The evolution of monitoring system: the INFN-CNAF case study

The evolution of monitoring system: the INFN-CNAF case study

Stefano Bovina[1], **Diego Michelotto**[1]
[1] INFN CNAF, Viale Berti Pichat 6/2, 40126, Bologna, Italy
E-mail: `stefano.bovina@cnaf.infn.it, diego.michelotto@cnaf.infn.it`

Abstract. Over the past two years, the operations at CNAF, the ICT center of the Italian Institute for Nuclear Physics, have undergone significant changes. The adoption of configuration management tools, such as Puppet, and the constant increase of dynamic and cloud infrastructures have led us to investigate a new monitoring approach. The present work deals with the centralization of the monitoring service at CNAF through a scalable and highly configurable monitoring infrastructure.

The selection of tools has been made taking into account the following requirements given by users: (I) adaptability to dynamic infrastructures, (II) ease of configuration and maintenance, capability to provide more flexibility, (III) compatibility with existing monitoring system, (IV) re-usability and ease of access to information and data. In the paper, the CNAF monitoring infrastructure and its related components are hereafter described: Sensu as monitoring router, InfluxDB as time series database to store data gathered from sensors, Uchiwa as monitoring dashboard and Grafana as a tool to create dashboards and to visualize time series metrics.

1. Introduction

CNAF [1] is the national center of INFN (Italian Institute for Nuclear Physics) for the Research and Development in INFN Information and Communication Technologies. As the central INFN computing facility, CNAF is engaged with the management and development of the most important information and data transmission services that support INFN activities at a national level.

Since the creation of the distributed computing system at geographical scale known as "Grid", CNAF has been deeply involved in the development and management of Grid middleware and infrastructure in the international framework (Worldwide LHC Computing Grid) [2].

Since 2003, CNAF is also involved in the hosting and management of the Italian Tier-1 for the high-energy physics experiments carried out in the Large Hadron Collider in Geneva, providing resources, services and support for all the activities related to data storage and distribution, data processing, Monte Carlo simulations, and data analysis. Moreover, CNAF represents a key computing facility for many other astro-particle and neutrino-physics experiments and one of the most important centers for distributed computing in Italy.

The CNAF management activities are structured in seven functional units with a total amount of about 50 employees. In addition to these groups, the User Support team helps research groups and experiments to exploit the Data Center infrastructure and its resources (see Table 1 for details).

Table 1. CNAF resources

Resource	Amount
Core	~22000
Disk storage	~22PB
Tape storage	~48PB
Racks	$>$ 180
KHEPSPEC06	~250
Apparent power	5000 kVA

In order to provide reliable services, is necessary detect and prevent any type of issue on our systems. To ensure this behavior is necessary to have an efficient monitoring system.

The monitoring systems of large datacenters has to take into account both the great amount of heterogeneous systems (more than 1500 between servers and network devices) and the need for homogeneous management approaches. The constant increase of resources and services to be monitored, has imposed the review of the current monitoring infrastructure to make it more efficient, sustainable and scalable.

2. The monitoring issue

Monitoring systems currently used at CNAF are Nagios [3] and Lemon [4]. Lemon is a metric gatherer and viewing tool developed at CERN [5], but it is no longer maintained and does not support newer operating systems such as CentOS 7. Nagios is a monitoring tool adopted for many years at CNAF and currently widely used, but the poor scaling properties, the old-style user interface (UI), the old-style design not suited to manage a dynamic infrastructure, and the lack of integration with configuration management (CM) tools, make it not suitable to be used in a modern computing center.

With the increase of services and resources, the monitoring infrastructure has become even more crucial for the data center, while the work force and budget needed to manage the monitoring infrastructure have been reduced.

For the above reasons, and with the goal of making the monitoring infrastructure more efficient and sustainable, a complete redesign has been performed by adopting configuration management tools like Puppet [6]. This choice was made with the aim of rationalizing resources, reducing the time for the maintenance, and obtaining a highly reliable and scalable setup.

To achieve these results, a team (called Bebop) has been formed, and assigned the task of managing the monitoring infrastructure and facilitating migration from legacy systems.

The selection of the adopted tools has been made, taking into account the user requirements:

- Cloud oriented monitoring system;
- Horizontal scaling and highly available monitoring system;
- Service management via CM tools like Puppet;
- Capability to provide more flexibility in the configuration;
- Reuse of Nagios scripts;
- Reuse and easy access to information and data;
- Interaction with API;
- Modern UI and dashboard composer system;
- Separation of contexts among CNAF functional units.

After testing potential available monitoring systems, InfluxDB [7] has been selected as a datastore for metrics and Sensu [8] to manage the gathering of metrics and alerting systems. The choice of InfluxDB as a datastore for the metrics resides in the great elasticity of the data structure, the support of a large number of writing operations as well as the restrained disk space usage, a necessary feature to monitor a large number of systems at reduced cost.

On the other hand, Sensu has been selected for the following reasons, which allowed us to save a consistent amount of time:

- Reuse of Nagios and Lemon scripts;
- Management and configuration with Puppet;
- Composable and extensible;
- Highly scalable.

Thanks to the compatibility with Nagios probes, Sensu has been chosen as a natural substitute of Nagios.

3. State of the art of monitoring at CNAF

The actual implementation of the monitoring system (see Figure 1 for details), consists on a set of centralized components needed for its operation like RabbitMQ [9], Redis [10], InfluxDB, and HAProxy [11]. The centralization of these components is necessary to implement a highly reliable setup, and reduce operating and maintenance costs, providing the user with a set of endpoints for RabbitMQ and InfluxDB, hiding the complexity of the underlying infrastructure.

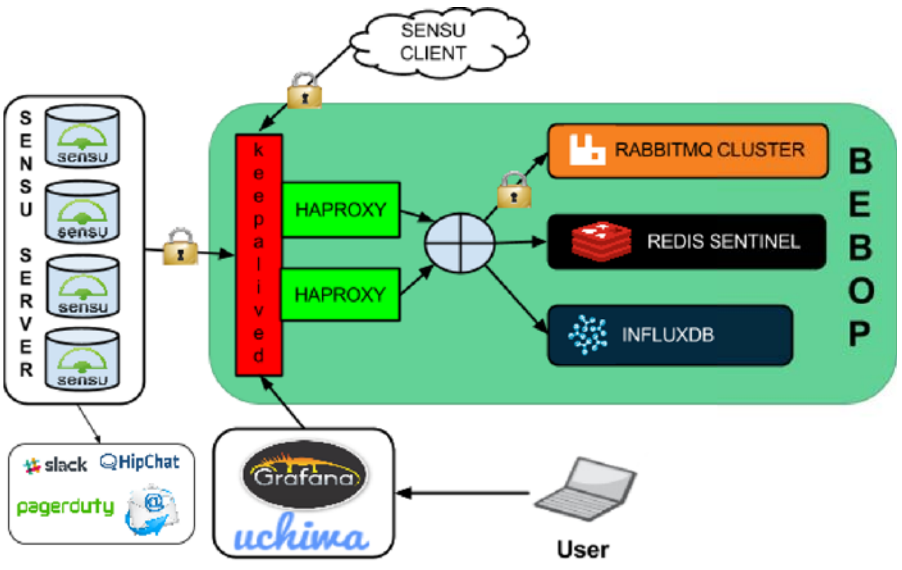
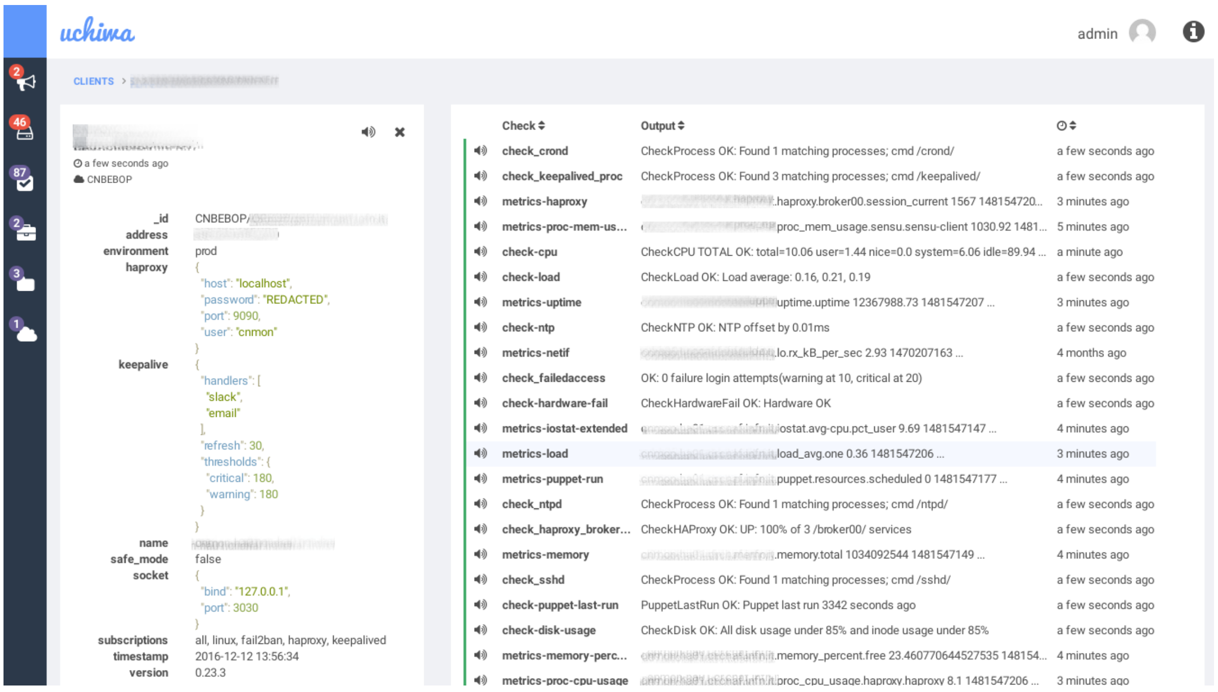


Figure 1. Monitoring system architecture

The components used to deploy the complete monitoring infrastructure are here briefly described:

- **RabbitMQ** is a message broker software that implements the AMQP protocol (Advanced Message Queuing Protocol), used by Sensu for communication between clients and servers. In particular, this software allows the server to schedule the execution of checks on client and enables the client to communicate the results to the server. In order to achieve a highly reliable configuration, we have chosen a cluster setup with mirrored queues.
- **Redis** is an in-memory database that also provides persistence of data on disk, used by Sensu for saving data about state of clients and their checks. To achieve an highly reliable configuration, we have chosen a setup based on Redis Sentinel technology.
- **Sensu** is an infrastructure, application monitoring and telemetry solution that allows reusing monitoring checks and plugins from legacy monitoring tools such as Nagios. Sensu was designed as a replacement for an aging Nagios installation and was specifically conceived to solve monitoring challenges introduced by modern infrastructure platforms with a mix of static, dynamic, and ephemeral infrastructure at scale (i.e. public, private, and hybrid clouds). We are currently using Sensu for:
  - Checking systems status;
  - Sending alerts and notifications;
  - Gathering metrics from systems that have to be sent to InfluxDB.
- **Uchiwa** [12] is an open source dashboard for Sensu. It provides a modern UI (see Figure 2) to easily visualize the state of our systems and a simple way to interact with Sensu server through RESTful API (provided by Sensu API service) for common tasks such as silencing hosts and checks related notification.



```
[redacted]
a few seconds ago
CNBEBOP

         _id   CNBEBOP/[redacted]
     address   [redacted]
 environment   prod
     haproxy   {
                 "host": "localhost",
                 "password": "REDACTED",
                 "port": 9090,
                 "user": "cnmon"
               }
   keepalive   {
                 "handlers": [
                   "slack",
                   "email"
                 ],
                 "refresh": 30,
                 "thresholds": {
                   "critical": 180,
                   "warning": 180
                 }
               }
        name   [redacted]
   safe_mode   false
      socket   {
                 "bind": "127.0.0.1",
                 "port": 3030
               }
subscriptions  all, linux, fail2ban, haproxy, keepalived
   timestamp   2016-12-12 13:56:34
     version   0.23.3
```

Check	Output	⏲
check_crond	CheckProcess OK: Found 1 matching processes; cmd /crond/	a few seconds ago
check_keepalived_proc	CheckProcess OK: Found 3 matching processes; cmd /keepalived/	a few seconds ago
metrics-haproxy	[redacted].haproxy.broker00.session_current 1567 148154720...	3 minutes ago
metrics-proc-mem-us...	[redacted]proc_mem_usage.sensu.sensu-client 1030.92 1481...	5 minutes ago
check-cpu	CheckCPU TOTAL OK: total=10.06 user=1.44 nice=0.0 system=6.06 idle=89.94 ...	a minute ago
check-load	CheckLoad OK: Load average: 0.16, 0.21, 0.19	a few seconds ago
metrics-uptime	[redacted]uptime.uptime 12367988.73 1481547207 ...	3 minutes ago
check-ntp	CheckNTP OK: NTP offset by 0.01ms	a few seconds ago
metrics-netif	[redacted].lo.rx_kB_per_sec 2.93 1470207163 ...	4 months ago
check_failedaccess	OK: 0 failure login attempts(warning at 10, critical at 20)	a few seconds ago
check-hardware-fail	CheckHardwareFail OK: Hardware OK	a few seconds ago
metrics-iostat-extended	[redacted]iostat.avg-cpu.pct_user 9.69 1481547147 ...	4 minutes ago
metrics-load	[redacted]load_avg.one 0.36 1481547206 ...	3 minutes ago
metrics-puppet-run	[redacted]puppet.resources.scheduled 0 1481547177 ...	4 minutes ago
check_ntpd	CheckProcess OK: Found 1 matching processes; cmd /ntpd/	a few seconds ago
check_haproxy_broker...	CheckHAProxy OK: UP: 100% of 3 /broker00/ services	a few seconds ago
metrics-memory	[redacted].memory.total 1034092544 1481547149 ...	4 minutes ago
check_sshd	CheckProcess OK: Found 1 matching processes; cmd /sshd/	a few seconds ago
check-puppet-last-run	PuppetLastRun OK: Puppet last run 3342 seconds ago	a few seconds ago
check-disk-usage	CheckDisk OK: All disk usage under 85% and inode usage under 85%	a few seconds ago
metrics-memory-perc...	[redacted].memory_percent.free 23.460770644527535 148154...	4 minutes ago
metrics-proc-cpu-usage	[redacted]proc_cpu_usage.haproxy.haproxy 8.1 1481547206 ...	3 minutes ago

Figure 2. Uchiwa dashboard showing checks detail of a host

- **InfluxDB** is an open source database optimized to handle time series data with no external dependencies. InfluxDB is targeted at use cases for DevOps, metrics, sensor data, and real-time analytics. Some of the key features that InfluxDB currently supports are:
  - SQL like query language;
  - HTTP(S) API;
  - Store billions of data points;
  - Database managed retention policies for data.

  To allow the extrapolation, viewing and analysis of data on a high timespan, appropriate retention policies (*one_ week*, *one_ month*, *six_ months*, *infinite*) and continuous queries (*populate_ one_ month*, *populate_ six_ months*, *populate_ infinite*) have been create to downsampled data, reducing its granularity. Each one of these retention policy retain data respectively for one week, one month, six month and infinitely. After these periods, older data are deleted (except in the infinite retention policy). New monitoring data retrieved by Sensu sensors are written by default into *one_ week* retention policy. Every 15 minutes, the *populate_ one_ month* continuous query downsample the data retrieved in the last 15 minutes (stored in the *one_ week* retention policy) and save downsampled data in the *one_ month* retention policy. Every 30 minutes, the *populate_ six_ months* continuous query downsample the data retrieved in the last 30 minutes (stored in the *one_ week* retention policy) and save downsampled data in the *six_ months* retention policy. Finally, every 1 hour, the *populate_ infinite* continuous query downsample the data retrieved in the last hour (stored in the *one_ week* retention policy) and save downsampled data in the *infinite* retention policy.
  In this way we have all the data retrieved during the last 7 days in the *one_ week* retention policy (with a granularity of one point every 5 minutes for each measurements), in the *one _ month* retention policy we have data of the last 30 days (with a granularity of one point every 15 minutes for each measurements), in the *six_ months* retention policy we have data of the last six months (with a granularity of one point every 30 minutes for each measurements), and finally, we have all the data (with a granularity of 1 point every 1 hour for each measurements) in the *infinite* retention policy.
- **Grafana** [13] is an open source dashboard composer and is most commonly used for visualizing time series data (see Figure 3). Some of the key features are:
  - Support for several datasources: InfluxDB, Graphite [14], Elasticsearch [15] etc.
  - Fully-interactive;
  - Editable graphs;
  - Create variables that are automatically filled with values from your database;
  - Variables in metric queries and panel titles can be reused;
  - Automatically repeat rows or panels for each selected variable value;
  - Multi tenancy and LDAP integration.

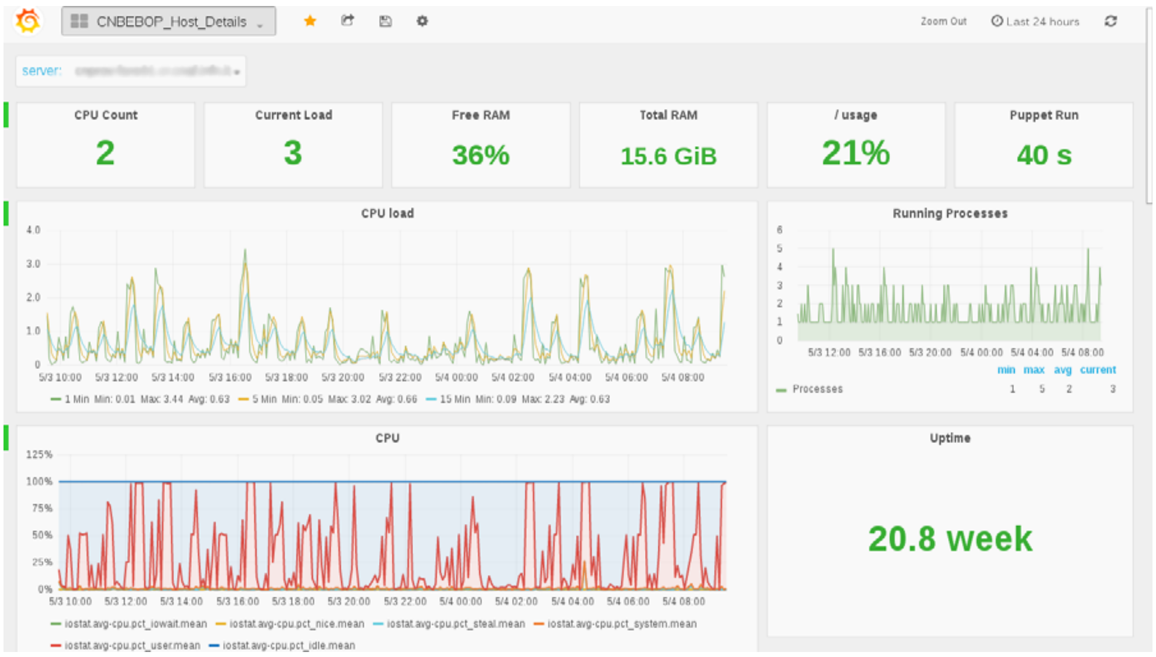

CNBEBOP_Host_Details
Zoom Out
Last 24 hours
server:
CPU Count
2
Current Load
3
Free RAM
36%
Total RAM
15.6 GiB
/ usage
21%
Puppet Run
40 s
CPU load
1 Min Min: 0.01 Max: 3.44 Avg: 0.63
5 Min Min: 0.05 Max: 3.02 Avg: 0.66
15 Min Min: 0.09 Max: 2.23 Avg: 0.63
Running Processes
min max avg current
Processes 1 5 2 3
CPU
iostat.avg-cpu.pct_iowait.mean
iostat.avg-cpu.pct_nice.mean
iostat.avg-cpu.pct_steal.mean
iostat.avg-cpu.pct_system.mean
iostat.avg-cpu.pct_user.mean
iostat.avg-cpu.pct_idle.mean
Uptime
20.8 week

Figure 3. Grafana dashboard showing metrics detail of a host

4. Components interaction and Puppet automation

A brief overview on how each component interacts with the others is hereafter summarized for Sensu checks workflow:

- A Check request is scheduled/published by Sensu-server or Sensu-client on RabbitMQ;
- Sensu-client executes a service check;
- Service checks emit status information and telemetry data as check results;
- Check results are published by Sensu-client on RabbitMQ;
- Sensu-server process check results, saving a copy of the latest result into Redis and creating a corresponding event;
- Sensu-server process events executing one or more event handlers;
- Sensu-server applies any event filters defined for an event handler;
- Sensu-server applies any event data mutators defined for an event handler (assuming the event was not filtered out);
- Sensu Server executes event handler (assuming the event was not filtered out).

A similar workflow can be used for Sensu metrics with the only difference that a particular handler is able to send data to InfluxDB server.

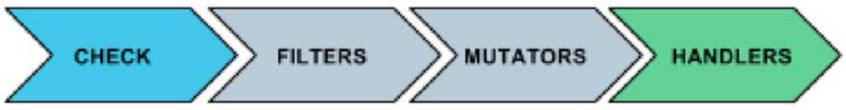


Figure 4. Schematic overview of Sensu workflow

After having collected all the data, with Uchiwa the operator can view hosts and checks status through the Sensu API and display metrics through graphs by the adoption of custom dashboards in Grafana.

Our infrastructure is continuously growing, and with the advent of the cloud paradigm, has also become a dynamic infrastructure, that can automatically provision and adjust itself as workload demands change. For this reasons is necessary being able to automate infrastructure monitoring, installation and configuration.

The Puppet community has developed a wide set of templates to configure Sensu and the use of such templates, has helped us save time and effort.

Today, reconfiguring Sensu clients and server as well as modifying the general configuration have become easier through our provisioning system.

5. Conclusion and future works

A new monitoring infrastructure is in production at CNAF. The monitoring service has been centralized using a scalable, modular and highly reliable monitoring solutions, completely managed with Puppet.

Currently, 1500 nodes are monitored and a great deal of use cases have been dealt with:

- Custom extension to optimize the writing of data in InfluxDB;
- Aggregation of check results and daily reports;
- Notification based on a percentage of failing host on a specific check;
- Notification routing and escalation.

The monitoring environment is only the first step of a more ambitious plan. The set of nodes is growing daily and to avoid possible scalability problems some performance optimization have already been applied. However, some fine-tuning may be required to achieve better performance and to improve data consumption.

Thanks to the complete compatibility of the adopted solutions with existing systems and the full integration of the configuration management systems, the team Bebop has been able to achieve the goal of deploying a new monitoring architecture at CNAF, reducing human errors as well as time required for the integration and maintenance of systems.

6. References

[1] INFN-CNAF website, last seen February 3rd 2017, https://www.cnaf.infn.it/
[2] WLCG website, last seen February 3rd 2017, http://wlcg.web.cern.ch/
[3] Nagios web site, last seen February 3rd 2017, https://www.nagios.org/
[4] Lemon web site, last seen February 3rd 2017, http://lemon.web.cern.ch/lemon/
[5] CERN web site, last seen February 3rd 2017, https://home.cern/
[6] Puppet web site, last seen February 3rd 2017, http://puppetlabs.com
[7] InfluxDB web site, last seen February 3rd 2017, https://influxdata.com/
[8] Sensu web site, last seen February 3rd 2017, https://sensuapp.org/
[9] RabbitMQ web site, last seen February 3rd 2017, https://www.rabbitmq.com/
[10] Redis web site, last seen February 3rd 2017, https://redis.io/
[11] HAProxy web site, last seen February 3rd 2017, http://www.haproxy.org/
[12] Uchiwa web site, last seen February 3rd 2017, https://uchiwa.io/
[13] Grafana web site, last seen February 3rd 2017, http://grafana.org/
[14] Graphite web site, last seen February 3rd 2017, https://graphiteapp.org/
[15] Elasticserch web site, last seen February 3rd 2017, https://www.elastic.co/