Infrastructure

What a DR test should actually prove.

A green backup report proves a job ran. It doesn't prove you can run the business on what it saved. A disaster recovery test should answer five questions, with a stopwatch running.

Toltec engineering | 23 September 2026 | 4 minute read

The short version

- Agree a recovery time and a recovery point for each system, with the business rather than IT alone.
- Prove the data, the order, the access, the time and the people.
- Restore into an isolated network and time every stage.
- Write down what happened, and fix it before the next test.
- Test the failback, not just the failover.

Start with two numbers

For every important system, agree a recovery time objective, which is how long the business can be without it, and a recovery point objective, which is how much recent data it can afford to lose.[1] They are business decisions, so agree them with the people who run the business: a finance system might need recovery within four hours with no more than fifteen minutes of data lost; an archive might allow three days for recovery and one day of data loss. Without these numbers, a test can only tell you what happened, not whether it was good enough.

The five things a test must prove

The data comes back, and it is correct

Restore into an isolated network, so nothing collides with production. Check integrity properly: run consistency checks on databases, such as DBCC CHECKDB for SQL Server, open a sample of files, and have someone who uses the system confirm that recent records are there.

It comes back in the right order

Systems depend on each other. Identity and DNS usually come first, then databases, then applications, then remote access. Write the order down and follow it. In our experience, most failed tests are dependency failures rather than backup failures.

People can reach it

A restored server nobody can use hasn't recovered anything. Check DNS changes and how long old records stay cached, certificates, VPN or remote access, licence activation, and multi-factor authentication when the identity system you normally rely on is the thing that's down.

It fits inside the recovery time

Time every stage, from the decision to invoke the plan to the first user confirming that it works. Compare the total with the objective, not with the backup product's restore speed. If it does not fit, investigate the design, recovery sequence, runbook and the team's readiness.

It works without the one person who knows

Have someone other than the usual engineer follow the runbook. Make sure the credentials for the backup system are reachable when the domain and the password manager are both unavailable, and that at least one copy of the backups can't be deleted by an attacker holding admin rights.

Choose the right kind of test

Test	What it proves	How often
Tabletop walkthrough	Roles, contacts and decisions, and whether the runbook still matches the estate.	Twice a year
Component restore	One system's data and timings, restored to an isolated network.	Monthly, rotating through systems
Full failover	The order, the access and the total time for a complete service.	Once a year, and after major change

Write down what happened

Record the date, the scope and the people involved; the start and end time of each stage; everything that went wrong; and the fix, owner and deadline for each problem. Compare the result with the objectives. That record is what insurers, auditors and your own board will ask for, and it is the best starting point for the next test.

Failures we see most often

- Licence keys or activation servers missing from the recovery plan.
- Certificates that have expired in the recovery copy.
- Applications with server names or IP addresses written into their configuration.
- Service account passwords changed in production but not in the runbook.
- The runbook itself stored on a server that needs recovering.
- DNS records cached for a day, so users keep reaching the old system.
- Nobody has tested failing back.

What we would do differently

Plan the failback as carefully as the failover. After a real incident, getting back to normal often takes longer than the recovery itself, and it is the part most plans leave blank.

References

1. SP 800-34 Rev. 1, Contingency Planning Guide for Federal Information Systems NIST, Swanson and others, May 2010

Read the current edition: toltec.io/note-dr-testing.html