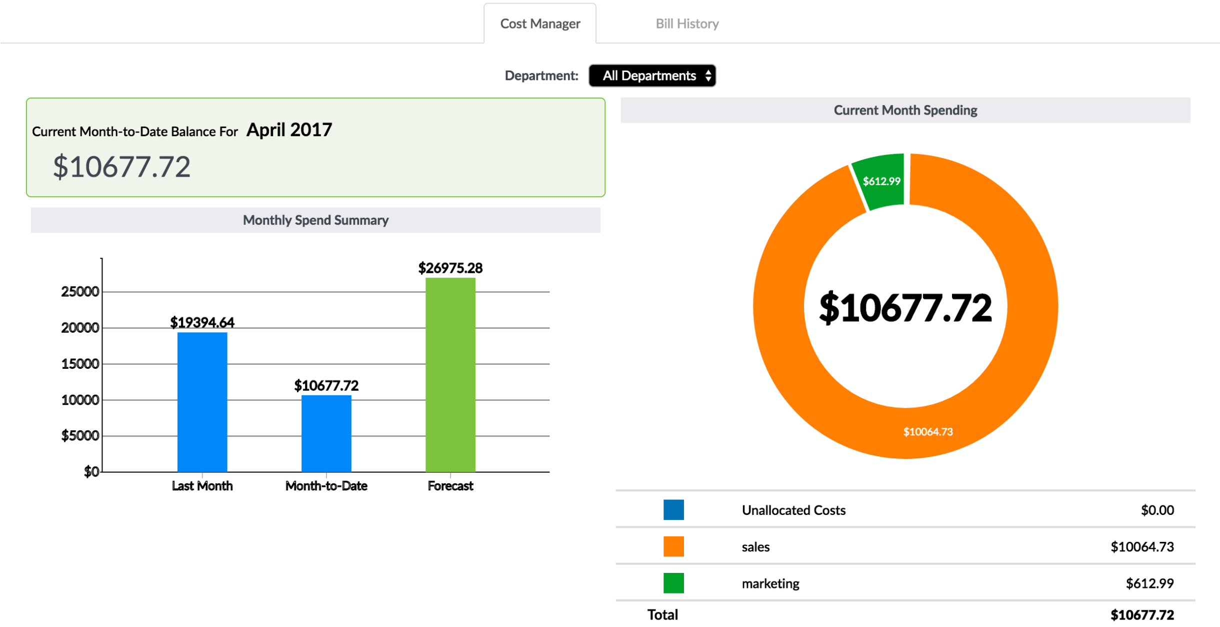


| sales | | |
|---|---|---|
| Department Name | Department Id | Total Cost |
| sales | ...0242ac120005 | $17809.75 |

| admin | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Project Id | Active Instances | Flavors | | | OS License | | Cost ($) | | Department Cost | | VCPUs | VCPU Usage (Hrs) | Acti |
| 46d205d10925 | 39 | Name | Count | Cost ($) | Name | Cost ($) | Compute | $4092.00 | Unallocated Costs | $0.00 | 50 | 31250.38 | |
| | | m1.medium | 1 | $297.60 | mitaka-controller | 267.84 | Floating IP | $2073.00 | sales | $17809.75 | | | |
| | | m1.small | 9 | $1339.20 | liberty-controller | 178.56 | Data Transfer | $9939.42 | | | | | |
| | | m1.tiny | 23 | $1711.20 | mitaka-compute | 148.80 | Storage | $0.75 | | | | | |
| | | m1.large | 2 | $744.00 | liberty-compute | 148.80 | Subnet | $59.52 | | | | | |
| | | | | | ubuntu | 238.08 | Network | $610.90 | | | | | |
| | | | | | cirros | 52.08 | OS License | $1034.16 | | | | | |

In addition to monthly invoices, the month-to-date cost and projected cost for the current month are displayed in the Cost Manager tab of the Chargeback page. Figure 182 on page 272 shows an example of the Cost Manager tab details.

### Figure 182: Cost Manager Chargeback Details



Department: All Departments

Current Month-to-Date Balance For April 2017

$10677.72

| | | |
|---|---|---|
| 🟦 | Unallocated Costs | $0.00 |
| 🟧 | sales | $10064.73 |
| 🟩 | marketing | $612.99 |
| Total | | **$10677.72** |

See the following topics for information about configuring chargeback costs.

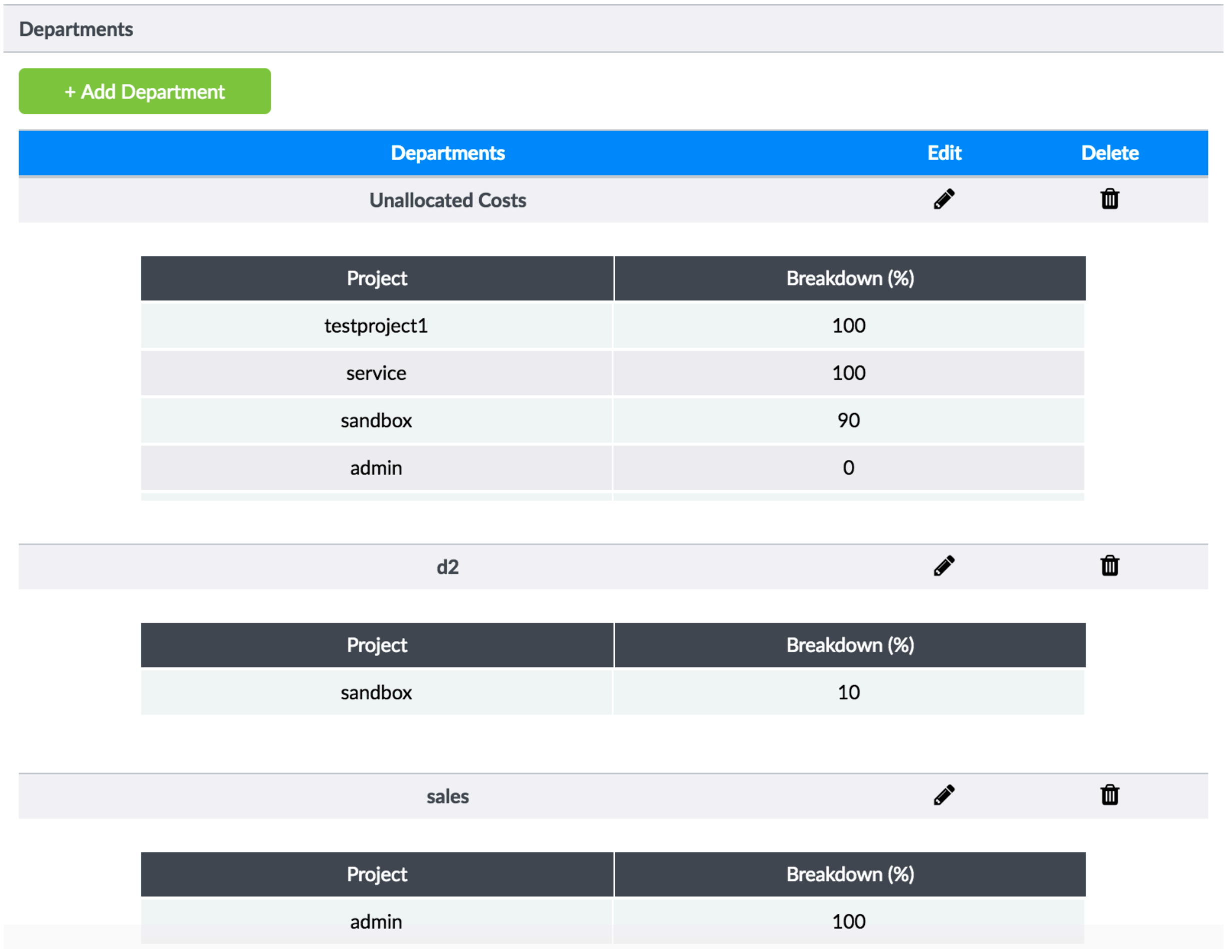
## Configure Departmental Shared Costs

The costs accrued by a project may be charged to one or more departments. When multiple departments share the financial cost of a project, an administrator can split the cost of a project among multiple departments.

To configure the percentage of project cost, select **Settings > Chargeback > Departments**. The Departments table displays each department, as shown in Figure 184 on page 275. Click a department name to show or hide the department details that display the percentage of per-project cost that will be charged to the department.

There is a default department called Unallocated Costs which accrues costs for any project that has not been assigned to any department. Figure 183 on page 274 shows the default department unallocated costs.

**Figure 183: Default Department Unallocated Costs**



| Departments | Edit | Delete |
|---|---|---|
| Unallocated Costs | ✎ | 🗑 |

| Project | Breakdown (%) |
|---|---|
| testproject1 | 100 |
| service | 100 |
| sandbox | 90 |
| admin | 0 |

| Departments | Edit | Delete |
|---|---|---|
| d2 | ✎ | 🗑 |

| Project | Breakdown (%) |
|---|---|
| sandbox | 10 |

| Departments | Edit | Delete |
|---|---|---|
| sales | ✎ | 🗑 |

| Project | Breakdown (%) |
|---|---|
| admin | 100 |

To add a new department:

1. Click **Add Department** and type a name for the new department.
2. Click **Add**.

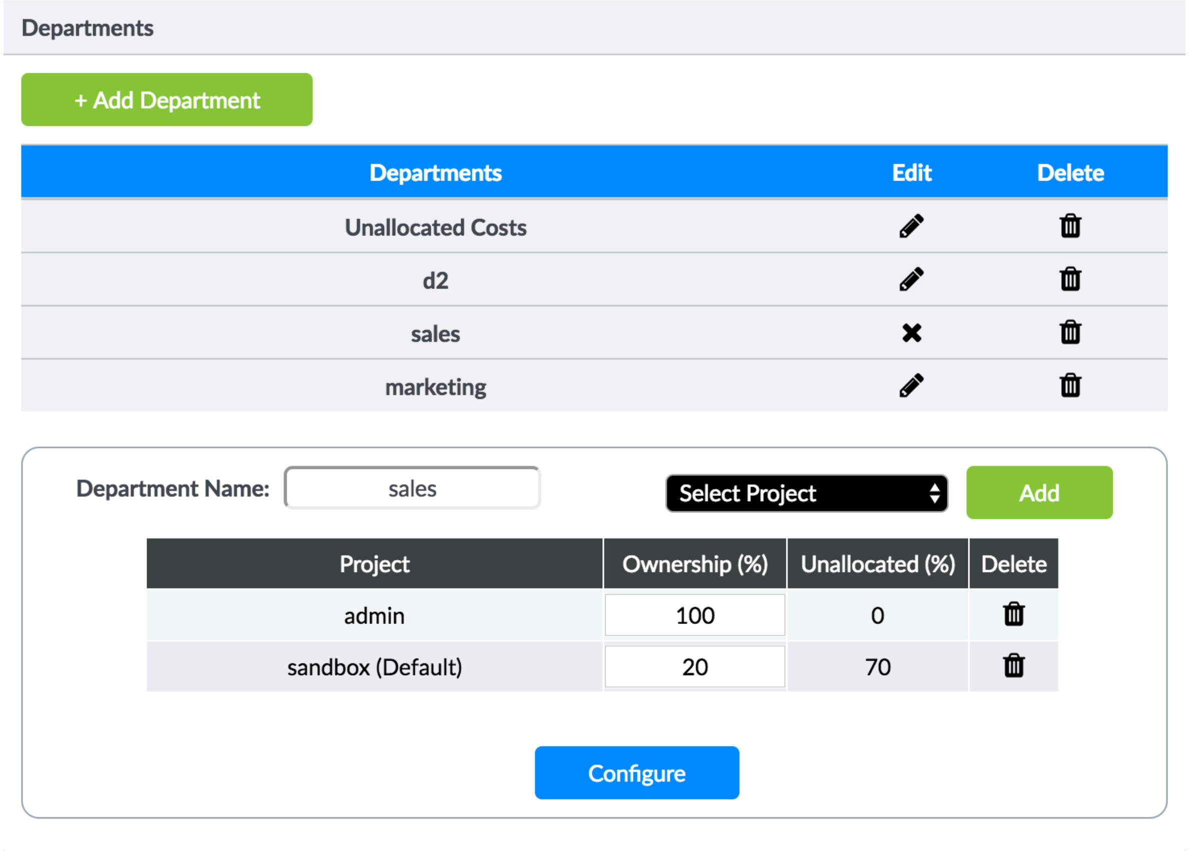
   The new department appears in the Department table.

To configure the list of projects assigned to a department:

1. Select the pencil icon to edit a department.

   A department configuration box appears following the Departments table, as shown in Figure 184 on page 275.

Figure 184: Configure Projects Assigned to Departments



Departments

| Departments | Edit | Delete |
|---|---|---|
| Unallocated Costs | ✎ | 🗑 |
| d2 | ✎ | 🗑 |
| sales | ✖ | 🗑 |
| marketing | ✎ | 🗑 |

Department Name: sales

| Project | Ownership (%) | Unallocated (%) | Delete |
|---|---|---|---|
| admin | 100 | 0 | 🗑 |
| sandbox (Default) | 20 | 70 | 🗑 |

**Project column** Lists each project for which the department accrues cost.

**Ownership (%) column** Indicates the percentage of a project's cost that is assigned to the department being configured. The ownership percentage value can be edited.

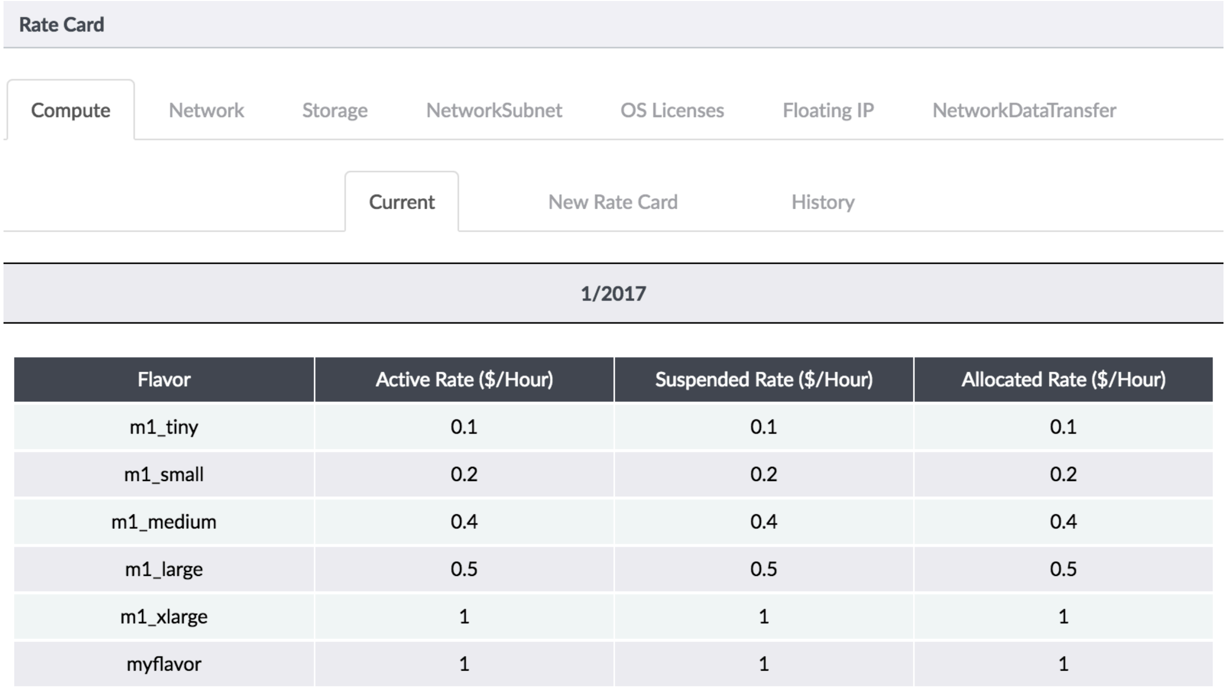
**Unallocated (%) column** Indicates the percent of a project's cost that is not allocated to any department.

2. To add a project to the table, select the project in the Select Project drop-down list and click **Add**.
3. After editing the department configuration, click **Configure** to save changes.
4. (Optional) To cancel changes without saving, click the **x** icon in the Edit column of the Departments table.

## Configure Rates Charged by Using the Rate Card

The rate charged for resources is configured in the rate card. Figure 185 on page 276 shows the resource hourly rate card per flavor by active, suspended, or allocated rates.

**Figure 185: Resource Hourly Rate Card per Flavor by Active, Suspended, or Allocated Rates**



1/2017

| Flavor | Active Rate ($/Hour) | Suspended Rate ($/Hour) | Allocated Rate ($/Hour) |
|---|---|---|---|
| m1_tiny | 0.1 | 0.1 | 0.1 |
| m1_small | 0.2 | 0.2 | 0.2 |
| m1_medium | 0.4 | 0.4 | 0.4 |
| m1_large | 0.5 | 0.5 | 0.5 |
| m1_xlarge | 1 | 1 | 1 |
| myflavor | 1 | 1 | 1 |

To configure the rate charged for resources:

1. Select **Settings** > **Chargeback** > **Rate Card**.
2. Select a tab for a resource type to display and to configure the rate card for that resource. The descriptions of the tabs are as follows:

   **Current** Shows the current rate card and the date that the rate went into effect.

   **History** Shows previous rate cards for a resource type, organized as a list by the effective date of the past rate card.

   **New Rate Card** Allows you to configure a new rate card for a resource.

   **Effective Date** Shows the month and year when the new rate card will start being used. The effective date must be later than the currently configured rate card.

3. Select **Save** to save your changes.

## Configure Compute Costs

Compute cost is charged by the hour that an instance is in one of the following states: active, suspended, or allocated. The compute cost is based on the amount of compute resources (CPU, memory, local storage) that is allocated (statically) for an instance on a host. The hourly rate is configured for each flavor type.

Instance states are defined as follows:

**Active** An instance is running on a compute host. Corresponding OpenStack state is *Active*.

**Suspended** An instance has been paused or suspended. Runtime state of such an instance has been preserved in memory or on disk. Compute resources assigned to such an instance are still allocated on a compute host. Corresponding OpenStack states are *Paused*, *Suspended*.

**Allocated** An instance is stopped or shut off. Runtime state of such an instance has not been preserved, but its disk image is still present on a compute host. The compute resources assigned to such an instance are still allocated on a compute host. Corresponding OpenStack states are *Shutoff*, *Stopped*.

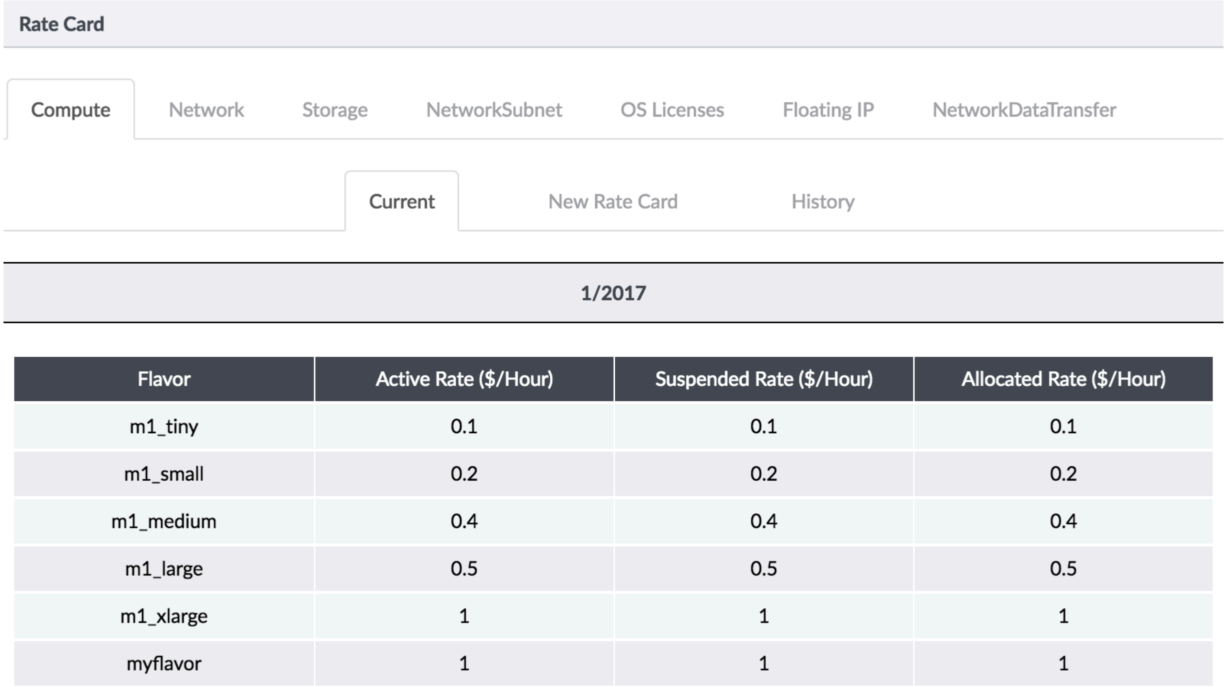
To configure compute costs:

1. Select **Settings** > **Chargeback** > **Rate Card**.

2. Select the **Compute** tab to display and to configure the price for each flavor type and instance states.

   The Current tab displays the rate card that is active. Select the **New Rate Card** tab to configure a new rate card. Figure 186 on page 278 shows the Compute tab with the active rates.

**Figure 186: Compute Cost Current Tab**



1/2017

| Flavor | Active Rate ($/Hour) | Suspended Rate ($/Hour) | Allocated Rate ($/Hour) |
|---|---|---|---|
| m1_tiny | 0.1 | 0.1 | 0.1 |
| m1_small | 0.2 | 0.2 | 0.2 |
| m1_medium | 0.4 | 0.4 | 0.4 |
| m1_large | 0.5 | 0.5 | 0.5 |
| m1_xlarge | 1 | 1 | 1 |
| myflavor | 1 | 1 | 1 |

## Configure Network Interface Costs

An instance can be charged an hourly cost for the use of network interfaces attached to the instance. Contrail Insights discovers the list of network interfaces from the cloud management system, such as Nova in OpenStack.

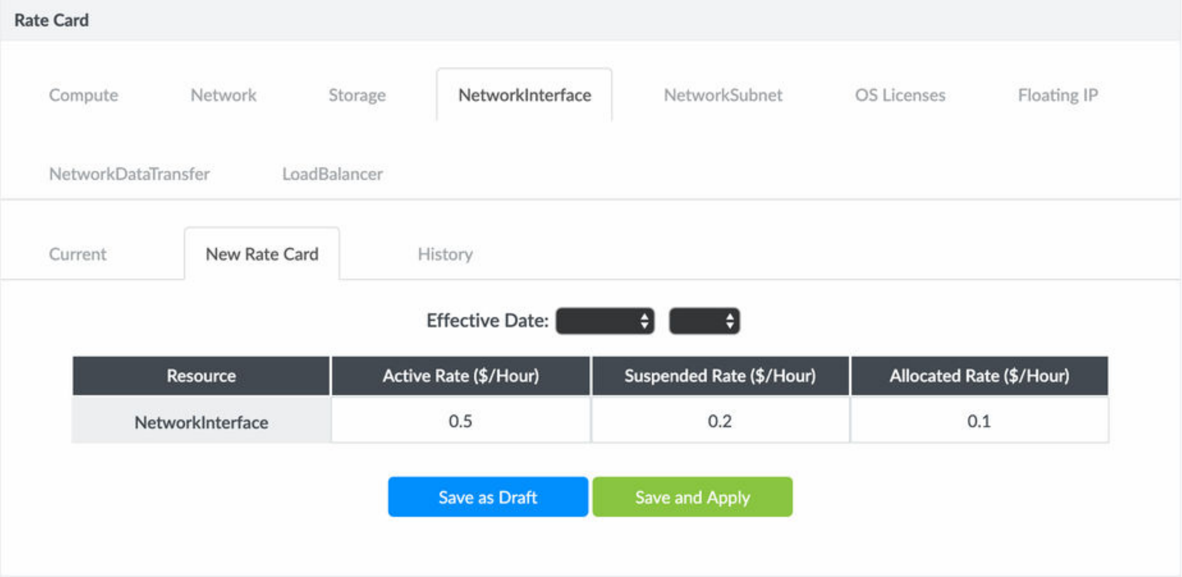
Network interface costs can be charged for active, suspended, and allocated instances. Figure 187 on page 279 shows a network interface rate card with active, suspended, and allocated instances hourly rates. Refer to instance states in *Compute Cost Current Tab* for a detailed description of the instance states.

To configure network interface costs:

1. Select **Settings** > **Chargeback** > **Rate Card**.

2. Select the NetworkInterface tab to display and to configure the price for each instance state.

Figure 187: Network Interface Rate Card with Active, Suspended, and Allocated Hourly Rates



| Resource | Active Rate ($/Hour) | Suspended Rate ($/Hour) | Allocated Rate ($/Hour) |
|---|---|---|---|
| NetworkInterface | 0.5 | 0.2 | 0.1 |

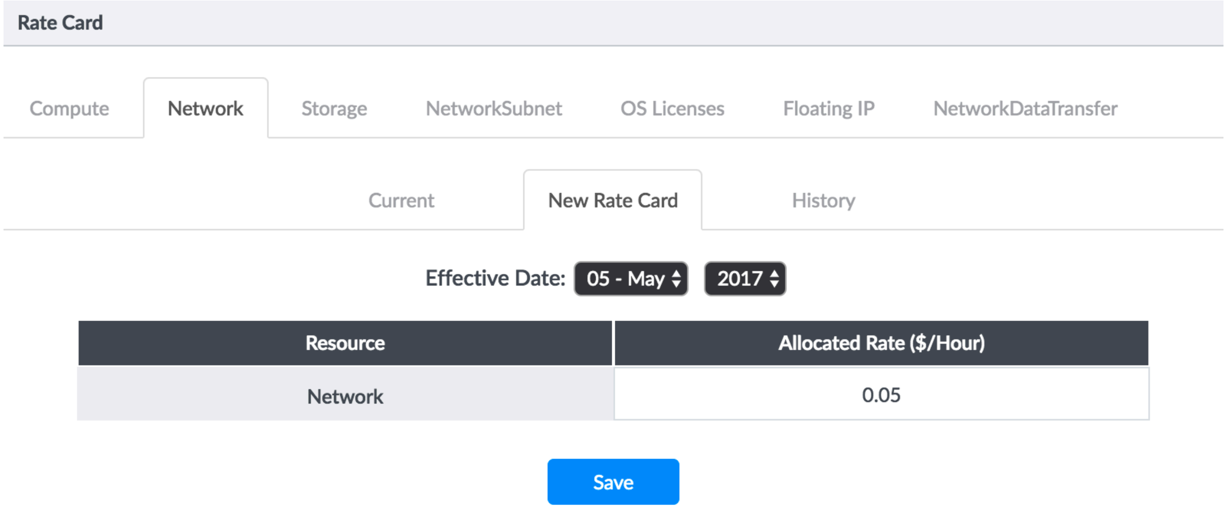
## Configure Network Resource Costs

Network resources that can be charged are virtual networks, virtual subnets, floating IP addresses, and data transfers. To configure network resource prices, select the **Network** tab.

### Virtual Network

A virtual network allocated to a project is charged on an hourly basis, as shown in Figure 188 on page 280.

Figure 188: Virtual Network Rate Card with Allocated Hourly Rate



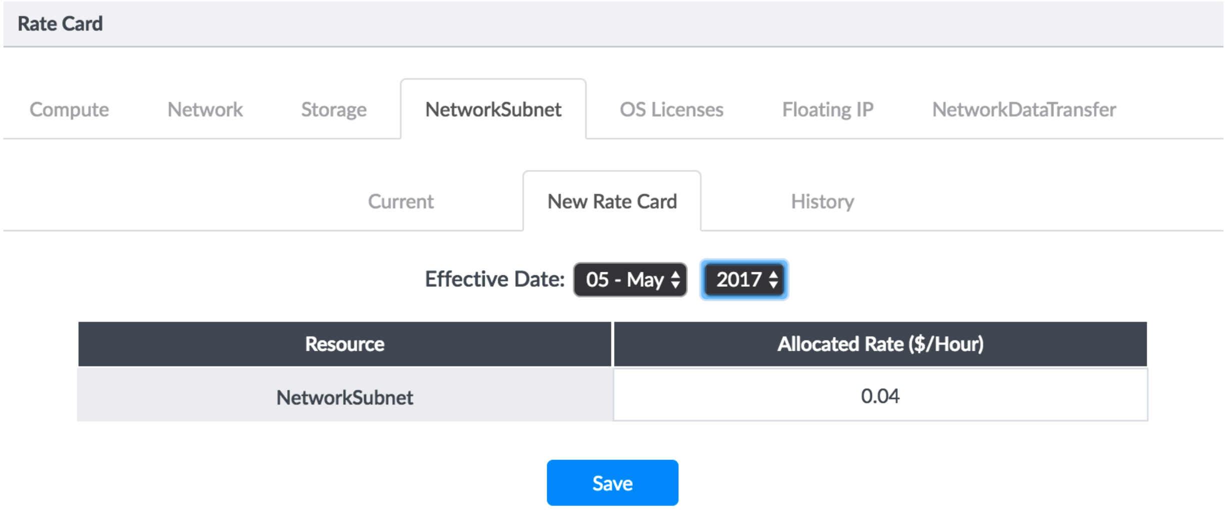
Effective Date: 05 - May 2017

| Resource | Allocated Rate ($/Hour) |
|---|---|
| Network | 0.05 |

## Virtual Network Subnet

A virtual network subnet allocated to a project is charged on an hourly basis. Figure 189 on page 280 shows an example.

Figure 189: Virtual Network Subnet Rate Card with Allocated Hourly Rate



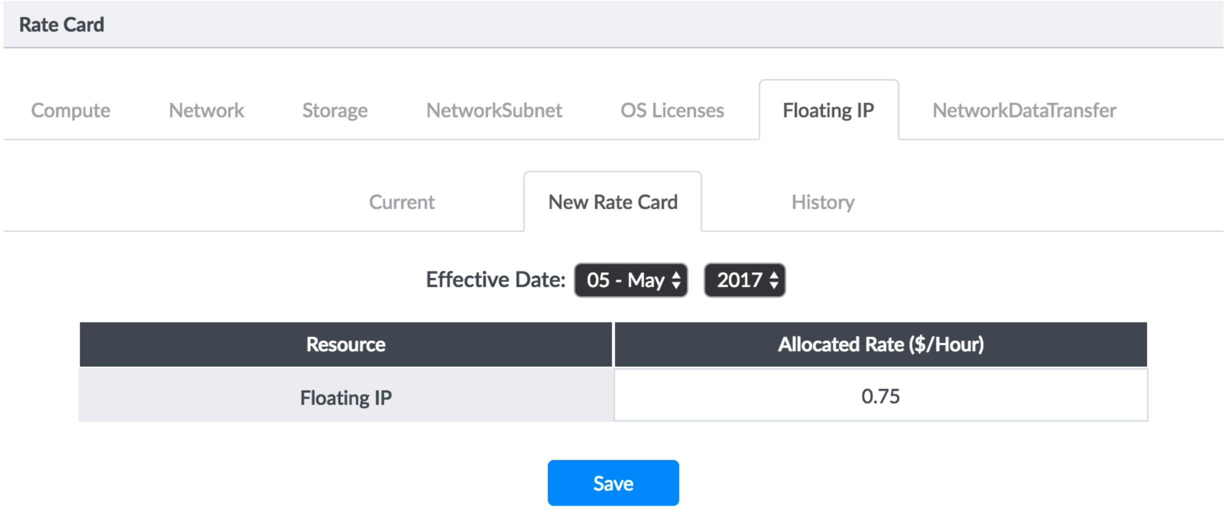
Effective Date: 05 - May 2017

| Resource | Allocated Rate ($/Hour) |
|---|---|
| NetworkSubnet | 0.04 |

### Floating IP Addresses

Each floating IP address is charged on a hourly basis for allocation of the IP address. The rate is expressed as cost per hour ($/hour) per IP address. Figure 190 on page 281 shows an example of an allocated rate for each floating IP address.

#### Figure 190: Floating IP Address Rate Card with Allocated Hourly Rate



Effective Date: 05 - May 2017

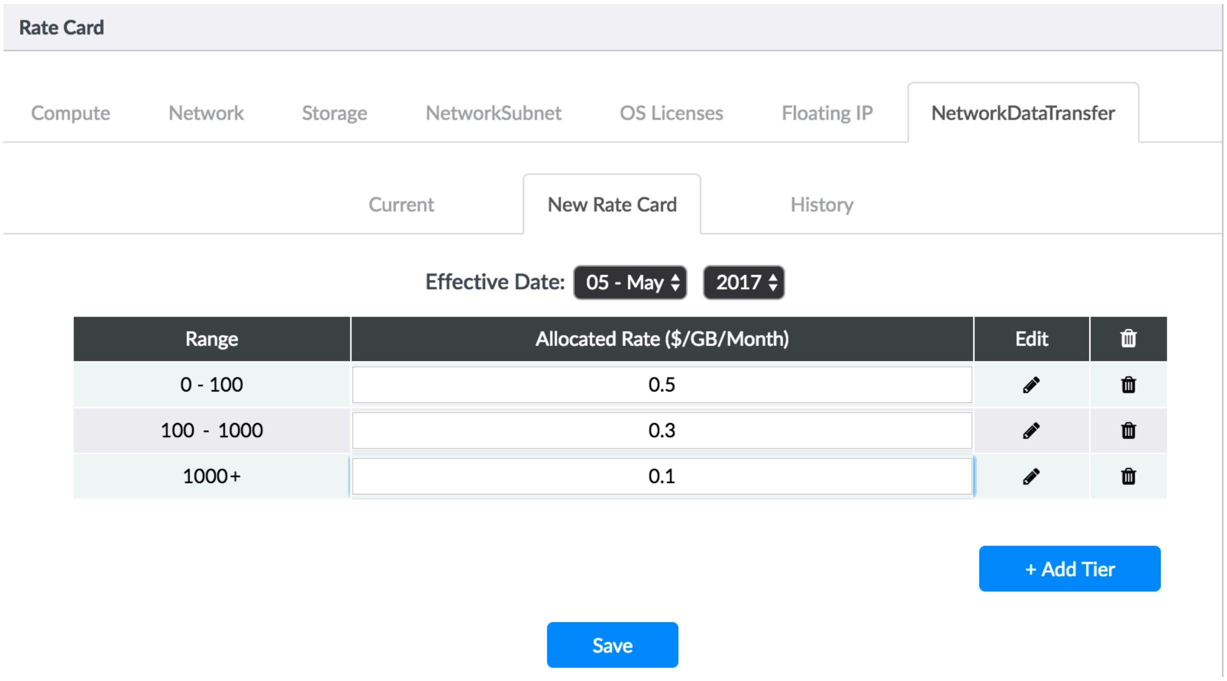
| Resource | Allocated Rate ($/Hour) |
|---|---|
| Floating IP | 0.75 |

### Network Data Transfer

Network data transfer cost is calculated according to a progressive, tiered pricing model on a per instance basis. Charges are calculated in gigabyte units of data transmitted by an instance (egress bytes). Data transfer amount is always rounded up to the next whole gigabyte. There is no charge for data received by an instance (ingress bytes).

An administrator creates price tiers by clicking **Add Tier**. See Figure 191 on page 282. The first tier starts from 0 GB. Each subsequent tier starts on the first gigabyte after the previous tier, and ends at a user-specified byte count. The final rate tier applies to all data transfer amounts that exceed the second-to-last tier. Figure 191 on page 282 shows a network data transfer rate card with allocated rates by gigabyte ranges.

Figure 191: Data Transfer Rate Card with Allocated Monthly Rates by Gigabyte Ranges



Effective Date: 05 - May 2017

| Range | Allocated Rate ($/GB/Month) | Edit | 🗑 |
|---|---|---|---|
| 0 - 100 | 0.5 | ✎ | 🗑 |
| 100 - 1000 | 0.3 | ✎ | 🗑 |
| 1000+ | 0.1 | ✎ | 🗑 |

### Example: Network Data Transfer

In Figure 191 on page 282, three tiers are configured: 0-100 GB, 100-1000 GB, and 1000+ GB. Suppose an instance transmits 399.4 GB of data during a billing period. The data transfer cost is calculated for 400 GB of data as follows: 100 GB * $0.50/GB + 300 GB * $0.30/GB = $140.

## Configure Load Balancer Costs

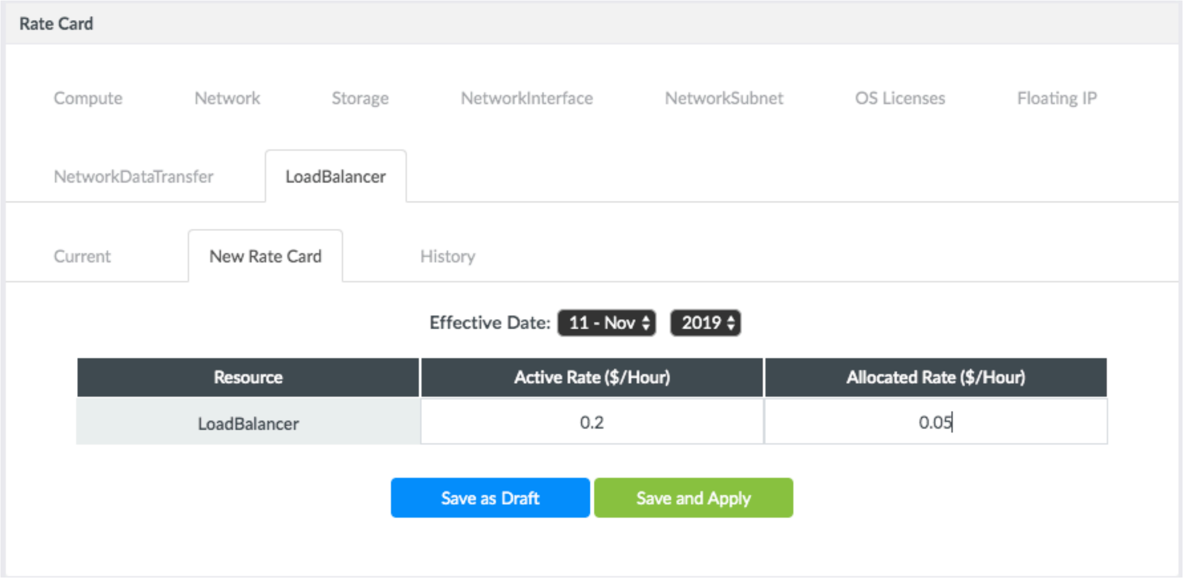
Contrail Insights discovers the list of configured Load Balancers from the cloud management system, such as Octavia in OpenStack. A load balancer is charged on an hourly basis if it is in one of the following states: active or allocated. These states are defined as follows:

- Active—Provisional status of `loadbalancer` is ACTIVE.
- Allocated—Provisional status of `loadbalancer` is one of: PENDING_CREATE, PENDING_UPDATE, PENDING_DELETE.

To configure load balancer costs:

1. Select **Settings** > **Chargeback** > **Rate Card**.
2. Select the LoadBalancer tab to display and to configure the price for each load balancer state.

Figure 192: Load Balancer Rate Card with Active and Allocated Hourly Rates



Effective Date: 11 - Nov 2019

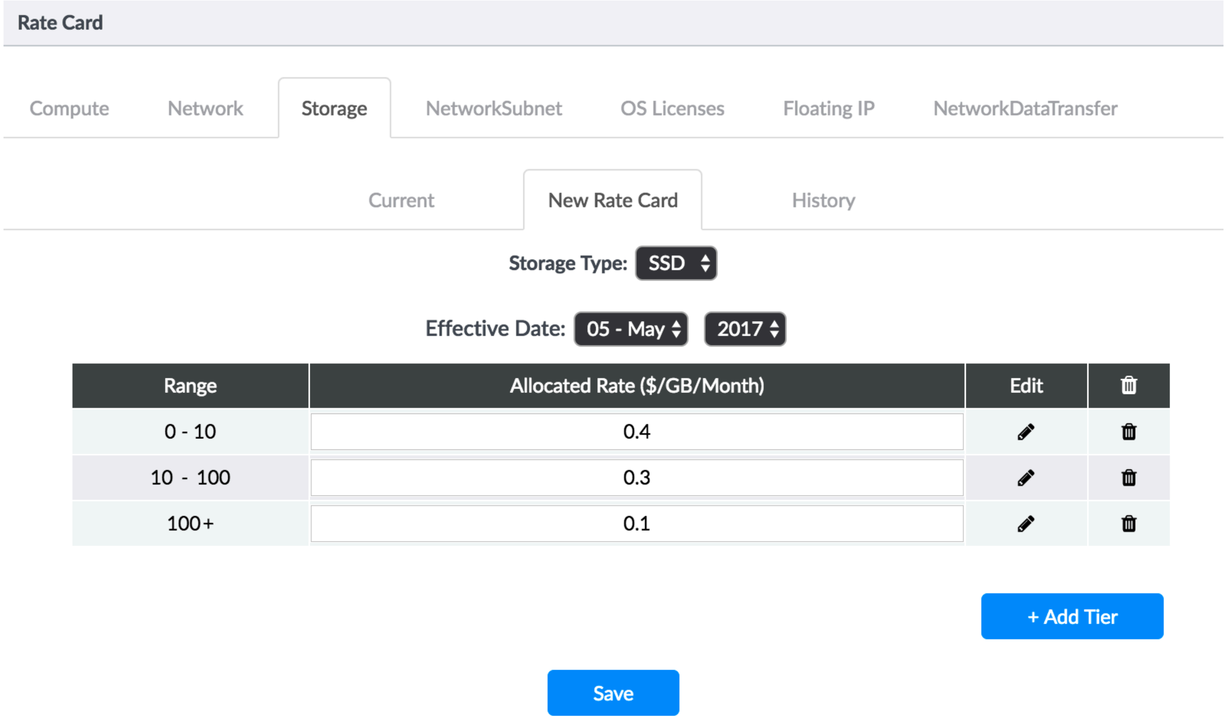
| Resource | Active Rate ($/Hour) | Allocated Rate ($/Hour) |
|---|---|---|
| LoadBalancer | 0.2 | 0.05 |

## Configure Storage Costs

Storage cost is calculated using a progressive, tiered pricing model on a per project basis. Storage cost is charged to a project hourly for the total volume storage allocated by a project. If a project allocates a 500 GB volume, but consumes only 100 GB in that volume, then the project is charged for the entire 500 GB allocation. Figure 193 on page 284 shows the Storage tab.

A rate card may be configured for each storage type. The storage types are discovered by Contrail Insights from the cloud management system (such as, Cinder in OpenStack). Figure 193 on page 284 shows a storage rate card with allocated rates by gigabyte ranges.

**Figure 193: Storage Rate Card with Allocated Monthly Rates by Gigabyte Ranges**



Storage Type: SSD

Effective Date: 05 - May 2017

| Range | Allocated Rate ($/GB/Month) | Edit | 🗑 |
|---|---|---|---|
| 0 - 10 | 0.4 | ✎ | 🗑 |
| 10 - 100 | 0.3 | ✎ | 🗑 |
| 100+ | 0.1 | ✎ | 🗑 |

To configure storage prices:

1. Select **Settings** > **Chargeback** > **Rate Card**, then select the **Storage** tab.

2. Select the **Storage Type** for which the rate card applies.

3. Click **+Add Tier** to add a new storage tier.

   Specify the end size in gigabytes and the cost per gigabyte in the tier.

4. Click **Save** to save the rate card.

### Example: Storage

Consider a rate card in which three tiers are configured: 0-10 GB, 10-100 GB, and 100+ GB. Suppose a project allocates a 25 GB volume for 10 hours, and subsequently allocates an additional 200 GB volume for 20 hours.

For each of the first 10 hours, the project is charged for 25 GB, calculated as follows:

```
(10 GB * $0.40/GB/hour) + (15 GB * $0.30/GB/hour) = $8.50/hour
```

For each of the next 20 hours, the project is charged for 225 GB, calculated as follows:

```
(10 GB * $0.40/GB/hour) + (90 GB * $0.30/GB/hour) + (125 GB * $0.10/GB/hour) = $43.50/hour
```

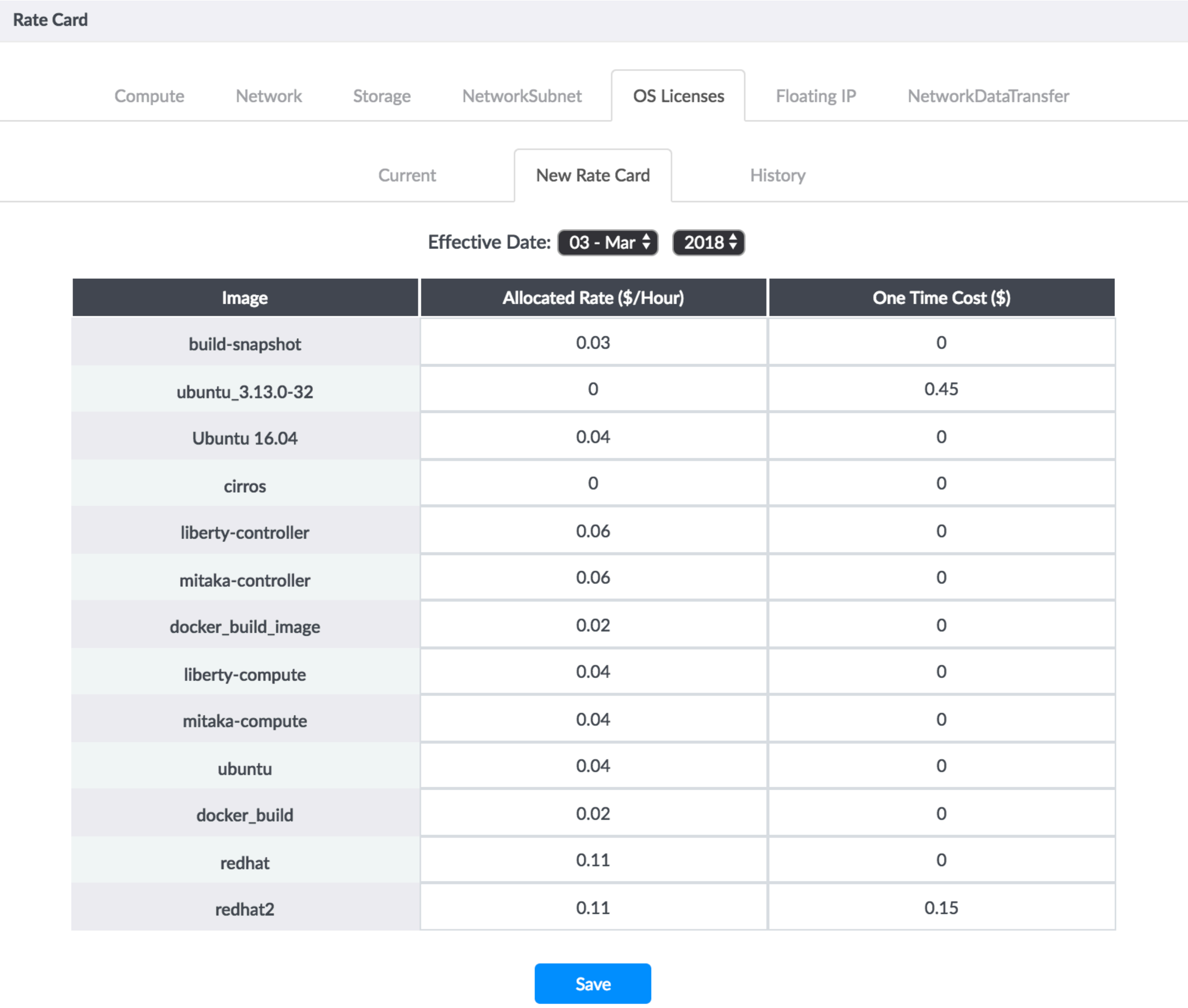
In total, for the 30 hours, the project is charged: $8.50 * 10 + $43.50 * 20 = $955.

## Configure OS License Rates

Each instance can be charged an OS license cost to use a particular OS image to boot the instance. The OS license cost is assigned to each disk image that might be used to create an instance. Contrail Insights discovers the list of images from the cloud management system, such as Glance in OpenStack.

OS license cost can be charged both on a hourly basis and as a one-time cost. The Allocated Rate is a cost per hour that is accounted for each hour that an instance is provisioned with a particular image. The One-Time Cost is charged each time that an instance is created that uses a particular image. Figure 194 on page 286 shows an OS license rate card with allocated hourly rates and one-time cost per image used.

Figure 194: OS License Rate Card with Allocated Hourly Rates and One-Time Cost per Image Used



Effective Date: 03 - Mar 2018

| Image | Allocated Rate ($/Hour) | One Time Cost ($) |
|---|---|---|
| build-snapshot | 0.03 | 0 |
| ubuntu_3.13.0-32 | 0 | 0.45 |
| Ubuntu 16.04 | 0.04 | 0 |
| cirros | 0 | 0 |
| liberty-controller | 0.06 | 0 |
| mitaka-controller | 0.06 | 0 |
| docker_build_image | 0.02 | 0 |
| liberty-compute | 0.04 | 0 |
| mitaka-compute | 0.04 | 0 |
| ubuntu | 0.04 | 0 |
| docker_build | 0.02 | 0 |
| redhat | 0.11 | 0 |
| redhat2 | 0.11 | 0.15 |

## Configure SNAT Logical Routers Network Data Transfer Costs

Starting with Contrail Insights Release 3.3.5, you can configure SNAT logical routers network data transfer costs. Logical routers are OpenStack resources that are associated with an OpenStack project. Contrail Insights collects network metrics for Source Network Address Translation (SNAT) logical routers and charge per tenant basis. Similar to network data transfer, the cost for configuring SNAT logical router is calculated as per a tiered pricing model.

Charges are applied by calculating the units of gigabyte (GB) of data transferred by an SNAT logical router (egress bytes) between the start and end timestamps of the chargeback report. The units of data transferred is always rounded up to the next whole GB. There is no charge levied for data received by the instance (ingress bytes).

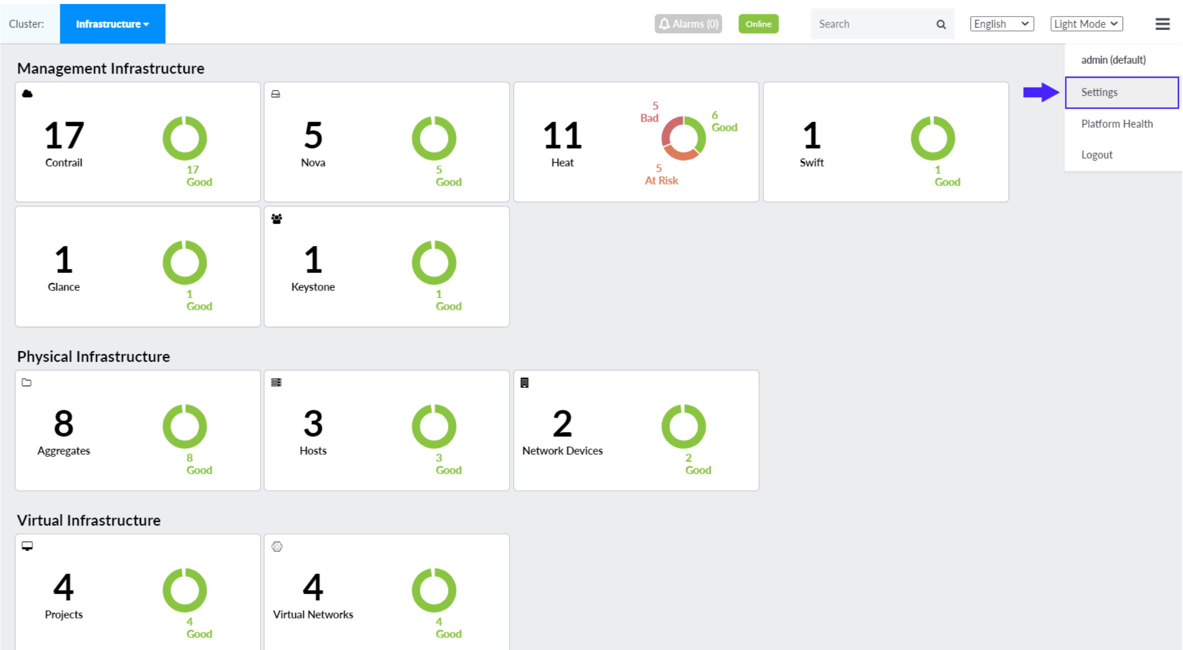
Consider the following example. An administrator creates a price tier by clicking **+ Add Tier** as shown in Figure 196 on page 288. The first tier or range starts with 0 GB. Every subsequent tier starts with the first GB following the previous tier and ends at the user-specified byte count. The final tier rate applies to all data transfer amounts that exceed the second-to-last tier. Once completed, you click **Save and Apply** to apply the new rates.

Follow these steps to configure SNAT logical router data transfer costs.

1. Click **Settings** as shown in Figure 195 on page 287.

   The Appformix Settings page is displayed.

**Figure 195: Click Settings to view Appformix Settings page**



2. Click **Chargeback** on the Appformix Settings pane.

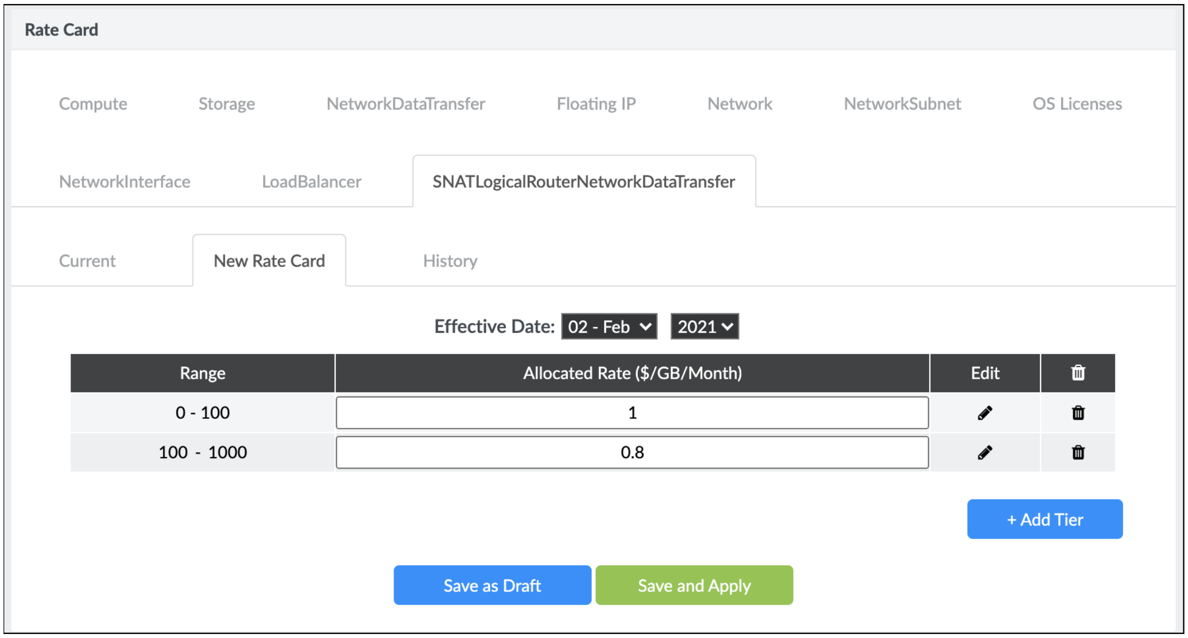
   The Rate Card page is displayed.

3. Click the **SNAT Logical Router Network Data Transfer** tab and then the **New Rate Card** tab.

4. From the New Rate Card view, click **+ Add Tier** as shown in Figure 196 on page 288.

   You can add more than one range by clicking **+ Add Tier**, and allocate a price range for each range that you add.

Figure 196: View SNAT Logical Router Network Data Transfer Rate Card



Effective Date: 02 - Feb 2021

| Range | Allocated Rate ($/GB/Month) | Edit | 🗑 |
|---|---|---|---|
| 0 - 100 | 1 | ✎ | 🗑 |
| 100 - 1000 | 0.8 | ✎ | 🗑 |

5. After you have added the required range(s), click **Save and Apply** to save and apply the new rate card.

   The new rate card is now applied.

## Monitoring Cost of Service Instances

Starting with Contrail Insights Release 3.3.7, you can configure and monitor costs of service instances.

Follow these steps to configure and to monitor costs of service instances:

1. Click the hamburger button and click **Settings**.

   The **Appformix Settings** page is displayed.

2. Click **Chargeback** on the **Appformix Settings** page.

   The Rate Card page is displayed.

3. Click the **ServiceInstance** tab and then the **New Rate Card** tab.

4. Enter the following information:

   a. From the **Effective Date** drop-down lists, select *month* from the first drop down, and select *year* from the next drop down.

      The effective date determines when the new rate card will come in to effect.

  b. Enter the **active rate charge** in the **Active Rate($/Hour)** column.

  c. Enter **no charge rate** in the **Nocharge Rate ($/Hour)** column.

5. Do any one of the following:

   - Click **Save as Draft** to save the new rate card as draft.

   - Click **Save and Apply** to save and immediately apply the rate card.

**Release History Table**

| Release | Description |
|---|---|
| 3.3.5 | Starting with Contrail Insights Release 3.3.5, you can configure SNAT logical routers network data transfer costs. |

## RELATED DOCUMENTATION

*Alarms*

*Capacity Planning*

*Charts*

*Health Monitor*

*Heat Map*

*Metrics Collected by Contrail Insights*

*Notifications*

*Extensibility Using Plug-Ins*

*Reports*

*Service Monitoring from the UI*

6

CHAPTER

# Contrail Insights APIs

# Using Contrail Insights APIs

IN THIS SECTION

## Contrail Insights APIs

Contrail Insights exposes a set of APIs for users to perform operations on the Contrail Insights Platform as needed. These APIs are published by Contrail Insights on a Swagger UI page. Users can access this UI through the Contrail Insights Dashboard to view and access Contrail Insights APIs. From the top right of the Contrail Insights Dashboard, select **Settings > API Documentation** to view the links to the Swagger UI for Contrail Insights APIs.

## Prerequisites for API Usage

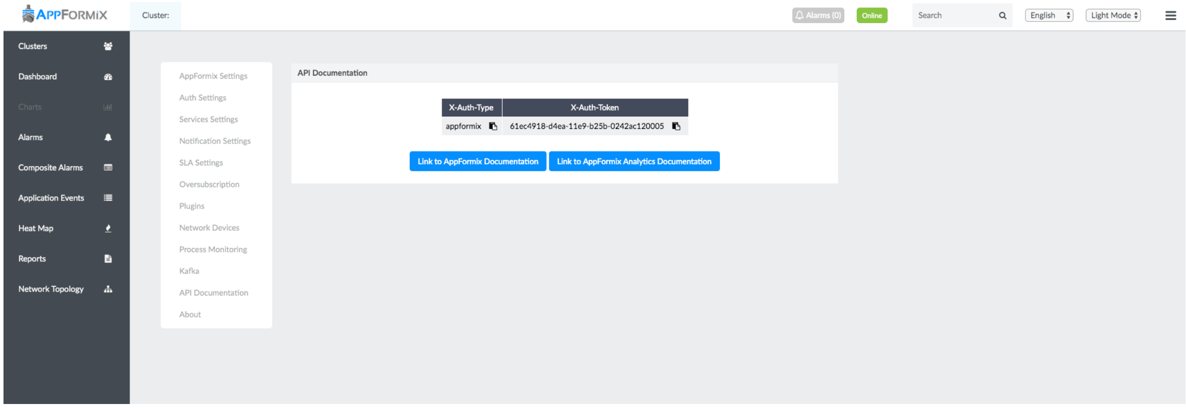
Contrail Insights APIs require authentication by means of the headers: `X-Auth-Type` and `X-Auth-Token`. These two values are available on the Contrail Insights Dashboard page. See Figure 197 on page 292.

To configure authentication for Contrail Insights APIs:

1. Open the Contrail Insights Dashboard in a Web browser. For example:

```
http://<contrail-insights-platform-hostname>:<port>/appformix/#/settings/api_docs
```

**Figure 197: Required Authentication Headers for contrail-insights APIs: X-Auth-Type and X-Auth-Token**



| X-Auth-Type | X-Auth-Token |
|---|---|
| appformix | 61ec4918-d4ea-11e9-b25b-0242ac120005 |

2. Select **Link to AppFormix Documentation** from the Dashboard page to view the main Contrail Insights Platform APIs.

   Select **Link to AppFormix Analytics Documentation** to view the Analytics APIs.

## Example of using Contrail Insights APIs

The following steps demonstrate POST and GET calls on the `/aggregates`Contrail Insights API by means of Swagger. Similar actions can be done on other APIs.

### POST /aggregates

1. This REST call creates a new aggregate on Contrail Insights. Click the **POST /aggregates** section in the Swagger UI to view the required fields for this POST call.

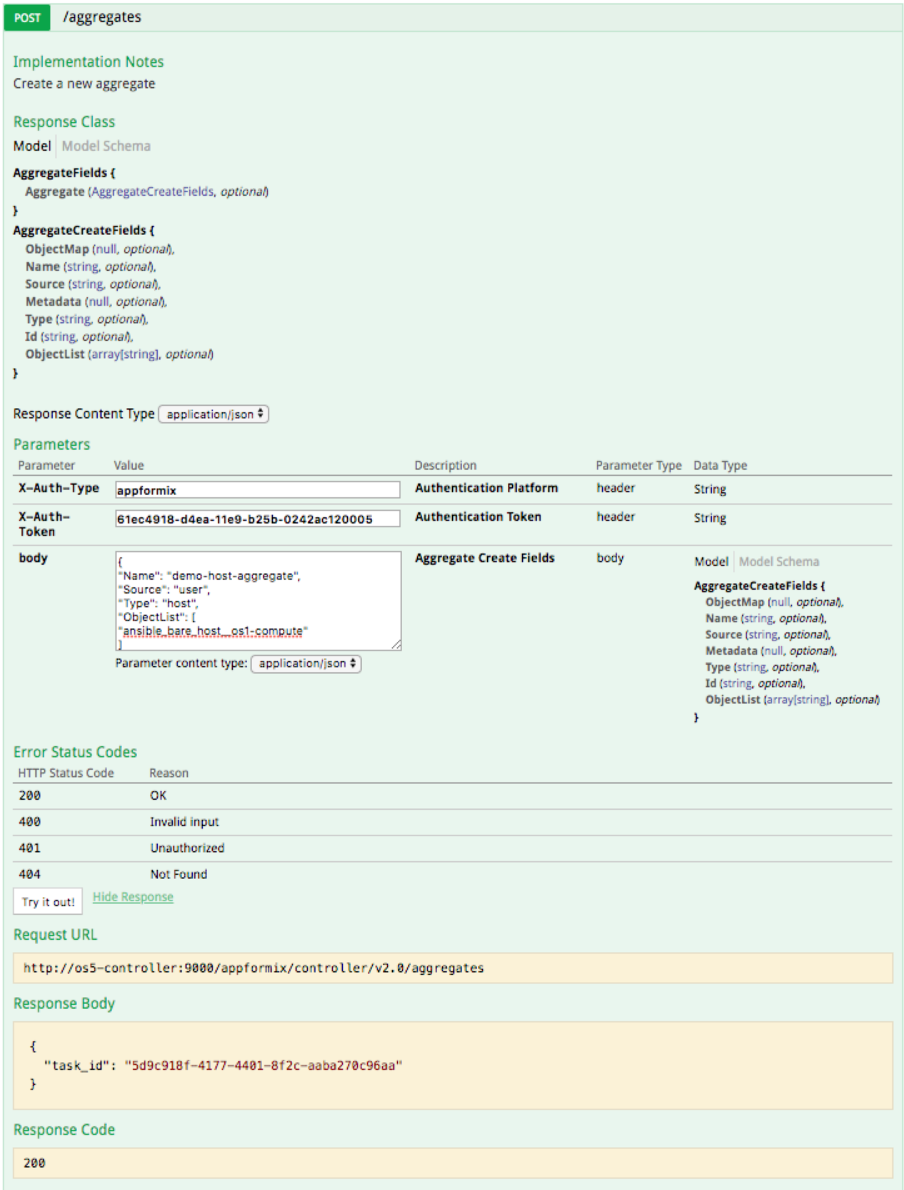
2. Use the following example headers and body required to create a new aggregate.

```
# Headers
X-Auth-Type: appformix
X-Auth-Token: <appformix_token>
# Body
{
    "Name": "demo-host-aggregate",
    "Source": "user",
    "Type": "host",
    "ObjectList": [
        "ansible_bare_host__os1-compute"
    ]
}
```

3. Enter the above information for **POST /aggregates API**, then click **Try it out!** This action creates a new aggregate named **demo-host-aggregate** on Contrail Insights. Refer to Figure 198 on page 294.

> **NOTE**: To populate the text area with all the relevant fields, click **Model Schema** and then on the snippet below it.

### Figure 198: Create New Aggregate with POST /aggregates API Call



POST /aggregates

Implementation Notes

Create a new aggregate

Response Class

Model | Model Schema

```
AggregateFields {
  Aggregate (AggregateCreateFields, optional)
}
AggregateCreateFields {
  ObjectMap (null, optional),
  Name (string, optional),
  Source (string, optional),
  Metadata (null, optional),
  Type (string, optional),
  Id (string, optional),
  ObjectList (array[string], optional)
}
```

Response Content Type application/json

Parameters

| Parameter | Value | Description | Parameter Type | Data Type |
|---|---|---|---|---|
| X-Auth-Type | appformix | Authentication Platform | header | String |
| X-Auth-Token | 61ec4918-d4ea-11e9-b25b-0242ac120005 | Authentication Token | header | String |
| body | {<br>"Name": "demo-host-aggregate",<br>"Source": "user",<br>"Type": "host",<br>"ObjectList": [<br>"ansible_bare_host__os1-compute"<br>]<br>Parameter content type: application/json | Aggregate Create Fields | body | Model \| Model Schema<br>AggregateCreateFields {<br>ObjectMap (null, optional),<br>Name (string, optional),<br>Source (string, optional),<br>Metadata (null, optional),<br>Type (string, optional),<br>Id (string, optional),<br>ObjectList (array[string], optional)<br>} |

Error Status Codes

| HTTP Status Code | Reason |
|---|---|
| 200 | OK |
| 400 | Invalid input |
| 401 | Unauthorized |
| 404 | Not Found |

Try it out! Hide Response

Request URL

```
http://os5-controller:9000/appformix/controller/v2.0/aggregates
```

Response Body

```
{
  "task_id": "5d9c918f-4177-4401-8f2c-aaba270c96aa"
}
```

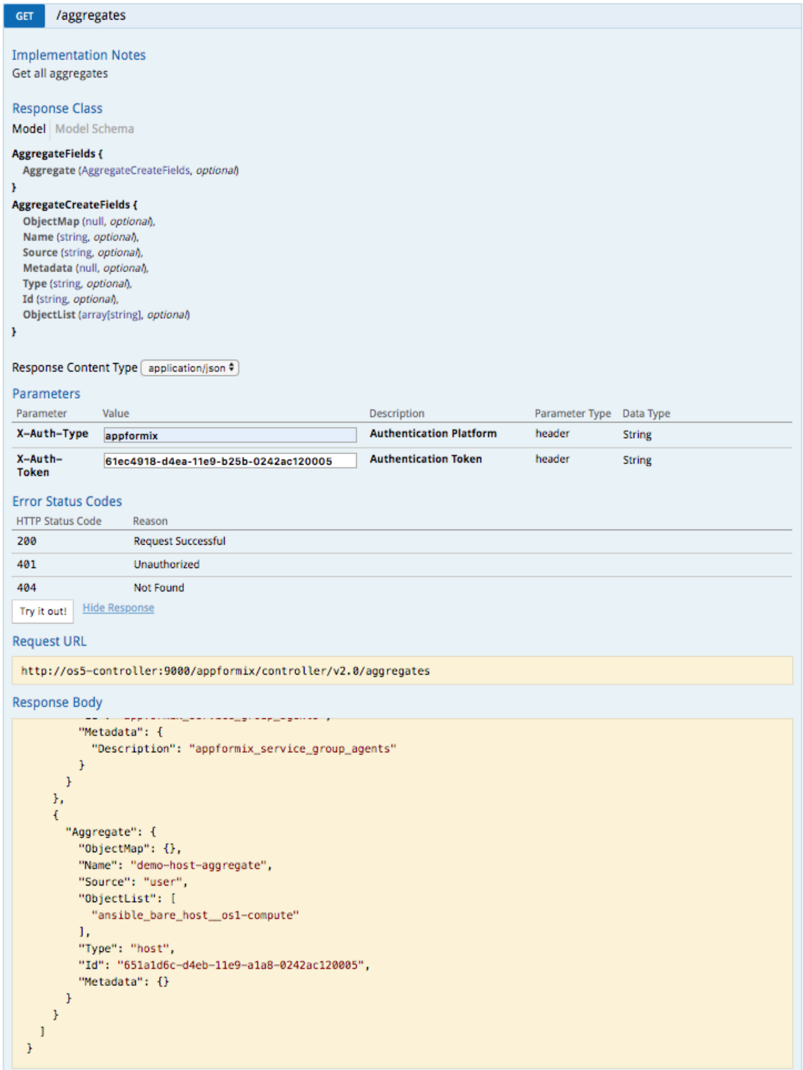
Response Code

```
200
```

## GET /aggregates

This REST call lists all of the aggregates present on the Contrail Insights Platform. To verify if the new aggregate from the above POST call is created successfully, provide headers for **GET /aggregates** API in the Swagger UI, then click **Try it out!** The output displays the new aggregate details. See Figure 199 on page 295.

Figure 199: Verify New Aggregate with GET /aggregates API Call



Implementation Notes

Get all aggregates

Response Class

```
AggregateFields {
  Aggregate (AggregateCreateFields, optional)
}
AggregateCreateFields {
  ObjectMap (null, optional),
  Name (string, optional),
  Source (string, optional),
  Metadata (null, optional),
  Type (string, optional),
  Id (string, optional),
  ObjectList (array[string], optional)
}
```

Response Content Type: application/json

Parameters

| Parameter | Value | Description | Parameter Type | Data Type |
|---|---|---|---|---|
| X-Auth-Type | appformix | Authentication Platform | header | String |
| X-Auth-Token | 61ec4918-d4ea-11e9-b25b-0242ac120005 | Authentication Token | header | String |

Error Status Codes

| HTTP Status Code | Reason |
|---|---|
| 200 | Request Successful |
| 401 | Unauthorized |
| 404 | Not Found |

Request URL

```
http://os5-controller:9000/appformix/controller/v2.0/aggregates
```

Response Body

```
          "Metadata": {
            "Description": "appformix_service_group_agents"
          }
        }
      },
      {
        "Aggregate": {
          "ObjectMap": {},
          "Name": "demo-host-aggregate",
          "Source": "user",
          "ObjectList": [
            "ansible_bare_host__os1-compute"
          ],
          "Type": "host",
          "Id": "651a1d6c-d4eb-11e9-a1a8-0242ac120005",
          "Metadata": {}
        }
      }
    ]
  }
```