REFERENCE REVIEWED 29 OCT 2026

Drill: should this page someone?

Five alert cards to sort into page, ticket or nothing, then check against the burn-rate rules. About 30 minutes. Built on alerting principles and burn-rate alerting.

01

How to run this drill

- One facilitator and two to six participants. Print the page: give out the first sheet, keep the facilitator notes on the second.
- Read the handout aloud, give the group 15 minutes, then go through the notes card by card. Disagreement is useful.
- No scores and no record of who said what. Finish with: what surprised you, and which card would page in our real set-up today?

02

Handout

The service has a 99.9% availability SLO over 30 days and the SRE Workbook's burn-rate alerts: page at 14.4× over 1 hour and 6× over 6 hours, ticket at 1× over 3 days, each with a short window.

It is 03:00. For each card, decide: page the on-call engineer now, open a ticket, or do nothing. Then say which alert rule, if any, would actually fire, and when.

1. Checkout requests started failing at 2% 10 minutes ago and are still failing at that rate.
2. Search has failed 0.3% of requests, steadily, for 2 days.
3. One of 40 web hosts is at 97% CPU. Latency and errors are normal.
4. The primary database's disk will be full in about four hours at the current growth rate.
5. Login failed every request for one minute, then recovered by itself.

FACILITATOR NOTES

1. **Page, but not yet.** 2% is a 20× burn: the whole budget would be gone in 1d 12h. The 14.4× page fires after 43m 12s, so nothing has fired at 10 minutes. Ask: is waiting 43m 12s acceptable? That delay is the price of not paging for short blips.
   `1.44% ÷ 2% × 1 h = 43m 12s`
2. **Ticket.** A 3× burn is under the 6× page, so no page fires. The ticket fires once the 3-day average passes 0.1%.
   `0.1% ÷ 0.3% × 3 d = 1d; spent so far: 20% of the budget`
3. **Nothing, or a dashboard.** No user impact: it is a cause, not a symptom. Ask what would make it page-worthy.
4. **Page.** The SRE book says to worry about causes only when they are very definite and very imminent, and treats a database disk filling in four hours as saturation worth paging for. Ask whether automation could have handled it.
5. **A page fired; was it useful?** 100% errors cross the 14.4× threshold after 51.8s, ignoring scrape and evaluation delays, so someone was paged just as login recovered. The 5-minute window clears the alert 5m 56s after the start. It spent 2.31% of the budget: worth a look in the morning, perhaps not at 03:00.

> Discuss: which of these would page in your real set-up today, and which of those you would want to?

Sources: Google, Site Reliability Engineering, 1st ed., ch. 28 “Accelerating SREs to On-Call and Beyond” (disaster role playing, “Wheel of Misfortune”, in which a game master plays out an outage live while one or two on-callers ask questions and act; this drill adapts that format as a discussion from a handout rather than following it) and ch. 6 “Monitoring Distributed Systems”; The Site Reliability Workbook, ch. 5 “Alerting on SLOs”. The scenario is invented for practice.