

# 4 SRE Golden Signals (What they are and why they matter)

JULY 29, 2021



SRE's Golden Signals are four key metrics used to monitor the health of your service and underlying systems. We will explain what they are, and how they can help you improve service performance.

## What are the Four Golden Signals?

- **Latency:** the time it takes to serve a request.
- **Traffic:** the total number of requests across the network.
- **Errors:** the number of requests that fail.
- **Saturation:** the load on your network and servers.

## SRE and Monitoring

In SRE practices, the major responsibilities of software engineering and operations are to build and improve service reliability and to do so in the most efficient, toll-free way. To improve service reliability, SRE teams generally proactively monitor their service(s) to identify areas that need closer inspection and potential engineering time to improve.

By proactively monitoring, the team can make a decision on which issues are important and need prioritization. This creates a roadmap for improving the service. . The team can also find issues that are less serious and could go on a backlog of items to inspect during regular working hours.

**Monitoring is a critical part of SRE practices.** It's often the starting point for managing the overall system and services reliability. With dashboards of reports and charts, the team can keep an 'eye out' for anything unusual. With aggregated data updating dashboards in real-time, one can determine if the 4 golden signals are all green. Monitoring data can be tracked through code pushst to see how each release affects your service.

## Monitoring SRE's Golden Signals

The dilemma of complex distributed systems is that despite being complex, they should be easy to monitor. However, in a complex microservice architecture, it's difficult to identify the root cause of an issue as different technologies use different components that require expert oversight.

The golden signals can help consolidate the data received from your many microservices into the most important factors. By reflecting on the most foundational aspects of your service, the four golden signals are the basic bulding blocks of an effective monitoring strategy. They improve the time to detect (TTD) and the time to resolve (TTR).

## Latency (Request Service Time)

Latency is the time it takes a system to respond to a request. Both successful and failed requests have latency and it's vital to differentiate between the latency of successful and failed requests. For example, an HTTP 500 error, triggered because of a connection loss to the database might be served fast. Although, since HTTP 500 is an error indicating failed request, factoring it into the overall latency will lead to misleading calculations. Alternatively, a slow error can be even worse as it factors in even more latency. Therefore, instead of filtering out errors altogether, keep track of the error latency. Define a target for a good latency rate and monitor the latency of successful requests against failed ones to track the system's health.

## Traffic (User Demand)

Traffic is the measure of how much your service is in demand among users. How this is determined varies depending on the type of business you have. For a web service, traffic measurement is generally HTTP requests per second, while, in a storage system, traffic might be transactions per second or retrievals per second.

By monitoring user interaction and traffic in the service, SRE teams can usually figure out the user experience with the service and how it's affected by shifts in the service's demand.

## Errors (Rate of Failed Requests)

Error is the rate of requests that fail in any of the following ways:

- **Explicitly:** for example, HTTP 500 internal server error.
- **Implicitly:** for example, HTTP 200 success response coupled with inaccurate content.
- **By policy:** for example, as your response time is set to one second, any request that takes over one second is considered an error.

SRE teams can monitor all errors across the system and at individual service levels to define which errors are critical and which are less severe. By identifying that, they determine the health of their system from the user's perspective and can take rapid action to fix frequent errors.

## Saturation (Overall Capacity of the System)

Saturation refers to the overall capacity of the service or how "full" the service is at a given time. It signifies how much memory or CPU resources your system is utilizing. Many systems start underperforming before they reach 100% utilization. Therefore, setting a utilization target is critical as it will help ensure the service performance and availability to the users.

An increase in latency is often a leading indicator of saturation. Measuring your 99th percentile response time over a small time period can provide an early indicator of saturation. For example, a 99th percentile latency of 60 ms indicates that there's a 60 ms delay for every one in 100 requests.

## What Should You Do with the Golden Signals?

What makes the golden signals "golden" is that they measure things that represent the most fundamental aspects of your service's functions. Monitoring large systems gets complicated because there are too many components to monitor, which means more issues and more alerts. To allow developers to focus on other projects, the maintenance burden on the engineers should be minimal.

The golden signals are mainly used for:

- **Troubleshooting various components of the system to find the root cause and fix the problem.**
- **Alerting the response team about an incident when the signals drop too low, so they can identify the problem and work towards its remediation.**
- **Capacity planning to consistently monitor and improve things over time.**

The broad idea is to use your current alerting methods on the signals and track the progress. However, it's harder to alert using the golden signals as they don't exactly have a static alerting threshold. Ideally, you should use static thresholds (such as high CPU usage and low memory) but set them realistically to avoid false alerts. For example, the latency of more than 10 seconds, error rates over three per second, etc.

Basic alerts normally compare the threshold against the average values, but we recommend using percentile or median values. Median values are less sensitive to outliers (big and small) and they reduce the probability of false alerts. In contrast, percentiles show how significant a given value is.

## Anomaly Detection

Basic alerting is good enough in normal circumstances, but ideally, you want to use anomaly detection to catch unusual behavior, fast. For example, if your web traffic is 5 times higher at 3 am or drops down to zero in the middle of the day. Besides catching anomalies, it also sets tighter alerting bands to find issues faster than the static thresholds.

In theory, anomaly detection sounds easy enough, but it can be challenging. Since it's a fairly new concept, few on-premise monitoring systems currently provide the option. Tools like Prometheus, InfluxDB, DataDog, and SignalFx are some of the tools that offer anomaly detection.

## Tools to Monitor the Golden Signals

Selecting monitoring tools to monitor the golden signals is another critical step in the SRE journey. You can choose between open-source tools and paid tools depending on your specific needs. Both open-source and paid monitoring systems come equipped with dashboards for default metrics and you can also define alerts and notifications.

## Open-source Monitoring Tools

Open-source monitoring tools are a great option if you have a limited tooling budget. In the open-source tools, the source code is accessible to the user, which means that they can customize it to their needs and integrate it into their system. However, customization is not simple and requires time and domain knowledge. Finally, the security, availability, and updates of the tool are also your own responsibility.

A few great open-source monitoring tools include:

- Nagios
- Zabbix
- Prometheus + Grafana
- Cacti
- Icinga
- Netdata

## Managed Monitoring Tools

On the other hand, managed tools come at a cost, but also offer robustness that's missing in the open-source monitoring tools. Here, you're not responsible for the security, updates, and availability of the monitoring system and also get professional support for their integration.

Some popular managed monitoring tools are:

- Datadog
- Prometheus
- AppDynamics
- New Relic
- Pingdom
- Honeycomb

## Beyond The Golden Signals

Golden signals are a great first step towards understanding and improving incident response. However, many organizations are employing proactive measures to learn more about their system. That includes running simulations to test their system and prepare engineers for various scenarios.

These techniques are a great way for SRE teams to learn more about the system and use the information to make it even more reliable.

## Chaos Engineering

Chaos engineering is a discipline that involves running experiments on a system to identify its weak spots and potential points of failure. Netflix practices creating failures using the **Chaos Monkey** - a tool invented by Netflix in 2011 to test the resilience of its IT infrastructure. It works by terminating random VM (virtual machines) instances and containers running in the production environment. This allows teams to see what would happen if these failures really occurred and practice their responses.

## Game Day

Gameday is another technique that involves simulating a failure event to test the system, processes, and the team's response. Unlike chaos engineering, game days are geared towards understanding the people and helping them prepare for big failure events. It's used by many tech giants including Amazon to improve incident response.

## Synthetic User Monitoring

Synthetic monitoring is the practice of monitoring your application by simulating users. It enables teams to create artificial users and simulate user behavior to determine their behavior flows. That way, teams can learn more about how the system responds under pressure.

## How can Blameless Help with the Measurement of the Golden Signals?

Blameless can help you make the most out of your monitoring systems and reach your reliability goals. To learn more, [request a demo](#) or sign up for the newsletter below.

## Get the latest from Blameless

Receive news, announcements, and special offers.

Email Address

Subscribe

## BLAMELESS

Optimize reliability of software and resilience of teams.



### Product

Product Overview

Incident Resolution

Incident Retrospectives

Reliability Insights

SLOs & Error Budgets

### Company

About Us

Newsroom

Events

Careers

Contact Us

### Learning

Blog: Failure Is Inevitable

SRE Industry Leader Perspectives

SRE vs. DevOps

Resources Library

The Complete Guide to SLIs, SLOs, and Error Budgets