



Belastingdienst

Ghost in the platform

25-03-2026

**How the Dutch Tax Authority built a service to scale
K8s to 99+ production applications**

Jerry van Hulst
Marcel Kerker



About the Dutch Tax Authority

- Part of the Dutch Government under the Ministry of Finance
- Core tasks: Tax collection, benefits and customs
- Over 28.000+ employees
- €300B+ collected in 2025
- Vast IT landscape with decades long history





About us

Jerry van Hulst
Product Owner @ Belastingdienst

Marcel Kerker
Consultant @ HCS Company
CNCF Ambassador





Our mission

To empower the Belastingdienst with the platform and tools to take full ownership of its digital future



The start of our journey

2017 - PoC with Kubernetes

- Maximum freedom for development teams
- Infrastructure team
- Automating the infrastructure

2019 – First applications on the new platform

- Early adopter phase
- Enablement team





Growing pains

Beyond the early adopter phase, developers started to struggle with:

- High cognitive load
- Lack of developer standardization
- Long lead time for new environments
- Enablement couldn't keep up with demand



Our problem

How do we reduce the cognitive load for developers, how do we increase standardization to speed-up workflows?



The Onboarding Journey



- **Prepare:** Ensure OpenShift suitability, gain Architecture Board approval, and align with the platform Manifesto.
- **Intake & Compliance:** Complete our onboarding questionnaire, followed by a kick-off meeting to align on tech, plans, and next steps.
- **Start on Non-Prod:** We assign a cluster, create your Harbor project, and validate your PaaS.
- **Go to Production:** Final validation of resources, monitoring and operations. The team formally commits to the Manifesto and deploys to Prod!



Cloud-Native Requirements & Our Approach

- **Restartability:** Applications must be able to handle unannounced restarts at any time.
- **Immutability:** Running containers are never patched; updates require deploying a completely new container.
- **Environment Parity:** Dev, Test, Acceptance, and Production environments are kept identical (goodbye "works-on-my-machine"!).
- **Shared Responsibility:** We provide the, but teams are responsible for using them.
- **Our Approach:** We don't enforce these rules rigidly, but proactively monitor, advise, and discuss deviations with the teams.





Introducing our custom operator

- Project-as-a-Service is our open-source solution for giving DevOps teams the means create a self-service, standardized, Kubernetes environment.
- A fully open-source solution
- A full Kubernetes environment in just one simple YAML
- GitOps as interface
- Hit the ground running
- Open Source powered



Project As A Service™



One YAML to rule them all

Everything you need in 1 YAML

- A project name
- RBAC
- Namespaces
- Resource quota
- Capabilities

```
apiVersion: cpet.belastingdienst.nl/v1alpha2
kind: Paas
metadata:
  name: my-new-app
spec:
  groups:
    ad-group:
      query: CN=admin-group
      roles:
        - admin
  capabilities:
    argocd: {}
    grafana: {}
  namespaces:
    prod: {}
  quota:
    requests.cpu: '10'
    requests.memory: 1Gi
    limits.memory: 5Gi
    requests.storage: 13Gi
```



Not just a namespace

A PaaS offers much more than just a namespace

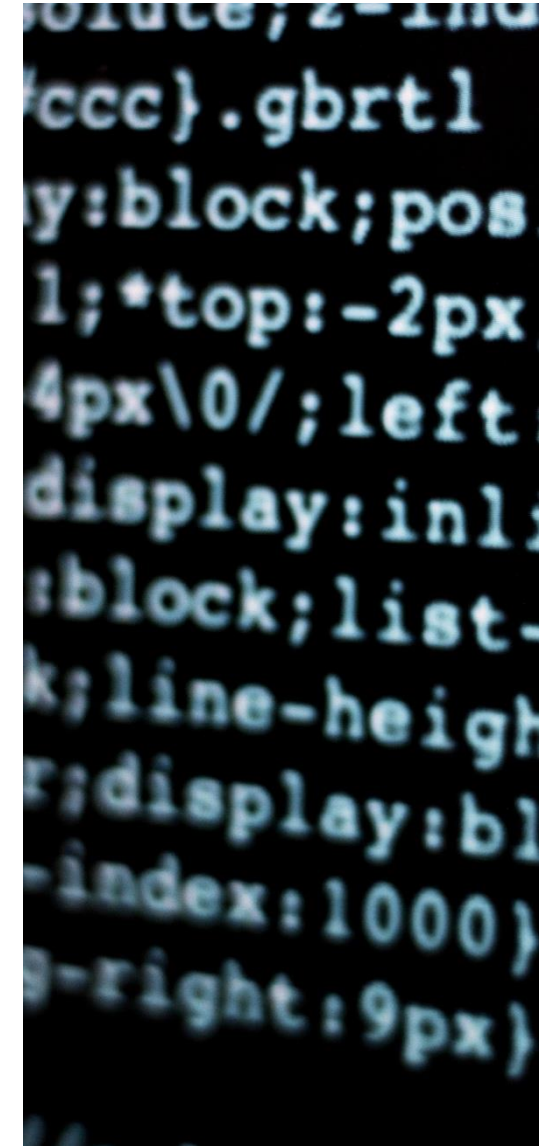
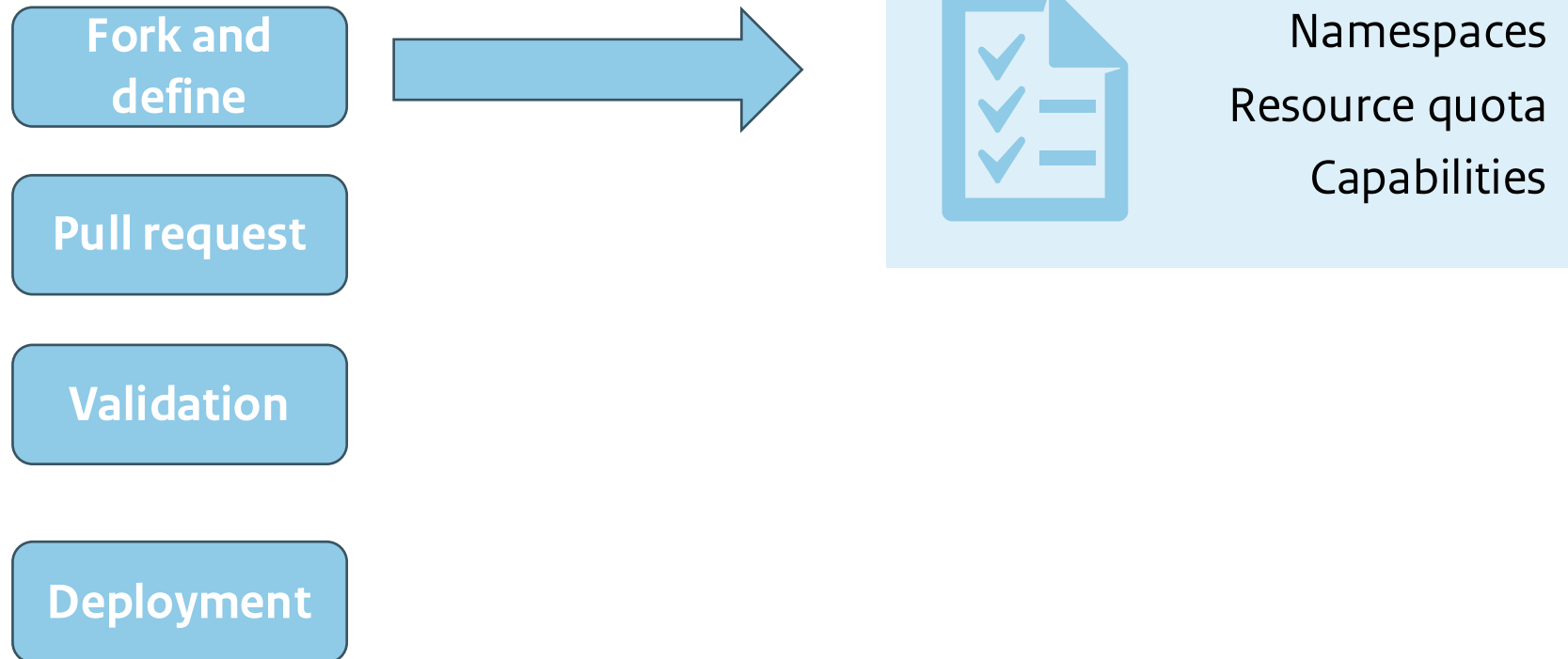
We offer pre-configured capabilities, ready for real workloads

- ArgoCD
- Keycloak
- Grafana
- Tekton
- Alloy
- Custom capabilities





PaaS Operator – Workflow





PaaS Operator – Pull request

Fork and
define

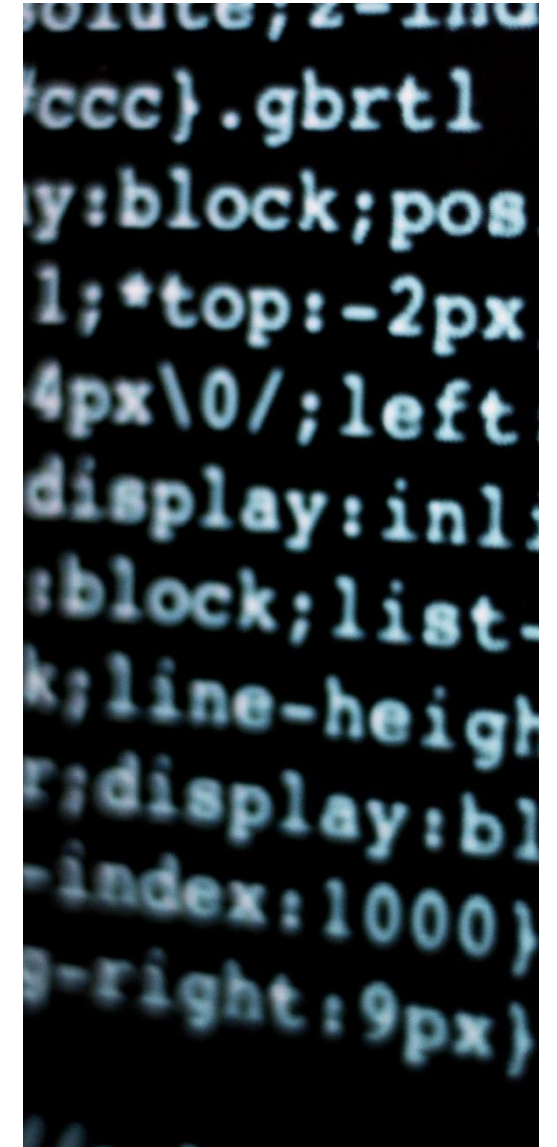
Pull request

Validation

Deployment

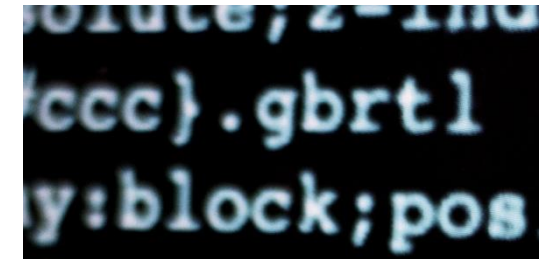


```
apiVersion: cpet.belastingdienst.nl/v1alpha2
kind: Paas
metadata:
  name: my-new-app
spec:
  groups:
    ad-group:
      query: CN=admin-group
      roles:
        - admin
  capabilities:
    argocd: {}
    grafana: {}
  namespaces:
    prod: {}
  quota:
    requests.cpu: '10'
    requests.memory: 1Gi
    limits.memory: 5Gi
    requests.storage: 13Gi
```





PaaS Operator - Validation



Functioneel Account CPET 3 hours ago [link](#)



Functioneel Account CPET **MERGED** master to master in commit [f0f6c07f917](#) [link](#) 1 hour ago



Functioneel Account CPET marked the pull request as **APPROVED** 1 hour ago

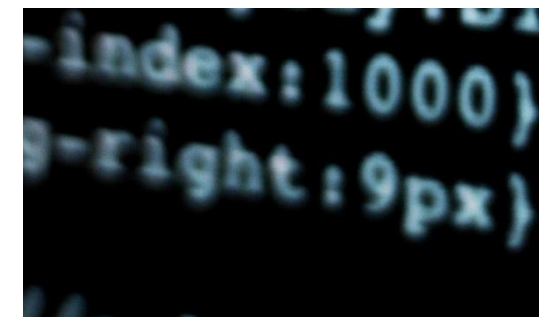


OPENED the pull request 2 hours ago

Reply · Resolve ...



OPENED the pull request 3 hours ago





PaaS Operator - Output

Fork and
define

Pull request

Validation

Deployment



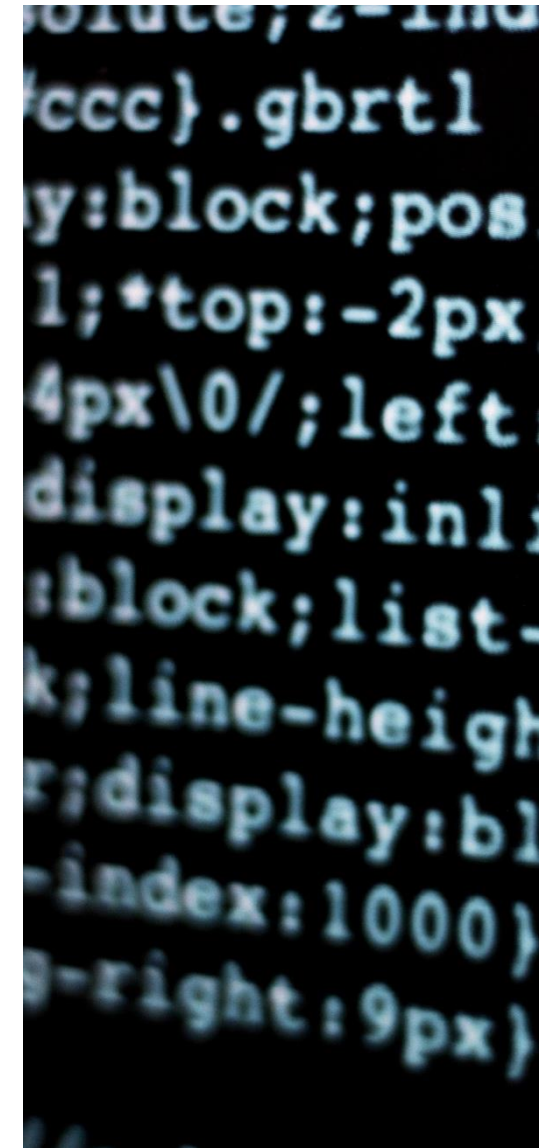
PaaS Project



Grafana



HARBOR





Open source powered

We believe in open source. Transparency and collaboration are important to us. This is why the PaaS operator is fully open source

For more information about the PaaS operator, check out the QR code

Or check the following URL

<https://github.com/belastingdienst/opr-paas>





The Ultimate Starting Point for CI/CD

The goal: A standardized, fully automated setup for a complete CI/CD pipeline on our OpenShift platform

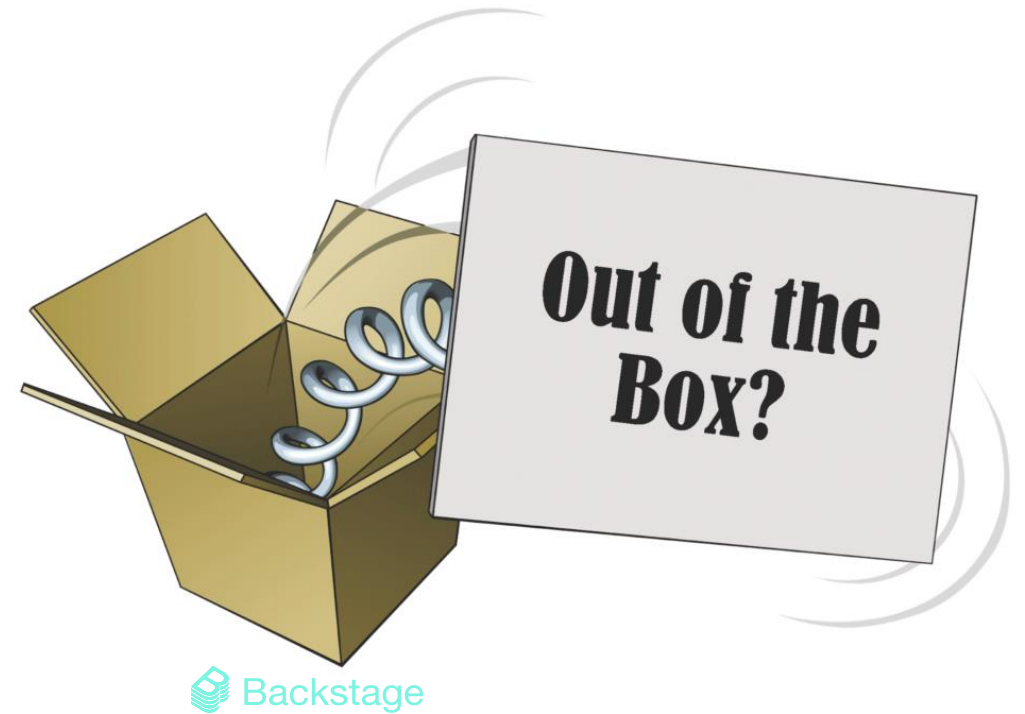
Core Integrations: Seamlessly connects Tekton, ArgoCD, Bitbucket, and other DevOps tool chains.

Versatility: Designed for both brand-new *cloud-native* applications and existing *cloud-aware* legacy apps.



What you get Out-of-the-Box

- **5 Dedicated Git Repositories:** Code, Test, Deploy, Tekton, and ArgoCD
- **Pipeline Infrastructure:** Fully configured Tekton pipelines and event triggers.
- **Deployment Setup:** Ready-to-go ArgoCD applications.
- **Security & Access:** SSH keypairs for Tekton and ArgoCD.
 - **Sealed Secrets for Git tokens, webhooks, and SSH keys.**
- **Optional Integrations:** Tokens for Harbor, Mattermost, Nexus, and Dependency-Track.
- **PaaS:** PaaS specifications and automatically creates a Pull Request to the PaaS repository.





Navigating the Golden Path

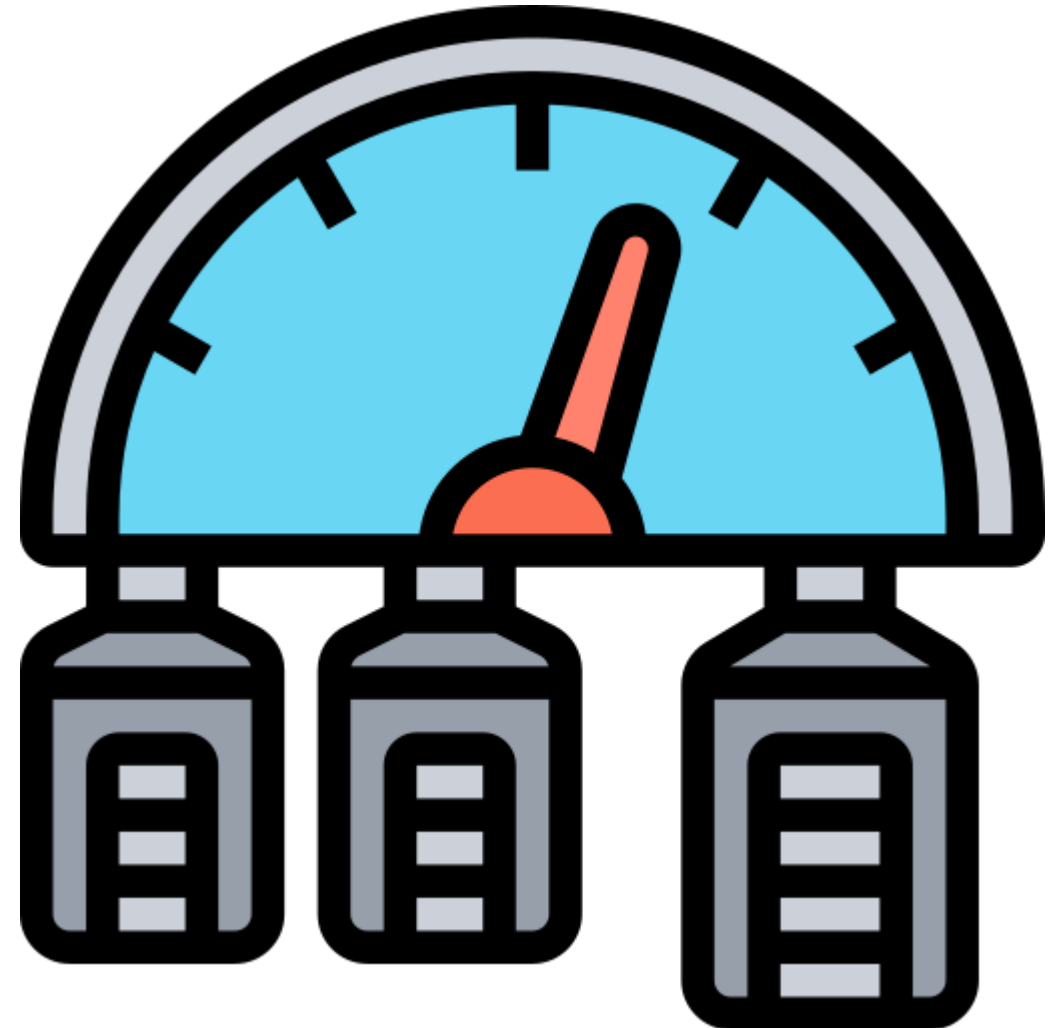
- **Opinionated Guidance:** Stop "reinventing the wheel" using our curated whitepapers and best practices.
- **A Ton of Recipes:** Ready-to-use templates and solutions to accelerate your development.
- **The Application Lifecycle:**
 - **Day 0 (Design):** Strategic architecture and framework choices (e.g., Java/Quarkus, TS/Angular, Python).
 - **Day 1 (Deploy):** Automated provisioning and seamless deployments via ArgoCD & GitOps.
 - **Day 2 (Operate):** Continuous observability, monitoring, and proactive right-sizing.





Developer Enablement: Empowering 99+ Teams

- **Enablement over Support:** We don't just solve tickets; we build self-sufficiency.
- **Communities of Practice:** Scaling knowledge across 99+ DevOps teams.
- **Container User Group (CUG):** Regular demos to showcase new platform features.
- **Containerization Days:** Plenary sessions and external speakers on new tech.
- **Self-paced workshops:** on Tekton, ArgoCD, IAM, RightSizing and Kustomize.
- **External:** Workshops with Red Hat and recommended certifications (CKAD/CKA, DO188/DO288).
- **Accelerator Hackathon:** Hands-on hackathons where Platform experts work side-by-side with your team for a day to fast-track your first app.





Our platform in numbers

What does our environment look like today?

40+ team members

82 teams
Over 160 applications

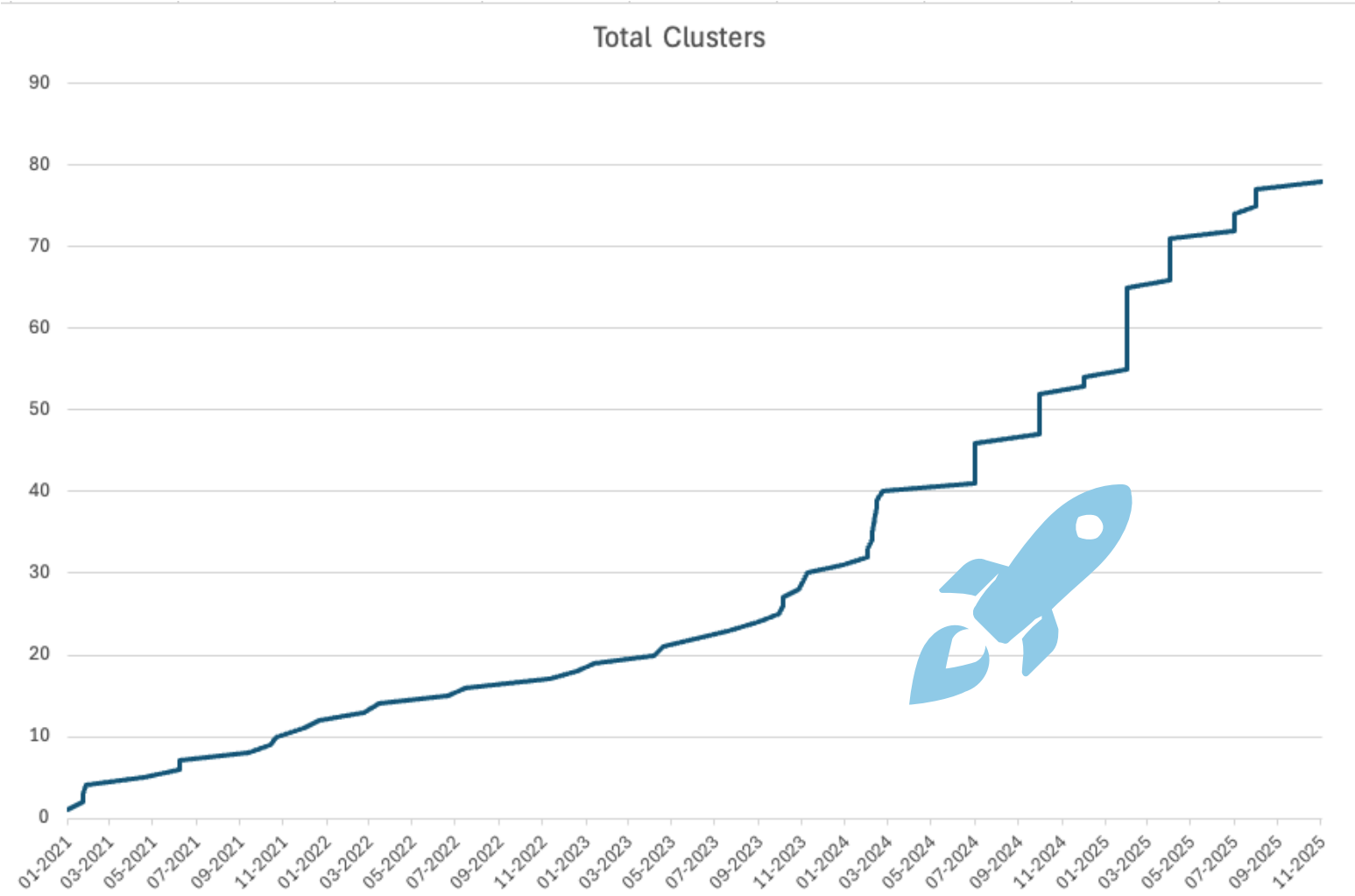
78 OpenShift clusters
900+ worker nodes
23.000 virtual cores
±100 TB memory

227 Dev PaaS
5300 namespaces
45.000 pods

125 Prod PaaS
2800 namespaces
22.000 pods



Platform growth





What's next for us?

- More integration with the internal developer portal / Backstage
- More golden path whitepapers
- AI-Powered support
- And most of all, we will keep listening to our users



Closing thoughts

- A platform is more than infrastructure
- Standardization enables adoption at scale
- Build for self-sufficiency, not support





Thank you!

Find us on GitHub, try the PaaS Operator, or contribute!

Scan the QR code or check the following URL
<https://github.com/belastingdienst/opr-paas>

