Engineer

Implemented backups and disaster recovery as code

Configured database backup retention as infrastructure-as-code, then audited the recovery position and documented the restore procedure — naming the remaining gaps rather than leaving them to be found during an incident.

2025

backups as code, and the recovery position written down

Situation. A product that lives on its data can't afford to lose any, and "there are backups somewhere" is a hope, not a recovery plan. The only backup worth having is one you know restores, into an environment you know you can rebuild. The product had neither half written down.

Task. Get the recoverable position defined instead of assumed — the data, the environment around it, and an honest account of how far that actually reaches today.

Action. Backup retention is configured in the Bicep beside the database it protects, so point-in-time restore is a property of the template rather than a setting someone once clicked in a portal. The environment around it is defined as infrastructure-as-code too, which is the quiet half people forget: restoring a database into an environment you'd have to rebuild by hand from memory isn't really recovery. Then the position was audited and written up — the restore procedure, the drill that would measure it, and the gaps still open: geo-redundancy is switched off, and the recovery-time objective is proposed rather than measured, because no drill has been run yet.

Result. Recovery stopped being a vague reassurance and became a documented position with its gaps named. That reads less impressively than "disaster recovery: done", and it is worth considerably more — whoever touches it next knows what is covered, what isn't, and exactly which drill closes the difference. A named gap is one you can close; an unnamed one is discovered during an incident.

Bring this experience to your project

Use “Implemented backups and disaster recovery as code” as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. Somebody responsible for the thing the whole company rests on: it is up, it is backed up, and it is not quietly rotting.

Explore our services Get in touch →

Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Implemented backups and disaster recovery as code — backups as code, and the recovery position written down

Explore our services Get in touch →

Could you recover fast if it all went down?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Site Reliability Engineer. It draws on Databases, PostgreSQL, DevOps, Infrastructure, Cloud and Reliability & Backups. And while you are here, we are rather good at Database Administration (DBA), Infrastructure as Code and Backup & Disaster Recovery.

Get in touch Explore our services

One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

See the whole record

Define the improvement you need

Bring a representative dataset, the questions it must answer and the current bottleneck. We can agree how to measure quality, throughput and recovery.

See what it costs Get in touch →

Could you recover fast if it all went down?

Ask about the decisions behind “Implemented backups and disaster recovery as code”. You will speak with the people who can explain the work and assess your own project.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/portfolio/implemented-backups-and-disaster-recovery-as-code/