Engineer Software

Published a readable Tor onion mirror

Published the site as a Tor onion mirror on a readable address beginning engineer, with the key generated off the host so it never reached this repository or a laptop, and left it the one front door of four that is never counted.

September 2026

a fourth front door, and the one nobody counts

Situation. The site already answered on three protocols from one build. A reader who needed to reach it without revealing that they had reached it was still making an exit-node round trip to an ordinary address, which is a weaker thing than a front door of its own. The obvious objection to a fourth door is the firewall, and it does not apply: the anonymity daemon reaches the network by making outbound circuits, so nothing dials in, no port opens, and neither firewall layer has anything to synchronise.

Task. The mirror had to serve the same pages as the ordinary address, be published wherever the other mirrors are published, carry an address a person can read aloud rather than a random string, and never let the private key that is the address touch this repository or a laptop.

Action. The role installs the daemon from the distribution's own package, writes a configuration with the outbound proxy port switched off, and declines to overwrite an identity it did not generate — so a key placed by hand is simply used. A readable address cannot be chosen, only found: the address is the public key in base32, so a prefix is bought by generating keypairs until one of them happens to encode the letters wanted, and each character costs thirty-two times the one before it. The prefix that was bought reads as the company's own name. The address is a declared value in the inventory and every converge asserts the host's real identity file matches it, so a swapped key with a stale declaration fails the run rather than going quiet. The key is caught by both secret guards, after it was measured that the obvious pattern for a key file matches a deploy key and misses this one. Retiring the random address the mirror had shipped with was a staged cutover rather than a switch, so the published address kept working while the readable one was being published. The step that looked

like the last one was not: the key landed correctly and the converge reported no change, because nothing yet named the new directory, so the new address was never published at all — the role now fails on an identity on disk that no service names.

Result. The site has four front doors onto one build, and this is the only one that is never counted. Gemini requests and Gopher connections are tallied per day with no address and no path; the onion has no counter and no flag to add one, because this site deliberately leaves that audience uncounted, and the blindness is the price of the posture rather than a gap in the reporting. The mirror also paid for itself as a second opinion: its first readers were on a different browser engine, which does not implement the scroll-driven animation the footer relied on, so every one of them got the colophon painted across the front page. That was a real defect on the ordinary site, found by the audience that had no other way to report it.

Bring this experience to your project

Use “Published a readable Tor onion mirror” as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. A site that is quick, findable and editable by your own staff, without a bill for every change of wording.

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. Published a readable Tor onion mirror — a fourth front door, and the one nobody counts

Explore our services Get in touch →

Need an address readers can reach without being seen?

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: Network and DNS Engineer. It draws on Web Development, Infrastructure, Linux & Servers, Security and System Administration. And while you are here, we are rather good at Website Development & CMS, Infrastructure as Code, System Administration and Security & Access Management.

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 →

Need an address readers can reach without being seen?

Ask about the decisions behind “Published a readable Tor onion mirror”. 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://software.engineer.company/portfolio/published-a-readable-tor-onion-mirror/