WHITEPAPER

LogicMonitor

5 infrastructure monitoring mistakes and how LogicMonitor addresses them

Enterprise organizations know that they need monitoring to ensure site uptime and keep their business running. Yet, many sites still suffer from outages first reported by their customers. This is often due to small mistakes made with monitoring systems. These monitoring mistakes are easy to make and even easier to overlook, but the consequences are detrimental. Here are some of the most common monitoring mistakes and how LogicMonitor addresses them.

Mistake #1: Relying on individuals and human-driven processes

Does this situation sound familiar?

- It is in the midst of a crisis – were you lucky enough to get notified?
- A change is made to your data center equipment – a new volume is added to your NetApp so that it can serve as high-speed storage for your web tier
- Moving quickly, you forget to add the new volume to your NetApp monitoring

When the dust settles, everyone is too busy breathing sighs of relief to worry about that new volume. Slowly but surely it starts exhibiting latency, due to high IO operations. No one is alerted, and customers are the first to notice, call in, and complain. Quite possibly, the CTO is the next to call.

This is a prime example of why monitoring best practices remove human configuration as much as possible. Not just because it saves people time, but because it makes monitoring – and hence the services monitored – more reliable.

When looking at solution features, choose a platform that:

Examines monitored devices continually for modifications. The ideal platform will automatically add new volumes, interfaces, docker containers, Kubernetes pods, load balancer VIPs, databases, and any other changes into monitoring and inform you via instant message (either in real-time or in batched notifications).

Provides filtering and classification of discovered changes. This helps to avoid alert overload.

Automatically adds new machines or instances to monitoring by scanning your subnets, or even your hyperscale cloud account regularly.

Spontaneously creates graphs and intelligent dashboards. A dashboard graph based on the sum of the sessions on ten web servers used to view the health of your service should automatically update when you add four more servers. Automation of this collection and representation ensures the continuity of your business overview.

Do not depend on manual monitoring updates to cover adds, moves, and changes.

Mistake #2: Considering an issue resolved when monitoring cannot detect recurrence

Outages occur even when you follow good monitoring practices. An issue should not be considered resolved, without ensuring that your monitoring system will detect the root cause or is modified to provide early warning.

For example, a Java application experiencing a service-affecting outage due to a large number of users overloading the system probably exhibited an increase in the number of busy threads. To prevent this from reoccurring, you should modify your JMX monitoring to watch for this type of increase. If you create an alert threshold on this metric or use a monitoring platform that supports dynamic thresholds, you can receive an advanced warning next time.

Early warning will provide your teams a window in which to avoid the outage: time to add another system to share the load or activate load-shedding mechanisms. Configuring alerts that respond to downtime will allow you to be proactive going forward. The next time you experience an outage, the root cause should never point to a repeated preventable event.

This is a very important principle. Recovery of service is the first step but does not mean the issue should be closed or dismissed. Make sure you are satisfied with the warnings your monitoring solution gave before the issue, as well as the alert types and escalations that triggered during the issue.

Sometimes, the issue might not be preventable through monitoring or alerts – catastrophic device failure does occur – but this process of evaluation should be undertaken for every service-impacting event.

Mistake #3: Alert overload

Alert overload and fatigue are detrimental. Too many alerts triggered too frequently result in your teams tuning them all out. You may run into a situation where critical production service outage alerts get pushed to scheduled downtime for eight hours because an admin assumes it was just another false alert.

LogicMonitor

Leverage your monitoring platform to prevent alert overloads by:

Adopting sensible escalation policies that distinguish between warnings and error or critical alert levels

There may be no need to wake people if NTP is out of sync, but if the primary database volume is seeing 200ms latency and transaction time is 18 seconds for an end-user, that is critical. You need to be on it, no matter the time.

Routing the right alerts to the right people

Don't alert the DBA about network issues and do not tell the networking group about a hung transaction.

Tuning your thresholds

Every alert must be real and meaningful. Tune your monitoring to get rid of false positives or alerts triggered on test systems.

Investigating alerts triggered when everything seems okay

If you find there was no outward issue, adjust thresholds or disable the alert.

Analyzing your top alerts by host or alert type

Investigate to see whether remedying issues in monitoring, systems, or operational processes can reduce the frequency of these alerts.

Ensuring alerts are acknowledged, resolved, and cleared

Hundreds of unacknowledged alerts are too difficult to allow easy parsing of an immediate issue. Use alert filtering to view only the groups of systems for which you are responsible.

Mistake #4: Monitoring system sprawl

You only need one monitoring system. Do not deploy one monitoring system for windows servers, another for Linux, another for MySQL, and another for storage. Even if each system is highly functional and capable, having multiple systems guarantees suboptimal datacenter performance. Your teams need one place to monitor as many different technologies as possible, ensuring they are aligned rather than pointing fingers at each other.

While it might be tempting to use the tools that are available directly from the technology vendor, this means your teams will be logging into different platforms regularly and getting a skewed view of the situation.

A central location to store your team's contact details is also vital. You do not want up-to-date information in the escalation methods of some systems but not others. Likewise, it's unhelpful if maintenance is correctly scheduled in one monitoring system but not in the one used to track other components of the same systems. These inconsistencies will lead to incorrectly routed alerts, ultimately resulting in alert overload. A system that notifies people about issues they cannot acknowledge leads to someone inevitably saying 'Oh...I turned my cell phone notifications off.'

A variant of this problem can be when your SysAdmins and DBAs automate things by writing cron jobs or stored procedures to check and alert on issues. The first part is great – the checking.

However, alerting should happen through your monitoring system. A better way is to have the monitoring system run the script and check the output, call the stored procedure, or read the web page. Otherwise, you've just created yet another place to adjust thresholds, acknowledge alerts, deal with escalations, and so on. Locally run, one-off alerting hacks will not incorporate these important monitoring system features. Furthermore, this approach creates monitoring silos, where some team members have access to certain data and alerts, while others do not.

Mistake #5: Not monitoring your monitoring system

Your monitoring solution can fail. Ignoring this fact only leaves you exposed. Companies invest significant capital to set up monitoring and understand the recurring cost in staff time, but then fail to monitor the system itself. Who knows when a hard drive or memory failure occurs, an OS or application crash happens, a network outage at your ISP, or a power failure? Don't let your monitoring system leave you blind to the health of the rest of your infrastructure. The monitoring system encompasses the complete system, including the ability to send alerts. If the outgoing mail and SMS message delivery connections are down, your monitoring system might detect an outage, but it is only apparent to staff watching the console. A system that cannot send alerts is not helping anyone.

False security is worse than having no monitoring system at all. If you do not have a monitoring system, you know you need to execute manual health checks. If you have an unmonitored system that is down, you're not executing health checks and you're unwittingly exposing the business to an undetected outage. If your teams develop a lack of faith in the reliability of your monitoring tool, they may well start to question the validity of alerts it produces.

Minimize your risk by configuring a check of your monitoring system from a location outside the reach of the monitoring system. Or, select a monitoring solution that's not only hosted in a separate location but also checking the health of their own monitoring solution from multiple locations.

The best way to address all of these mistakes is to find a comprehensive monitoring platform that does the work for you. LogicMonitor is a cloud-based unified observability platform that enables organizations to see what's coming before it happens. With advanced AIOps features, LogicMonitor helps teams proactively identify and resolve IT infrastructure issues before they can adversely affect business-critical systems and end-user performance. If you are interested in learning more about LogicMonitor's capabilities, see our platform demo or request a free trial.

About LogicMonitor®

LogicMonitor®'s SaaS-based observability and IT operations data collaboration platform helps ITOps, developers, MSPs and business leaders gain visibility into and predictability across the technologies that modern organizations depend on to deliver extraordinary employee and customer experiences. LogicMonitor seamlessly monitors everything from networks to applications to the cloud, empowering companies to focus less on troubleshooting and more on innovation. For more information, visit www.logicmonitor.com.

LogicMonitor