



HAMIDREZA
MADANI



سیگنال طلایی مانیتورینگ

برای Tech Lead ها

در کتاب Google's Site Reliability Engineering (SRE) handbook این سیگنال‌ها به عنوان پایه‌ی Observability (قابلیت مشاهده‌پذیری) معرفی شده‌اند.



زمان مطالعه: ۳ دقیقه



HAMIDREZA
MADANI

در کتاب SRE (Site Reliability Engineering) گوگل، این سیگنال‌ها به‌عنوان پایه‌ی Observability (قابلیت مشاهده‌پذیری) معرفی شده‌اند. طبق تجربه‌ی من، طراحی برای Production فقط نوشتن کد بهینه یا Scale کردن سرویس‌ها نیست بلکه همه‌چیز از «دید داشتن» یا همان Observability شروع می‌شود.

بدون دید، شما در تاریکی حرکت می‌کنید.





HAMIDREZA
MADANI

Latency

زمان پاسخ‌گویی

چیست: مدت زمانی که طول می‌کشد تا یک درخواست سرویس‌دهی شود. باید هم Latency موفق (مثلاً پاسخ OK 200) و هم Latency شکست‌خورده (Timeout یا خطای 5xx) را اندازه‌گیری کنید.

چرا مهم است: اولین چیزی که کاربر حس می‌کند همین است. یک اپلیکیشن سریع می‌تواند بسیاری از مشکلات را پنهان کند، اما یک برنامه کند حتی اگر درست کار کند، اعتماد کاربر را از بین می‌برد.





HAMIDREZA
MADANI

چطور پیاده سازی کنیم:

Generate: در Endpoint های سرویس، زمان پاسخ هر Request را ثبت کنید. بیشتر Framework ها مثل Spring، ASP.NET Core، Express یا net/http در Go این قابلیت را با Middleware دارند.

Ship: این داده ها را به یک سیستم مانیتورینگ مثل Prometheus، Datadog یا New Relic بفرستید.

Use: به جای Average روی Percentile ها (مثل P95 یا P99 Latency) Alert بگذارید، چون Outlier ها کاربر واقعی را آسیب می زنند.





HAMIDREZA
MADANI

Traffic

حجم ترافیک

چیست: میزان بار یا تقاضا روی سیستم، مثل
Requests per second، Queries per
second یا Messages per second.

چرا مهم است: Traffic کمک می‌کند الگوهای
بار، نیاز به Capacity و میزان استفاده‌ی واقعی
از سرویس را درک کنید.





HAMIDREZA
MADANI

چطور پیاده سازی کنیم:

Generate: تعداد Request ها، Query ها یا Message های عبوری از API، Database یا Queue را ثبت کنید.

Ship: تعداد تولید شده را به عنوان Metrics در یک سیستم Time-Series ذخیره کنید.

Use: ترافیک را با Latency و Error ها مقایسه کنید. مثلاً یک جهش ناگهانی در Traffic می تواند Latency را بالا ببرد.





HAMIDREZA
MADANI

Errors

نرخ خطا

چیست: درصد یا نرخ درخواست‌هایی که به شکست منجر می‌شوند، از جمله خطاهای 5xx، Timeoutها یا مشکلات منطقی در اپلیکیشن.

چرا مهم است: Errorها مستقیماً بیانگر درد کاربر هستند. قوی‌ترین سیگنال برای اینکه "چیزی خراب است" همین است.





HAMIDREZA
MADANI

چطور پیاده سازی کنیم:

Generate: لاگ‌های ساختارمند برای خطاها و Metrics مرتبط تولید کنید. Exception ها را فقط لاگ نکنید بلکه آن‌ها را دسته‌بندی کنید (Client Error، Server Error، Dependency Failure).

Ship: خطاها را به یک Log Aggregator مثل ELK، Splunk یا OpenTelemetry بفرستید و Metrics را به ابزار مانیتورینگ.

Use: روی Error Rate به صورت درصدی از کل ترافیک Alert بگذارید، نه به صورت عدد مطلق.





HAMIDREZA
MADANI

Saturation

میزان پرشدن منابع

چیست: شاخص ظرفیت سیستم یعنی اینکه منابع تا چه حد «پر» هستند. مثل CPU Usage، Thread Pool، Memory Usage، طول Queue یا Connection های Database.

چرا مهم است: سیستمی که منابعش نزدیک پر شدن است شکننده می شود. شاید هنوز پاسخ دهد، اما یک افزایش کوچک بار می تواند باعث Fail شدن شود.





HAMIDREZA
MADANI

چطور پیاده سازی کنیم:

Generate: از طریق VM، Kubernetes، Monitoring یا APM Agent ها Metrics زیر ساخت را جمع آوری کنید.

Ship: از Exporter هایی مثل Node Exporter یا cAdvisor یا سرویس های مانیتورینگ Cloud استفاده کنید.

Use: به جای شاخص های عقب مانده (مثل CPU وقتی به ۱۰۰٪ رسید) روی شاخص های پیش رو مثل طول Queue یا Connection Pool Utilization نظارت کنید.





HAMIDREZA
MADANI

به عنوان یک Tech Lead:

هر وقت تیم خواست یک سرویس جدید طراحی کند، اولین سوال تو نباید این باشد که کد را با چه Framework ای می‌نویسید؟

بلکه این است که چطور می‌خواهید Latency، Traffic، Errors و Saturation را ببینید؟

اگر از روز اول این چهار سیگنال را طراحی نکنید، بعدها هزینه‌اش را با Outage، Firefighting و مشتریان ناراضی خواهید پرداخت.

به عنوان یک Tech Lead، مسئولیت ما این است که Observability را از همان ابتدا در فرهنگ و طراحی سیستم بگنجانیم.





HAMIDREZA
MADANI

هر وقت تیم خواست یک سرویس جدید طراحی کند، اولین سوال تو نباید این باشد که کد را با چه Framework ای می‌نویسید؟

بلکه این است که چطور می‌خواهید Latency، Traffic، Errors و Saturation را ببینید؟

اگر از روز اول این چهار سیگنال را طراحی نکنید، بعدها هزینه‌اش را با Outage، Firefighting و مشتریان ناراضی خواهید پرداخت.

به‌عنوان یک Tech Lead، مسئولیت ما این است که Observability را از همان ابتدا در فرهنگ و طراحی سیستم بگنجانیم.





HAMIDREZA MADANI

Experienced Tech Lead

[linkedin.com/in/hrmadani](https://www.linkedin.com/in/hrmadani)