Engineer

Public services on a small server

Hosted the company’s public services on a single-core 512 MB host — a git forge, a web server for seven domains, Tor, two alternative-protocol servers and intrusion banning — treating 464 MB of usable memory as the design constraint; backup automation is a separate implementation.

September 2026

512 MB

running an entire company's estate

Situation. The company’s production host is a single-core cloud instance with 512 MB of memory, of which about 464 MB is usable, and 10 GB of disk. Everything the business runs in public sits on it: the web server terminating TLS for seven domains, the git forge, a Tor onion service, a Gemini server, a Gopher server, intrusion banning and separately implemented backup automation. The usual reaction to that list is to buy a larger box.

Task. The constraint had to be treated as an architectural input rather than a problem to spend money on, because the honest question was not whether a bigger machine would work but whether the design needed one.

Action. Memory became the argument that settled decisions. There is no monitoring agent, no metrics pipeline and no dashboard — reporting is a pull, seven commands that read the host and render Markdown, changing nothing and running only when asked. The supervisor is systemd rather than a second process manager layered on top of it, and the container plane is Quadlet units under the same supervisor rather than a daemon with its own. Web-panel platforms were ruled out at the design stage for the same reason. When the question came up of whether the host could carry an onion service, the answer came from a day of measured samples rather than an opinion: available memory never fell below about 310 MB of 464, swap sat at 2.6 per cent and the processor was 99.7 per cent idle.

Result. The recorded public-service setup ran on a machine that cost less per month than a lunch, and the design is better for the discipline rather than merely cheaper. What it cost is headroom for anything careless — mail is deliberately not on this box at all — it is written as an install scaffold awaiting a host of its own, because a mail server needs headroom this machine has already spent, and that is written down rather than discovered later.

Bring this experience to your project

Use “Public services on a small server” as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. The shape of a system worked out on paper first, so the parts fit and the cost is known before anyone writes code.

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. Public services on a small server — 512 MB running an entire company's estate

Explore our services Get in touch →

Paying for capacity you never use?

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 Platform Architecture, Solution Architecture, Infrastructure, Reliability & Backups, Linux & Servers, Performance Tuning and System Administration. And while you are here, we are rather good at Platform & Solution Architecture, Cloud Infrastructure & Migration, Infrastructure as Code and System Administration.

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

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Paying for capacity you never use?

Ask about the decisions behind “Public services on a small server”. 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/a-whole-company-on-one-small-server/