Infrastructure and security: design for the failure you will have

21 October 2026 · 6 min read · blogs.iictp.org/design-for-the-failure-you-will-have

Systems fail, and credentials leak, and people make mistakes. The fourth TPBOK domain, Infrastructure & Security, starts from that fact. The goal is not a system that never breaks, but one where a break is small, visible and quick to recover from.

TL;DR The short version

- The goal is not a system that never breaks, but one where a break is small, visible and quick to recover from.
- Limit the blast radius, set reliability targets with error budgets, and collect metrics, logs and traces so you can see failures.
- Practise recovery before you need it: a backup you have never restored is a hope, and incident reviews should fix the system, not blame the person.

Read online, listen to it, or change the text size: https://blogs.iictp.org/design-for-the-failure-you-will-have

Old principles, still the right ones

In 1975 Jerome Saltzer and Michael Schroeder set out design principles for protecting information. Most of them still read as practical advice today.

- **Least privilege.** Give every person and program only the access it needs.
- **Fail-safe defaults.** Access is denied unless explicitly granted.
- **Complete mediation.** Check every access, every time.
- **Economy of mechanism.** Simpler designs are easier to verify.
- **Open design.** Security should not depend on the design being secret.
- **Separation of privilege.** Sensitive actions need more than one condition.

Zero trust, now described in NIST SP 800-207, is these ideas applied to modern networks: do not trust a request because of where it comes from, verify it.

Limit the blast radius

Assume something will go wrong and decide in advance how far the damage can spread. Separate environments, narrow credentials that expire, and isolated components mean one failure stays one failure. Infrastructure defined as code and kept in version control makes environments reproducible, so you can rebuild instead of repair.

Measure reliability and spend it deliberately

The Site Reliability Engineering practice from Google describes service level objectives, a target for how reliable a service should be, and the error budget, the amount of unreliability you can afford before you stop shipping new features. It turns an argument about risk into a number both product and engineering can see.

> Recovery speed often matters more than avoiding every failure. Mean time to restore is something a team can improve every month. Failure-free operation is not.

You cannot fix what you cannot see

Observability rests on three kinds of signal: metrics that show trends, logs that record events, and traces that follow a request across services. Together they let you ask a question you did not plan for when an incident starts.

Practise recovery before you need it

- Write down how much data you can afford to lose (recovery point objective) and how long you can be down (recovery time objective).
- Test restores from backup. A backup you have never restored is a hope.
- Run a game day: break something in a safe environment and watch how the team responds.

After the incident

Hold a review that asks what in the system made the mistake easy, not who made it. Record what you will change, assign an owner, and follow up.

Infrastructure & Security is 20% of the ATP exam and 18% of the CTP exam. The specialist Cloud Architecture Practitioner (CAP) and IoT and Embedded Systems (IESP) exams give it 34% and 30%. See exam facts.

Sources and further reading

1. Beyer, B., Jones, C., Petoff, J. and Murphy, N. R. (eds.) (2016). Site Reliability Engineering. O'Reilly Media.
2. Rose, S., Borchert, O., Mitchell, S. and Connelly, S. (2020). Zero Trust Architecture. NIST Special Publication 800-207.
3. Saltzer, J. H. and Schroeder, M. D. (1975). "The Protection of Information in Computer Systems." Proceedings of the IEEE 63(9).

Published by IICTP, an independent technical credential body, on 21 October 2026. The current version of this post is at https://blogs.iictp.org/design-for-the-failure-you-will-have. More guides: blogs.iictp.org.