#KUBECON #CLOUDNATIVECON

Agentic SDLC for Open Source Projects

What maintainers need to know

Karena Angell • Joseph Sandoval

KubeCon CloudNativeCon North America 2026
MAINTAINER SUMMIT

This session is about the agents you run yourselves, and where your judgment still makes the call.

The TOC has a review queue too

200+

Active CNCF projects

13

Volunteer committee members

01 **Sandbox Applications**

New projects asking to join the CNCF.

02 **Due Diligence**

Evidence and adopter interviews before level progression.

03 **Health Reviews**

Shifting from reactive triage to continuous evaluation.

Principles of our tooling

01

Agents gather, humans decide

Deterministic checks first. A model drafts. A person decides and owns it.

02

Every finding cites evidence

The exact repo, file, issue or run it came from, or it does not count.

03

Unknown is never a pass

Missing evidence reads exclusively as unknown.

04

Keep the record and the cost

Inputs committed, outputs regenerable, cost strictly metered.

How our work actually flows

01

Inbound

Assigned issues, docs, portfolio readiness.

02

Agent Gathers

Signals checked against baseline. Sources linked.

03

Agent Drafts

Proposal drafted only when rule-triggered.

04

Person Decides

Interviews, judgment, vote. Accept or edit.

05

Record

Inputs committed. Triage doc acts as archive.

Agents do the gathering and drafting. **Every decision has a person's name on it.**

What a skill is, and when you need one

What it is

- Instructions on what to check and evidence required.
- Written once, read by any agent, reviewed like code.
- Your CONTRIBUTING.md is already half of one.

When to reach for one

- The exact same judgment call keeps coming back.
- Knowledge exists solely in one maintainer's head.
- You need consistent answers regardless of who runs it.

WHERE A HUMAN MADE THE CALL

A healthy project the numbers said was struggling

01

Agent Report

Flagged project for low pull-request volume and slow portfolio movement.

02

Human Context

It is a stable specification. Finished specifications are supposed to be quiet.

03

The Decision

Adjusted baselines. Specification projects are now banded against their own mean.

What an eval is, and when you need one

The Mechanism

- A folder of past cases where you know the answer.
- A few targeted checks the output has to pass.
- A table a person reads to fix the skill.

The Trigger

- Before you trust a skill on anything that ships.
- **When the output reads right, but you cannot verify it.**
- Every time the skill or the criteria change.

What an eval actually looks like

Test Case Definition

```
# evals/refarch-review/case-014.yaml
input:
submission: cncf/architecture#14
expected:
# what the human reviewer asked
must_flag:
- no multi-cluster failure handling
- no observability retention numbers
decision_owner: human
```

Evaluation Results

Case	Evidence	Gaps	Human Call
case-009	PASS	PASS	PASS
case-014	PASS	MISSED 1	PASS
case-021	UNKNOWN	PASS	FAIL

UNKNOWN never counts as a pass.

Gaps measure what the human caught, quoting the exact line.

It never initially looks suspicious, aka “confidentally correct”

3 real failures

0 looked uncertain in the output

CASE 01

Invented Session

A schedule summary invented a session: title, speakers, track, timing.

CASE 02

Broken Public Link

A file renamed mid-task broke a public link.

CASE 03

Chart Mismatch

A README that disagreed with what its own install chart pinned.

All three from building an inference test lab, October 2026, for KubeCon “Inference on Kubernetes” session on November 11, 2026 at 5pm.

What to watch once agents are running

What to watch

Cost per task.
Track which calls are expensive.

Override volume.
How often and what you override.

Quality drift.
Monitor deviations between runs.

How to see it

Distinct identity.
Give the agent its own repo account.

Propagated traces.
One trace ID passed at every hop.

Public corrections.
Keep the record where anyone can read it.

Lint your skills the way you lint your code

Frontmatter and structure

Check it before your agent runs.

Prompt injection & encoded payloads

Blocks invisible unicode and hidden execution instructions.

Download hooks & unvetted servers

Prevent curl-piped-to-shell in lifecycle hooks.

Token budget & committed secrets

Flag credential leakage into agent context.

Example linter: skillsaw, github.com/stbenjam/skillsaw (Apache-2.0)

Start with your review comments

Your past review comments are already corrections - now turn them into binary checks.

Sourcing

Does every single finding link directly to the file or issue it came from?

BINARY CHECK
Direct link attached?

Completeness

Does the output explicitly flag what is missing, as well as what is present?

BINARY CHECK
Missing & present flagged?

Boundaries

Does the agent strictly leave the community fit and health calls to a person?

BINARY CHECK
Left for human review?

What we see projects leaning on

DOMAIN	PROJECT	IMPLEMENTATION
POLICY	**Kubernetes**	Requires disclosure. Forbids AI co-authors via CLA check.
POLICY	**KubeVirt**	Accepts AI co-authors. Requires DCO and explicit PR checkboxes.
CONTEXT	**NATS**	Publishes formal agent guidance alongside governance models.
SCOPE	**Harbor Satellite**	Marks every design decision distinctly as shipped or proposed.

What to do with your project on Monday

STEP 01



Agree on the problem and scope before an agent writes anything.

STEP 02



Write down what you reject and why. That is your first skill.

STEP 03



Test against past PRs and dig into where the agent disagrees.

STEP 04



Give agents their own identity and log exactly what they touch.

STEP 05



Decide out loud where a human makes the call in your contributing guide.

Sources and further reading

TOPIC	PROJECT OR AUTHOR	REFERENCE
POLICY	**Kubernetes**	“Open source maintainership in the age of AI”, kubernetes.io/blog
POLICY	**KubeVirt**	ai-contribution-policy.md, github.com/kubevirt/community
CONTEXT	**NATS**	AGENTS.md and GOVERNANCE.md, github.com/nats-io/nats-server
SCOPE	**Harbor Satellite**	CLAUDE.md, github.com/container-registry/harbor-satellite
LINTING	**skillsaw**	Apache-2.0, github.com/stbenjam/skillsaw, skillsaw.org/ci
SKILLS	**kubevirt/ci-health**	.claude/skills and .gemini/skills, Apache-2.0
READING	**Background**	LF Research · CNCF TAG DX · Kyverno · OpenSSF Privateer · The New Stack

Thank you

Questions? Bring them to the TOC Q&A at 1:35.

Karena Angell · Joseph Sandoval