



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

# Pull Request Wars: The Flux Awakens

## Ephemeral Environments Strike Back



# A Long Time Ago, in a Cluster Far, Far Away...

## Stefan Prodan

*The Jedi Master of GitOps*

Principal Engineer @ ControlPlane

- ▶ Creator of **Flagger**, progressive delivery for K8s
- ▶ Core maintainer of the **CNCF Flux project**
- ▶ 20+ years of software engineering
- ▶ Architect of Flux Operator



## Matteo Bianchi

*The Rebel Engineer*

Solutions Engineer @ GitHub

- ▶ CNCF Ambassador
- ▶ Former startup CTO and (semi-retired) DevRel
- ▶ Open Source contributor in Kubernetes (RelEng)
- ▶ Former member of **K8s Release Team** (v1.31–v1.34)



# Platform Teams in the AI era

*“The more you tighten your grip on your CI/CD pipelines, the more developer experience slips through your fingers.”*  
- Probably Obi-Wan, if he was a Platform Engineer

## Challenges:

- ▶ **Pipelines are hard to maintain** and inconsistent across teams
- ▶ **Shared staging environments** are slow, flaky, and create bottlenecks
- ▶ **Long review cycles** mean production bugs survive too long
- ▶ App teams want **self-service** but platform teams need **governance**
- ▶ AI and agentic coding makes it so easy to open a PR. Proper testing has become fundamental to **avoid slop**

## The answer:

A declarative, PR-driven, GitOps-native preview environment owned by the platform, self-served by developers by applying a single label on a Pull Request.

# The Repos

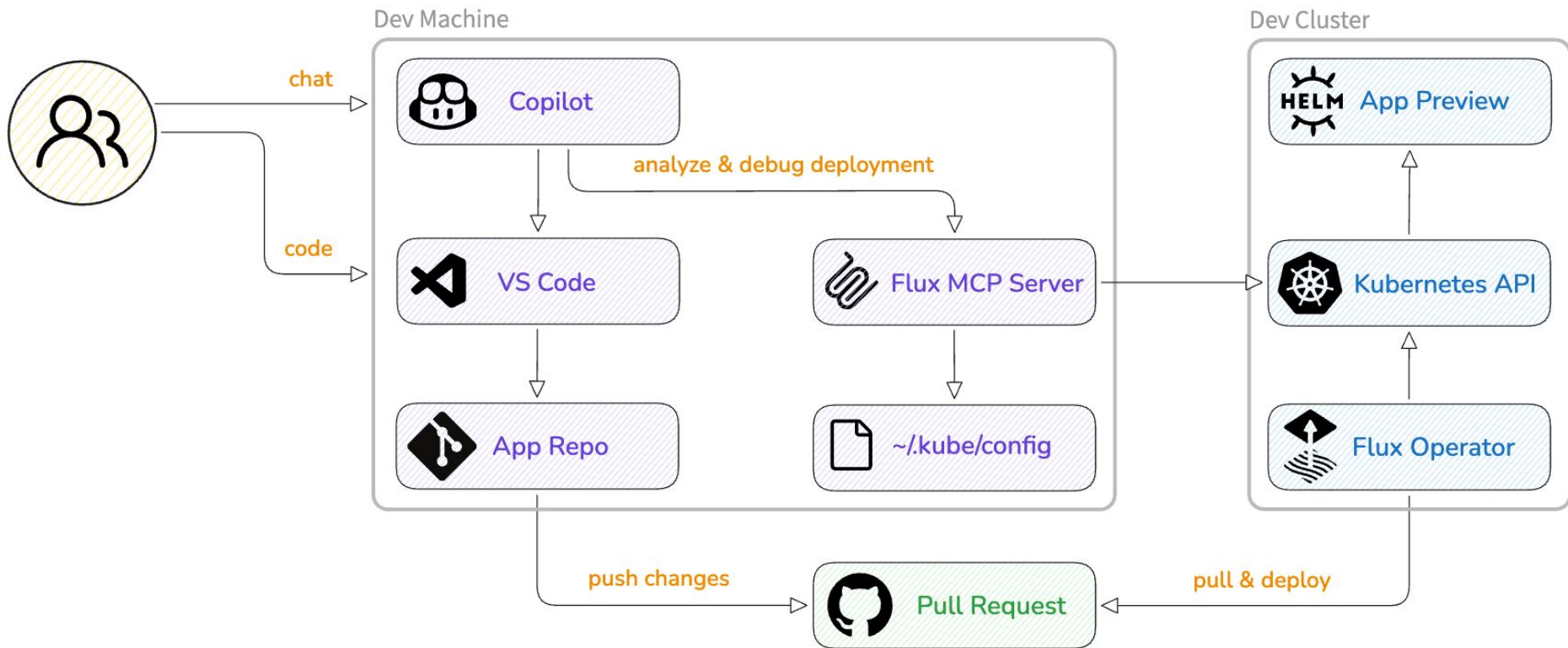
*Three repositories. One mission. Zero bespoke pipelines. Governance by default.*

- ▶ **platform** [[repo](#)] - Flux Operator config, cluster bootstrap, infra, ResourceSets, owned by the **platform team**
- ▶ **appx** [[repo](#)] - the app, where **developers** build and can open labeled PRs to get preview environments
- ▶ **charts** [[repo](#)] - generic Helm chart for every preview environment, also owned by the platform team

**Demo time!**  
**[github.com/pull-request-wars-fluxcd](https://github.com/pull-request-wars-fluxcd)**



# Dev Workflow: Diagram



# Dev Workflow: Explained

## From Code to Running Preview

1. **Developer** writes code in VSCode → pushes → opens a PR and labels it
2. **GitHub Copilot** assists in IDE with debug capabilities via Flux MCP Server
3. **Flux MCP Server** connects to K8s API on the dev cluster
4. **Flux Operator** pulls from PR and deploys App Preview
5. Flux posts commit status back → dev sees  or  in GitHub

**Key insight:** The PR acts as the single control plane.  
No dashboards, no tickets, no manual approvals.

## The (GitOps) Rebellion Wins

- ▶ **Pipeline maintenance:** High toil → Zero (one ResourceSet)
- ▶ **Feedback loop time:** Hours → Minutes (per-PR)
- ▶ **Governance:** Hard to enforce → Declarative, in Git
- ▶ **Self-service:** Needs platform team → Add a label, done
- ▶ **Cleanup:** Manual → Automatic on PR closed

### Choose your own adventure and infra:

- ▶  Ephemeral namespaces or entire clusters per PR
- ▶  Full ephemeral cluster (e.g. with vCluster)
- ▶  Flux MCP Server + GitHub Copilot for debug (with any model bus Opus 4.6 definitely does it better)
- ▶  Works on KinD, k3s, k0s, any K8s flavor

# Force of ResourceSets

## ResourceSetInputProvider

Watches GitHub PRs matching a filter:

```
apiVersion: fluxcd.controlplane.io/v1
kind: ResourceSetInputProvider
spec:
  type: GitHubPullRequest
  filter:
    labels:
      - "deploy/flux-preview"
  reconcileEvery: "1m"
```

### Template inputs per PR:

```
inputs.id, inputs.sha, inputs.branch, inputs.author, inputs.image,
inputs.repo, inputs.replicas
```

## ResourceSet

*A.k.a. the Clone Army*

Renders K8s resources per qualifying PR:

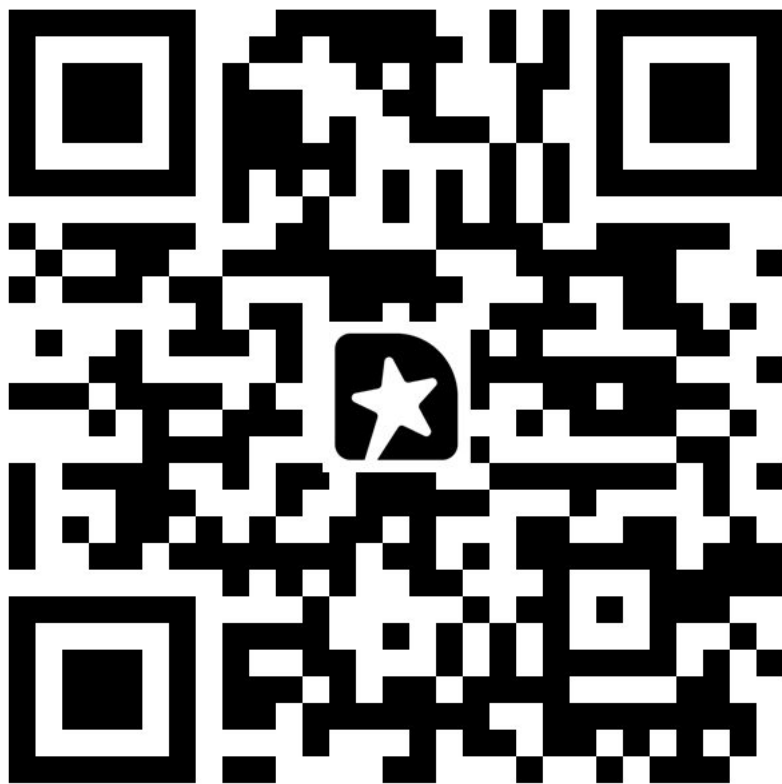
- ▶  **HelmRelease** — deploys from generic Helm chart
- ▶  **Provider** — posts PR comment status
- ▶  **Provider** — updates commit status checks
- ▶  **Alert(s) × 2** — reconciliation events

### Multi-tenancy boundaries:

- ▶ Isolated apps-preview namespaces with RBAC
- ▶ App teams never see or touch secrets
- ▶ Platform owns templates, RBAC, policies

# Resources

- MCP docs <https://fluxoperator.dev/mcp-server/>
- Flux operator repo <https://github.com/controlplaneio-fluxcd/flux-operator>
- Demo org (repos are within the org) <https://github.com/pull-request-wars-fluxcd>



Pull Request Wars: The Flux Awakens - Ephemeral Kubernetes  
Environments Strike Back