Monitoring Tools

Contents

[Overview 2](#_Toc429400494)

[Use Cases 2](#_Toc429400495)

[Sensu 3](#_Toc429400496)

[Sensu client plugins 3](#_Toc429400497)

[Sensu Client 4](#_Toc429400498)

[Sensu Server 4](#_Toc429400499)

[Uchiwa 5](#_Toc429400500)

[Flapjack 5](#_Toc429400501)

[Graphite 6](#_Toc429400502)

[Grafana 6](#_Toc429400503)

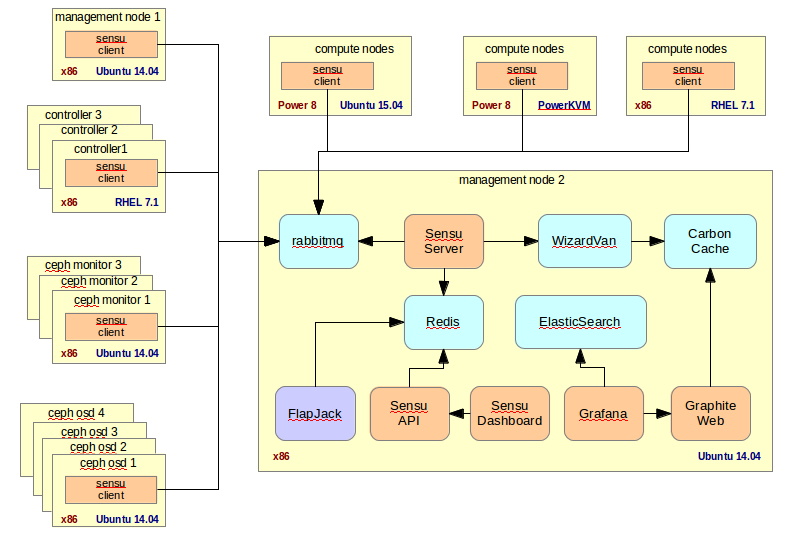
[Packages 7](#_Toc429400504)

[FAQs 8](#_Toc429400505)

[References 8](#_Toc429400506)

# Overview

This document describes various monitoring tools used for monitoring alerts in the datacenter. Installation and configuration of each comment can be found in the reference section. The figure below describe the components of monitoring tool framework, their interactions and deployment model. The sensu client running on client system periodically executes sensu client plugins and alerts are notified to sensu server. All the sensu alerts in the datacenter and all the sensu client registered can be viewed on sensu dashboard uchiwa. By adding flapjack handler in the sensu server, the alerts can be redirected to flapjack component. Flapjack can further be configured to send the alert to other systems like e-mail, SMS, AWS SNS service, Pageduty etc. Using WizardVan sensu handler the alert along with the time of occurrence will be sent to the carbon-cache component of Graphite. Graphite maintains PostgreSQL database to store all the alert and time series data .Through the graphite web, graphs of the alert from various client can be generated. Grafana is UI component to connect to graphite api and visualize alert data. Compared to Graphite web, Grafana provides good look and feel and other features to visualize the data.



# Use Cases

With the use of sensu community plugins, Nagios plugins, monitoring tools can be used to monitor databases, operating systems, applications, network equipment, protocols and more. Listing out some of the typical use case provide by the monitoring tool framework

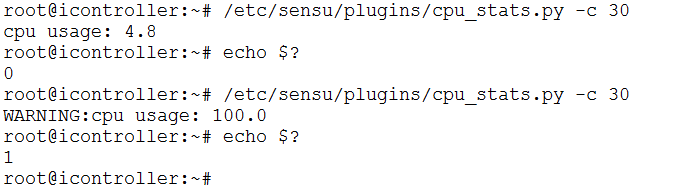
* Monitor CPU usage, memory usage, disk usage, TCP connection usage, network bandwidth consumption of all the systems in the datacenter
* Monitor application server availability and response time across all the nodes
* Custom sensu plugins can be developed in shell script, Ruby, Python, command line interface to monitor web page changes, stock price alerts, etc...

# Sensu

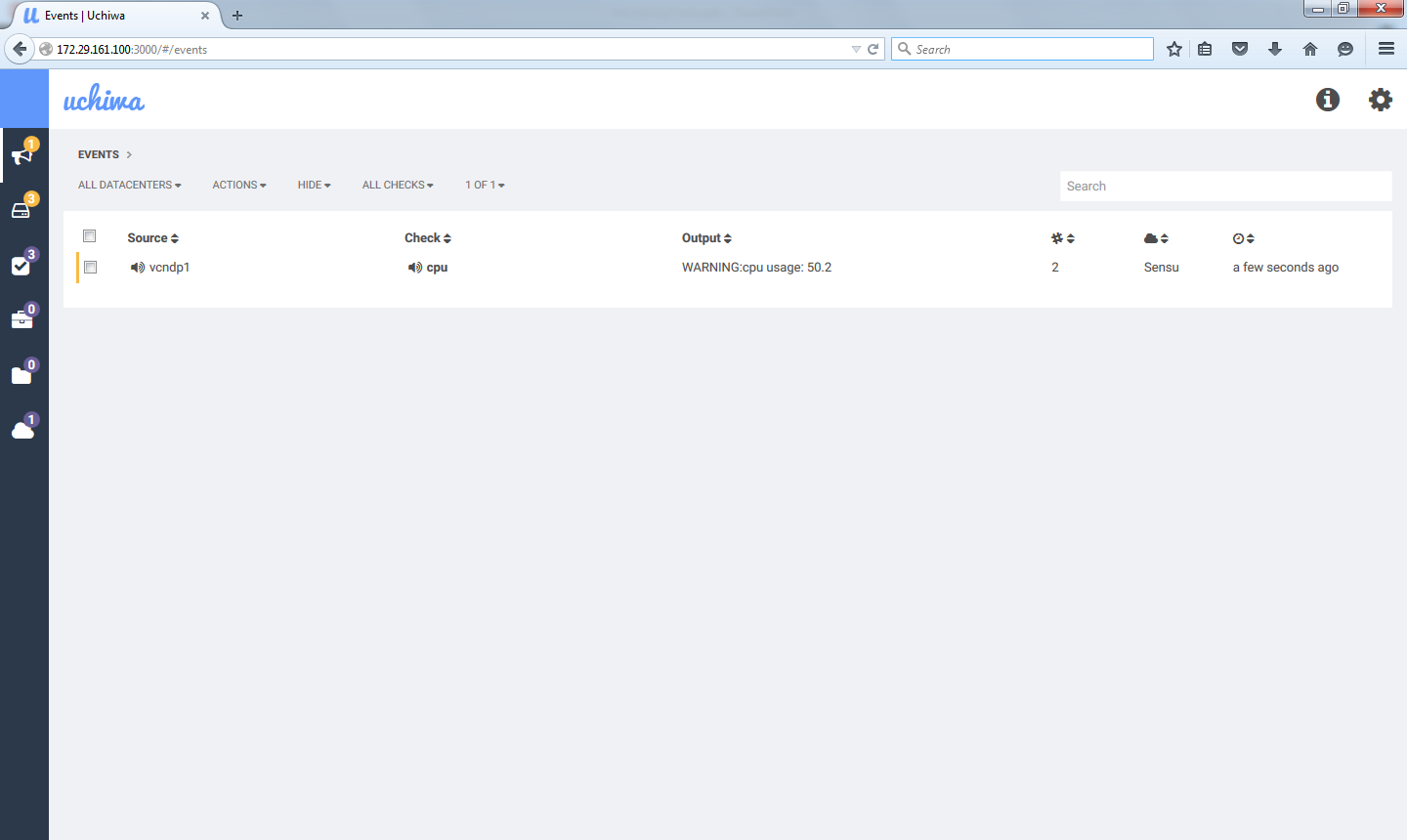
Sensu is the core component of the monitoring tool framework. Sensu includes the following components

## Sensu client plugins

Sensu client plugins can be command line interface, python, ruby, shell scripts. Publically available Nagios plugins and sensu community plugin can be configured in the monitoring tool framework. Sensu client plugins are executed periodically and whenever the plugin output is 1, alert is send to the sensu server and the same can be viewed in the uchiwa dashboard. The following screenshots demonstrate the use CPU usage sensu plugin cpus\_stats.py. This python emits the current CPU usage message and return a value of 1 if CPU is greater than the specified value of 30.



CPU alert seen on uchiwa dashboard.

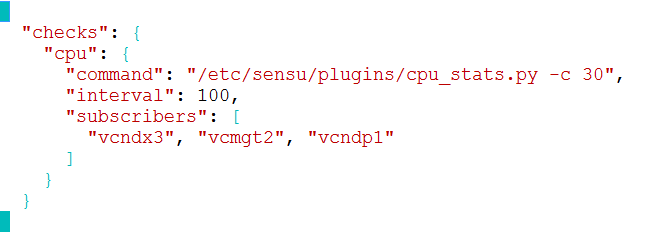


## Sensu Client

Sensu client is a light weight component installed on all compute, storage, and controller nodes to gather alerts and notify the sensu server. Sensu client communicates with sensu server through Rabbitmq configured in /etc/sensu/config.json file. Running as a daemon, client registers itself to sensu sever and pulls in all the all the checks that need to be executed and reports alert back to the server.

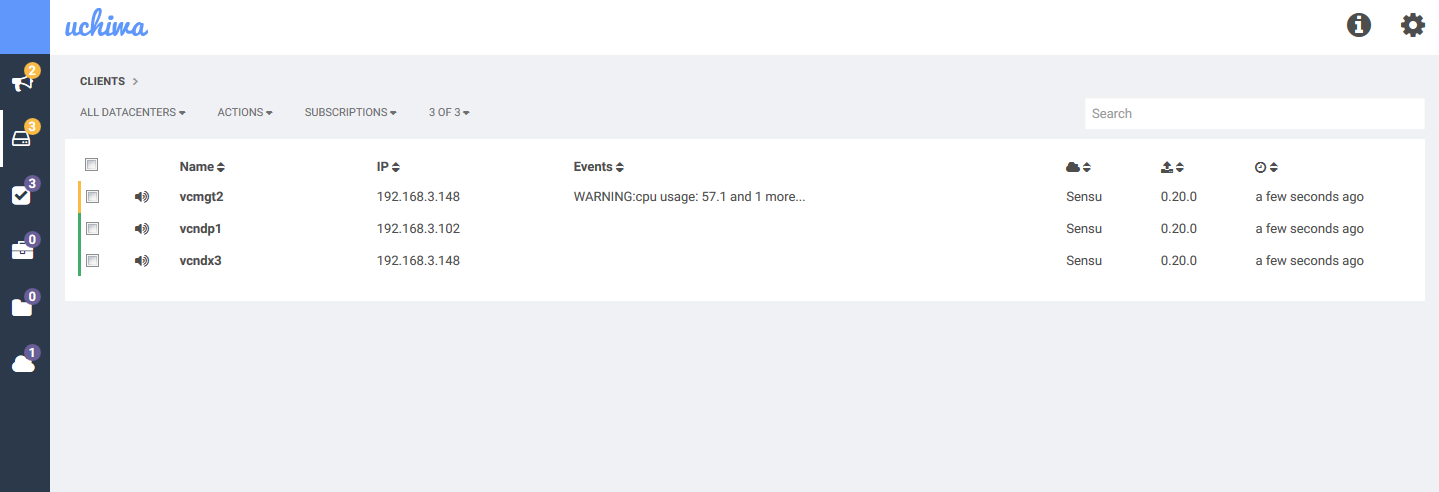
## Sensu Server

Sensu server is the core component of Monitoring framework. Sensu API provides REST based web services to pull alerts, register client and other info to external subsystems. The following snapshot shows the sample CPU check configured on sensu server at location /etc/sensu/conf.d/check\_cpu.json. This check specifies that python script cpu\_stats.py need to be executed on 3 nodes at interval of 100 second periodically.



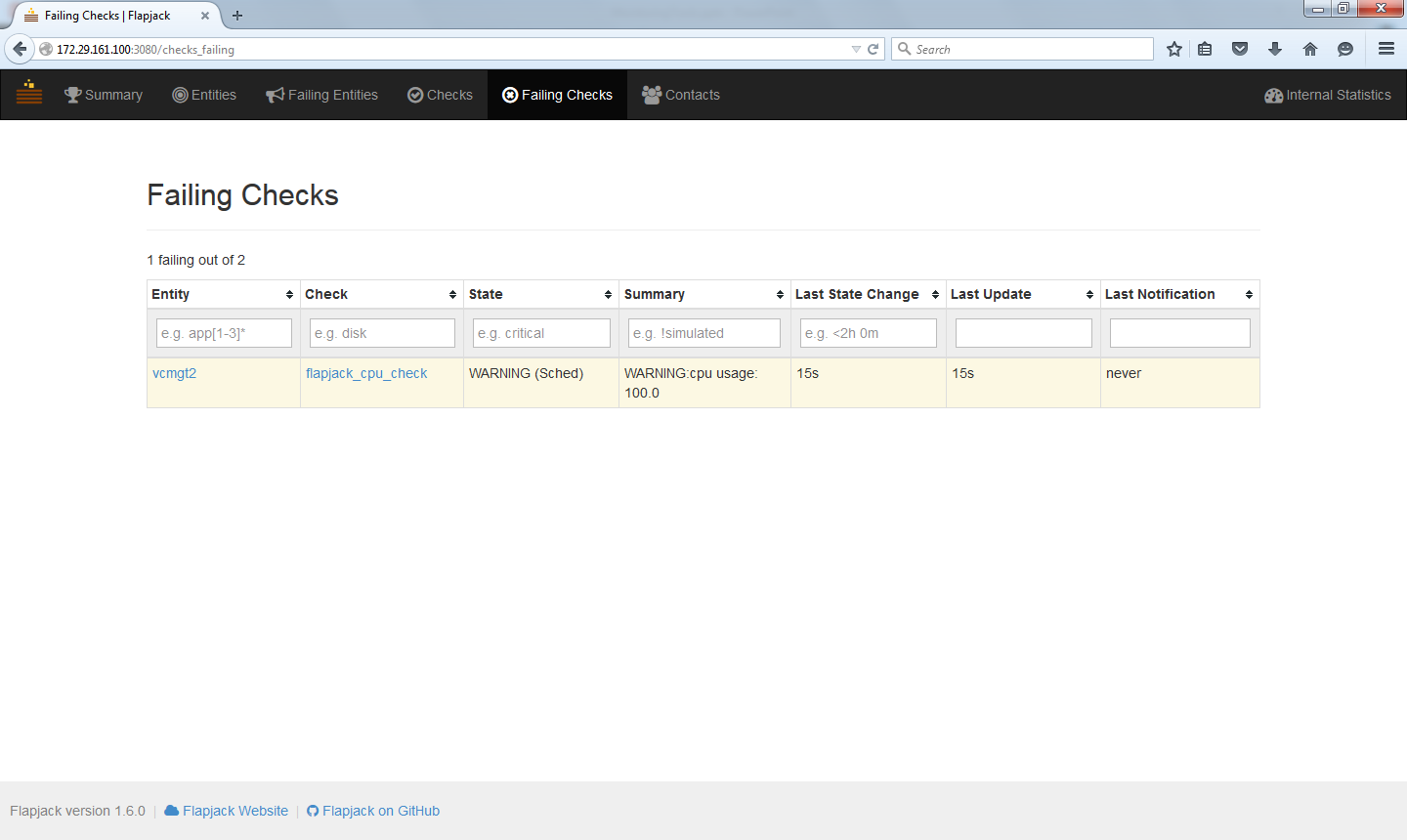
## Uchiwa

Uchiwa is the default dashboard for viewing the alerts and the list of clients subscribed to the sensu server. The following screen shot shows the list of registered sensu client, client version installed and events.



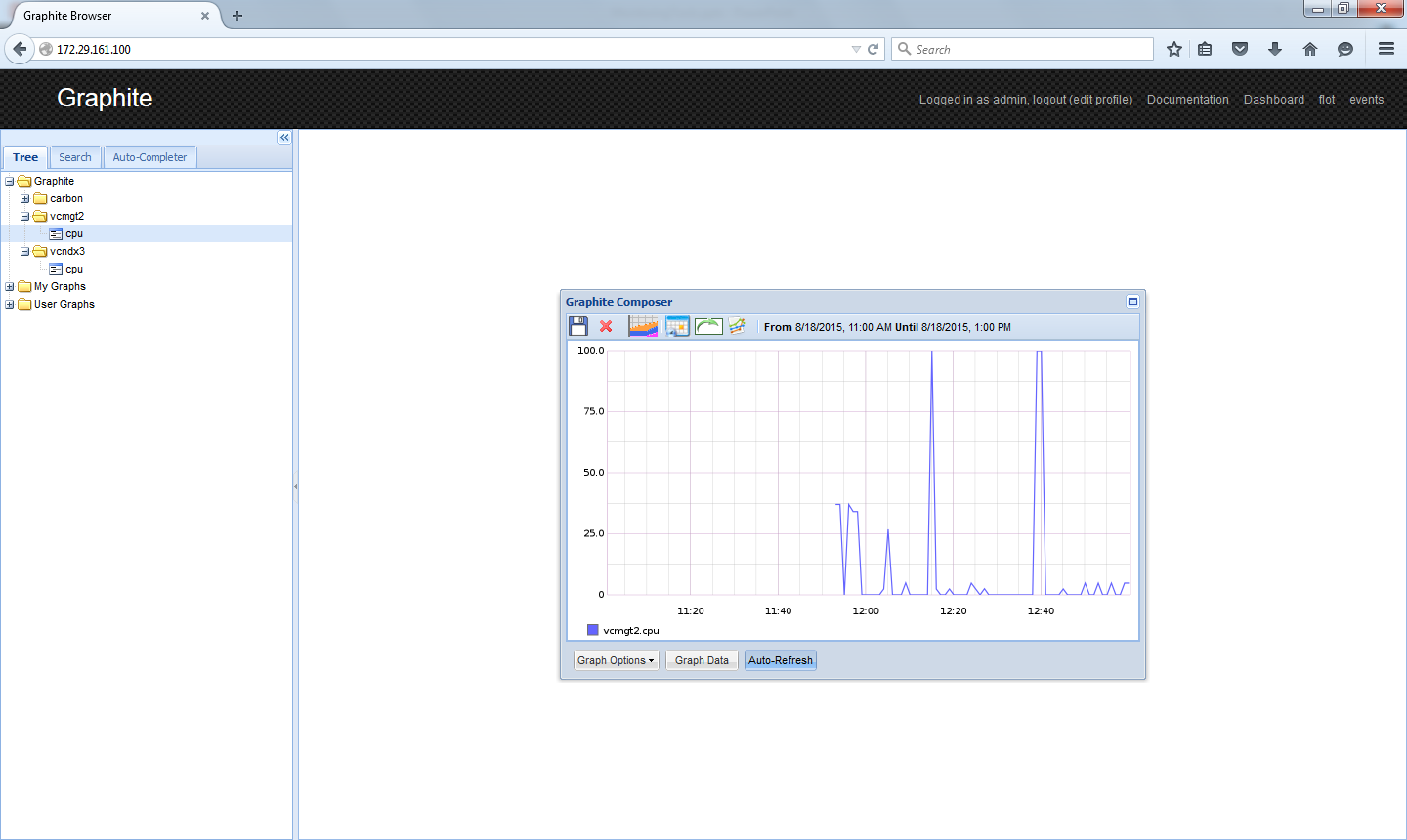
# Flapjack

Flapjack is alert routing framework and can be configured with sensu and Nagios as their backbends. The alert received form the backed system can be routed to various external subsystems including web console, email, SMS alert through Nexmo, Twilio, Pagerduty, AWS SNS, Slack, Jabber chat box, based on the configuration file /etc/flapjack/flapjack\_config.yaml. Flapjack web console screen on receiving an alert from sensu is displayed below.



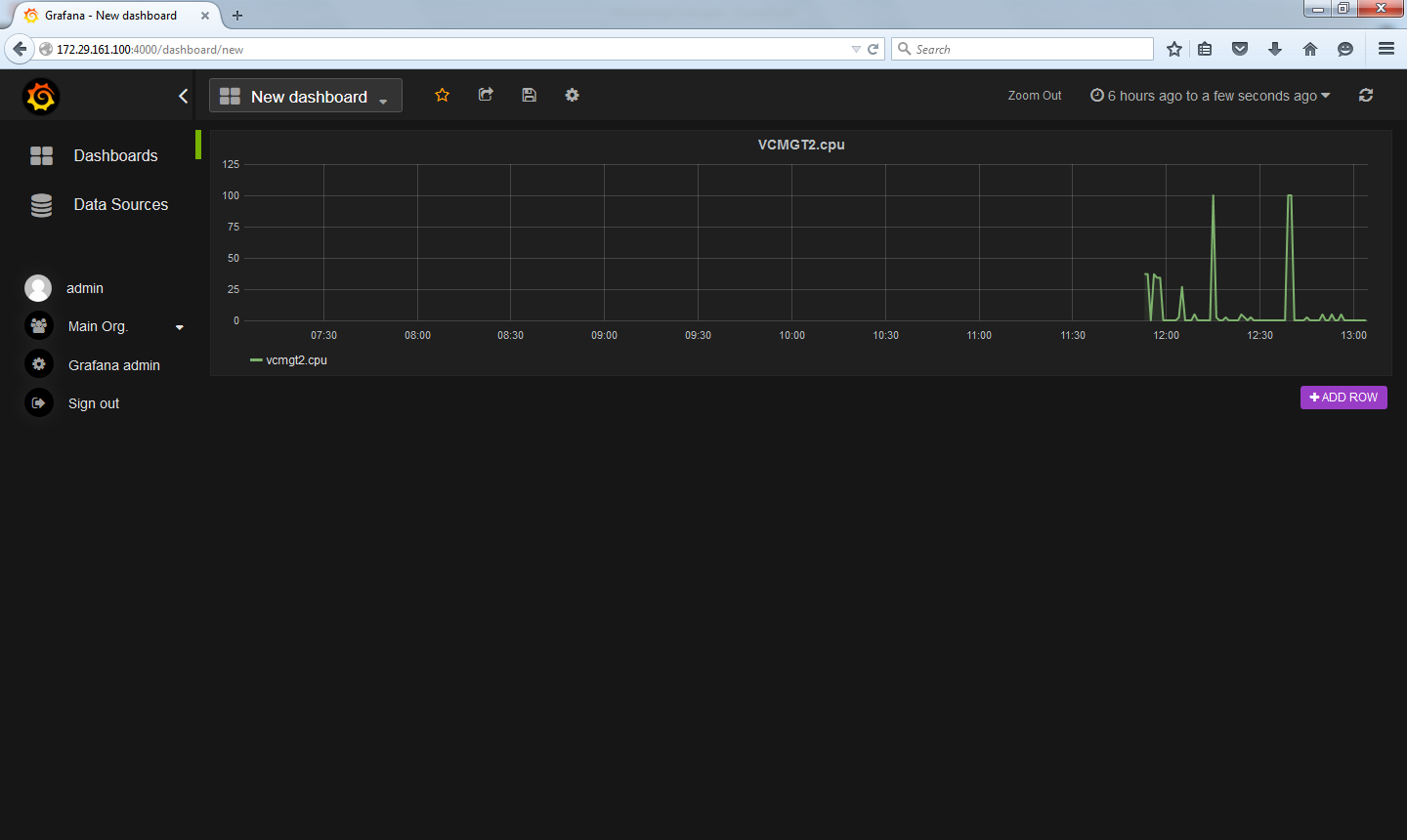
# Graphite

One of the drawback of sensu framework is that it won’t keep the history of the alerts received. By adding sensu based graphite plugin running on the sensu client system, time series data can be send to the graphite system. Graphite uses PostgreSQL to store time series numeric data, carbon daemon running on port 2003 to collect the time series data. It includes an UI component to render graphs and view stats. As Grafana provides rich user interface than Graphite UI, graphite is used to store time series data in Monitoring tools framework. The screen shot of CPU usage graph on Graphite is shown below.



# Grafana

Grafana is feature rich metrics dashboard and graph editor for Graphite, Influx DB & OpenTSDB. Grafana supports features like click and drag, zoom in and zoom out, generates graphs based on bars, lines, point. As Grafana provides good look and feel, monitoring tool framework end users will be using Grafana instead of Graphite UI for data visualization. Graphite can be configured as data source to Grafana by selecting the “Data source” button on the left side of Grafana web portal. Sample screen shot of Grafana showing the CPU usage pulled from Graphite data source is shown below.



# Packages

Monitoring tools installation includes server and client components. Server component includes following packages:

1. Sensu Server components

* Sensu includes Sensu Server, API
* Rabbit MQ
* Redis
* Uchiwa

1. Flapjack
2. Graphite

* Graphite web
* Graphite Carbon
* PostgreSQL
* Apache server

1. Grafana

Client components includes the following:

1. Sensu package which includes sensu client.

Packages for x86 system are available publicly through apt or yum repositories. For power ppc64, client packages need to be built from source by installing compilers and other tools.

# FAQs

**Is it possible to create bootable glance image with pre-installed sensu client, so that VM booting through glance image has sensu client installed and configured**

Yes, by installing sensu client and properly configuring the sensu config file /etc/sensu/config.json, it is possible to create pre-installed glance image with sensu client. This can also be achieved through CI/CD tools which will install and configure sensu client post VM creation.

# References

1. Sensu installation guide <https://sensuapp.org/docs/latest/installation-overview>
2. Graphite installation guide <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-graphite-on-an-ubuntu-14-04-server>
3. Wizardvan plug-in for sensu https://github.com/opower/sensu-metrics-relay
4. Sensu and flapjack integration <https://ianunruh.com/2014/05/monitor-everything-part-5.html>
5. Grafana installation <http://docs.grafana.org/installation/>

/etc/sensu/config.json

{

"rabbitmq": {

"host": "localhost",

"vhost": "/sensu",

"user": "sensu",

"password": "secret"

},

"redis": {

"host": "localhost"

},

"api": {

"port": 4567

}

}

/etc/sensu/conf.d/check\_cpu.json

{

"checks": {

"cpu": {

"command": "/etc/sensu/plugins/cpu\_stats.py -c 30",

"interval": 100,

"subscribers": [

"sensu-client"

]

}

}

}

/etc/sensu/conf.d/default\_handler.json

{

"handlers": {

"default": {

"type": "pipe",

"command": "cat"

}

}

}

/etc/sensu/conf.d/config\_relay.json

{

"relay": {

"graphite": {

"host": "127.0.0.1",

"port": 2003

}

}

}

/etc/sensu/uchiwa.json

"sensu": [

{

"name": "Sensu",

"host": "localhost",

"port": 4567,

"timeout": 5

}

],

"uchiwa": {

"host": "0.0.0.0",

"port": 3001,

"interval": 5

}

}

Update MAX\_QUEUE\_SIZE to 200 in /etc/sensu/extensions/handlers/relay.rb

Client side config:

"/etc/sensu/config.json"

{

"rabbitmq": {

"host": "10.143.142.201",

"vhost": "/sensu",

"user": "sensu",

"password": "secret"

}

}

"/etc/sensu/conf.d/client.json"

{

"client": {

"name": "sensu-client",

"address": "10.143.142.203",

"subscriptions": [

"sensu-client"

]

}

}