

## v3.4-OPS-002 – ZeroOps: Autonomous Twin Lifecycle & Self-Operating Clusters

Document Title	ZeroOps: Autonomous Twin Lifecycle & Self-Operating Clusters
Version	v3.4
Document ID	v3.4-OPS-002
Date	2025-03-22
Author	Take Back Your Data – Sovereign Ops Division
Document Type	Public / Certification / Internal

### 1. Purpose & Scope

This document defines the principles of ZeroOps within MaxOneOpen, focusing on self-operating clusters, autonomous twin lifecycle control and human-optional operations. It ensures forks can run, heal and scale without administrator intervention.

### 2. ZeroOps Principles

- No persistent human admin needed beyond setup
- Twin infrastructure must operate, scale and recover autonomously
- Manual override must be possible but optional
- Operation logs must be ZK-verifiable and policy-bound

### 3. Lifecycle Automation Flow

Phase	Trigger	Autonomous Action
Spawn	Manifest + load threshold	Instantiate twin & sync state
Operate	Policy rule or active user	Self-manage and log stats
Scale	Load, token, peer metrics	Activate clone or edge instance
Suspend	TTL or inactivity	Archive state & notify peers
Retire	Policy breach or expiry	ZK-proof offboard & revoke rights

### 4. Self-Operating Cluster Logic

- Forks must expose cluster graph with autonomous twin routing
- Clusters handle request routing, fallback, and version mismatch resolution
- Scaling patterns are defined per namespace and schema context
- Forks must support quorum-based override for edge clusters

## 5. Certification Triggers

- Forks must demonstrate full lifecycle execution under ZeroOps
- Twin suspension, scaling and migration must be policy-driven
- Forks lacking automated fallback or routing disqualify from certification

## 6. Certification Relevance

MaxOneOpen-certified forks must prove autonomous twin lifecycle handling and support for human-optional operation. Clusters without fallback, audit, and ZK-proofed operation traces are non-compliant.