StackGen

FAILURE MODE TAXONOMY

SRE FAILURE MODE TAXONOMY 30 FAILURE MODES • 8 FAMILIES

What keeps breaking— and how often?

Classify incidents consistently and uncover the failure patterns shaping your reliability risk.

DERIVED FROM THE STATE OF RELIABILITY 2026 ANALYSIS OF:

27K+	178K+	392	2018-26
classified incidents	status-page entries	companies	date range

A practical reference for SRE, platform and reliability teams.

SRE Failure Mode Taxonomy

SSOR 2026 QUICK REFERENCE • TAXONOMY V0.9 • 30 ACTIVE FAILURE MODES • 16 ROOT CAUSES • SOFTWARE & ONLINE SERVICES ONLY

HOW TO USE

1. Classify every incident

At post-incident review, tag each resolved incident with an FM code and the matching Root Cause (RC) code. Build this into your runbook so it's automatic.

2. Track your FM mix over time

Your FM distribution reveals your reliability archetype — Dependency-Driven (FM-01 dominant), Velocity-Driven (FM-09/10), Scale-Driven (F2), or Data-Integrity-Driven (F4). The mix tells you where to invest.

3. Watch Family 6 proactively

FM-17 (AI output quality), FM-19 (agentic failures), and FM-33 (GPU fleet) are early warning signals for AI-induced risk. These are growing fast and rarely appear in traditional SRE playbooks — track them now, before they become your dominant pattern.

Failure mode families – Active catalog • % = Share of ~27,000 classified incidents (SSOR 2026 corpus)

F1 Network & Connectivity — 54%

FM-01 Cross-Org Cascade — 52%
Upstream provider failure propagates to dependent consumers. Modal failure mode corpus-wide — 3.2× slower recovery than internal failures.

FM-02 Connectivity Disruption
Network partition, DNS failure, BGP misroute, or routing loop.

FM-03 AZ / Region Failure
Single availability zone or cloud region becomes unavailable.

FM-04 CDN / Edge Failure
Edge layer becomes unavailable or serves stale / incorrect content.

FM-05 DDoS / Traffic Flood
External attack or organic surge overwhelms capacity.

F2 Capacity & Resource — 8%

FM-06 Memory Leak / OOM
Process consumes increasing memory until crash or OOMKilled.

FM-07 Disk / Storage Exhaustion
Disk fills; writes fail, causing cascading read errors and crashes.

FM-08 CPU / Thread Saturation
Worker pool or CPU pegged; tail latency degrades under sustained load.

FM-13 Resource Exhaustion — 6%
Queue backup, connection pool depletion, or rate-limit hit.

FM-25 Autoscaling Pathology
Autoscaler over- or under-scales, causing oscillation or capacity gap.

F3 Change-Induced — 16%

FM-09 Deploy-Induced Regression — 9%
Code or config change introduces a new defect at deployment time.

FM-10 Config-Induced Failure — 4%
Misconfiguration causes outage without an application code change.

FM-11 Schema / Data Migration Failure
DB migration corrupts data or blocks reads and writes mid-rollout.

FM-12 Untracked Vendor Change
Upstream changes API behavior or defaults without customer notice.

FM-35 In-Flight Compatibility Break — 2%
Two incompatible versions co-exist during phased rollout or migration.

F4 Data & State — 12%

FM-21 Phased Data Recovery — 8%
Service recovers but data requires multi-step reconciliation or replay.

FM-26 Silent Data Corruption — 3%
Data is written but incorrect; no immediate error raised at write time.

FM-27 Stale Cache / Stale Read
Cached or replicated data served after the authoritative source changes.

FM-28 Runaway Write / Amplification
Write storm fills storage or overwhelms downstream systems.

F5 Platform & Control Plane — 4%

FM-29 Service Discovery / LB Failure
Traffic misrouted or backends unreachable via load balancer or discovery.

FM-30 Control Plane / Orchestration Failure — 3%
Kubernetes, Terraform, or similar orchestration fails to schedule or deploy.

FM-31 Natural Disaster / Physical Event
Data center power, cooling, or physical access disruption.

F6 AI-Specific — 2%

FM-17 AI Service Output Quality Degradation — 1%
Model serves degraded, hallucinated, or empty responses. Spans LLMOps / MLOps / DataOps / DevOps.

FM-19 Agentic / Tool-Use Failure — Provisional
Autonomous agent stalls, loops, or takes destructive action in production.

FM-33 GPU/Accelerator Fleet Heterogeneity — Provisional
Mixed GPU generations cause training divergence or inference inconsistency.

F7 Security & Auth — 1%

FM-14 Credential / Auth Failure
Auth tokens, TLS certificates, or API keys expire, rotate, or are revoked.

FM-15 Security Incident Response
Breach, active exploit, or disclosed CVE triggers emergency IR procedure.

FM-16 Supply Chain / Dependency Compromise
Malicious or tampered package introduced via a dependency.

F8 Hardware — 5%

FM-23 Hardware Failure — Node — 4%
Single physical device (disk, NIC, DIMM) fails; affects pods or VMs on that host.

FM-24 Hardware Failure — Systemic
Fleet-wide hardware defect or vendor recall; multiple nodes fail simultaneously.

WATCHLIST

Pending promotion — zero primary hits in the classified corpus:

- FM-32 AI Training Pipeline Failure
- FM-34 High-Perf Network Fabric Failure

Turn incident history into reliability insight

Classify every resolved incident

Assign the most relevant Failure Mode code during the post-incident review.

EXAMPLES:

- **FM-01** – Cross-Organization Cascade
- **FM-09** – Deploy-Induced Regression
- **FM-10** – Config-Induced Failure

Use a consistent primary code across services and teams.

Track your failure-mode mix

Review the distribution monthly or quarterly to identify whether your environment is primarily:

- Dependency-driven
- Velocity-driven
- Capacity-driven
- Data-integrity-driven
- AI-risk-driven

The pattern shows where reliability investment is most needed.

Watch emerging risks

Track FM-17 (AI output quality), FM-19 (agentic tool-use failures), and FM-33 (GPU fleet inconsistencies) separately.

These patterns may not appear in traditional SRE classification systems — but they can become increasingly important as AI workloads enter production.

PRACTICAL PROMPT

"Which three failure modes accounted for the greatest share of incidents during your last quarter?"

From the State of Reliability 2026

The Failure Mode Taxonomy is one part of a connected incident-learning framework developed through the analysis of public reliability data.

Failure Mode	What failed?
Root Cause	Why did it fail?
Remediation	What was done about it?
Context Atlas	What information was required to diagnose and act?

Use the framework to create a more consistent language for postmortems, incident trends, and reliability investment.

DOWNLOAD THE FULL FRAMEWORK

RC Cheat Sheet
Root Cause Taxonomy

RM Cheat Sheet
Remediation Taxonomy

Context Atlas
Diagnostic Context Reference

Download The State of Reliability 2026 full report.

Follow StackGen for the report release and additional reliability research.

stackgen.com/state-of-reliability-2026