



Learnings from Building a Managed Keycloak Offering

Andreas Müller-Niggemann, 08.10.2026

The shape of the next 30 minutes

adorsys

The situation

A customer wants Keycloak — run by us.

The roadmap

What topics do we need to take care of?

The outcome

A decision — we did not expect

About me

adorsys



Andreas Müller-Niggemann

Director Cloud Technologies

adorsys · Nuremberg

- Cloud Consulting and Strategy at adorsys
- 25 years spanning Software Development, DevOps, Product Owner SaaS & PaaS
- AWS, On-Prem, Kubernetes, CNCF, Automation, AI
- Running, Biking, Swimrun

Why I am on this stage

We were asked to run Keycloak for one customer. We said yes, and then had to work out what we had actually promised.

Everything in the next half hour is what that turned into: a project plan, technical options and a decision how we tackle it.

The numbers are ours.

Focus:

Making technology adoption and managed services work, technically and commercially

Customer request

adorsys

From: Markus Weber · Product Owner Tax, <CUSTOMER_NAME>

Subject: Quick one – Managed Keycloak?

Hi team, hope you're well! Quick question.

We love the Keycloak setup you built for our customer portal last year. Works great. But our ops team is fully booked with the current go-live efforts, and honestly nobody here really wants to own an identity server long-term.

So: could you just run it for us? Managed Keycloak, basically – **dedicated** for us. Shouldn't be a big deal, right? It's open source, it's already running, it's basically just a container. Keep it up, click “update” now and then, done.

A few small things:

- Available 24/7 — something like 99.9% would be great
- Keep our custom login page and the little authenticator plugin
- Compliance will have a few questions. Nothing dramatic, I think
- Budget is tight – Subscription on a user-basis would be great (MAU)
- Live by end of quarter — can you send a quick offer?

Honestly, this can't be that hard.

Cheers, Markus

Who asked

A product owner moving towards production and no team to run identity. Not an ops lead — nobody is volunteering for the pager.

What he is really asking for

Availability, patching, a custom extension, compliance answers and an exit — priced as a hosting fee.

Our disclosure

adorsys sells the buy side. Every number in this talk is ours, and it closes with the cases where we would tell you to build.

The Situation · Project or product?

adorsys

There are two honest answers to that email - and they build two different companies.

— same request, billed by effort or by subscription.

Option 1) Answer it as a project

Scoped engagement, day rates, one customer's requirements. Ends when the invoice is paid.

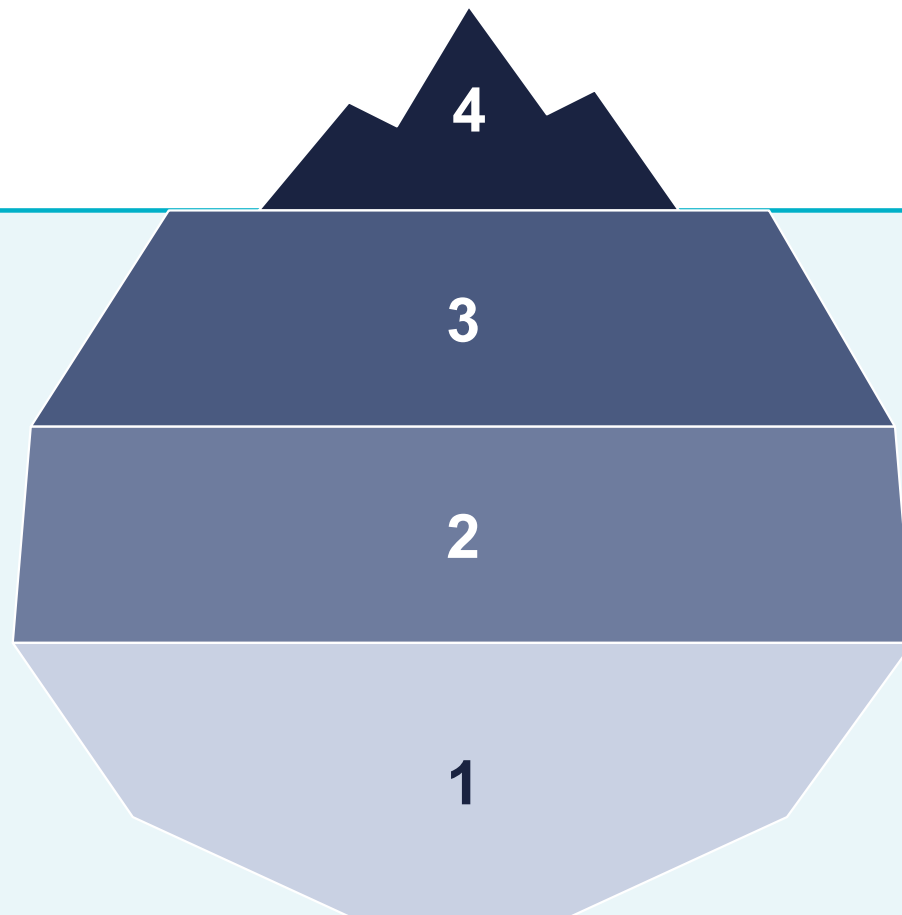
Option 2) Answer it as a product

A service definition, a fleet, an on-call rota, an audit trail. Ends when the company does.

Assumption: We are talking about a SaaS Product

Managed service · Customers see only the tip

adorsys



4 · IAM outcomes

Every login works, end to end. This is what the customer buys.

Below the waterline: what it takes

3 · Configuration

Realms, clients, flows and IdPs

2 · Operations

Availability, upgrades, backup, incidents

1 · Hosting

Infra Setup, the process runs, the OS is patched

Selling the tip means owning everything below it.

Building a service · 10 building blocks

adorsys

We assumed this was mostly a technical problem. Exactly one of these ten blocks is technical.

1 · Service definition

Hosting, operations, configuration or outcomes

2 · Platform & infrastructure

Clusters, Postgres, regions, capacity

TECH

3 · Automation & lifecycle

Config as code; upgrade pipeline before customer #2

4 · Security

Patch cadence, secrets, break-glass approval

5 · Operations

On-call, backup, restore, incident response

6 · Support

Triage first — most tickets are their OIDC config

7 · Compliance & legal

Evidence calendar, DORA, DPA, exit

8 · Commercial

Pricing model, SLA credits, contract terms

9 · Customer lifecycle

Onboarding, change requests, offboarding

10 · Organization

Roles, rota, who owns what at 3 a.m.

Building a service · 3 out of 10!

Ten blocks. Lets look into the first 3.

1 · Service definition

Scope: what the customer really wants

2 · Platform & infrastructure

Shared or dedicated per customer

3 · Automation & lifecycle

Build from code, survive every release

4 · Security

5 · Operations

6 · Support

7 · Compliance & legal

8 · Commercial

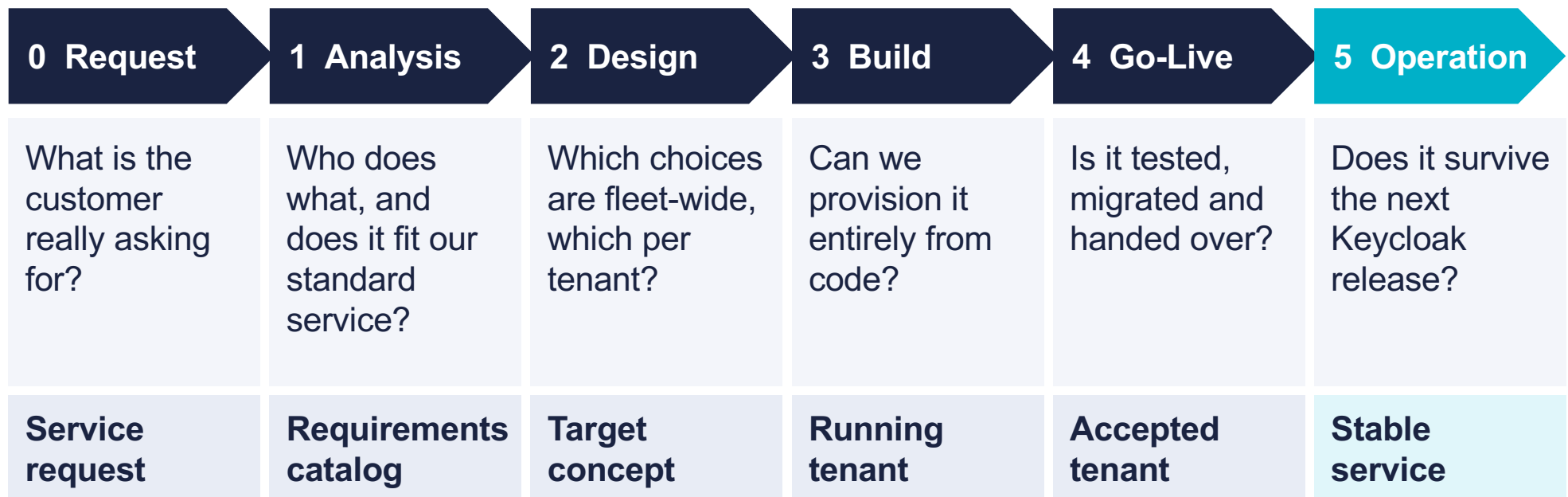
9 · Customer lifecycle

10 · Organization

adorsys

From request to rollout · The last step never ends

adorsys



Steps 0–4 run once per tenant. Operation repeats with every release.

A project ends at Go-Live. A product starts there.

1 · Analysis · First settle who does what

adorsys

What the email asks for

Provide us an IDP as a service (based on Keycloak)

Availability, patching, a custom extension, compliance answers and an exit.

Priced as monthly fee.

What the analysis has to settle

Identity sources & apps	Which IdPs, user stores and client apps connect, and who integrates them.
Tenancy & sizing	Does the tenant fit the fleet's isolation unit and capacity profile?
Responsibility split	Who owns realm config, custom SPIs and themes after Go-Live. 1/2/3rd Level
Compliance scope	Of course: NIS2, DORA, ISO27001

Gate: does this tenant fit the standard service?

If the answer needs exceptions, it is a project. Output of this step: the requirements catalog.

2 · Design · Decide once, for the whole fleet

adorsys

Decided once, for the fleet	Decided per tenant
Unit of isolation Realm or instance per tenant	Realm & client model Mapped onto the standard
Upgrade authority We set the release schedule	Login flows Chosen from supported flows
Customization policy What may run in our runtime	Federation & IdPs Customer's directories, our connectors
Security concept & SLA One standard, one price list	Theming Inside the customization policy

Every item that moves to the right turns the product back into a project.

Output of this step: the target concept, a tenant-specific instance of the fleet design.

Two ends of one spectrum · Product or project

adorsys

Highly standardized SaaS

One product, many customers

Customized managed application

One customer, their own setup



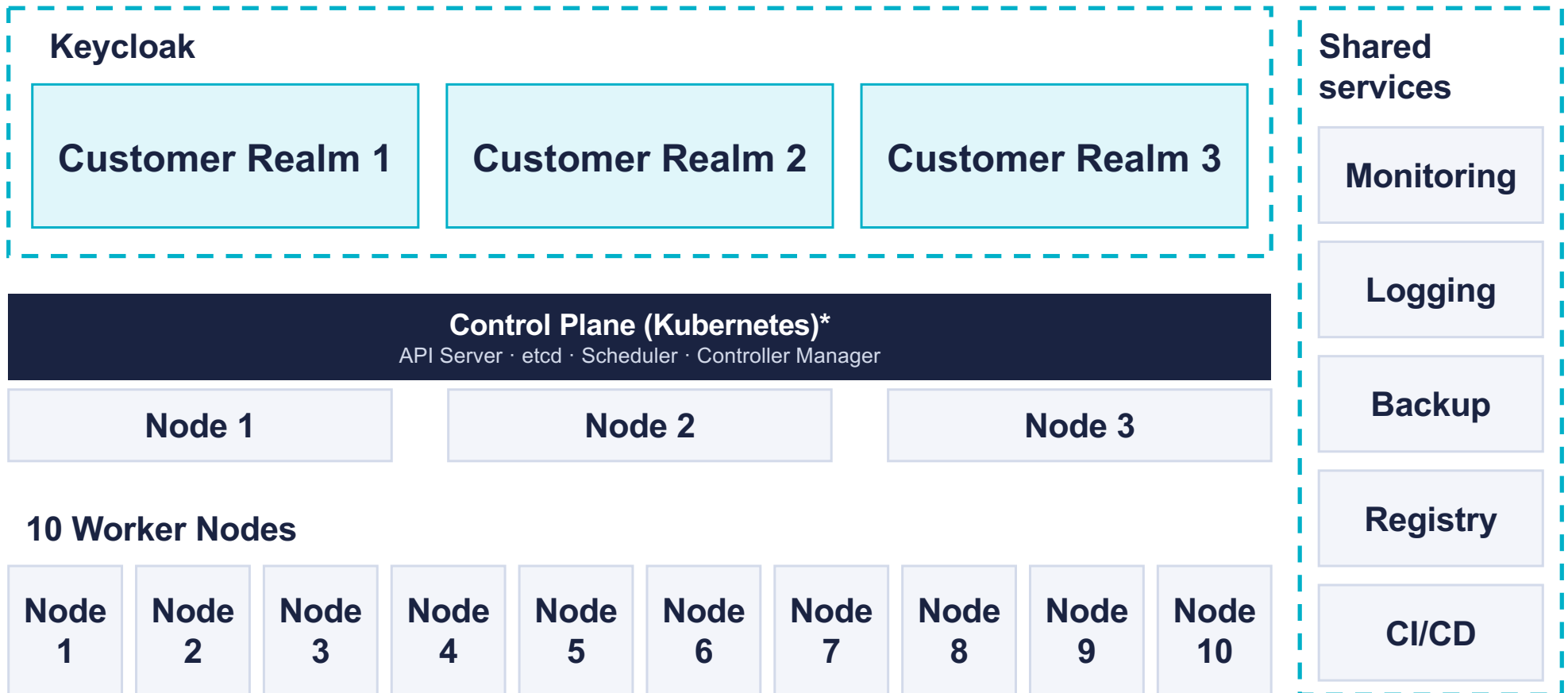
One version and schedule for all	Releases	Release timing agreed per customer
Self-service, from a standard template	Onboarding	A project per customer
Standard themes and configuration	Customization	Own login page and extensions
Shared cluster and shared services	Infrastructure	Dedicated stack per customer
Price per tenant or per user	Pricing	Priced on effort, set in the contract

Every customer-specific feature moves the offering to the right.

On the right, each customer brings its own release, test and support effort.

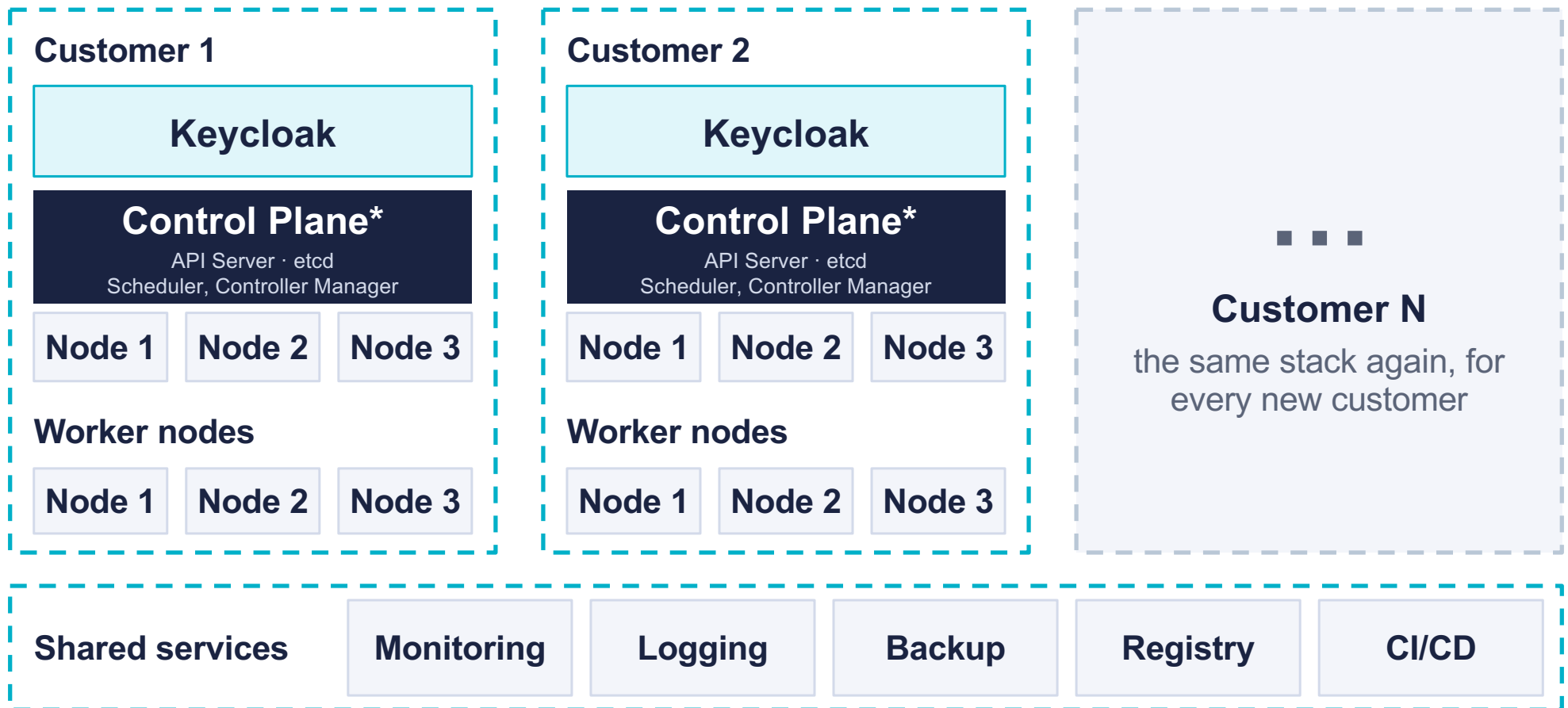
Infrastructure – Option 1 – Shared Infra

adorsys



* Control plane: a Kubernetes cluster running on 3 nodes.

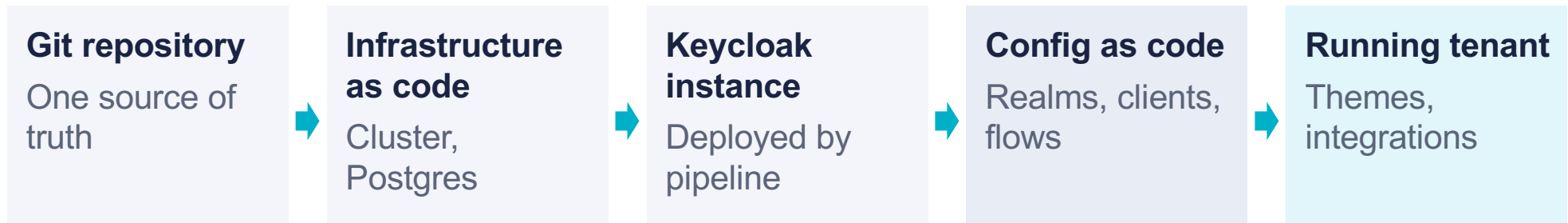
Infrastructure – Option 2 – Dedicated Infra



* Control plane: a Kubernetes cluster running on 3 nodes.

3 · Build · How we do it: deployed from code

adorsys



>50%

of respondents in the 2024 Keycloak community survey manage realm config with Terraform.

No update path

Plain JSON import and the realm-import Operator cannot cleanly update an existing realm.

Our deployment

Every deploy runs our pipeline. keycloak-config-cli applies the realm config from Git.

A change made by hand in the console is lost at the next upgrade.

Output of this step: a running tenant that can be rebuilt from Git at any time.

4 · Go-Live · Ownership moves to operations

adorsys

Three steps before the pager

1	Load & security tests Prove the tenant at expected peak before real users arrive.
2	User migration Users, credentials and IdP links move from the old system.
3	Runbook handover The project team hands over to on-call. The runbook is the contract.

Who owns what from day one

We own	• Infrastructure and Kubernetes • Keycloak process, HA, backup • Upgrades and release schedule • Incident communication
Customer owns	• Clients and OIDC integrations • Users, credentials, IdPs
Shared	• Realm configuration • Custom SPIs and themes

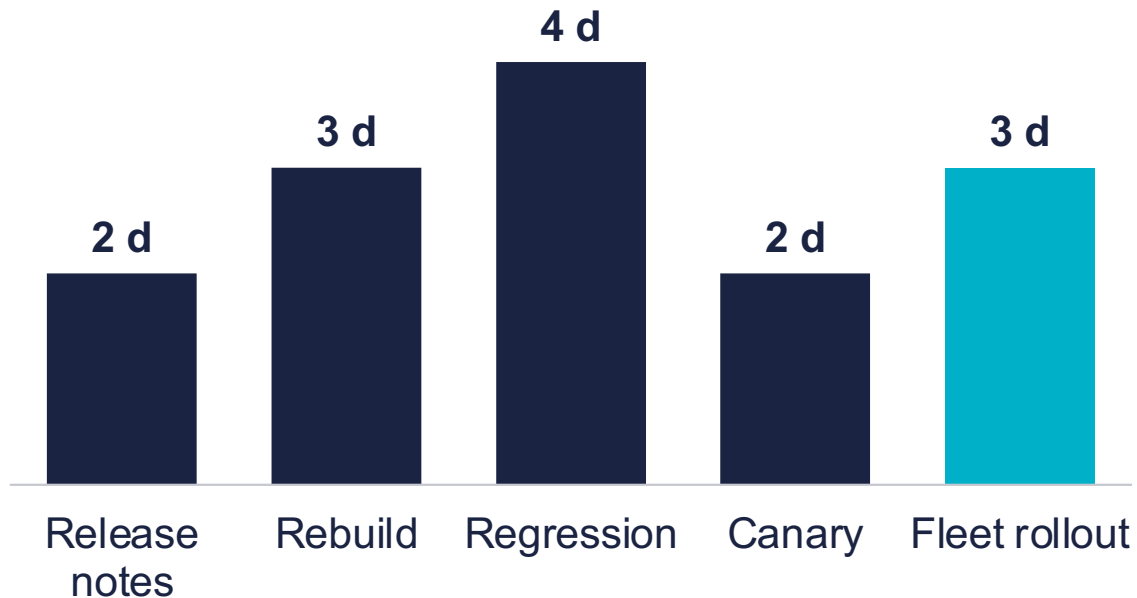
Most tickets after Go-Live are about the customer's OIDC configuration.

Output of this step: an accepted tenant, owned by operations, with triage set up for the shared items.

5 · Operation · Every release costs 14 days

adorsys

Engineer-days per Keycloak minor release, by step



× 4 releases a year

56 engineer-days

before a single ticket or incident

No LTS: two minors behind, a CVE fix is an upgrade, not a patch.

Since 26.6: patch releases support zero-downtime upgrades.

Only the fleet rollout grows with tenant count.

The other 11 days are verification, and they cost the same for one instance or eighty.

The process runs on three clocks at once

adorsys

Per tenant	Per release	Per year
Steps 0–4 ≈13 days¹ engineer time to onboard one tenant	Step 5 14 days* to absorb one Keycloak minor, four times a year *Scope to future optimization	Compliance €70,000 run-rate for registers, audits and exit evidence

A project runs one clock and stops. A SaaS offering runs all three, indefinitely.

Each clock needs its own owner, its own automation and its own budget line.

¹ €3,470 onboarding per tenant-year × 3-year tenant life ÷ €800 per engineer-day. Source: adorsys cost model, 2026.

Back to the request · A contract, not a product

adorsys

What Markus asked for

Managed Keycloak, priced as a hosting fee

99.9% availability, 24/7

His login page and authenticator plugin kept

Compliance answers

Live by end of quarter



What we signed: a Managed Application Contract

Scope

His Keycloak, his extensions. One customer, not a multi-tenant product.

Release testing

Regression and rollout for every Keycloak minor are contracted effort.

Support between releases

Patches, CVE fixes and incidents sit inside an agreed support budget.

We decided against a SaaS-like Keycloak offering.

The contract terms put a price on the testing and support work between releases – that was not acceptable for both sides.

What took the most time · Nothing technical

adorsys



Processes

Service requests

- Infrastructure, access and network changes each needed a request.
- Each request waited in another team's queue.



Certifications

ISO and audit evidence

- Registers, audits and exit evidence had to exist before go-live.



Scope clarification

What does he really want?

- The request said “managed Keycloak”.
- The real need was his login page, his plugins and compliance answers.

The Keycloak work was the predictable part.

Every one of these delays sat between people, teams and paperwork, not in the code.

Six steps, three clocks, every tenant,
every release.

**What can we as Community do –
to grow adaption in Business?**