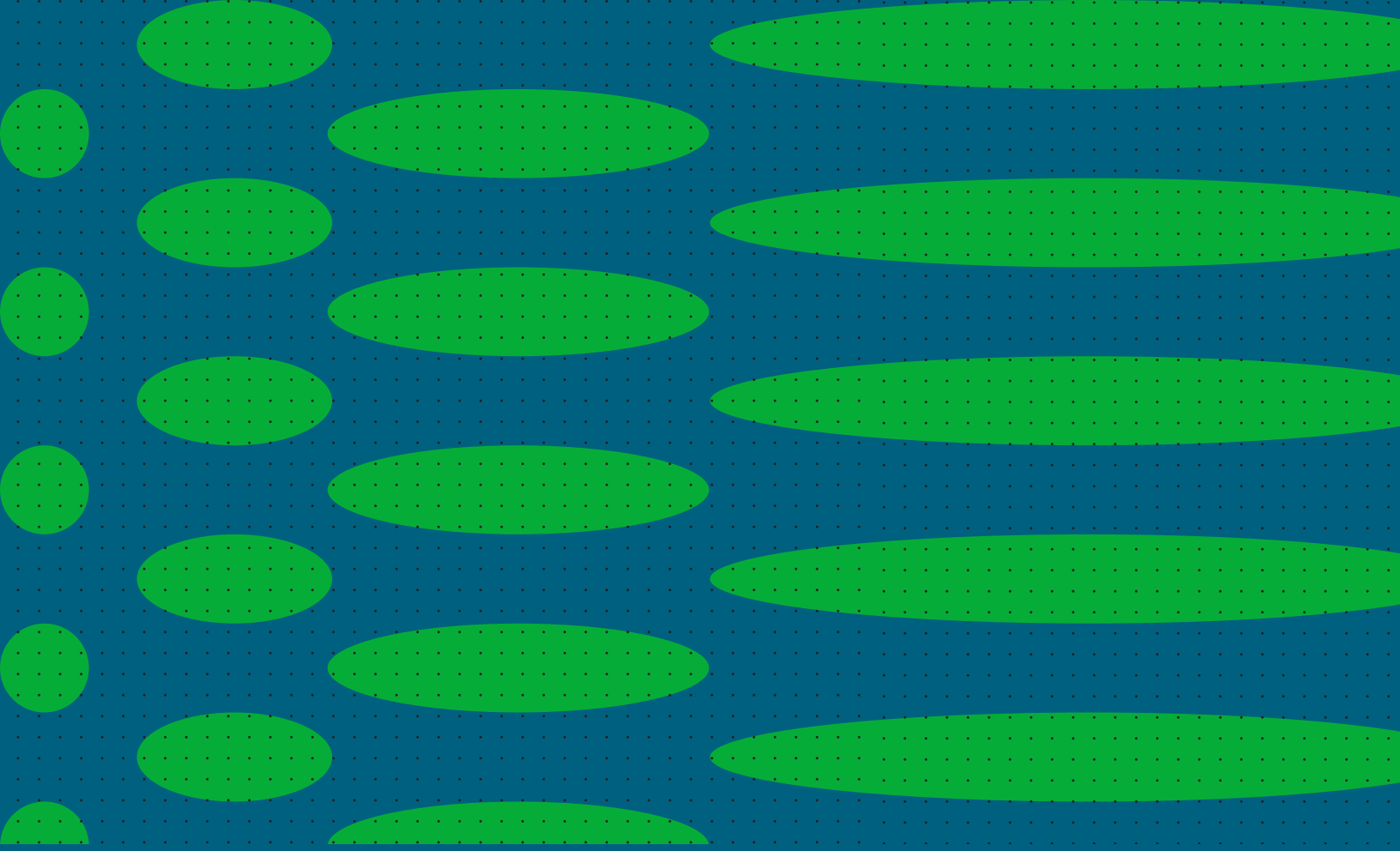


PagerDuty

Best Practices for Monitoring

Reduce outages & downtime

Introduction

Effective monitoring has a huge impact on your business and customers. Without the proper systems in place, your product might go down, giving you an unreliable business that your customers can no longer trust. A lot of damage can occur when your team misses an issue in your infrastructure that negatively impacts customers' experiences.

That's why it's worth spending the time to develop an intelligent, streamlined way to monitor and detect issues before they become critical, and respond to system events efficiently. You'll save yourself serious headaches in the future, keep customers happy, and avoid lost revenue from downtime or churn.

But we're all busier than we have any right to be. It's tempting to just cobble together an ad hoc alert system in your free time or not create one at all. Don't fall into that trap. This guide is here to help.

This ebook is designed to teach you best practices for monitoring system events. These best practices will reduce the length and impact of outages, as well as help you prevent them — which means better business results. You'll be able to create and implement an effective monitoring strategy in less time, without losing sleep. Read on to learn how.

Table of Contents

What To Monitor

You've made the commitment to monitor your systems, or to monitor them better. Choosing what to monitor is a critical first step. When you pick the right events to monitor, you ensure that:

- You know about mission-critical issues before customers or your boss
- You always have an accurate view of system performance
- You aren't caught unprepared when key infrastructure fails

Those are all compelling reasons to carefully consider this decision upfront. But how do you actually do this? There are a number of metrics that may be important to the customer experience. To identify these, you'll need to break down any available metrics into one of two categories:

WORK METRICS

These metrics measure what your system produces, such as queries, website visits or revenue.

RESOURCE METRICS

These metrics measure what it takes to produce work metrics and may include items like CPU, network, disk, or memory.

It's tempting to alert on the resource metrics, but you only want to receive alerts for work metrics. High memory usage that isn't also resulting in a degradation in one of your work metrics doesn't need to wake someone up at night. The only time you'll want to establish alerts for resource metrics is if one or more are leading indicators of system failure.

Then, narrow down your list of work metrics to actionable work metrics. For instance, an actionable work metric for a web server might be how many pages you serve without errors per second. A non-actionable work metric, on the other hand, might be how many 404s you serve per second. This isn't actionable because it entirely depends on what people are doing on your site.

Finally, periodically review your list of metrics and their associated alerts. This review should happen regularly with your team, whether on a weekly, bi-weekly or monthly basis. A good place to start is to look at all of the events that alerted (or the most common ones if there are too many) and find out which ones were not actionable. It's worth taking extra time to review and iterate on your monitoring metrics, so you avoid headaches down the road.

Now that you know what metrics to monitor, you need to separate the events by urgency so you can structure alerts accordingly. Rank which ones matter most to your business, and then identify which team members should be notified about each one. Also, you should identify escalation rules, so that in the event the first line of defense doesn't act on an issue within a given amount of time, the issue will route to the second line of defense, or the third, and so on.

With metrics, urgency priorities and team member assignments and escalations in place, you're already most of the way there.

Additional Metrics for Mobile Apps

Mobile apps face some concerns that web services don't, like carrier latency and different OS versions. If your service includes mobile apps, there are two other metrics you need to monitor:

UPTIME

This measures the percentage of app loads that do not crash. To stay competitive in public app stores, uptime must be above 99%.

RESPONSIVENESS

Additionally, the company says the responsiveness of your app should be under one second to satisfy user expectations.

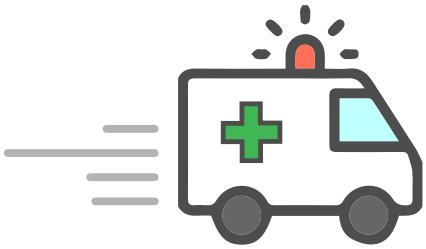
Ensure Your Team Responds Quickly

Now that you know what to monitor and what to avoid, how do you make sure your team responds quickly to the alerts you've set up? There are four key strategies to cultivate a lightning-fast response team.

DITCH EMAIL ALERTS

Email alerts are a huge liability. Even if they're the norm at your company, it's time to ditch them in favor of SMS, voice or persistent mobile push. Here's why:

- They're too easy to miss, and don't help you prioritize. Anyone who faces a daily avalanche of email knows how easy it is to miss important communications. The stakes are extremely high when it comes to your critical infrastructure, and alerts shouldn't be sharing space with the latest Amazon sale or chain email in your inbox.
- You can't assign email to someone. No one should be left wondering if an issue has been resolved or assigned when fast resolution is critical.
- Emails can't be aggregated. When all hell breaks loose, prepare to drown in tons of individual notifications that make your inbox virtually unusable.
- They don't provide team visibility. You can't view all emails in a single dashboard along with other notifications and metrics. That dramatically reduces transparency for all involved and on-call team members.
- They don't provide metrics. And as we've seen, having the right metrics is key to keeping your infrastructure up and running. The only thing you'll measure with email is the blood pressure spike from all the frustration they cause.



Aggregate All Events from Different Systems in One Place

Monitoring success depends in large part on aggregating events and metrics in one place. Non-actionable alerts should be easily suppressed via rules that span multiple integrations, and actionable, related alerts should be grouped into incidents with all the relevant context in a single location. By aggregating all events in a single location, you will have a single place to tune alerting thresholds and customize alert and incident behavior and workflows, which significantly reduces noise and enables you to focus on what matters. Cutting down on responder noise means a far more effective and less burnt out team.

Aggregating events in a central location also improves visibility and collaboration, as it enables full-stack visualizations that surface actionable insights on infrastructure-wide and service health. This also makes it a lot easier to spot patterns and anomalies so you can determine whether there is a noisy service and your alert thresholds need improved tuning, if there is a simmering issue that needs proactive remediation, etc. Make sure all the right team members have access to this dashboard — it will give everyone a real-time snapshot of system and issue status.

Create a Culture of Accountability & Transparency

The only way to create the right culture of transparency and dedication to providing the best customer experience is by treating incident response time as a key performance metric. When you do, it encourages a set of processes that make operations much more transparent.

Having tools that drive accountability can go a long way. You'll want to use a system that allows you to quickly acknowledge incidents when you work on them, or easily re-assign to the right expert.

This establishes clear owners and common understanding of who is the Incident Commander that will be driving the coordination of the response. Your system should also support the ability to route issues requiring a response to the people that are responsible for fixing them, rather than notifying everyone at the same time.

The net result of implementing these tactics is a monitoring culture that puts real-time accountability and transparency first.

Use Analytics for Continuous Improvement

Metrics are a major way to continually make changes that improve your organization, whether those changes are cultural, operational, or at the system-level. That's why operationally mature teams use analytics to gain valuable insight into their work, optimize and load balance their capacity, and drive cultural change. Here are some ways to use analytics for continuous improvement:

- Track Time to Response and escalations to establish a culture of high achievement and benchmark your team's performance.
- Log incidents escalated over time to see how well you're decreasing escalations in the long run.
- Use raw incident count metrics to weed out low-quality alerts, automate common fixes and build runbooks that ultimately help you lower incidents per responder to combat alert fatigue.
- Measure your team by Mean Time to Resolution and use this data to identify ways to decrease downtime.

What To Avoid

No matter what your monitoring needs are, there are some things to avoid as you set up a monitoring system.

NOT MAKING MONITORING HABIT-FORMING

The earlier you get your new team members integrated, the better. Some companies even have them start shipping code on day one. This means you should think about how accessible your incident resolution systems are to new hires. Are they simple? Could they be used without special training? Is it easy to extract the right metrics that must be acted on? If not, monitoring won't become a regular habit at your company, and that leads to dropped alerts.

CHOOSING THE WRONG TYPE OF NOTIFICATION

Monitoring isn't complete if you don't have multiple ways to notify on-call engineers. It's not an excuse to sleep through an outage because an email was lost in the shuffle. In fact, email is rarely a good option. Critical incident notifications should instead come via SMS, voice or persistent mobile push. Ideally, choose a platform that dynamically routes issues and selects the right notification method based on severity.

NOT HAVING A SERVER CHECKLIST

When you set up alerts, it's easy to forget to set them up for each and every server you have. Make a checklist to remember them all. Often overlooked servers include secondary, new and temporary, or cloud servers.

TOO MUCH TIME BETWEEN CHECK INTERVALS

Ten minutes between alerts might seem like a short time, but not when your entire system is at risk. Whatever monitoring system you use, make sure very frequent (for instance, one minute) check intervals are set up.

NOTIFYING THE WRONG PERSON

Implement a system or process to change who receives notifications when normal on-call schedules are disrupted by vacations, holidays or personal issues. After all, a great monitoring system doesn't get you very far if all those timely alerts go to someone on a trans-Atlantic flight or off the grid in the great outdoors. This is also where escalations come in handy: missed alerts are set up to automatically pass on to someone else, so that nothing falls through the cracks.

Wrap Up

LEVELING UP YOUR MONITORING ACTIVITIES

These best practices are designed to help you take your full-stack monitoring and incident resolution processes to the next level. To achieve that, the most operationally mature teams do the following:

- Measure the right metrics from the start.
- Avoid common pitfalls, such as failing to tune alert thresholds.
- Optimize their team's efforts to decrease response time.
- Track and optimize their team's efforts to decrease response time.
- Continually monitor, tweak and iterate after their monitoring and incident resolution system is in place.

Starting with the best practices in this guide, your team will be able to up level your operational maturity as well.