



Publication Year	2022
Acceptance in OA @INAF	2022-12-15T14:46:48Z
Title	Monitoring the performance of the SKA CICD infrastructure
Authors	DI CARLO, Matteo; Harding, Piers; Yilmaz, Ugur; Maia, Dalmiro; Ribeiro, Bruno; et al.
DOI	10.1117/12.2627025
Handle	http://hdl.handle.net/20.500.12386/32753
Series	PROCEEDINGS OF SPIE
Number	12189

PROCEEDINGS OF SPIE

[SPIDigitalLibrary.org/conference-proceedings-of-spie](https://spiedigitallibrary.org/conference-proceedings-of-spie)

Monitoring the performance of the SKA CICD infrastructure

M. Di Carlo, P. Harding, U. Yilmaz, D. Maia, B. Ribeiro, et al.

M. Di Carlo, P. Harding, U. Yilmaz, D. Maia, B. Ribeiro, D. Nunes, D. Regateiro, J. B. Morgado, M. Paulo, M. Santos, M. Dolci, "Monitoring the performance of the SKA CICD infrastructure," Proc. SPIE 12189, Software and Cyberinfrastructure for Astronomy VII, 121891O (29 August 2022); doi: 10.1117/12.2627025

SPIE.

Event: SPIE Astronomical Telescopes + Instrumentation, 2022, Montréal, Québec, Canada

Monitoring the performance of the SKA CICD Infrastructure

Di Carlo M.^a, Harding P.^b, Yilmaz U.^b, Maia D.^d, Ribeiro B.^c, Nunes D.^c, Regateiro D.^c, Morgado J.B.^d, Paulo M.^e, Santos M.^e, and Dolci M.^a

^aINAF Osservatorio Astronomico d'Abruzzo, Teramo, Italy

^bSKA Observatory, Macclesfield, UK

^cAtlar Innovation, Portugal

^dCICGE, Faculdade de Ciências da Universidade do Porto, Portugal

^fCritical Software, Portugal

ABSTRACT

The selected solution for monitoring the SKA CICD (continuous integration and continuous deployment) Infrastructure is Prometheus and Grafana. Starting from a study on the modifiability aspects of it, the Grafana project emerged as an important tool for displaying data in order to make specific reasoning and debugging of particular aspect of the infrastructure in place. Its plugin architecture easily allow to add new data sources like prometheus and the TANGO-controls framework related data sources has been added as well. The main concept of grafana is the dashboard, which enable to create real analysis. Thanos is also used for high availability, resilience and long term storage retention for monitoring data. In this paper the monitoring platform is presented which take advantage of different datasources and a variety of different panels (widget) for reasoning on archiving data, monitoring data, state of the system and general health of it.

Keywords: diagnostic, SKA, Prometheus, System team, monitoring, CICD

1. INTRODUCTION

The Square Kilometre Array (SKA) project has selected as development process the SAFe Agile (Scaled Agile framework) that is an incremental and iterative process with a specialized team (known as the Systems Team) devoted to supporting the Continuous Integration, Continuous Deployment (CICD),¹ test automation and quality. In the last 3 years the team has been busy with building an infrastructure for supporting the CICD together with a monitoring solution for it.

The SKA infrastructure consists of a standard footprint of VPN/SSH JumpHost gateway (called Terminus), Monitoring, Logging, Storage and K8s services to support the GitLab² runner architecture, and MVP testing facilities as shown in Fig.1 used to support DevOps and Integration testing facilities. In specific, the logging solution selected is Elasticsearch,³ the storage solution is Ceph⁴ and the (central) artefact repository (CAR) is Nexus.⁵ In relation to CAR, it is important to notice that only artefacts produced from the gitlab pipeline are allowed to be stored into the CAR and only if the pipeline has been triggered for a git tag. In all other cases, the gitlab artefact repository is used.

2. PROMETHEUS

The selected monitoring solution is Prometheus. It is a client server architecture where prometheus represents the client which read (scrape in its language) timestamped information from many servers (called targets or exporters). All data are then stored in a disk in TSDB format.⁶ Fig. 2 reports a diagram that illustrates the monitoring architecture in SKA. The components of the diagram where the most important are:

Further author information: (Send correspondence to Di Carlo M.)

Di Carlo M.: E-mail: matteo.dicarlo@inaf.it

Dolci M.: E-mail: mauro.dolci@inaf.it

Harding P.: E-mail: P.Harding@skatelescope.org

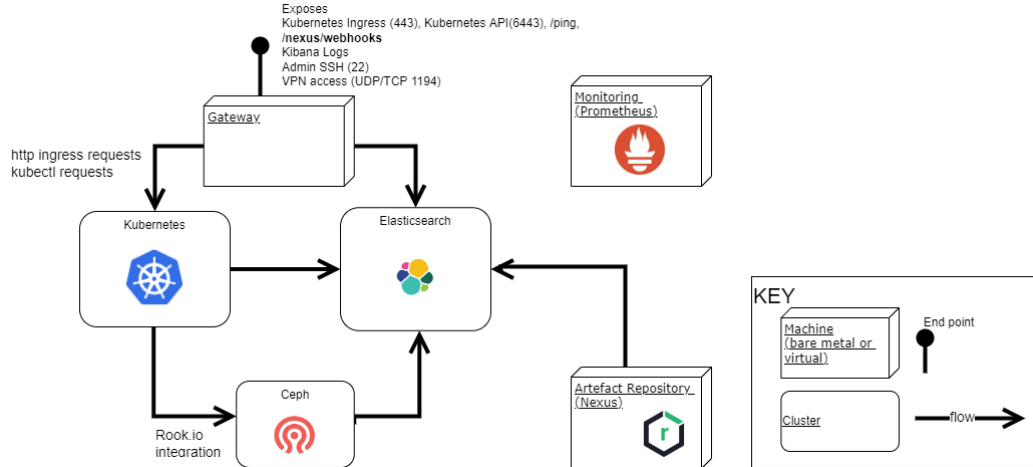


Figure 1. Simplified infrastructure

- the prometheus server which is composed by the retrieval component, the TSDB storage, and the http server for information retrieval,
- the jobs/exporters components from which prometheus takes information as time series and
- the data visualization and export which are external tools that integrates with prometheus with a specific query language called promql.

Each exporter provides time series uniquely identified by its metric name and some optional key-value pairs (called labels). It is important to note that the exporter must give all the information about its monitoring that is it must have all the information related to the “instrument”.

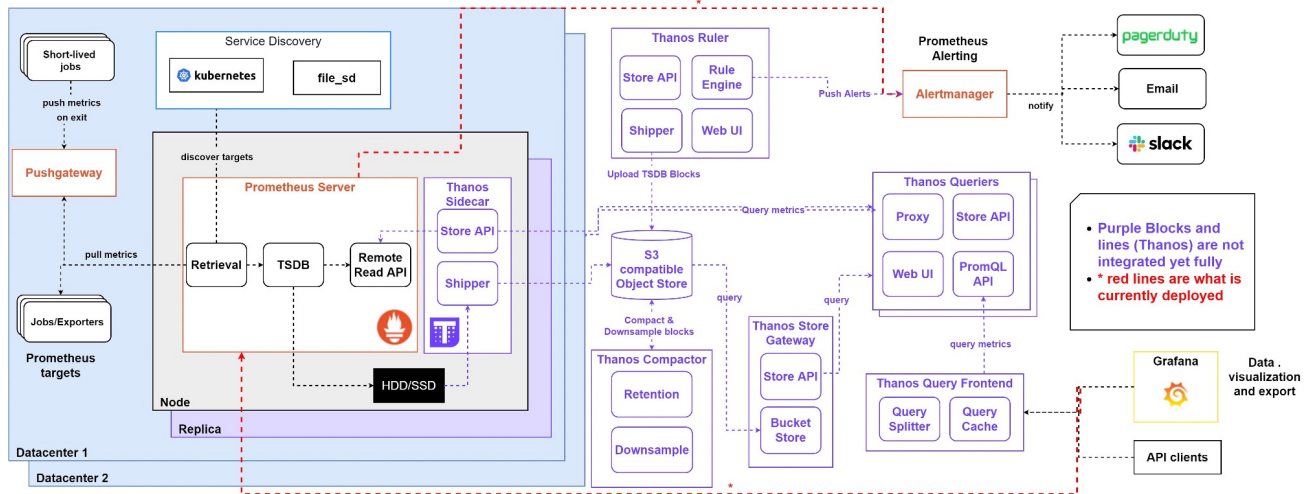


Figure 2. Detailed Monitoring Architecture

2.1 Exporters

One of the most important quality of the prometheus monitoring solution is the modifiability. It is in fact very easy to add new exporter since they are basically simple http server which provide a well defined metrics information.

In SKA, a set of commonly used exporters has been selected to be deployed (or enabled) in each node of the infrastructure. The most important are:

- Node exporter;⁷
- Container runtime metrics: enabled depending of the runtime (containerd, docker or podman);
- kubernetes⁸ exporters with the help of kube-state-metrics⁹ project;
- Elasticsearch metrics³
- Ceph metrics⁴

There are also custom exporters since writing them is very easy. Since the monitoring and controls project is built on top of the TANGO-controls framework,¹⁰ one of the custom exporters made is the TANGO-exporter¹¹ (Fig. 3 shows the result of a query to this exporter). Other custom exporters collect metrics the logging infrastructure to check if logging is performed correctly in each node.

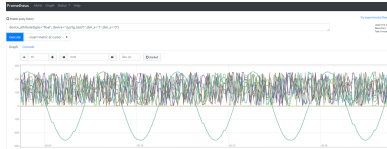


Figure 3. TANGO exporter graph view from Prometheus GUI

The configuration of the exporters in the Prometheus server happens with the help of a simple python script (prom-helper).¹² The script creates a process for 1) labelling each node in the openstack virtualization¹³ and 2) retrieving the labels and create the configuration file for the Prometheus server. The server run in a docker container and share a volume for the configuration files. The first step of the process happens every time a new node is added to the infrastructure while the second happen in a cron job of the node where the Prometheus container is running.

2.2 Alerts

An important part of the monitoring solution selection is the ability to inform the maintainers (basically the system team) of any possible cause of problems. For this purpose, it has been adopted many alerts coming from various community (for examples from the project kubernetes-mixin¹⁴). The alert manager sends alert messages to a slack channel¹⁵ so that required (and for the moment) manual actions can be performed. Fig.4 shows an example message.

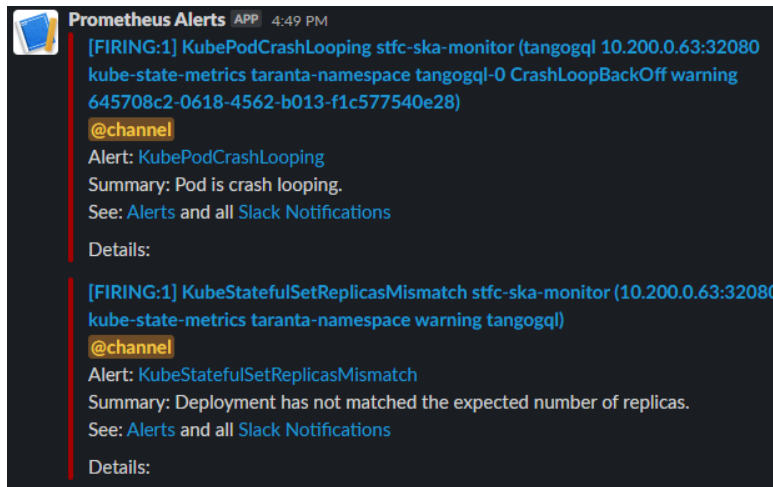


Figure 4. An alert message in Slack

3. THANOS

Thanos is an open source project that provides global query view, high availability, long term data storage with historical lookup capabilities.¹⁶ Thanos is used in conjunction with Prometheus instances and as a middle layer to provide its querying features to compose a centralised solution for monitoring. It has many components that are highlighted in Fig. 2 for replication, storage integration and centralised, load-balanced query functions. Its main components and how they are used in SKA are explained below. Note that, SKA is currently progressing towards implementing the architecture in Fig. 2.

- **Thanos Sidecar** is the main component which queries Prometheus, makes it available for querying and uploads TSDB blocks to object store. This is deployed next to each Prometheus container where possible. If it's not possible to query it from outside, it's deployed elsewhere where it's configured to use Prometheus remote-write API to access the data.
- **Thanos Store Gateway** presents metrics from storage which is used for historical queries
- **Thanos Ruler** is the equivalent of Prometheus Ruler that is used for managing rules for Thanos
- **Thanos Compactor** is responsible for downsampling and retention for the data for effective long term storage
- **Thanos Querier** aggregates, deduplicates the underlying metrics data and implements Prometheus API and exposes it for dashboards and other tools

4. GRAFANA

Another quality of Prometheus is its ability to integrate with many visualization engines. One of them is Grafana,¹⁷ an engine for displaying data on web coming from many data sources. Working with Grafana means to create dashboards in order to perform a particular analysis on a set of monitoring data. From an architectural point of view, it is a plugin architecture where a plugin can be:

- a panel (the basic visualization building block in Grafana),
- a data source (such as Prometheus, MySQL or Elasticsearch) or
- an app (combination of panels and data sources for a specific purpose).

This architecture can be personalized very well for giving the developers and testers the ability to diagnose any kind of problem that can happen in a production system. Fig. 5 shows the resulting model (in UML, every box is an object) of the customization made. The starting point is the Grafana object which is basically an aggregation of plugins and is associated with one or more dashboards (a collection of panels). A plugin can be a panel, an app or a data source. The data sources used are Prometheus, MySQL databases, one for the TANGO archiver¹⁸ and another one for the TANGO database (which represents the CORBA naming server) and the Elasticsearch database. In this model it is shown a simple customization made for displaying information in Grafana: the TANGO-Attribute¹⁹ panel.

Fig. 6 shows some panels which display different values from many different data sources.

5. DASHBOARDS

All the selected tools for the SKA infrastructure come with one or more dashboards that help in analysing some aspect of the entire CI/CD infrastructure. In this section we present some of the dashboards that we adopted from various data sources together with two dashboards that have been created for SKA specific analysis. Fig. 7, Fig. 8 and Fig. 9 show some dashboards adopted from the Grafana community for:

- the node-exporter,²⁰

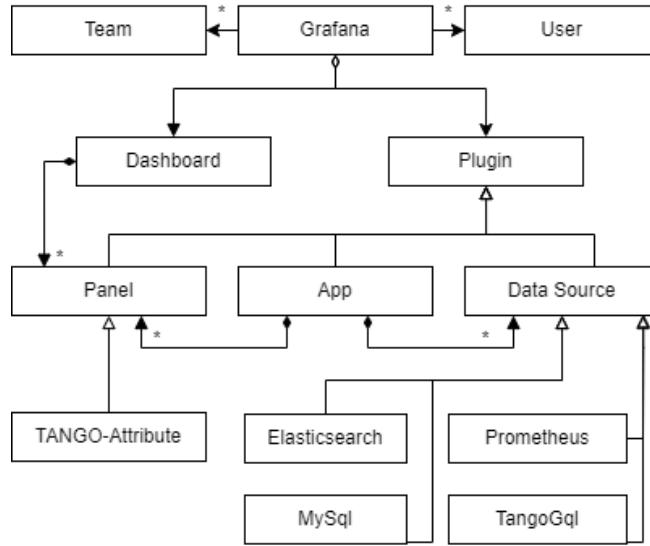


Figure 5. Grafana UML model



Figure 6. Panels taken from the SKA grafana monitoring web application

- the health of the elasticsearch cluster,²¹
- the health of the docker containers,²²

Fig. 10 shows a custom dashboard for monitoring the information inside the mysql TANGO-controls database.

Recently kibana has been used for making some specific analysis like evaluating who are the producers of log messages that arrives to the log database from kubernetes namespaces. The analysis want to understand the cause of a major problem in the elasticsearch cluster by understand why some specific shard were failing. For this purpose a kibana dashboard has been created and it is shown in fig. 11.

Fig. 12 shows a dashbord taken from the project kubernetes-mixin.¹⁴

6. CONCLUSION AND FUTURE WORK

This paper presented an overview of the tools selected by the SKA project in order to monitor the performance of the infrastructure put in place for CICD purposes. In specific Prometheus with Thanos and Grafana have



Figure 7. Node exporter dashboard

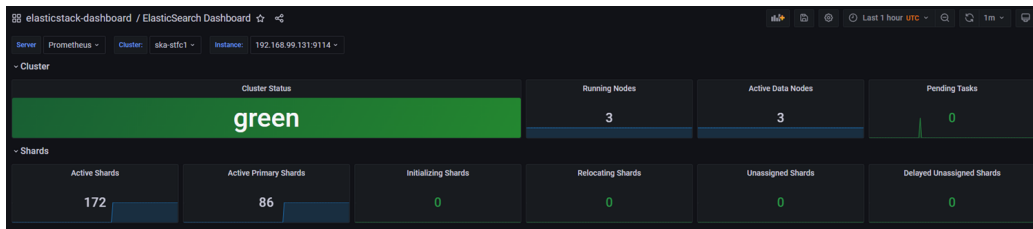


Figure 8. Elasticsearch dashboard for cluster health



Figure 9. Docker metrics dashboard for container health

been selected for this scope. The main rationale for this choice is the modifiability aspect of both of them. In fact adding new exporters for Prometheus or adding a new dashboard is very easy. Besides the community around those two projects is very big and active and this gives enough trust on them. The potential of this diagnostic tool resides overall on the analysis that it enables. Considering the work done for "Achieving a rolled-up view of SKA TM health status and state: definition and analysis of aggregation methods",²³ and considering that each aspect of the lifecycle of every SKA element will be reported as timestamped data in a Prometheus server or in one of the data sources available in the TANGO framework, the aggregation methods provided in the paper can be easily realized.

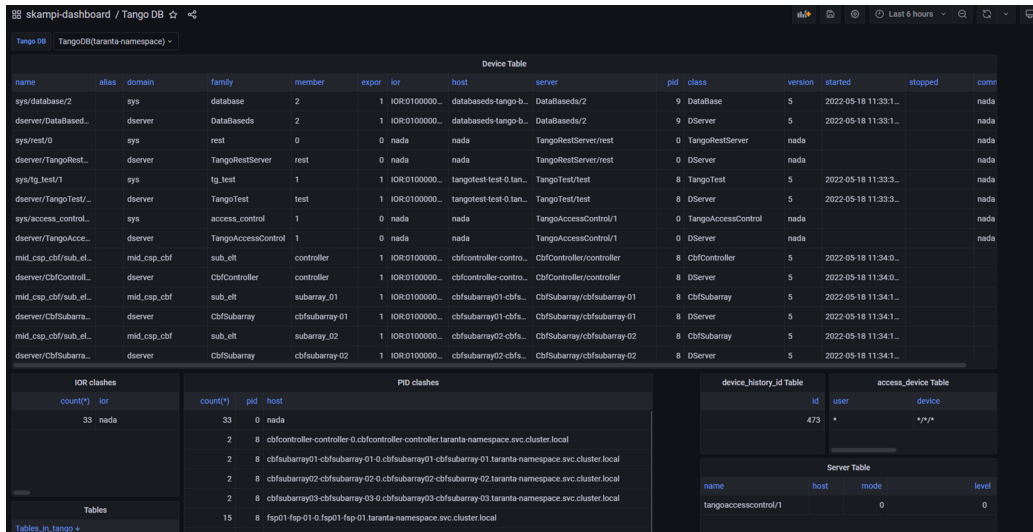


Figure 10. TANGO-controls database dashboard

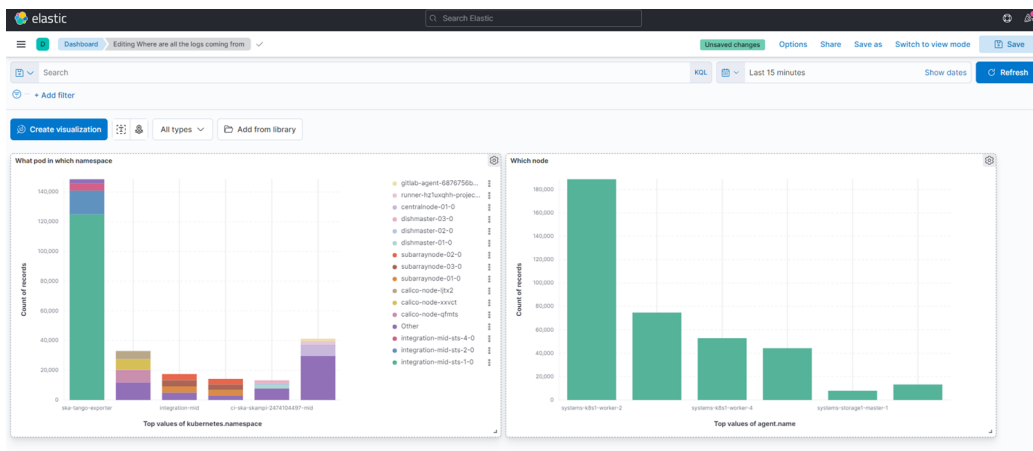


Figure 11. Kibana dashboard



Figure 12. Kubernetes pods dashboard per node

The metrics and dashboards gathered should be available for a long time (at least for the duration of an observation) and should be available from different parts of world for developers, scientists and engineers. The architecture explained is here are in its trial phase while it's getting integrated completely to make this possible and will be investigated for performance, reliability and maintainance aspects.

ACKNOWLEDGMENTS

This work has been supported by Italian Government (MEF - Ministero dell'Economia e delle Finanze, MIUR - Ministero dell'Istruzione, dell'Università e della Ricerca).

REFERENCES

- [1] “Ci-cd practices at ska,” International Society for Optics and Photonics, SPIE (2022).
- [2] “Gitlab.” <https://gitlab.com/>. (Accessed: 5 May 2022).
- [3] “Elasticsearch.” <https://www.elastic.co/>. (Accessed: 5 May 2022).
- [4] “Ceph storage.” <https://ceph.io/>. (Accessed: 5 May 2022).
- [5] “Nexus.” <https://www.sonatype.com/nexus/repository-pro>. (Accessed: 5 May 2022).
- [6] “Tsdbs format.” <https://github.com/prometheus/prometheus/tree/release-2.22/tsdb/docs/format>. (Accessed: 19 May 2022).
- [7] “Node exporter.” https://github.com/prometheus/node_exporter. (Accessed: 19 May 2022).
- [8] “Kubernetes.” <https://kubernetes.io/>. (Accessed: 5 May 2022).
- [9] “kube-state-metrics.” <https://github.com/kubernetes/kube-state-metrics>. (Accessed: 19 May 2022).
- [10] “Tango-controls framework.” <https://www.tango-controls.org/>. (Accessed: 5 October 2020).
- [11] “Ska tango exporter.” <https://gitlab.com/ska-telescope/ska-tango-exporter>. (Accessed: 19 May 2022).
- [12] “Ska cicd deploy prometheus.” <https://gitlab.com/ska-telescope/sdi/ska-cicd-deploy-prometheus>. (Accessed: 19 May 2022).
- [13] “Openstack.” <https://www.openstack.org/>. (Accessed: 5 May 2022).
- [14] “Prometheus monitoring mixin for kubernetes.” <https://github.com/kubernetes-monitoring/kubernetes-mixin>. (Accessed: 19 May 2022).
- [15] “Slack.” <https://slack.com>. (Accessed: 5 May 2022).
- [16] “Thanos.” <https://thanos.io>. (Accessed: 23 June 2022).
- [17] “Grafana.” <https://grafana.com/>. (Accessed: 19 May 2022).
- [18] “Tango archiver.” <https://tango-controls.readthedocs.io/en/latest/tools-and-extensions/archiving/HDB++.html>. (Accessed: 19 May 2022).
- [19] “Ska tango attribute panel.” <https://gitlab.com/ska-telescope/ska-tango-attributes>. (Accessed: 19 May 2022).
- [20] “Node exporter full.” <https://grafana.com/grafana/dashboards/1860>. (Accessed: 19 May 2022).
- [21] “Elasticsearch dashboard.” <https://grafana.com/grafana/dashboards/878>. (Accessed: 19 May 2022).
- [22] “Docker and system monitoring.” <https://grafana.com/grafana/dashboards/893>. (Accessed: 19 May 2022).
- [23] Dolci, M., Canzari, M., Carlo, M. D., and Smareglia, R., “Achieving a rolled-up view of SKA TM health status and state: definition and analysis of aggregation methods,” in [*Observatory Operations: Strategies, Processes, and Systems VII*], Peck, A. B., Seaman, R. L., and Benn, C. R., eds., **10704**, 182 – 191, International Society for Optics and Photonics, SPIE (2018).