KubeCon CloudNativeCon Europe 2026

#KubeCon #CloudNativeCon

Metal3.io's Path to CNCF Incubation: Governance, Processes, and Community

Kashif Khan

Metal3.io Journey to CNCF Incubation

- How we prepared Metal3.io's governance, security, and processes for CNCF Incubation
- Lessons other projects can reuse for moving from Sandbox → Incubation

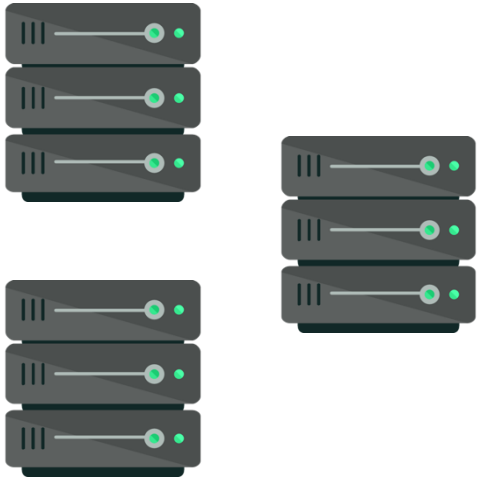
Who am I?

Kashif Khan
Maintainer, **Metal3.io**
Co-chair, **CNCF TAG Infrastructure**
Open Source Architect, **Ericsson**

https://www.linkedin.com/in/kashifnizamkhan/

Project Overview

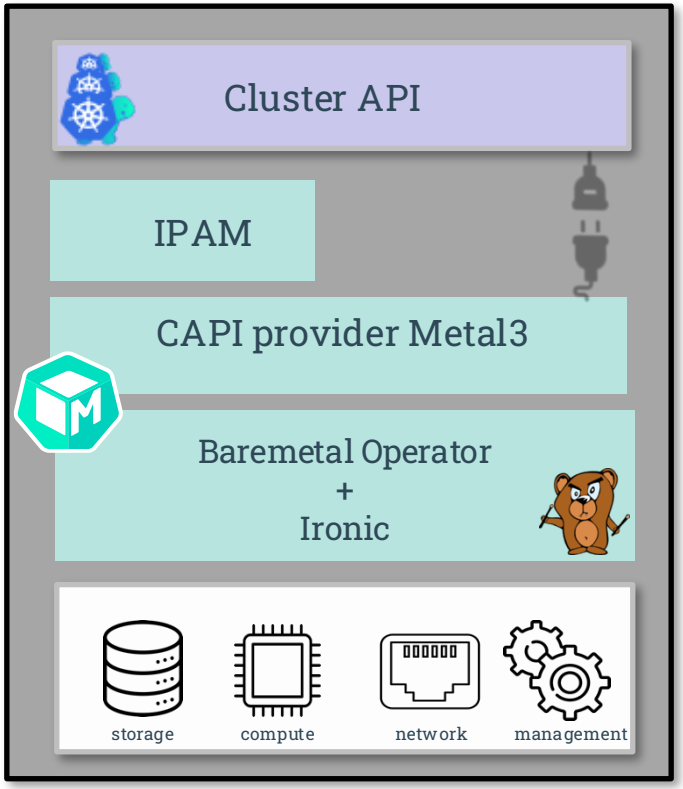

```yaml
apiVersion: metal3.io/v1alpha1
kind: BareMetalHost
metadata:
  finalizers:
  - baremetalhost.metal3.io
  generation: 1
  labels:
    cluster.x-k8s.io/cluster-name: test1
  name: node-0
  namespace: metal3
  ownerReferences:
  - apiVersion: infrastructure.cluster.x-k8s.io/v1alpha4
    controller: true
    kind: Metal3Machine
    name: test1-controlplane-s6tdz
spec:
  bmc:
    address: ipmi://192.168.111.1:6230
    credentialsName: node-0-bmc-secret
  bootMACAddress: 00:8e:50:0e:e8:3a
  bootMode: legacy
```



Project Overview

- Metal3.io (“Metal Kubed”) provides Kubernetes-native bare-metal host management
- Core components:
  - Baremetal Operator (BMO)
  - Cluster API Provider Metal3 (CAPM3)
  - IP Address Manager (IPAM)
  - Ironic Standalone Operator (IrSO)
  - Ironic Image

- Uses OpenStack Ironic under the hood, but packaged as a Kubernetes-native stack



Early history, path to sandbox

Motivation for Joining CNCF

- Align bare-metal infrastructure work with the broader Kubernetes ecosystem
- Neutral governance for shared assets
- Discoverability and trust for users choosing a bare-metal solution
- Access to CNCF services: infra, programs, and community channels

The “Code First, Governance Later” Start

- Project bootstrapped in early 2019 focused on building useful code
- Governance and formal processes were deliberately de-prioritized at start
- After ~1 year, multiple companies were actively contributing
- That growth triggered the need to “get serious” about governance and long-term community health

Sandbox Foundation Proposal

- Wrote a dedicated “Community Future / CNCF Sandbox” proposal
- Explicit goals:
  - Align policies and processes with CNCF ecosystem projects
  - Transfer key assets to a neutral foundation as appropriate
  - Improve visibility in the CNCF ecosystem to attract collaborators
- Non-goals: no code changes, no change to existing maintainer process

The CNCF Sandbox Application

- Clear description of scope:
  - Kubernetes-native bare-metal host management
  - CAPM3 as a Cluster API infrastructure provider
  - BMO as the BareMetalHost controller
- Infrastructure:
  - Dedicated Prow-based CI on donated hardware
  - Jenkins jobs for integration tests
  - Website, project-infra, and docs repos explicitly documented
- Governance and maintainers:
  - OWNERS files per repo for committers/maintainers
  - Documented process for adding/removing maintainers in metal3-docs

Path to Incubation

Incubation Requirements

- Governance and Maintainers
- Engineering Principle
- Security
- Contributors and Community
- Ecosystem

Governance and Maintainers: Creating a Dedicated Community Repo

- New “community” repo created to centralize:
  - Governance and policy docs
  - Contributor ladder and roles
  - Adopters, roadmap, vendor-neutrality
- This separation made:
  - Governance review easier (one canonical place)
  - CNCF and TAGs able to see how the whole org is run, not just a single code repo (specially for a multi-repo project like metal3.io which doesn’t have any main repo)

Governance and Maintainers: Governance Documents

- Governance overview:
  - Metal3 Project Governance doc cataloguing all top-level policies:
    - Code of Conduct
    - Contribution Guide
    - Contributor Ladder
    - Contributor Role definitions
    - DCO and License
    - Maintainers Guide
    - Project Roadmap
    - Project Adopters
- Maintainers Guide:
- DCO + Apache-2.0 license across the org for consistency

Contributors and Community: Formal Contributor Ladder

- Contributor Ladder doc defines levels:
  - Community Contributor
  - Organization Member
  - Reviewer
  - Approver
- Each level has:
  - Requirements (activity / behavior expected)
  - Qualifications (what you must have done)
  - Privileges (what you can do in the project)
- Promotion process:
  - Org Membership via issue in the community repo
  - Reviewers/Approvers via OWNERS PRs for the specific repo

Contributors and Community: Community Health & Transparency

- Communication channels:
  - Kubernetes Slack: #cluster-api-baremetal
  - metal3-dev mailing list
  - Weekly community meeting
  - meeting notes
  - YouTube playlist for recordings
- Adoption & roadmap:
  - ADOPTERS doc
  - ROADMAP doc
- External health indicators:
  - CLOMonitor
  - LFX Insights
  - CNCF project health

Engineering Principles

- **Suggested**
  - Roadmap change process is documented and community-driven
  - Proven history of regular, high-quality releases across components (CAPM3, Baremetal Operator, IPAM, Ironic Image)
- **Required**
  - Project Goals
  - Use Cases
  - Roadmap
  - Architecture
  - Release Process

Security Governance

- Separate “Contributor Role” document:
  - Explains specific roles (e.g., Security Team) and expectations
- Security Team specifics:
  - Dedicated security mailing list: metal3-security@googlegroups.com
  - Role: respond to all security reports, manage disclosure process
  - Designated Security Lead (documented by name in the role doc)
  - Access model: security-specific GitHub team (metal3_security_team)

Security: Policy, Insights, and Self-Assessment

- Central Security Policy:
  - Public “Security Policy” page in the Metal3 user guide / book
  - Linked from self-assessment and **security-insights.yml** metadata
- CNCF TAG-Security alignment:
  - Comprehensive Metal3 Security Self-Assessment document
  - This serves as the basis for the joint TAG-Security assessment used in incubation

Ecosystem

- **Adoption**
  - Public adopters list maintained
    Additional active adopters contribute without being formally listed
- **Ecosystem Usage**
  - Adopted by multiple open source and industry projects
    Airship, OpenShift, Sylvia-Projects etc
- **Adoption Maturity and Integration**
  - Used by 3+ independent adopters
  - Integrates with Cluster API

Quick start guide

- Started with a separate repo that had a mix of bash and ansible
  - Limitation: only supports VM emulation out of box
  - Inflexible (Ansible hidden behind bash)
- Iteration 2: a separate quick start documentation
  - Targeting both operators and developers
  - Supports both emulation and real hardware
  - But: duplicates both the old quick start and the installation guide
- In progress now:
  - Single quick start guide, CI tested
  - Less manual actions: simpler BMO start-up, new operator to manage Ironic

CNCF Incubation Evidence

- Governance & maintainers
  - 17 active maintainers, lifecycle documented (onboarding, offboarding, emeritus)
  - Governance and vendor-neutrality codified in the metal3-io/community repo
  - Clear mapping from OWNERS/permissions to documented roles
- Contributors & community
  - 426 contributors from ~60 organizations (DevStats)
  - 900+ stars / 450+ forks across repos, steady 1-year contribution rate
  - Adopters tracked in ADOPTERS.md plus a list of OSS projects using Metal3 (CAPI, Airship, OpenShift, etc.)
- Engineering & docs
  - Public roadmap and roadmap process, per-repo release pages
  - Detailed user guide and install docs (book.metal3.io)
  - Documented release process in metal3-docs
- Security
  - Public security policy and dedicated Security Team
  - Completed Metal3 security self-assessment used for TAG-Security
  - OpenSSF Best Practices *passing* badge (this is repo specific)

New Process

Governance Review vs Technical Review (New CNCF Model)

- Today's CNCF process:
  - Governance review: TAG Contributor Strategy / governance reviewers
  - Technical review: domain TAGs and TAG-Security for security aspects
  - TOC evaluates the reviews
- How Metal3's work maps to that:
  - Governance review:
    - community repo (governance, ladder, roles, roadmap, adopters)
    - vendor-neutrality policy and maintainer data
  - Technical review:
    - metal3-docs, architecture descriptions (BMO, CAPM3, IPAM, Ironic)
    - CNCF Sandbox application and design docs
  - Security review:
    - security self-assessment + org-wide security metadata
- Even though Metal3 started under an earlier process, the artifacts line up cleanly with this new two-track review model.

Learnings

What Worked Well for Metal3

- Create a dedicated community/governance repo per org early
- Treat:
  - Governance
  - Contributor ladder
  - Security processes
    as first-class “features” of the project
- Draft CNCF applications and proposals in public in your own repo
- Use self-assessment style docs for:
  - Security posture
  - Architecture overview
  - Project health narrative
- Automate what reviewers will ask for:
  - security-insights metadata validation
  - maintainers CSV alignment with CNCF

Metal3-Specific Lessons Learned

- If We Were Starting Again Today, we would:
  - Introduce governance and community repo earlier, not a year in
  - Design security metadata and self-assessment with TAG-Security in mind from day one
  - Align release and CI patterns with CNCF expectations from the start
  - User-first quick start guide instead of focusing on the CI usage
- But the staged approach still worked:
  - Code first, then governance and foundation proposal
  - Sandbox application as a forcing function
  - Incubation preparation as a deep health check

A Practical Checklist for Your Project

- Governance & Community
  - Public governance doc listing all key policies
  - Contributor ladder and roles (including Security Team, if relevant)
  - Maintainers guide and OWNERS-based process
  - Roadmap and Adopters docs
- Security
  - Public security policy and contact
  - Self-assessment (security posture, data flows, threat model)
  - Org-wide security metadata (similar to security-insights.yml)
- Operations
  - Documented release process and cadence
  - CI that enforces reviews, tests, and DCO
  - Clear separation between community repo, project-infra, and code
- CNCF Alignment
  - Draft application checked against current CNCF criteria
  - Evidence for governance, technical, and security reviews collected in one place

Summary

Key Takeaways

- CNCF readiness is as much about governance and security as it is about code
- A dedicated community repo made our story easy to review and reason about
- The security self-assessment and governance docs aligned naturally with CNCF's newer governance/tech review split
- Investing in clear roles, ladders, and release/CI processes increases trust, sustainability, and adoption


KubeCon
CloudNativeCon
Europe 2026