

TECHNICAL WHITEPAPER

OpenKubes Anywhere

Unified Kubernetes Operations Across
Local, Bare Metal, Edge and Cloud

Version 1.0 · June 2026

Kubernauts GmbH · kubernauts.de

Audience: CTOs, Platform Engineers, Architects, Investors

Table of Contents

- 1. Executive Summary**
- 2. Problem Statement**
- 3. Why Now?**
- 4. Platform Architecture**
 - 4.1 Control Plane Design
 - 4.2 GitOps Pipeline
 - 4.3 Cluster Lifecycle Management
 - 4.4 Infrastructure as Code
- 5. Environment Coverage**
 - 5.1 Local Development
 - 5.2 Bare Metal & Sovereign Cloud
 - 5.3 Edge Computing
 - 5.4 Public Cloud
- 6. Security Framework**
- 7. Observability Stack**
- 8. AI & MLOps Integration**
- 9. Robotics & Industrial AI**
- 10. Implementation Roadmap**
 - 10.1 Current State (GA)
 - 10.2 Near-Term H2 2026 (Planned)
 - 10.3 Mid-Term 2027 (Experimental)
 - 10.4 Long-Term 2028+ (Vision)
- 11. Reference Architecture**
- 12. Why OpenKubes Anywhere?**
- 13. Conclusion**

1. Executive Summary

OpenKubes Anywhere provides a unified Kubernetes operating model across local development environments, bare-metal datacenters, edge locations, and public clouds — from a single GitOps-driven control plane. Where organizations today operate fragmented toolchains per environment, OpenKubes Anywhere replaces them with one declarative, continuously-reconciled platform built entirely on CNCF-graduated projects: ArgoCD, Cluster API, and Crossplane.

This whitepaper describes the current technical implementation, architecture decisions, supported environments, security posture, and the planned development roadmap through 2028. It is intended for CTOs evaluating platform consolidation, platform engineers designing multi-environment Kubernetes strategies, and technical architects assessing cloud-native infrastructure frameworks.

Capability	Status	Technology
GitOps Control Plane	GA	ArgoCD, Flux
Cluster Lifecycle Management	GA	Cluster API
Infrastructure as Code	GA	Crossplane
Bare Metal Provisioning	GA	Metal3, iPXE
Edge Fleet Management	GA	K3s, ArgoCD
Public Cloud (EKS/AKS/GKE)	GA	Cluster API Providers
AI/MLOps Platform	GA	Kubeflow, KServe, Ray
Robotics / Industrial AI	Beta	ROS2, Open-RMF
Multi-Tenancy & Policy Engine	GA	Keycloak, Kyverno
Unified Observability	GA	Prometheus, Grafana, OTel

Table 1: OpenKubes Anywhere capability matrix

2. Problem Statement

Enterprise Kubernetes adoption has matured significantly. Organizations no longer ask *whether* to run Kubernetes — they ask *how* to run it consistently across increasingly diverse infrastructure footprints. This creates several compounding operational challenges:

Toolchain fragmentation. Each environment (local, bare metal, edge, cloud) historically requires separate deployment tooling, CI/CD pipelines, and lifecycle processes. Platform engineers maintain multiple configuration repositories, credentials stores, and monitoring stacks.

Security model divergence. RBAC policies, network policies, and compliance controls are defined independently per environment. This creates audit gaps, increases compliance overhead, and introduces inconsistent security postures across the fleet.

Cluster drift. Without a declarative, continuously-reconciled control plane, cluster configurations diverge over time. Manual changes, emergency patches, and environment-specific overrides accumulate into a state that is difficult to reproduce or audit.

Slow developer feedback loops. Developers require access to Kubernetes environments that accurately reflect production. Without a consistent local-to-cloud operating model, environment parity is difficult to achieve and maintain.

Edge operational complexity. Edge Kubernetes deployments face unique challenges: intermittent connectivity, resource constraints, remote fleet management, and the need for offline-capable GitOps workflows.

OpenKubes Anywhere addresses all of these challenges through a single, declarative, GitOps-native platform engineering framework built entirely on CNCF-graduated and incubating projects.

3. Why Now?

Several converging market forces have created an urgent and concrete need for a unified Kubernetes operating platform. The timing for OpenKubes Anywhere is not coincidental — it reflects a fundamental shift in enterprise infrastructure requirements:

Broadcom / VMware disruption. The Broadcom acquisition of VMware has triggered a large-scale re-evaluation of enterprise virtualization strategies. Thousands of organizations are actively migrating from vSphere to Kubernetes-native alternatives. OpenKubes Anywhere — with KubeVirt and bare metal support — is purpose-built for this transition, offering a drop-in replacement for VM workloads without abandoning operational discipline.

AI infrastructure demand. Generative AI and LLM workloads require GPU-accelerated, distributed infrastructure that spans edge, on-premises, and cloud environments. Organizations cannot afford separate operational models for AI and non-AI workloads. OpenKubes AI integrates natively into the same platform that runs production application workloads — no separate MLOps silo required.

Edge computing growth. Industrial IoT, autonomous systems, and smart manufacturing are driving rapid growth in edge Kubernetes deployments. IDC projects over 55 billion connected edge devices by 2027. Managing these fleets at scale requires exactly the kind of GitOps-driven, offline-capable fleet management that OpenKubes Edge provides.

Sovereign cloud requirements. Regulatory frameworks (NIS2, DORA, GDPR, BSI C5) are driving European and global enterprises to require infrastructure sovereignty — the ability to run workloads on infrastructure they fully control, including air-gapped and classified environments. OpenKubes Anywhere is designed for sovereign deployment from day one.

Platform engineering adoption. The CNCF Platform Engineering Working Group and Gartner both highlight platform engineering as a top infrastructure priority for 2025–2027. Organizations are consolidating from bespoke Kubernetes toolchains to standardized Internal Developer Platforms (IDPs). OpenKubes Anywhere provides the technical foundation for an enterprise-grade IDP.

Market Driver	Impact	OpenKubes Response
VMware/Broadcom disruption	Mass VM-to-K8s migration	KubeVirt + Bare Metal support
AI/LLM infrastructure demand	GPU workloads across all envs	OpenKubes AI stack
Edge computing growth	55B+ edge devices by 2027	K3s + offline GitOps fleet mgmt
Sovereign cloud regulations	NIS2, DORA, BSI C5 compliance	Air-gap + RBAC + Crossplane
Platform engineering adoption	IDP consolidation trend	Unified control plane + CLI

Table 2: Market drivers and OpenKubes Anywhere response

4. Platform Architecture

OpenKubes Anywhere is structured as a layered platform engineering framework. Each layer is independently replaceable and composable, enabling organizations to adopt the platform incrementally or integrate components into existing toolchains.

4.1 Control Plane Design

The OpenKubes control plane is a management Kubernetes cluster that hosts all platform components. It serves as the single source of truth for cluster fleet state, application deployments, and infrastructure resources. The management cluster runs:

- ArgoCD — continuous delivery and GitOps reconciliation engine
- Cluster API (CAPI) — declarative cluster lifecycle management
- Crossplane — infrastructure resource provisioning via Kubernetes-native API
- Keycloak — identity provider and OIDC broker
- Kyverno — policy engine for governance and compliance
- Prometheus + Grafana + OpenTelemetry Collector — observability stack

4.2 GitOps Pipeline

All configuration is stored in Git. The Git repository is the single source of truth. ArgoCD continuously reconciles the desired state (Git) with the actual state (cluster). The pipeline follows a structured promotion model:

Stage	Description	Tooling
Source	Infrastructure and app manifests in Git	Git (GitHub/GitLab/Gitea)
Validate	Policy checks, schema validation, security scanning	Kyverno, Conftest, Trivy
Sync	ArgoCD Application reconciliation per environment	ArgoCD, ApplicationSets
Verify	Health checks, smoke tests, status gates	ArgoCD Health Checks, Argo Rollouts
Promote	Automated or manual promotion across environments	ArgoCD Image Updater, Argo Workflows

Table 2: GitOps pipeline stages

4.3 Cluster Lifecycle Management

Cluster API (CAPI) provides a declarative API for cluster provisioning, scaling, upgrading, and deletion. OpenKubes ships with tested CAPI provider configurations for:

- CAPM3 (Metal3) — bare metal cluster provisioning via BMC/IPMI/Redfish
- CAPA — Amazon EKS and EC2-based clusters

- CAPZ — Azure AKS and managed node pools
- CAPG — Google GKE clusters
- CAPD — local Docker-based clusters for development and CI

ClusterClass templates provide opinionated, reusable cluster topologies that encode organizational standards for node sizing, networking, security, and add-ons.

4.4 Infrastructure as Code with Crossplane

Crossplane extends Kubernetes with Composite Resource Definitions (XRDs) and Compositions that abstract cloud provider APIs into organization-specific infrastructure primitives. Platform teams define reusable abstractions (e.g. 'DatabaseCluster', 'ObjectStoreBucket') that developers consume via standard Kubernetes manifests — without requiring cloud-provider specific knowledge.

5. Environment Coverage

5.1 Local Development — The Same GitOps Workflow from Laptop to Production

[k3s](#) · [Multipass](#) · [KubeVirt](#) · [bonsai](#) · [GitOps](#)

OpenKubes Local is one of the platform's most distinctive capabilities. Developers run a full GitOps-managed Kubernetes environment on their workstation — not a simulation, not a stripped-down approximation, but the same cluster topology, the same ArgoCD ApplicationSets, and the same Kyverno policies as production. When a developer pushes to Git, the same reconciliation loop that runs in the datacenter runs on their laptop.

Built on k3s (lightweight Kubernetes), Multipass (VM isolation on macOS, Linux, and Windows), and KubeVirt (VM workloads alongside containers), OpenKubes Local enables full-stack development including legacy VM-based applications. bonsai provides opinionated local cluster bootstrapping with pre-configured GitOps, observability, and security tooling — a production-grade developer environment in minutes. This is a uniquely strong value proposition: *what runs locally runs everywhere*.

5.2 Bare Metal & Sovereign Cloud

[Cluster API](#) · [Metal3](#) · [ArgoCD](#) · [iPXE](#)

OpenKubes Bare Metal delivers declarative cluster lifecycle management for on-premises datacenters and sovereign infrastructure. Metal3 (Cluster API Provider for Metal3) handles bare metal host discovery, inspection, provisioning, and deprovisioning via BMC interfaces (IPMI, Redfish).

Nodes are provisioned via iPXE network boot, eliminating manual OS installation. Ironic manages the hardware inventory. Once provisioned, clusters are managed identically to cloud clusters via CAPI and ArgoCD. This model is fully air-gap capable for sovereign and classified environments.

5.3 Edge Computing

[K3s](#) · [GitOps](#) · [Offline](#) · [Fleet Management](#)

OpenKubes Edge addresses the operational challenges of running Kubernetes at factories, warehouses, retail sites, and distributed industrial locations. K3s provides a production-grade, resource-efficient Kubernetes distribution for constrained hardware.

ArgoCD agent mode enables pull-based GitOps reconciliation that tolerates intermittent WAN connectivity. Fleet management via ApplicationSets enables consistent, scalable deployment across thousands of edge sites from a single management cluster.

5.4 Public Cloud

[EKS](#) · [AKS](#) · [GKE](#) · [Crossplane](#)

OpenKubes Cloud provides unified lifecycle management for managed Kubernetes services across AWS, Azure, and Google Cloud — without vendor lock-in. Cluster API providers (CAPA, CAPZ, CAPG) manage cluster creation, node pool scaling, and Kubernetes version upgrades through the same declarative API used for bare metal.

Crossplane Compositions abstract cloud-provider managed services (RDS, Azure Database, Cloud SQL) into portable resource definitions. Applications consume infrastructure resources via standard Kubernetes CRDs regardless of the underlying cloud provider.

6. Security Framework

OpenKubes implements a zero-trust security model across the entire platform stack. Security is not an add-on — it is baked into the platform architecture at every layer.

Layer	Component	Capability
Identity	Keycloak	OIDC/OAuth2 identity brokering, SSO, MFA, LDAP integration
Access Control	Kubernetes RBAC + Kyverno	Fine-grained RBAC, policy-as-code, admission control
Network	Cilium / Calico	Network policies, mTLS, zero-trust east-west traffic
Supply Chain	Cosign + Trivy	Image signing, SBOM generation, vulnerability scanning
Secrets	Vault / ESO	Secret management, dynamic secrets, external secret operator
Audit	Falco + OpenSearch	Runtime threat detection, audit log aggregation, alerting
Compliance	Kyverno + OPA	CIS benchmarks, PCI-DSS, SOC2, policy enforcement

Table 3: Security framework components

7. Observability Stack

Full-stack observability is included out of the box and deployed consistently across every environment. The observability stack is built on open standards (OpenTelemetry, Prometheus Remote Write, OTLP) ensuring portability across environments and cloud providers.

Metrics — Prometheus. Cluster, node, workload, and custom application metrics. Thanos for long-term storage and cross-cluster querying.

Dashboards — Grafana. Pre-built dashboards for cluster health, workload performance, GitOps sync status, and security posture.

Logs — OpenSearch + Fluentd. Centralized log aggregation with full-text search, audit log retention, and compliance-ready log management.

Traces — Tempo + OpenTelemetry. Distributed tracing for application and infrastructure components. OTLP-native collection and storage.

Alerting — Alertmanager + PagerDuty. Multi-channel alerting with routing, silencing, and escalation policies.

AI Insights — OpenKubes AI Ops (Beta). Anomaly detection and predictive alerting using ML models trained on cluster telemetry.

8. AI & MLOps Integration

OpenKubes AI provides a fully integrated MLOps and AI platform engineering stack, deployable across edge, datacenter, and cloud environments through the same GitOps-driven workflows used for all other platform components.

Component	Technology	Purpose
ML Pipeline Orchestration	Kubeflow Pipelines	Reusable, portable ML pipelines with version control
Experiment Tracking	MLflow	Experiment metadata, model registry, artifact storage
Distributed Training	Ray / Kubeflow Training	Large-scale model training across GPU node pools
Model Serving (Standard)	KServe	Serverless model inference with autoscaling
Model Serving (High-Perf)	Triton Inference Server	NVIDIA GPU-optimized multi-model serving
LLM Inference	vLLM / Ollama	Efficient LLM inference for open-weight models
Vector Database	Qdrant / Milvus	Embedding storage and semantic search
Feature Store	Feast	Feature engineering and serving for ML pipelines
GPU Scheduling	NVIDIA GPU Operator	GPU resource management across heterogeneous nodes

Table 4: AI/MLOps stack components

9. Robotics & Industrial AI

OpenKubes Robotics enables the same operational model used for cloud-native applications to manage fleets of autonomous industrial systems. A factory floor becomes just another environment in the GitOps pipeline — provisioned, secured, observed, and lifecycle-managed through the same control plane as any cloud cluster. For CTOs in manufacturing, logistics, and industrial automation, this means one platform team can operate both IT and OT workloads without separate toolchains or expertise silos.

ROS2. Robot Operating System 2 — containerized ROS2 nodes managed via Kubernetes Deployments and DaemonSets. Supports multi-robot coordination and lifecycle management.

Open-RMF. Open Robotics Middleware Framework — fleet management for heterogeneous robot fleets. Deployed on Kubernetes for high availability and scalability.

DDS Networking. Data Distribution Service for real-time, deterministic publish-subscribe communication between ROS2 nodes across Kubernetes pods.

Multus CNI. Multi-homed networking enabling pods to connect to multiple networks simultaneously — required for industrial fieldbus, OT network, and IT network isolation.

SR-IOV. Single Root I/O Virtualization for hardware-accelerated, low-latency networking. Enables near-bare-metal network performance for time-sensitive workloads.

GPU Scheduling. NVIDIA GPU Operator and device plugins for scheduling AI inference and computer vision workloads on GPU-equipped industrial edge nodes.

10. Implementation Roadmap

The OpenKubes Anywhere roadmap is organized into four horizons reflecting current general availability, near-term delivery, mid-term development, and long-term platform vision.

10.1 Current State — Generally Available ■ GA

- ✓ GitOps-driven cluster lifecycle via ArgoCD + Cluster API
- ✓ Bare metal provisioning via Metal3/Ironic with iPXE boot
- ✓ Edge Kubernetes fleet management with offline GitOps support
- ✓ Public cloud support: EKS, AKS, GKE via CAPI providers
- ✓ Infrastructure as Code via Crossplane with AWS, Azure, GCP providers
- ✓ Zero-trust security: Keycloak, Kyverno, Cilium, Falco
- ✓ Full observability: Prometheus, Grafana, Tempo, OpenSearch, OTel
- ✓ AI/MLOps stack: Kubeflow, MLflow, KServe, Triton, Ray
- ✓ Local development: k3s + Multipass + KubeVirt

10.2 Near-Term — H2 2026 ■ Planned

OpenKubes UI (Alpha). Unified web dashboard for cluster fleet management, GitOps sync status, and observability across all environments. Built on React with a Kubernetes-native API backend.

Cluster API v1.10 support. Adoption of CAPI v1.10 features including ClusterClass topology mutation hooks and improved MachineHealthCheck semantics.

AI Ops Integration. Integration of anomaly detection models into the observability stack for predictive alerting on cluster health metrics and workload performance degradation.

vCluster Multi-Tenancy. Virtual cluster support via vCluster enabling self-service namespace isolation with full Kubernetes API compatibility for development teams.

SOPS + External Secrets Operator GA. Unified secret management across environments with SOPS-encrypted Git secrets and ESO for cloud-provider secret store integration.

OpenKubes CLI (okctl). Unified CLI for cluster bootstrapping, environment onboarding, GitOps status inspection, and platform upgrade management.

10.3 Mid-Term — 2027 ■ Experimental

Autonomous Cluster Remediation. ML-driven remediation engine that detects cluster health anomalies and triggers automated corrective GitOps workflows without human intervention.

Multi-Cluster Service Mesh. Istio/Cilium-based multi-cluster service mesh enabling transparent, mTLS-encrypted service-to-service communication across all environment types.

OpenKubes Marketplace. Self-service catalog of pre-validated platform add-ons, application templates, and infrastructure compositions. GitOps-native one-click deployment.

Cost Management Integration. Kubecost integration with cross-environment cost attribution, chargeback reporting, and rightsizing recommendations.

Confidential Computing. Intel TDX / AMD SEV integration for confidential VM workloads on bare metal and cloud environments.

OpenKubes Robotics GA. Full general availability of the robotics platform with certified ROS2 distributions, Open-RMF fleet management, and industrial AI workload templates.

10.4 Long-Term — 2028+ ■ Vision

The long-term platform vision centers on autonomous, self-healing infrastructure that minimizes operational toil while maximizing developer productivity and platform reliability:

- Intent-driven infrastructure — declare business outcomes, platform resolves infrastructure topology
- AI-native cluster scheduling — workload placement optimization using ML models
- Cross-environment workload mobility — live migration of stateful workloads across environments
- Quantum-safe cryptography — post-quantum TLS and key management across the platform
- Carbon-aware scheduling — workload scheduling optimized for renewable energy availability
- OpenKubes SaaS offering — fully managed control plane with bring-your-own infrastructure model
- OpenKubes IMP (Research) — Infrastructure Memory Platform: Git-native infrastructure memory and recovery framework for autonomous platform operations. Enables the platform to learn from historical cluster state, detect regressions, and autonomously restore known-good configurations.

11. Reference Architecture

The diagram below illustrates the OpenKubes Anywhere platform layers — from the Git source of truth at the top, through the unified management control plane, down to the four target environment types. All environments are reconciled continuously through the same GitOps pipeline.



Figure 1: OpenKubes Anywhere platform architecture — Git to workload cluster across all environments

The reference table below maps each architecture layer to its deployment model:

Layer	Component	Deployment Model
Git Source of Truth	GitHub / GitLab / Gitea	SaaS or self-hosted, HA configuration
Management Cluster	Kubernetes (RKE2 / K3s)	3-node HA control plane, bare metal or cloud VM
GitOps Engine	ArgoCD	HA deployment on management cluster, SSO via Keycloak
Cluster Lifecycle	Cluster API + Providers	Deployed on management cluster, provider per env type
Infra Provisioning	Crossplane + Providers	Deployed on management cluster, XRDs per org
Identity	Keycloak	HA deployment, external PostgreSQL backend
Observability	Prometheus + Thanos	Federation across workload clusters, long-term in object store
Secrets	Vault + ESO	HA Vault cluster, ESO deployed on workload clusters
Workload Clusters	CAPI-managed clusters	Per environment, bootstrapped via ArgoCD ApplicationSets

Table 5: Reference architecture component mapping

Expected Business Outcomes

The following outcomes reflect the measurable business impact of adopting a unified Kubernetes operating model. These are based on platform engineering patterns observed across enterprise Kubernetes deployments:

Operating Model	Business Outcome	Impact
One GitOps pipeline	Reduced operational complexity	Fewer tools, less context-switching, lower training overhead
One security model	Lower compliance effort	Single audit scope, consistent policy enforcement across all envs
One observability stack	Faster incident response	Unified dashboards, correlated alerts, no data silos
One lifecycle model	Faster platform delivery	Standardized cluster provisioning from days to minutes
GitOps-native IaC	Reduced configuration drift	Declarative state eliminates manual change accumulation
Sovereign deployment	Regulatory compliance	NIS2, DORA, BSI C5, GDPR without cloud-provider dependency
Open Source foundation	No vendor lock-in	Full portability across cloud providers and on-premises infra

Table 6: Expected business outcomes from OpenKubes Anywhere adoption

12. Why OpenKubes Anywhere?

Enterprise Kubernetes platforms are not a new category. OpenShift, Rancher, Anthos, and Azure Arc are all mature, well-supported products. The question a CTO, architect, or investor will ask is not *what* OpenKubes Anywhere does — it is *why OpenKubes Anywhere instead of the alternatives*. The answer lies in four foundational design decisions that distinguish it from every commercial platform in the market.

GitOps-first by design, not by integration

Most platforms offer GitOps as an add-on — an optional integration layer on top of a control plane that was designed for imperative operations. OpenKubes Anywhere is built from the ground up on the GitOps operating model. Git is not a deployment trigger; it is the system of record. Every cluster, every infrastructure resource, every security policy, and every application is defined declaratively in Git and continuously reconciled via ArgoCD. There is no imperative escape hatch, no click-ops console as the source of truth, and no configuration that lives outside version control.

CNCF-native, not CNCF-compatible

Commercial platforms typically wrap CNCF projects in proprietary abstractions that create soft lock-in: custom resource schemas, proprietary APIs, and vendor-specific tooling that diverges from upstream. OpenKubes Anywhere uses CNCF projects directly — ArgoCD, Cluster API, Crossplane, Prometheus, OpenTelemetry — without proprietary wrappers. Platform teams work with the same tools and APIs they would use in a self-built platform, but pre-integrated, pre-tested, and production-validated. When upstream projects evolve, OpenKubes evolves with them — without a vendor migration path.

Sovereign and air-gap ready from day one

Cloud-hosted control planes (Anthos, Azure Arc) introduce a dependency on the provider's infrastructure for cluster management. For organizations operating under NIS2, DORA, BSI C5, or classified requirements, this is a disqualifier. OpenKubes Anywhere runs entirely on infrastructure the customer controls — including fully air-gapped environments with no outbound connectivity. The management cluster, ArgoCD, and all platform components operate on-premises. This is not a workaround; it is the default deployment model.

AI and Robotics as first-class platform citizens

Most Kubernetes platforms treat AI/ML as a separate concern — managed by a different team, on different infrastructure, with different operational tooling. OpenKubes AI and OpenKubes Robotics are not separate products bolted on after the fact. They run on the same clusters, managed through the same GitOps pipeline, secured by the same zero-trust framework, and observed through the same unified observability stack. A platform team operating OpenKubes Anywhere manages cloud-native

applications, LLM inference workloads, and factory floor robotics systems under one unified operating model.

Capability	OpenKubes	OpenShift	Rancher	Anthos	Azure Arc
GitOps-first (native)	■	Partial	Partial	Partial	Partial
CNCF-native (no wrappers)	■	■	Partial	■	■
Air-gap / Sovereign cloud	■	■	■	■	■
Bare metal (Metal3/CAPI)	■	■	Partial	■	■
Edge fleet management	■	Partial	■	Partial	■
AI/MLOps integrated	■	Partial	■	Partial	Partial
Robotics / Industrial AI	■	■	■	■	■
Open Source (no lock-in)	■	■	Partial	■	■
Multi-cloud without lock-in	■	Partial	■	Partial	■

Table 7: Competitive feature comparison. ■ = native/full support, Partial = available with limitations or add-ons, ■ = not supported. Feature assessments reflect the default platform architecture and native capabilities without third-party extensions. All referenced products are trademarks of their respective owners.

OpenKubes Anywhere does not compete on features alone. It competes on philosophy:

Open by default. Declarative by design. Sovereign by architecture. Extensible without vendor permission.

For organizations that have outgrown cloud-provider managed Kubernetes and need a platform that works the same way from a developer's laptop to a factory floor to a multi-cloud production environment — OpenKubes Anywhere is the only platform built for that operating model from the ground up.

13. Conclusion

OpenKubes Anywhere represents a significant step forward in enterprise platform engineering. By unifying Kubernetes operations across every environment through a single GitOps-driven control plane, organizations can eliminate operational fragmentation, reduce platform complexity, and accelerate developer productivity.

Built entirely on CNCF-graduated and incubating projects — ArgoCD, Cluster API, Crossplane, Prometheus, OpenTelemetry — OpenKubes Anywhere avoids vendor lock-in by design. Platform teams retain full control over their infrastructure while benefiting from the velocity of a pre-integrated, production-validated platform engineering framework.

The roadmap reflects a clear trajectory: from today's operational unification toward tomorrow's autonomous, AI-driven infrastructure that proactively manages itself. The foundation built today — declarative, versioned, auditable — is the prerequisite for everything that follows.

Kubernetes solved application portability. OpenKubes Anywhere solves operational portability. The future is not cloud-only. The future is Anywhere.

Kubernauts GmbH	Im Mediapark 4C, 50670 Cologne, Germany
Web	https://kubernauts.de/en/openkubes/openkubes-anywhere/
Phone	+49 221 379 90 680