Proactive Monitoring: Definition and Best Practices

Source: Link

Constantly monitoring your systems and networks is critical to keeping your applications healthy. You can monitor either proactively or reactively. Although both methods are effective, proactive monitoring is preferable to reactive monitoring, and unfortunately, we seem to spend more time being reactive than proactive.

Proactive monitoring is preferable because it can prevent potential issues before they happen and possibly lead to a major downtime. It also helps ensure your end users receive seamless services while minimizing unnecessary revenue loss on your part.

What is Proactive Monitoring?

Proactive monitoring simply means constantly attempting to identify potential issues before they create major challenges for your business. Since proactive monitoring anticipates issues, you can address these issues before an application crashes or performance degradation sets in.

Aside from detecting issues, proactively monitoring your infrastructure(s) and applications allows you to make strategic rather than tactical business decisions. An example of such a decision is knowing when to change components to accommodate a recent increase in users' activities. Capacity planning and optimization are two disciplines you can execute on if you're leveraging the information, you collect proactively. Not only does it minimize user impact, but it's also the most cost-effective way to use your IT investments.

Proactive Monitoring Best Practices

Establish Your Baseline

What does "normal" mean to you? To get the most out of your monitoring practices, you must first understand how your business and infrastructures function under specific types of loads at various points in time. Once you understand this, you'll be able to quickly identify what's "abnormal" and what you need to pay close attention to. You can also plan for potential critical resource limitation issues and remediate issues potentially leading to outages.

You can't establish a baseline based on predictions because "normal" varies depending on the business and its use case. Even if your prediction is based on experience, it's still a guess that may or may not be correct. To get an accurate baseline tailored to your business and infrastructures, you'll need to speak with your developers, IT professionals, and stakeholders.

It’s critical to understand what metrics you must collect to achieve infrastructure visibility. For example, while a gaming application and search engine both require a fast response time, the gaming application’s response time will be faster than the search engines. You also need to examine your historical records for trends, peaks, occurrences, and seasonality to gain insight into established patterns.

Setting this baseline allows you to detect issues like unexplained changes in traffic, which may indicate a performance issue in your full-stack environment. You can set thresholds and receive alerts to warn you when infrastructure utilization is reaching capacity or application response is starting to degrade. I’ll touch on this in the next section.

Establishing a baseline should be a routine procedure. This will help you keep your baseline metrics up to date on any changes or dramatic fluctuations as your business and infrastructure scale over time. Generally, setting up periodic reviews helps you optimize performance and meet your key performance indicators (KPIs).

Identify Problem Areas

Like a guard who keeps a watchful eye to secure a location, proactive monitoring tools keep an eye out for indicators suggesting if we don’t intervene, we’ll have health or performance issues. These tools can inform you whenever any of your metrics fall on either side of the extreme end of the spectrum. But how can this be done? By making use of alerts and baselines.

Before you can set up alerts, you must first identify your potential problem areas. In the previous section, I mentioned your baseline gives you a clear picture of your current status and what you need to pay close attention to. You have critical infrastructure constantly monitored like your firewalls. And then you have others such as resource usage that are routinely checked and are your potential problem areas.

When you use alerts, your monitoring tools will notify you when your system's status approaches abnormal. You don't want to wait to be alerted when your infrastructure or application is already impacting users. By setting warning alerts based on your baselines, it allows you to anticipate the need for configuration changes, additional resources, better load balancing, etc.; for example, you'll know when you're due for a load reduction or an upgrade.

These alerts are often the starting point for any investigation into potential problems, thereby allowing you to mitigate risks. You can use your knowledge of potential problem areas to set alerts and determine which to prioritize. An alert set to draw your attention to, for example, when any of your components are nearing capacity helps you be aware of potential memory issues.

These types of alerts can also help you make better long-term decisions. You just need to configure your monitoring tools to track and receive alerts on such indicators in real time. A game of observation and spot the difference, setting alerts while monitoring will help you swing into action before end users are impacted.

Monitor Your Infrastructure

Infrastructure monitoring is a critical aspect of proactive monitoring practices. However, because of the complexity of today's modern enterprise IT infrastructure, it's become more challenging. Monitoring your infrastructure today entails looking at your physical and virtual machines (VMs), cloud environments, and "as-a-service" platforms in real time. These systems are often distributed across multiple hosts, sites, and data centres.

Here are some important things to know to achieve your infrastructure monitoring objective.

Find Key Metrics to Monitor

To understand your entire system, you need to capture your infrastructure's performance and key metrics at normal. Remember this varies depending on the use case and application. A deployment frequency of a few minutes will be fine for a decision support system (DSS), but not for an online transaction process (OLTP).

Having these baseline metrics is key when developing a well-designed monitoring strategy. I'll go into more details about this in the following sections. Examples of other key metrics are RAM, CPU usage, response time, storage, throughput, and quality of experience (QoE).

Optimize Your Resource Use

Always aim for efficiency and optimization. Underutilization of and redundancy in your computational resources is a wasteful expense and will result in your business goals not being met. However, consuming more resources than you anticipated because of a design flaw could impact users and your business and, because you must act quickly, cost you more to resource the issue and raise costs unnecessarily. The key is to track and analyse your resources as you use them.

Continuous Integration

Changes incompatible with third-party components often result in buggy code. Your monitoring tools should work in tandem with existing processes to give complete visibility into your infrastructure. In this regard, a continuous integration/continuous delivery (CI/CD) pipeline with load testing on the application is important.

Application Monitoring

With your baseline, you'll need to set custom performance thresholds for each segment of your infrastructure. Using hooks or API endpoints in your application can help implement application-level monitoring efficiently. You can then detect performance degradations in real time and take pre-emptive measures on flagged infrastructure.

Pick the Right Tool to Get Seamless Monitoring

The goal is that you spend most of your day meeting business goals and less time on your monitoring tool. Proactive monitoring can be daunting at first, but it's well worth the effort. Thus, to experience seamless monitoring, you need to find the right monitoring tool.

Your monitoring tool should be simple to use and require little maintenance time. This tool should also provide you with granular insight into your application's performance. Also, context matters. Knowing and getting an alert that your infrastructure is having an issue isn't the same as understanding why.

This is where picking the right tool comes in. By integrating SolarWinds® tools like Loggly® and AppOptics™, you can identify issues and understand why they happened. There's a certain ease that only an evaluative tool like Loggly that requires little configuration and quick remediation can give. To learn more about how Loggly and AppOptics metrics and logs together can be used to help you plan for capacity changes and optimize your environment, read this blog post.

This is particularly important when purchasing third-party applications. Regardless of the metrics you choose to monitor, don't neglect locally installed monitoring software or third-party monitoring tool. This will determine your ability to be efficient and proactive when a situation arises. You can always request help from your third-party application providers. Every great third-party application will always have resources and support for you at your disposal.

It’s also important to ensure your monitoring tool itself is up and running, especially after turning it off following a deployment process. Doing this verifies the tool is indeed looking out for issues.

Design a Scalable Monitoring Strategy

At the end of the day, your team’s proactive monitoring capabilities come down to its monitoring practices. With a seamless monitoring tool, your key players will be able to identify flaws and resolve issues early. But to increase your chances of success whenever there’s downtime, you’ll need an effective, scalable monitoring strategy.

Here are a few strategies to consider:

- Having a defined outline is always a good place to start. This helps you avoid potential issues and serves as a jumping-off point for any investigation and evaluation.
- Setting up a process on how alerts should be resolved ensures your team knows what to do if an alert goes off. Alerts are completely ineffective if not acted on.
- Having monitoring objectives gives you insights into your workflow and the effectiveness of your team. This raises the question of how quickly you can debug. Is your monitoring tool capable of meeting your objectives? If not, what’s hindering the process, and how can you speed up troubleshooting?
- We tend to ignore alerts when we’re bombarded by them. Fine-tune your setup process with tools like SolarWinds Snap Agent to help prioritize alerts.
- Other times, this ineffectiveness could be due to a lack of capacity (possibly team size). As your business scales, it becomes logical to delegate responsibilities to a managed services provider.

Monitoring has gotten to the point where tasks can be automated with tools like AppOptics, allowing you focus on other business operations. These tools give visibility by allowing you to drill down into your infrastructure and—better still—into specific lines of code. To get a more comprehensive full-stack monitoring snapshot of your infrastructure, integrate tools that work in tandem. When push comes to shove, your system is only as good as its implementation, strategies, and tools.

In conclusion

Proactive monitoring doesn't stop at getting insights and setting alerts. It feeds into picking the right tool, serving insights to stakeholders and your team so you can have a cohesive, scalable monitoring strategy, and preventing issues before end users are impacted. While this is by no means an exhaustive list, it should give you an idea of some best proactive monitoring practices to consider. It's best to prepare for what's to come rather than wait for the nightmare to creep up on you.