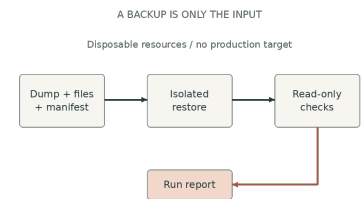


SYSTEMS TOOLING

Backup restore checks

Test whether a backup can be restored, not just whether a file exists.



PROPOSED ARCHITECTURE

Recovery tool brief; backup contract to define

THE PROBLEM

A successful backup job says little about the restored application. This command-line tool restores a PostgreSQL dump and an uploaded-file archive into disposable local resources, checks them together, and produces a report tied to the exact backup set.

SCOPE

- Use a Python CLI to create an isolated database container and temporary file tree.
- Verify manifests, checksums, referential integrity, and configured read-only application probes.
- Capture failures and timings, then remove only resources created by the run.

PLANNED DELIVERABLES

- Restore CLI with structured and human-readable reports
- Valid, corrupt, and mismatched backup fixtures
- Local runner configuration and a recovery drill guide

CONSTRAINTS

Inputs

A documented dump format and matching file manifest

Isolation

No production credentials or production restore target

Archives

Reject path traversal and unsafe symbolic links

Evidence

Report restore success separately from application checks

VERIFICATION PLAN

Damaged backup

Truncate a dump or remove a referenced file; report the specific failure.

Cleanup

Interrupt each stage; cleanup must leave unrelated files and containers intact.

Recovery fixtures

Restore a known dataset and compare its expected records and file digests.

NEXT STEP

Define the backup-set manifest and the application checks that matter.

The runner checks a defined backup format and application contract. Results apply to the tested backup set and environment.