Monitoring CERN Windows Infrastructure using Open Source products

Pablo Martín Zamora[1*], *Michalis* Kalliafas[2]

[1] European Organization for Nuclear Research (CERN), Geneva, Switzerland
[2] University of Piraeus, Piraeus, Greece

Abstract. More than 500 servers are actively managed by the Windows infrastructure team on the CERN site. They run critical services for the laboratory, from controlling some of the accelerator systems to directory services, storage, and databases. Visibility of the state of these servers at all times is critical for smooth operation of their services. This paper describes the implementation of an open-source ecosystem based on icinga2, focused on how to monitor Windows-based services, and the technical choices and configurations that were made to transparently deploy and manage an open-source monitoring infrastructure in the organization.

1 Introduction

The project began at the end of 2020, as an initiative to look for alternative solutions that could replace Microsoft System Center Operations Manager 2012 R2 (SCOM 2012 R2) [1]. This effort was done in the context of a CERN wide campaign to reduce dependencies on commercial software in order to avoid future vendor lock-ins.

A small proof of concept was implemented using "icinga2" [2], to explore and evaluate whether this solution would meet the technical requirements to replace SCOM. There were several requirements that would need to be met such as seamless integration within the Windows servers, easy agent deployment and the possibility to manage configurations centrally through a modern user interface. Icinga2 met all these, standing out in the opensource monitoring landscape with features such as a native icinga2 Windows agent, an extensive support for PowerShell scripts [3] and support for configuration management tools such as Puppet (the IT supported configuration management framework at CERN), a rich API that could facilitate the integration of the system with IT Central monitoring stack and the possibility to manage all the icinga2 configurations easily via a powerful web interface.

We proceeded to implement and customize the solution, the main goals to achieve were: to seamlessly migrate all the services being monitored by SCOM 2012 R2, to work on the technical challenges of agent deployment, to integrate icinga2 with CERN monitoring and notifications stack (IT Central monitoring service, notifications, E-mails) and to expand the out-of-the-box offer by developing custom functionality such as tailored monitoring scripts that could fit into the complex ecosystem of Windows-based IT services.

* Corresponding author: pablo.martin.zamora@cern.ch

In 2022, the tool was deployed in the production infrastructure, and since then it has become the de-facto monitoring tool for Windows Infrastructure services running in the IT Department.

2 Icinga2 at CERN

2.1 System Architecture

Icinga2 offers different architectural approaches to deploy its components, depending on the size and requirements of an organization [4]. It was decided to go for a simplified setup as the most convenient to support around 500 Windows servers, as this would remove unnecessary complexity in maintaining the system and spending time looking after the infrastructure.

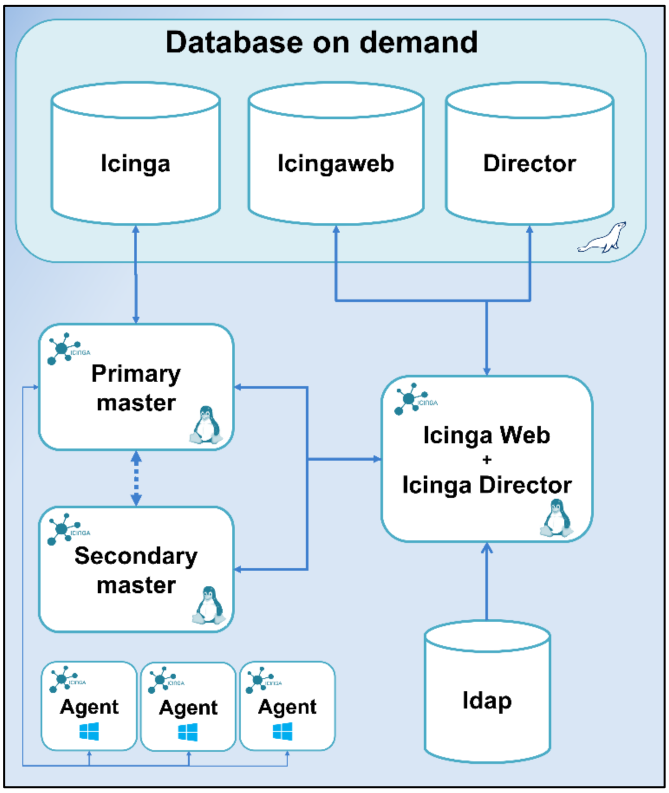


Fig 1. Overview of icinga2 architecture at CERN

The architecture consists of three virtualized servers running CentOS 7 [5]. The main components are an active and a passive metric collector servers known in the icinga2 argot as “masters”. These master servers operate in a highly-available setup, ensuring uninterrupted data collection. In this setup, if one master becomes unavailable, the primary master role is seamlessly taken over by the other available master at that time ensuring that the icinga2 agent that runs on Windows clients can always contact one of the available masters to deliver metrics. To improve resilience against unexpected events, each master is deployed in a different physical area of CERN’s Computer Centre, making the solution resilient to common business continuity scenarios such as prolonged power cuts.

The second component is the icinga2 Web server, which acts as the main website frontend for administrators and users of icinga2, enabling their interactions with other components of the system. The Web server leverages several administrative capabilities, the most notable being the “icinga2 Director” [6]. The icinga2 Director allows for a centralized management of all icinga2 configurations via a user-friendly web interface. It helps icinga2 administrators manage the system, which is especially relevant when dealing with many connected nodes and multiple people responsible for distinct groups of nodes or services. The Director allows for different ACLs per administrator and to track changes to agents, checks and configurations using its version control, which makes it trivial for an accidental change to be reversed or for a non-working configuration to be rolled back almost immediately after it has been deployed.

The Icinga2 Web server is integrated with Microsoft’s Active Directory [7] as the LDAP directory to enable user-based authentication. It was a deliberate decision not to integrate it with CERN’s Single Sign-On (SSO), in order to maintain visibility over critical components in the event of general-wide incidents and eliminate an additional layer of dependency for the system.

Both the Web server and the master servers communicate with their respective MySQL databases, which are hosted on CERN's On-Demand Database Service (DBoD) [8]. There is a database shared by all the masters, a database for the Web server and a database for the icinga2 Director component.

The deployment of the system architecture is fully automated in three environments (dev, QA, and production) using Puppet, which helps recreating any master or web server when needed and facilitates future migrations of the infrastructure.

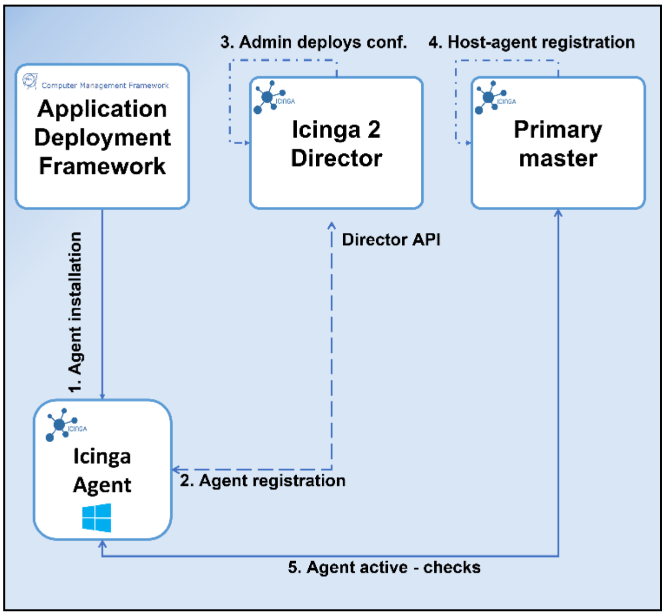


Fig 2. icinga2 agent installation workflow

To allow clients to send metrics to the icinga2 masters, a light icinga2 agent is installed locally in each Windows server, which provides the capability to execute PowerShell scripts directly on the local host, leveraging the power to execute tasks, check registry keys, WMI settings or make use of any PowerShell module installed in the machine.

The agent installation begins with the distribution of the agent to the Windows server park. To do this, we use CERN’s in-house configuration management application called

“Computer Management Framework” (CMF) [9]. CMF automates the installation of the package into the node and executes a series of PowerShell scripts to configure and set up the icinga2 agent with the environment parameters. Most of the scripting work uses the icinga2 PowerShell modules provided by the icinga2 repositories.

In the next step of the process, the agent calls the icinga2 Director API to register itself with the system. At this point, an administrator will see the incoming request for registration of this new node in the icinga2 Web interface and approve it, which pushes the configuration for the node into the master servers and authorizes the agent to begin sending metrics. Once the agent installation is completed, the agent begins communicating with the master and its interactions and check runs can be seen in the icinga2 Web interface.

2.2 System overview

2.2.1 Web interface

The Web interface serves as a centralized hub for icinga2 administrators and provides a comprehensive view of the health of the services and how they are performing. It is extensively used during operations such as patching, migrations and rapid assessment of configurations in the infrastructure, ensuring real-time monitoring.

Through the icinga2 Director extension, the Web interface offers a high level of granularity, helping classify nodes into categories such as “hostgroups”, which serve as the logical groupings where different sets of icinga2 monitors can be applied. This feature profits from each IT services’ hierarchy and separation and allows for different icinga2 checks or scripts to be assigned to certain hostgroups but not others. By logically classifying machines within icinga2, we can be more precise and target notifications to certain groups of administrators and assign specific monitors to certain parts of the infrastructure.

A relevant part of the Web interface is the ability to create separate dashboards per administrator and/or functionality. Examples of this are a dashboard that shows all the unpatched Windows servers, a dashboard that shows hardware issues, etc.

For instance, we monitor critical Windows infrastructure, such as Active Directory and Microsoft SQL databases [10], which are accessible only to Windows administrators within icinga2. On the other hand, we also monitor the infrastructure associated with CERN's Engineering licence servers, which is exclusively accessed by Engineering license service administrators. This ensures that each administrator receives and has access to the relevant information for their services without the risk of interfering with others' monitoring and operations.

Other examples of Windows’s services monitored with icinga2 are CERN’s PLM (Project Lifecycle Management) infrastructure, Microsoft DFS servers, Engineering License servers (for which we monitor Windows and Linux devices), CERN’s Print servers, Microsoft Hyper-V clusters and Web servers, including Internet Information Services and Drupal websites.

2.2.2 Metrics, notifications and automations

There are two types of metric collections or checks consumed by icinga2: remote-based checks and local agent-based checks. Among these, the latter is the most frequent approach used on Windows devices.

A local agent-based check consists in using icinga2 with its out-of-the-box support for PowerShell and built-in modules to query anything inside a machine. The base checks are then extended using custom scripts, in order to perform more advanced monitoring checks on a given node. As an example, a local agent-based check can be used to monitor whether a scheduled task has been executed successfully or to verify if a node is up to date with patches or to monitor the successful completion of backups to a certain location.

Remote based checks on the other hand, are used less often, but they remain useful to query machines that cannot have an agent installed (such as aliases, virtual load balancers, etc.). They can be used to determine which ports are opened, whether a website is available by monitoring its URL or checking certificates lifetime and expirations. The remote-based checks are developed using Python, while the local agent-based checks are mostly written in PowerShell.

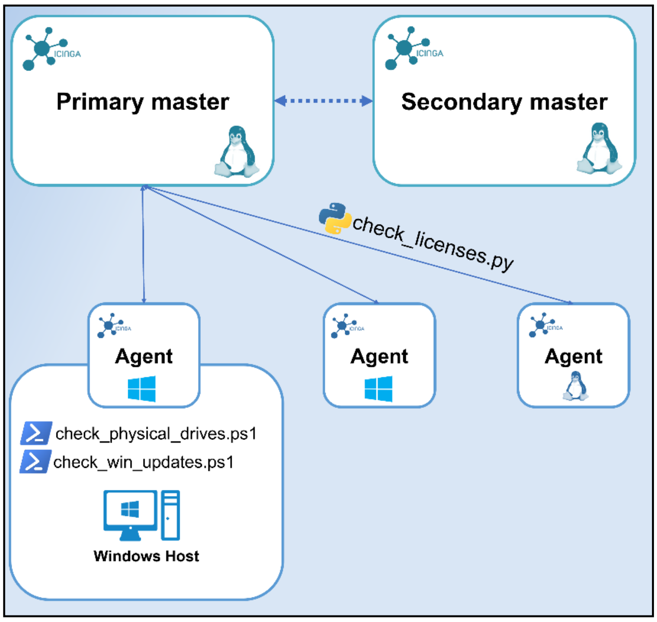


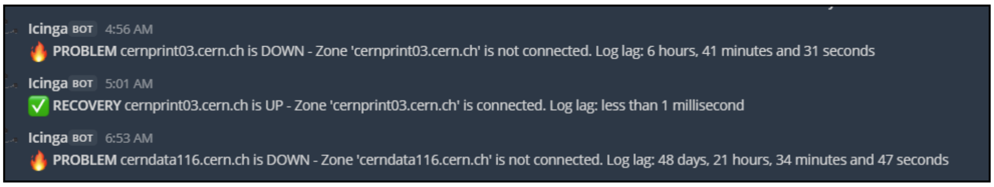
Fig 3. Overview of icinga2 metrics collection

At CERN, we have developed robust check functionality adapted to monitor our Microsoft on-premises services such as Active Directory (to check database replication, LDAP ports and DNS records for example) and SQL Server (to check memory consumption and successful backup restores) as well as to raise an alarm when licensed software expires, and physical hardware events occur.

By integrating these functionalities into a unified framework and a single view, we gained the ability to monitor all aspects within the same dashboard and leverage alarm and notification capabilities within the system itself.

The core of icinga2 is to gather metrics from servers, aggregate them in the icinga2 masters and display their status in the icinga2 web. Based on what the status of a service is, a notification is raised to prompt administrators of that service to take a closer look. Notifications can also be enriched with data coming from the local-agent checks, so an administrator immediately knows the status of a flag, or a key by looking at the email notification without having to query the target server.

Notifications are structured using a templating model, where settings from a parent notification are inherited by default and can be overridden by child notification templates. This flexibility allows to adapt each notification and alarm to different use cases. For example, in the case of critical events, such as when a Domain Controller replication for Active Directory is not functioning properly, we have configured the system to send email, SMS, and Mattermost [11] notifications. However, for routine or low-importance events, it may be sufficient to view the status directly on the icinga2 web interface and not raise any notification.


Icinga BOT 4:56 AM
🔥 **PROBLEM** cernprint03.cern.ch is DOWN - Zone 'cernprint03.cern.ch' is not connected. Log lag: 6 hours, 41 minutes and 31 seconds

Icinga BOT 5:01 AM
✅ **RECOVERY** cernprint03.cern.ch is UP - Zone 'cernprint03.cern.ch' is connected. Log lag: less than 1 millisecond

Icinga BOT 6:53 AM
🔥 **PROBLEM** cerndata116.cern.ch is DOWN - Zone 'cerndata116.cern.ch' is not connected. Log lag: 48 days, 21 hours, 34 minutes and 47 seconds

Fig 4. icinga2 Mattermost notification of nodes going down and up.

A good example of automation that we have implemented at CERN is the automatic remediation of icinga2 alarms by attempting to heal abnormal states on the monitored nodes. To accomplish this, we make use of an icinga2 functionality known as "Event Handlers". An Event Handler is a tailored PowerShell script that will be executed when certain conditions are met during a monitoring check.

For instance, if a node reports issues with a specific service running a database, we can create an associated "Event Handler" to attempt tasks such as restarting the service, deleting a certain file, or even restarting the machine after a certain number of failed attempts. This approach is used in production to resolve issues with websites hosted on Microsoft Internet Information Services (IIS) [12]. It is common for these sites to temporarily lose access to the underlying storage hosting their configuration files, resulting in intermittent unavailability until an administrator has manually restarted the IIS service running on the monitored Web server. To address this, an Event Handler is triggered when such event is detected, the IIS service is restarted, and the website is back online.

It is this kind of automation that makes icinga2 very powerful, and by leveraging this functionality, we achieve efficient and automated healing capabilities, ensuring quick resolution of common issues, avoiding tedious manual tasks, and minimizing downtime.

2.2.3 CERN Integrations

An important part of the project was the integration of icinga2 with the IT Central monitoring stack. To do this, we aggregate relevant events from icinga2, such as disk failures, memory errors or anomalies in Active Directory replication for example. These "interesting" events are exported to a dedicated influx database, and then plotted for visualization purposes using tools like Grafana [13]. This allows us to report valuable service performance indicators and gain insights into the performance trends on Windows servers over time.

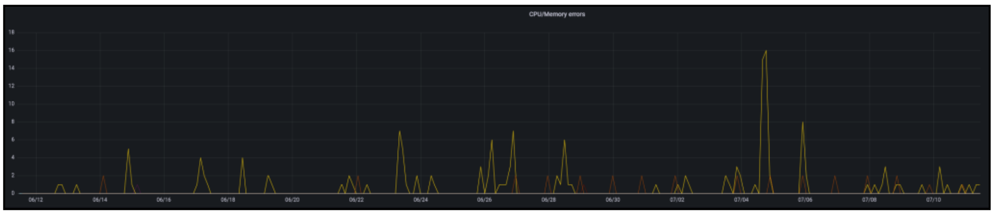


Fig 5. Example of memory errors reported by two problematic servers in a period of a month.

Another noteworthy integration is the integration of "Service Availability metrics" with our IT Service Management (ITSM) tool. We have developed logic within icinga2 to query various icinga2 host groups and services using the icinga2 API. Based on the outcome of these queries (such as the number of available nodes in a certain host group or the status of important services), the system makes a categorization for each service. This allows us to determine whether a service is functioning correctly, experiencing degraded performance, or completely unavailable.

These metrics are exported to IT Central Monitoring and then ingested into Service Now [14], our ITSM tool where they can be accessed by CERN users to check the availability of IT services.

3 Future steps

We continue to build new monitoring checks and logic into icinga2, to address new use cases and improve the visibility over our existing Windows services.

In the next year we will need to decide the path to migrate the icinga2 architecture to another Linux distribution to align with CentOS 7 end-of-life. It will be an interesting exercise, as the icinga2 RPMs for RHEL are paywalled [15] and icinga2 is no longer distributed for CentOS.

A prototype to migrate icinga2 to another Linux distribution such as Debian or Ubuntu, will be explored, with the downside that these distributions are not centrally supported at CERN, losing the advantage of relying on all the built-in tooling that currently exists for CentOS/RedHat such as puppet, monitoring, etc.

4 Conclusion

When we initially implemented icinga2, the primary objective was to remove the dependency on a complex commercial monitoring system and simplify service maintenance and operations. In the process, we successfully implemented an open-source monitoring framework for Windows services, effectively decommissioning SCOM, and expanded monitoring of our services. This was achieved through a combination of custom configurations such as deployment scripts, puppet code and PowerShell scripts tailored to our particular use cases.

The main challenge of the implementation was to orchestrate the deployment of the infrastructure. This work was extensively refined with the help of Puppet configurations to achieve full automation of the deployment and client management.

An important milestone of the project was the integration of icinga2 with CERN's IT monitoring stack. This integration proved particularly valuable for notifications and alarms, with administrators relying heavily on SMS, email, and Mattermost; and visualizations based on Grafana where we could analyse behaviour of our Windows systems over time.

In summary, by implementing icinga2 to monitor Windows Infrastructure, we have significantly reduced the time spent on the maintenance of our monitoring infrastructure. This allows us to focus on delivering value by evolving our services and implementing new functionalities while spending less time looking after the system.

References

[1] Microsoft, System Center Operations Manager, version 2012 R2, 2012. Available from https://www.microsoft.com/en-US/download/details.aspx?id=40844 [accessed 2023-09-04]

[2] Icinga GmbH, Icinga2. Available from https://icinga.com/docs/icinga-2/latest/doc/01-about/ [accessed 2023-09-04]

[3] Icinga, Icinga PowerShell framework. Available from https://github.com/Icinga/icinga-powershell-framework [accessed 2023-09-04]

[4] Icinga GmbH, *Distributed Monitoring with icinga2*. Available from https://icinga.com/docs/icinga-2/latest/doc/06-distributed-monitoring/ [accessed 2023-09-04]

[5] CERN, Latest production system updates for CERN CentOS 7. Available from https://linux.web.cern.ch/updates/cc7/prod/latest_updates/ [accessed 2023-09-04]

[6] Icinga GmbH, Icingaweb2 Module Director, Available from https://github.com/Icinga/icingaweb2-module-director [accessed 2023-09-04]

[7] Microsoft, Active Directory Domain Services Overview. Available from https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview [accessed 2023-09-04]

[8] CERN, Database on demand service, Available from https://dbod.web.cern.ch [accessed 2023-09-04]

[9] CERN, Computer Management Framework, version 2023. Available from https://cmf.web.cern.ch/cmf/Help/?kbid=001001 [accessed 2023-09-04]

[10] Microsoft, SQL Server. Available from https://www.microsoft.com/en-us/sql-server [accessed 2023-09-04]

[11] CERN, Mattermost. Available from https://mattermost.web.cern.ch/ [accessed 2023-09-04]

[12] Microsoft, Internet Information Services. Available from https://www.iis.net/ [accessed 2023-09-04]

[13] CERN, Monitoring dashboards. Available from https://monit-grafana.cern.ch/ [accessed 2023-09-04]

[14] Service Now, ITSM dashboard. Available from https://cern.service-now.com [accessed 2023-09-04]

[15] Icinga GmbH, Icinga2 RHEL subscription. Available from https://icinga.com/subscriptions/ [accessed 2023-09-04]