THE ONE RULE

AI reads and reasons. Humans run commands.

Let the model summarize, hypothesize, and draft commands. You read every command and you run it. Never let AI execute against production.

1 · Classify severity → priority

SEVERITY	PRIORITY	LOOKS LIKE
critical	P1	Customer-impacting outage, data-loss risk, full unavailability
warning	P2	Degraded service, rising errors, growing backlog
info	P3 / P4	Capacity trend, maintenance signal, non-urgent notice

2 · Work the command ladder — safest first, stop at diagnosis

SAFE · READ-ONLY

`kubectl get · journalctl · ss · ip · cat · grep · promtool query`

CAUTION · SMALL CHANGE

`kubectl exec · docker exec · edit non-prod config`

DESTRUCTIVE · LAST RESORT

`restart · delete · scale-to-zero · firewall · migrate · restore`

3 · Copy-paste prompts

SUMMARIZE THE FIREHOSE

```
"Here are the alerts, logs, and recent changes for an active production incident. Summarize what's happening in 5 bullets, list the top 3 hypotheses ordered by likelihood, and for each give the single read-only command to confirm or rule it out. Suggest no command that changes state."
```

CORRELATE WHAT CHANGED

```
"The spike started at <TIME> UTC. Here is the deploy and config-change history for the last 6 hours. What changed closest to that time, and by what mechanism could it cause this symptom?"
```

DRAFT COMMS

```
"Write a customer-facing status-page update for a degraded-<SERVICE> incident — no jargon, no root-cause speculation, ~3 sentences. Then a one-line internal update with current severity and what we're checking."
```

4 · Never

- × Paste secrets, tokens, or customer data into a model
- × Trust a confident command without reading it
- × Let it invent metric names — give real ones or placeholders
- × Run the "obvious" destructive fix before confirming cause