Modernizing our Approach Towards Network & Server Monitoring Systems

Brian Hungerman

Ventana, Roche Tissue Diagnostics

Background Information

The Imaging Team at Roche Tissue Diagnostics (RTD) relies on an old, outdated, and unwieldy network monitoring solution with little thought put into its user interface and user experience. This makes understanding network and server health both a chore and a time sink.

As RTD's Platform Software Engineering Intern, I was tasked with updating this system to help the Imaging Team better assess their servers' health via various parameters including:

- HTTP Connections
- Disk Space
- Memory Usage
- CPU Load
- Service Statuses

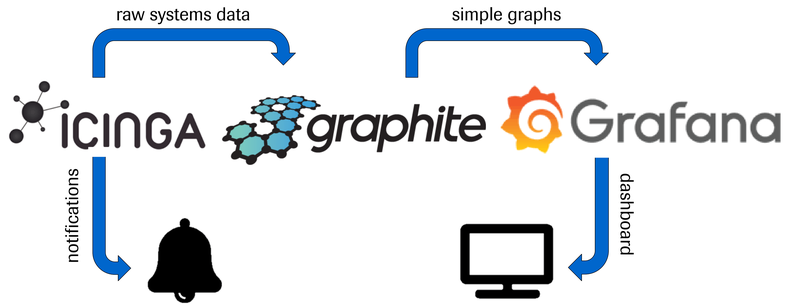
Through these parameters, an accurate assessment of the server and network can be ascertained. However, interpreting these values from Nagios- the current interface- takes time and is not obvious at a glance. Hence, my goal was to build an **intuitive visual dashboard** to replace Nagios.

The old Nagios Core web interface

Principle

The first goal was understanding the tech stack to be implemented. This required a cost-benefit analysis of the 20+ network monitoring solutions on the market to determine which path to go down. First narrowing down by solutions that were under the GNU Public License, then by their reputability and size of their community, a decision was made to use the following technologies:

- **Icinga 2**: data collection daemons and central parent client
- **Graphite**: data interpretation and graphing solution
- **Grafana**: data visualization and dashboard front-end

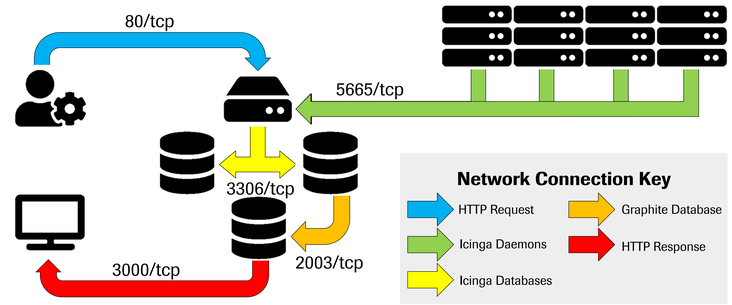


The new tech stack based around Icinga 2

Implementation

One of the main struggles in implementing this system was managing TCP ports. As other processes are running for external clients, it is crucial that this monitoring system **does not interfere with any existing network or database connections**.

To accomplish this, mapping out the existing and new ports and their purposes was essential. This requires a Network Data Flow Diagram (seen below), which cleanly lays out all proposed systems and ports to ensure there is no overlap between the proposed systems and existing ones. After this exercise, concerns over breaking existing connections were mitigated and construction began.



The proposed Network Data Flow Diagram

Documentation

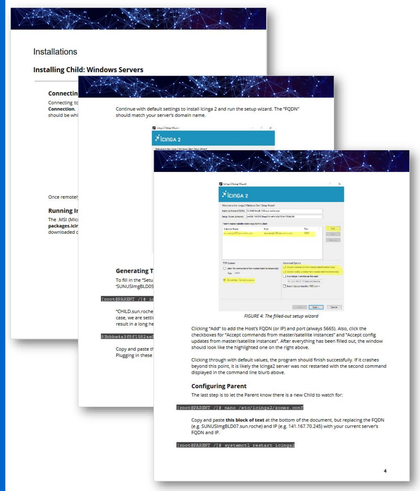


FIGURE 4: The filled-out setup wizard

An excerpt that details adding child servers

Sustainability of the system is imperative. To address this, a **comprehensive user manual** was written to aid in future deployments of and modifications to the system.

Through the 20 page user manual, anyone with even a limited understanding of networks and software engineering will be able to install, update, and visualize Icinga's monitoring daemons.

The document details every aspect of the work done throughout the internship, including:

- Installing the CentOS Icinga host
- Deploying the Icinga Web interface
- Adding new Windows child servers
- Adding new CentOS child servers

- Configuring Service Checks
- Configuring Custom Commands/Plugins

- Troubleshooting tips and tricks
- Additional resources and references

Optimization

Not only does the new system replace every aspect of the existing Nagios installation, but it also adds new features to allow for future development operations practices. The most impactful of these features is the sleek graphing powered by Graphite. With this program, any data point from any metric can be charted and interpreted in real time to allow for **quick debugging and analysis**.

Below are some of the key take-home statistics of the new solution. Every server the Imaging Team uses has real-time graphing of its key resources, uptime, and server ping. All that data from every one of the fifty servers is funneled into one easy-to-understand Grafana dashboard.

50	1200	23,100	1
servers	services	bits/second	dashboard

The beauty of the tech stack is in its **infinite scalability to RTD's growing needs**: servers can be added, removed, and modified seamlessly and zero downtime; additional data metrics can be implemented via plug-and-play Icinga and Nagios plugins; and specialized dashboards can be constructed in seconds to get specialized analytics on specific servers and server groups.

Finishing the internship, the door has been left open for future development operations expansions on the solid foundation of the tech stack chosen for the project. And, if it is especially well-received, this system can easily be expanded to encompass even more of Roche's servers to provide a comprehensive approach to server monitoring and maintenance.

Results

The finished dashboard allows for **at-a-glance diagnostics**, allowing for preventative measures to ensure high up-times of servers and quick response times on network problems that can slow development. The dashboard will be a unique visual art piece as well, displaying live data for all of Roche to see via a flat screen television mounted centrally in our Santa Clara location.

The finished product, accessible from anywhere