

DRAFT

BubbleOps vs Kubernetes

Infrastructure Lifecycle Automation vs Container Orchestration

Table of Contents

[Executive Summary \(Slide 3\)](#)

[Architectural Scope Comparison \(Slide 4\)](#)

[Full Capability Comparison Matrix \(Slide 5\)](#)

[Operational Surface Area Analysis \(Slide 6\)](#)

[Kubernetes Delivered Through BubbleOps \(Slide 8\)](#)

[Sovereignty & Air-Gapped Considerations \(Slide 10\)](#)

[Decision Framework \(Slide 12\)](#)

[Conclusion \(Slide 14\)](#)

[Call to Action \(Slide 15\)](#)

Executive Summary

Purpose of this Document

This document clarifies the architectural differences between:

- BubbleOps: A platform engineering and Day-2 automation layer for virtualized infrastructure
- Kubernetes: A container orchestration platform

It is intended for infrastructure architects, CTOs, and platform teams evaluating tooling decisions.

Core Thesis

Kubernetes orchestrates containers.

BubbleOps orchestrates entire infrastructure environments.

These tools operate at different layers of the stack.

For organizations operating Proxmox-based virtualized infrastructure, BubbleOps provides:

- Integrated Day-2 automation
- Infrastructure governance
- Lifecycle symmetry
- Sovereign deployment models

Kubernetes can be deployed inside BubbleOps-managed environments when container orchestration is required.

Architectural Scope Comparison

Abstraction Layer

Kubernetes

- Orchestrates container workloads
- Manages pods, services, ingress, and scaling
- Operates at the application orchestration layer

BubbleOps

- Orchestrates full multi-instance environments
 - Provisions VMs via PXE + Kickstart
 - Integrates identity, monitoring, backup, and config management
 - Manages lifecycle at infrastructure layer
-

Deployment Unit

Capability	BubbleOps	Kubernetes
Deployment Unit	Multi-instance “Bubble”	Pod / Container
Environment Scope	Full infrastructure	Application Workload

Full Capability Comparison Matrix

Infrastructure Control vs Application Orchestration

Capability	BubbleOps	Kubernetes
Primary Focus	Full environment lifecycle automation	Container workload orchestration
Deployment Unit	Multi-instance Bubble	Pod / Container
VM Provisioning	Native (PXE + Kickstart)	Not built-in
One-Click Kubernetes Cluster Deployment	Yes — full multi-node cluster	Requires external tooling
Cluster Teardown Symmetry	Full clean teardown (reversible hooks)	Manual or external automation required
Infrastructure Placement Governance	Policy-driven (cluster, network, storage)	Requires additional tooling
External System Orchestration	Built-in hookscripts	Requires operators/controllers
Identity-Integrated RBAC	Infrastructure-aware (FreeIPA)	API-object focused RBAC
Multi-Instance Parallel Provisioning	Native	Not environment-centric
Application Scaling	Via Kubernetes bubble	Core strength
Container Ecosystem	Through deployed clusters	Native
Full OS Control	Full operating system control	Container-level only
Sovereign / Air-Gapped Friendly	Designed for it	Possible, but operationally complex
Audit Logging of Environment Lifecycle	Blow & Burst tracking	Application-level logging

Operational Surface Area Analysis

Kubernetes Operational Layers

- Control plane
- etcd
- Container runtime
- CNI plugins
- Ingress controllers
- Cluster autoscalers
- External identity and backup integration

Cluster lifecycle must be managed separately.

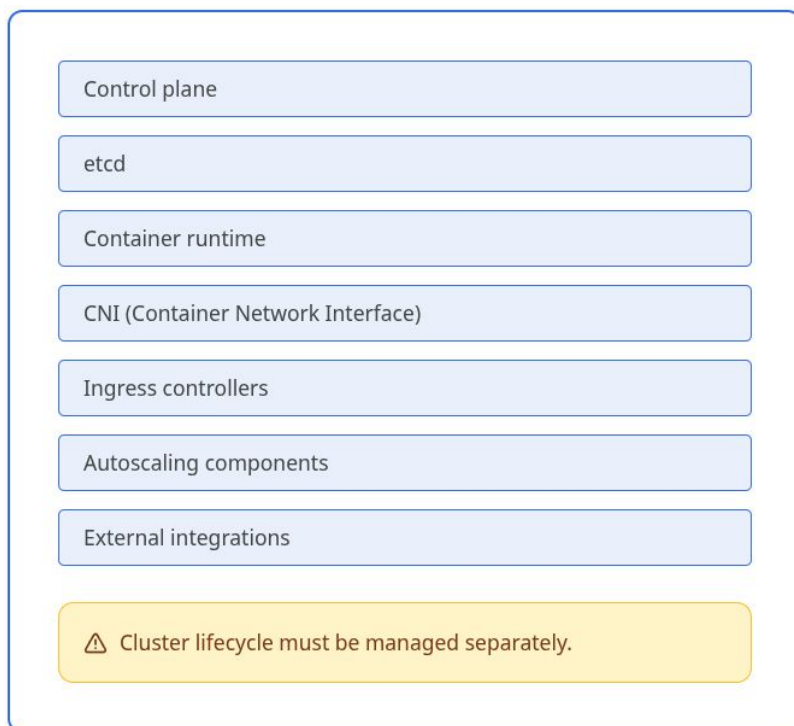
BubbleOps Operational Model

- Single automation control layer
- Direct integration with Proxmox
- Infrastructure-aware identity
- Backup enrollment at provision time
- Config management registration
- Governance at environment level

Lower operational fragmentation for VM-based environments.

Operational Surface Area Analysis

Kubernetes Operational Layers



BubbleOps Operational Model



Kubernetes Delivered Through BubbleOps

Kubernetes as a Managed Environment

BubbleOps can deploy full multi-node Kubernetes clusters as defined Bubbles.

This provides:

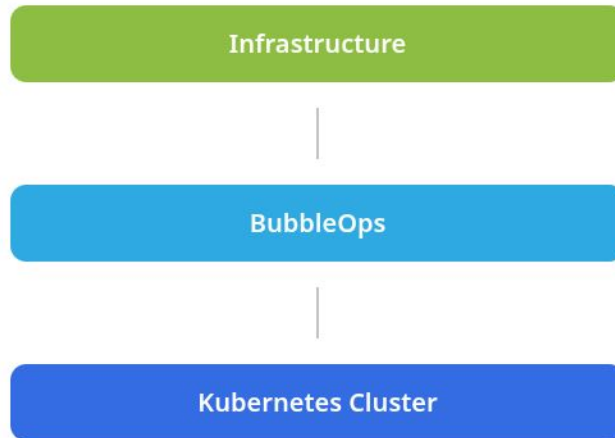
- Defined cluster templates
- Policy-driven placement
- Identity and monitoring enrollment
- Backup registration for cluster nodes
- Clean teardown symmetry
- Audit visibility of cluster lifecycle

Kubernetes becomes a workload layer inside a governed infrastructure envelope.

Kubernetes Delivered Through BubbleOps

Kubernetes as a Managed Environment

BubbleOps can deploy full multi-node Kubernetes clusters as defined Bubbles.



🔗 Provisioned + Governed + Tear-down symmetry

Key Capabilities

- ✔ Defined cluster templates
- ✔ Identity & monitoring enrollment
- ✔ Clean teardown symmetry
- ✔ Policy-driven placement
- ✔ Backup registration
- ✔ Lifecycle audit tracking

Sovereignty & Air-Gapped Considerations

Kubernetes in Sovereign Environments

Possible but requires:

- Manual registry configuration
 - Offline image management
 - Separate backup tooling
 - Cluster lifecycle scripting
-

BubbleOps in Sovereign Environments

Designed for:

- Customer-controlled infrastructure
- Air-gapped deployments
- No SaaS dependency
- No forced telemetry
- Infrastructure-first governance

Sovereignty & Air-Gapped Considerations



Kubernetes in Sovereign Environments

- Manual registry configuration
- Offline image handling
- Separate backup tooling
- Lifecycle scripting



BubbleOps in Sovereign Environments

- ✓ Customer-controlled infrastructure
- ✓ Air-gapped friendly
- ✓ No SaaS dependency
- ✓ No forced telemetry
- ✓ Infrastructure-first governance

Decision Framework

Choose Kubernetes When:

- Your architecture is container-native
- You operate microservices at scale
- You require pod-level auto-scaling
- You prioritize container ecosystems

Choose BubbleOps When:

- You operate Proxmox-based virtualized infrastructure
- You require integrated Day-2 automation
- You need governance at infrastructure level
- You prioritize sovereign deployment
- You want lifecycle symmetry

Use Both When:

- You need container orchestration
- But require infrastructure governance
- And want Kubernetes clusters provisioned and lifecycle-managed cleanly

Decision Framework

Choose Kubernetes When:

- Container-native workloads
- Microservices at scale
- Pod-level scaling requirements
- Container ecosystem adoption

Choose BubbleOps When:

- Proxmox infrastructure
- Day-2 automation required
- Infrastructure governance needed
- Sovereign deployment
- Lifecycle symmetry

Use Both When:

- ✓ You need container orchestration
- ✓ But require infrastructure governance
- ✓ And want clean lifecycle-managed clusters

Conclusion

Kubernetes and BubbleOps solve different problems.

Kubernetes orchestrates container workloads.

BubbleOps orchestrates infrastructure environments.

For organizations operating virtualized infrastructure, BubbleOps provides a direct and governed path to platform engineering maturity.

When container orchestration is required, Kubernetes can be deployed as a managed Bubble.

Infrastructure lifecycle and application orchestration do not need to compete... they can operate at complementary layers.

Call to Action

Start Your Sovereign Platform Journey

Learn how BubbleOps can automate and govern your infrastructure.

<https://bubbleops.io>