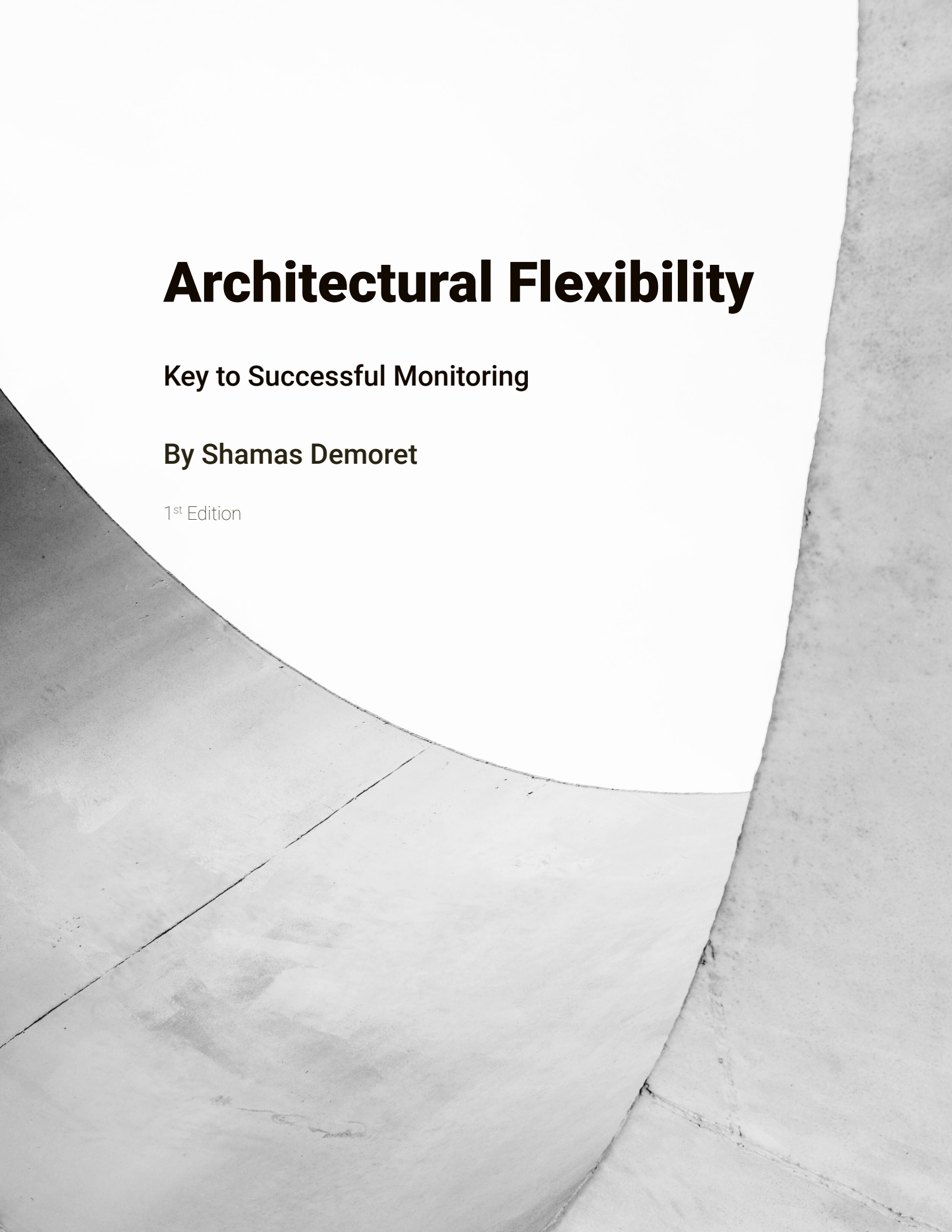
Architectural Flexibility

Key to Successful Monitoring

By Shamas Demoret

1st Edition

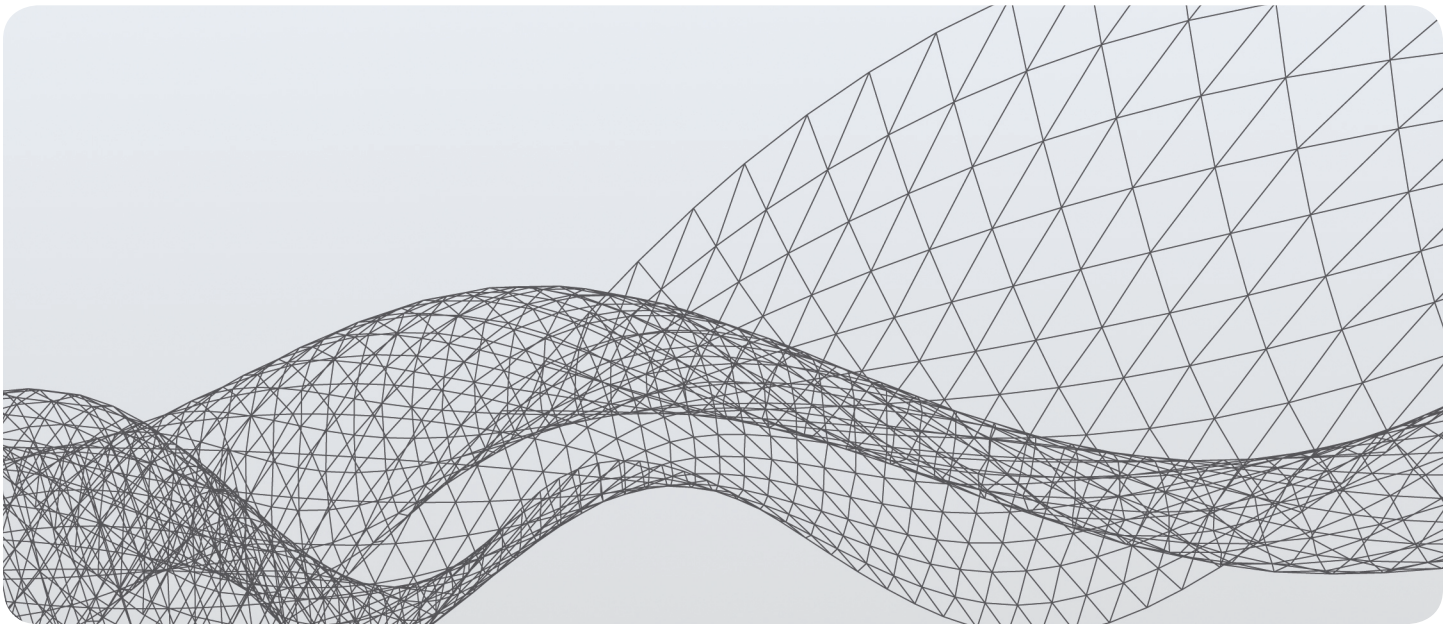


Introduction

Most modern IT infrastructures are a combination of directly managed and cloud hosted resources, and in order to successfully monitor both it is of key importance that the monitoring solution you employ provides a variety of architectural options. If your monitoring platform locks you into a narrow set of options for checking individual elements and distributing your monitoring, it may be time to explore other solutions.

It is also key that your monitoring solution provides the flexibility to monitor less common items such as proprietary applications by providing a framework for creating custom modules for these assets. A strong collection of out-of-box wizards is of great value, but won't likely cover the entire gamut of items you'll need to monitor to ensure the health of your entire infrastructure.

Beyond how individual infrastructure elements are checked, it is also often necessary to integrate with other systems, so a comprehensive monitoring system also needs to provide options for sending data upstream when problems are discovered.



Finally, free backup and development licenses should be provided so that it is not necessary to purchase additional licenses to have a failover install in case of primary failure, and do pre-production testing before rolling new versions and configurations into production.

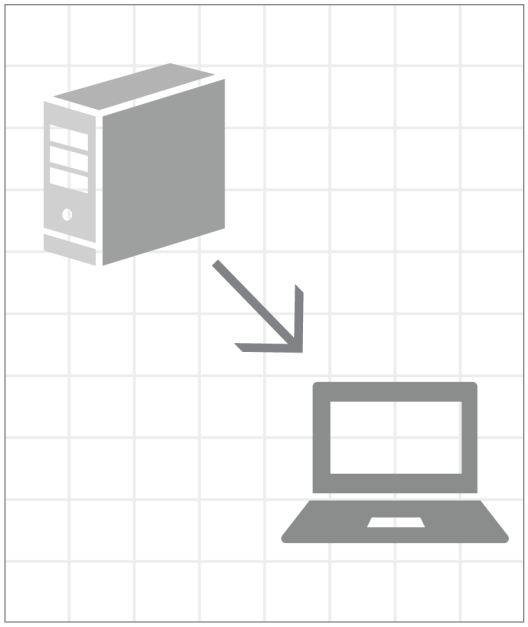
In this e-book we will discuss the many possible ways monitoring can be configured to meet the demands of even the most heterogeneous, distributed environments.

Part 1: Checking Individual Systems

A monitoring solution must provide multiple options for checking the health of a wide array of servers, operating systems, applications, network devices, websites, hypervisors, and other critical systems.

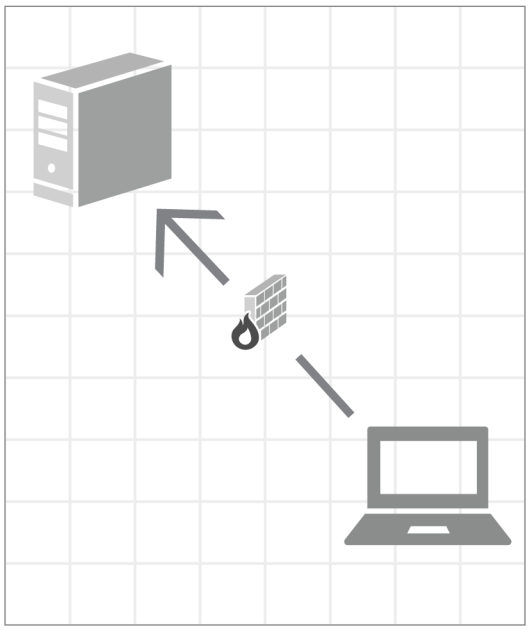
Modern environments tend to be heterogeneous, and may include on-prem physical and virtual, and cloud hosted assets, so it is critical that a variety of options are made available to ensure the flexibility necessary to ascertain the health of virtually anything, anywhere. In this section, we'll cover common options for checking individual systems.

Active Checks



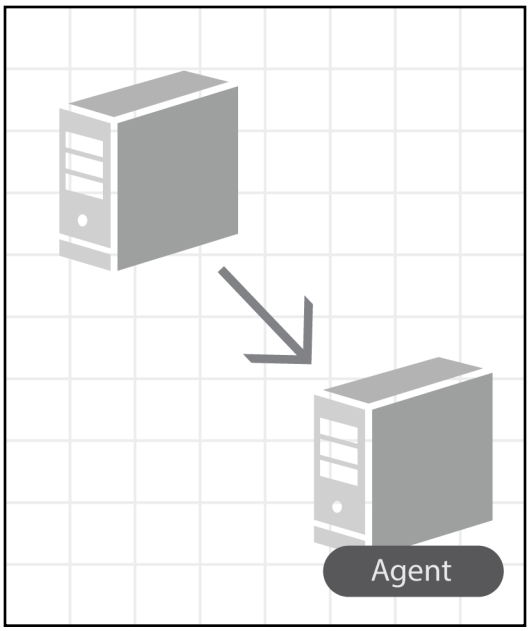
In an active check scenario, the monitoring server initiates communication with the monitored host, executing health checks on the host via an agent, or directly via a native protocol in an agentless scenario. This option is useful when direct inbound access to the target host from your monitoring server is available, and is often the simplest method. In environments composed of a single network, or environments with multiple locations connected via a VPN, active checks are an excellent option.

Passive Checks



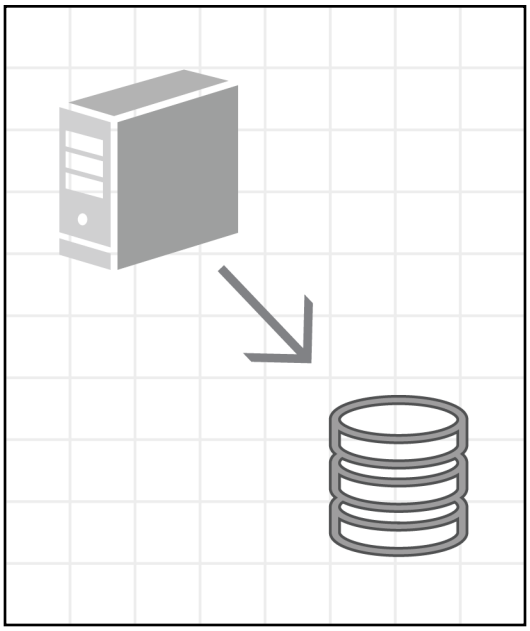
In a passive check scenario, the monitored host sends data upstream to the monitoring server; the monitoring server does not initiate communication with the host, so does not need inbound network access to it.
This option is useful when inbound access to the host's network is not available or desirable, but it is possible to send data outbound from the network. Passive checks may also reduce load on the monitoring server itself, since the monitoring application simply receives the data, rather than initiating a direct request for it.

Agent-Based Method



With an agent-based method, a lightweight agent is installed on the monitored host which enables the monitoring tool to connect to the host and check specific metrics. Agents often provide greater flexibility to monitor a wider array of specific metrics than an agentless method, since they are not limited to the metrics made available by a native protocol. However, agents may slightly increase administrative overhead, since they may need to be occasionally updated. A smart agent has even more utility, enabling your monitoring solution to scan hosts for items such as drives, services, and processes via a smart wizard that interacts with it during configuration.

Agentless Method



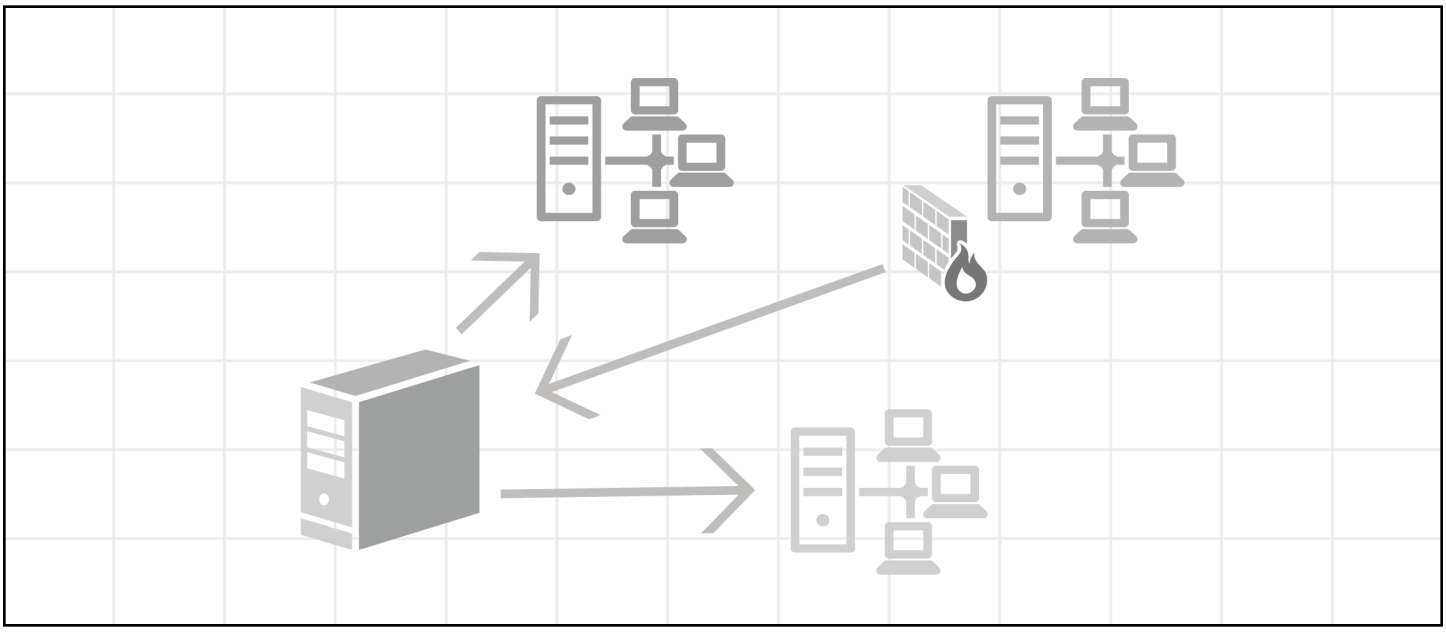
With an agentless method, a native protocol supported by the monitored host is leveraged to run checks. Examples include SNMP (Windows, Linux, network devices), WMI (Windows Management Instrumentation), and SSH (Secure Shell protocol, Linux machines). Although as stated above an agentless method will be limited to the specific metrics the native protocol can produce, agentless methods don't require direct updates outside of regular software and firmware updates. Native protocols may also enable you to scan your hosts for elements which can be monitored, much like a smart

Part 2: Distributed Monitoring

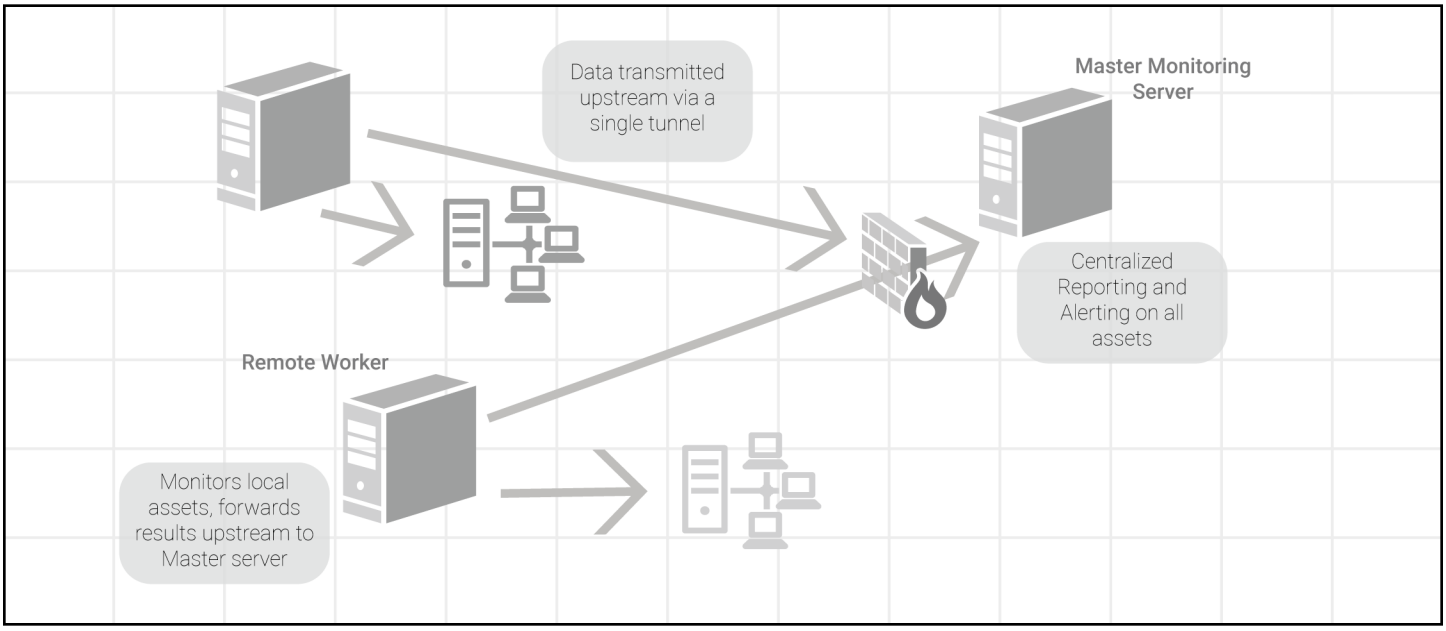
In addition to offering multiple options for checking individual hosts, your monitoring system should also support a variety of architectural options for large, secured, or geographically separated environments that may require multiple monitoring servers.

In a centralized model, all hosts are monitored from a single centralized monitoring server. The hosts could be local, remote, or cloud hosted, as long as your monitoring tool can communicate with them to run an active check, or they can communicate outbound to produce a passive check result.

This model is useful when a single monitoring server can handle, from a technical perspective, the load of checking all of the hosts in a deployment, or in scenarios where access to each of the monitored objects can be configured from one place. In this section we will discuss options which can be employed to meet the demands of other scenarios.



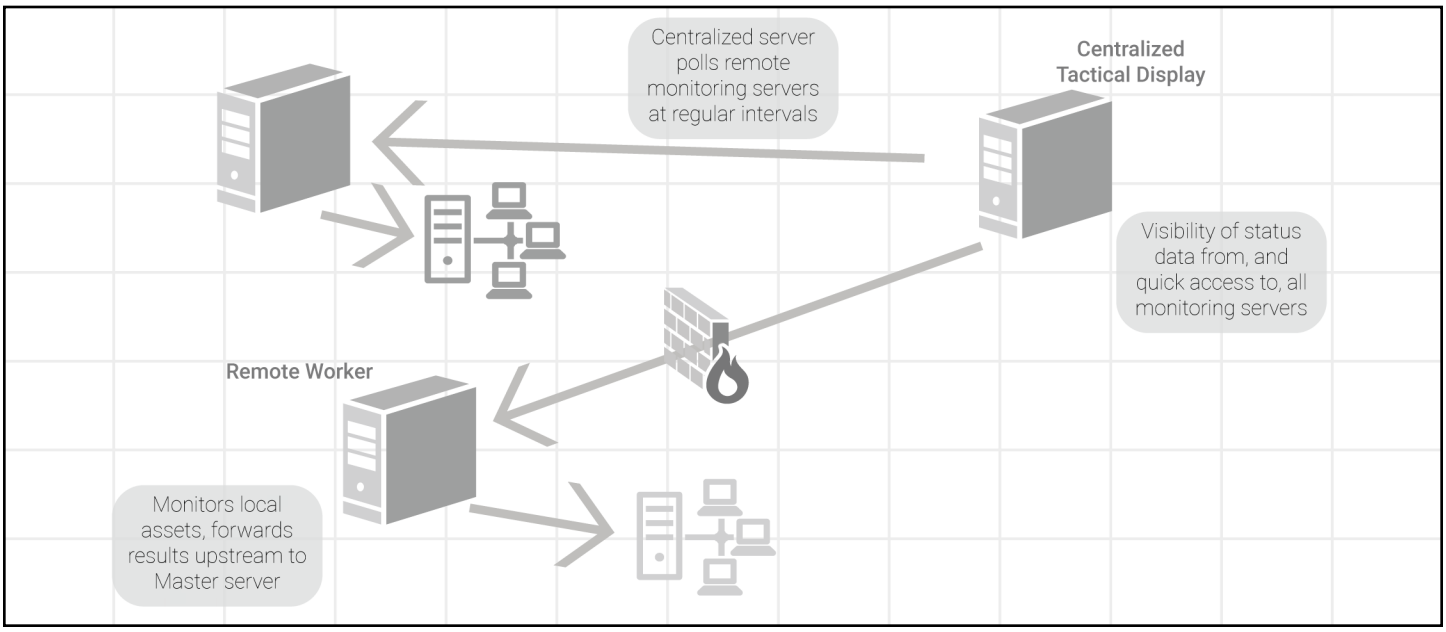
Option 1: Federated



In this model, multiple worker monitoring servers are employed. The remote monitoring servers check the hosts on their network, or in their region, then send the collected data northbound to a primary monitoring server via a single tunnel for centralized visibility and reporting.

Though this model is still limited to the capacity of the single master monitoring server, it simplifies network configuration by requiring only a single path be opened for the data, rather than configuring permission or each of the individual hosts being monitored at the remote sites.

Option 2: Single Pane of Glass



In large-scale deployments, even the federated model may be out of reach, since so many objects are being monitored that a single centralized monitoring server can't facilitate the load. In this case, the monitoring solution should provide an additional option which provides centralized visibility of status data from across the deployment, and quick access to the many individual monitoring servers for configuration and reporting.

Part 3: Customization, Integration, and Backup

Customization

Each infrastructure is unique, and there is often a requirement to monitor less common hardware or proprietary applications, so it is essential that your monitoring solution provides a framework for creating custom checks for items without a baked-in wizard. This framework should be well documented, and ideally a public platform should be made available which you and a community of other users can leverage to share resources.

Third Party Integration

Another important feature is the ability to pass data upstream to other applications, such as ticketing or SIEM (Security Information and Event Management) systems. This capability should provide the flexibility necessary to interact with the other tools in a variety of ways, so that collected status data can interact with a wide variety of third party intake mechanisms.

Examples of integration mechanisms include the ability to send an SNMP trap, and the ability to run a script to take actions and pass data, when status changes are detected.

Backup and Development

The fact that servers and applications are fallible is the reason monitoring tools exist, and monitoring solutions themselves, and their supporting hardware, are no exception. The ideal solution should provide a free failover license in case of primary failure, as well as a development install so that proposed changes and updates can be vetted in a sandbox before being put into production. Your monitoring solution is the application you rely on to keep the rest of your infrastructure secured and running smoothly, so it is vital that it remain functional.

Conclusion

As you can see, there are myriad ways to monitor individual items, and to architect a monitoring solution in larger or more complex deployments. A truly capable monitoring solution should empower you to choose from a variety of constructive options, instead of locking you into a limited set of defaults.

By providing architectural flexibility, the ability to integrate with other tools and create custom monitors, and free backup and testing licenses, your monitoring tool will help you keep things running smoothly and rest easy in even the most complex, challenging infrastructures.

Resources

Nagios XI is an excellent option which meets all of the criteria outlined in this document, so is well worth considering if you're seeking a new monitoring solution.
Nagios XI includes not only a variety of agent-based and agentless wizards to help you quickly configure monitoring of common items such as servers, operating systems, applications, websites, and network devices, but can also be expanded with over 4,200 free community plugins available on the Nagios Exchange community site, and custom plugins that you write using the simple guidelines.

Nagios also offers *Nagios Log Server*, an ELK-based log collection, querying, archiving, and alerting platform, as well as *Nagios Network Analyzer*, a flow data collection, querying, and alerting tool, both of which integrate easily with *Nagios XI*.

In larger environments, *Nagios Fusion* provides a centralized tactical overview which can be used to visualize status data from many individual *Nagios XI* servers, along with quick access to each for configuration and reporting tasks.

We offer free, fully functional 60 day trial versions of each solution, as well as weekly live webinars to help you explore each one individually, or see how multiple tools can work together to provide you with complete monitoring of your infrastructure:

Downloads

Webinars

You are also welcome to email sales@nagios.com for help with any questions you may have, we're happy to help!

About the Author

Shamas Demoret has been with the Nagios team since the early years, serving as a Sales Tech for most of his tenure. In this capacity he's had the opportunity to learn about the requirements of thousands of organizations, and help them explore the Nagios Solutions best suited to meet their requirements.

Through this experience he has learned that the one identifiable common thread among these companies is the desire to keep things secured and maximize uptime. Drilling down further always reveals that the monitoring, architectural, and security requirements of each organization are quite unique.

This ebook is inspired by the wealth of diversity he's encountered, and is designed to serve as a primer on the foundational architectural options a monitoring solution should make available in order to provide the necessary flexibility to be of universal value.