THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
KOREA

서울

Self-Healing Rollouts: Automating Production Fixes With Agentic AI

Kevin Dubois
Sr Principal Developer Advocate, IBM

kevindubois

Kevin Dubois

- ★ Sr. Principal Developer Advocate at IBM
- ★ Java Champion
- ★ Technical Lead, CNCF DevEx TAG
- ★ Agentic AI Foundation Ambassador
- ★ From Belgium 🇧🇪 / Live in Switzerland 🇨🇭
- ★ 🗣️ English, Dutch, French, Italian

Java Champions

Sessionize Most Active Speaker 2025, 2024, 2023

Agentic AI Foundation Ambassador 2026

Red Hat Certified Specialist — Containers and Kubernetes

Microsoft Certified Azure Fundamentals

Docker Certified Associate

@kevindubois.com

@thekevindubois

kevindubois

kdubois

THE LINUX FOUNDATION OPEN SOURCE SUMMIT KOREA

'IT대란' 국내 피해 기업 10곳…블루스크린 조치 방법은

문세영 기자 2024.07.21 10:22

공유
가 글자크기 -
가 글자크기 +
인쇄
ENG
AiR 질문하기

마이크로소프트(MS) 클라우드 서비스 장애로 서비스에 차질을 빚었던 제주항공이 20일 복구로 다시 원활한 탑승 수속이 이뤄지고 있다. 연합뉴스 제공.

AI 기사 요약 AI가 핵심만 알려드립니다!

- 크라우드스트라이크 보안 업데이트 충돌로 MS 윈도에 블루스크린이 발생해 국내 10개 기업이 피해를 입었습니다.
- 과기정통부와 한국인터넷진흥원이 비상대응팀을 꾸려 블루스크린 원인 규명과 피해 접수, 사이버 공격 모니터링을 진행했습니다.
- 정부는 안전 모드로 부팅해 특정 크라우드스트라이크 드라이버 파일을 삭제하는 등 복구 절차와 보안 유의사항을 안내했습니다.

From 2h ago

08.47 CEST

Crowdstrike identifies issue behind worldwide outage

Josh Taylor

Cybersecurity software firm Crowdstrike has posted in its support updates that it has identified the issue behind today's massive, global Windows outage.

It is an issue with "content deployment" that has since had those changes reverted and has advised a workaround for affected users.

We are still waiting for confirmation or a statement from Crowdstrike's media team. It's almost midnight in the US, so it's not clear when we might get more detail.

+ Show more

What could have mitigated the impact?

6. Template Instances should have staged deployment

Findings: Each Template Instance should be deployed in a staged rollout.

Mitigation: The Content Configuration System has been updated with additional deployment layers and acceptance checks

Staged deployment mitigates impact if a new Template Instance causes failures such as system crashes, false-positive detection volume spikes or performance issues. New Template Instances that have passed canary testing are to be successively promoted to wider deployment rings or rolled back if problems are detected. Each ring is designed to identify and mitigate potential issues before wider deployment. Promoting a Template Instance to the next successive ring is followed by additional bake-in time, where telemetry is gathered to determine the overall impact of the Template Instance on the endpoint.

Progressive Delivery

Progressive Delivery ...

kevindubois

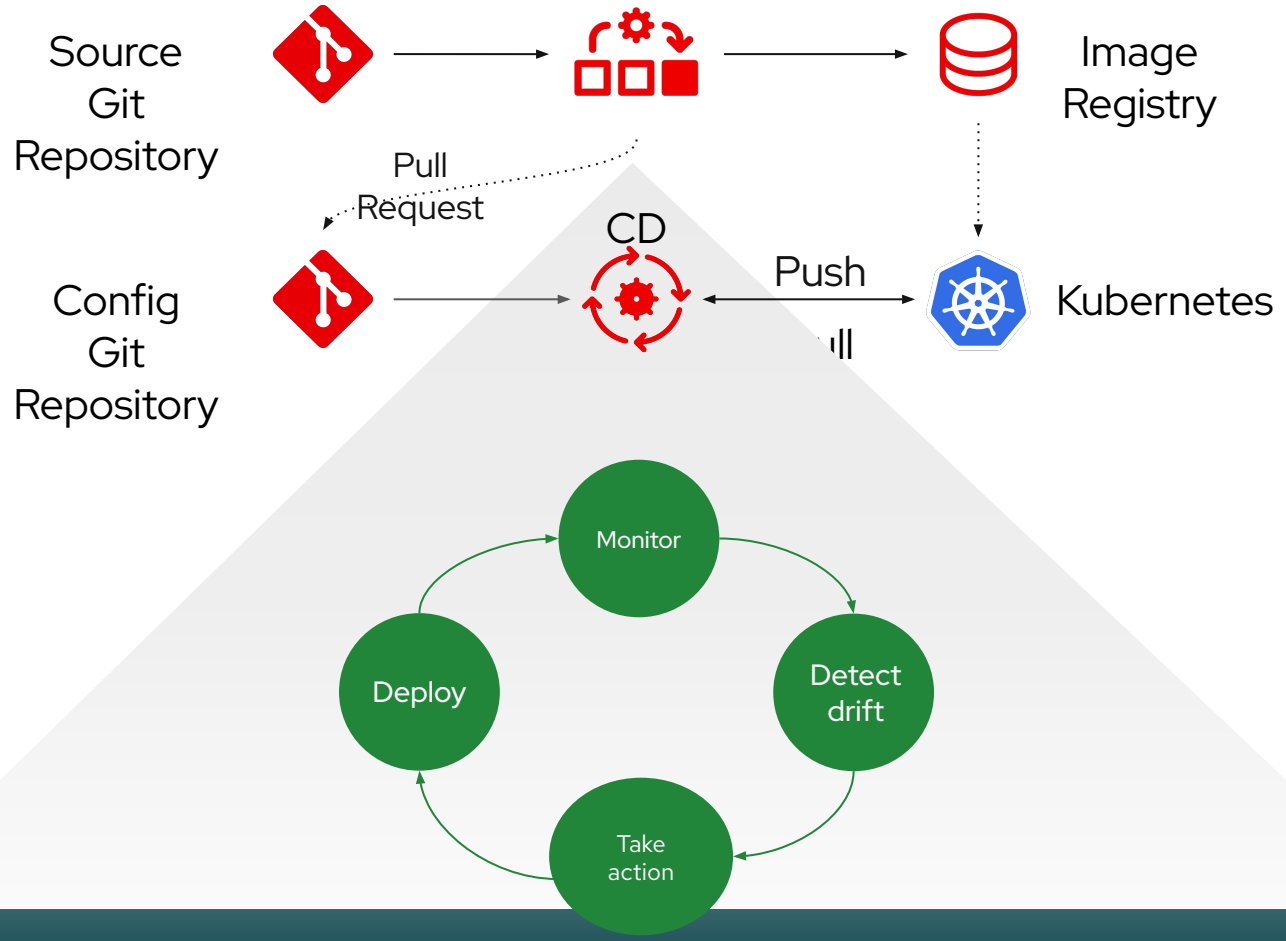
GitOps

Git is the single source of truth

Treat everything as code

Operations through Git workflows

GitOps Application Delivery Model

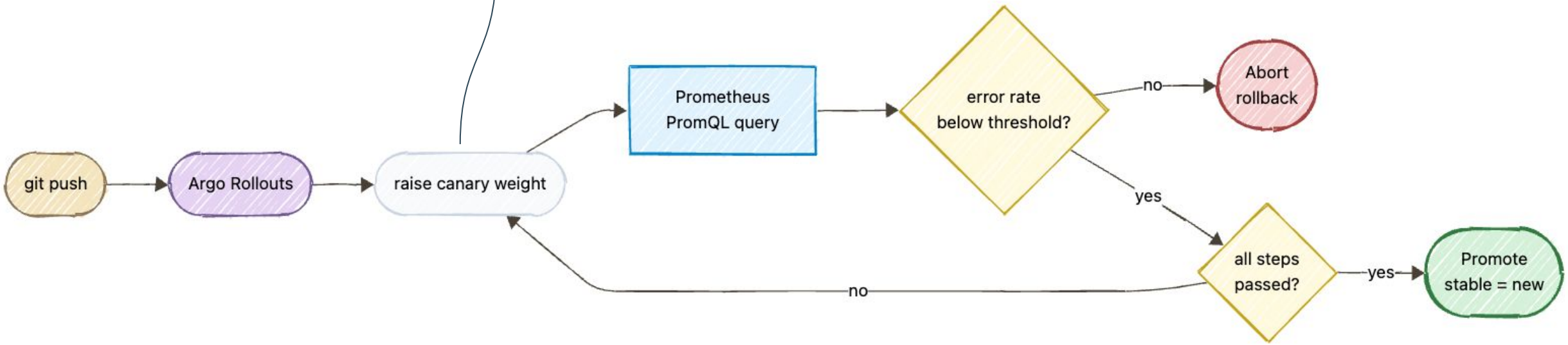


By default:
a big bang / all or nothing release

Argo Rollouts

flagger

Increase percentage of new (canary) version



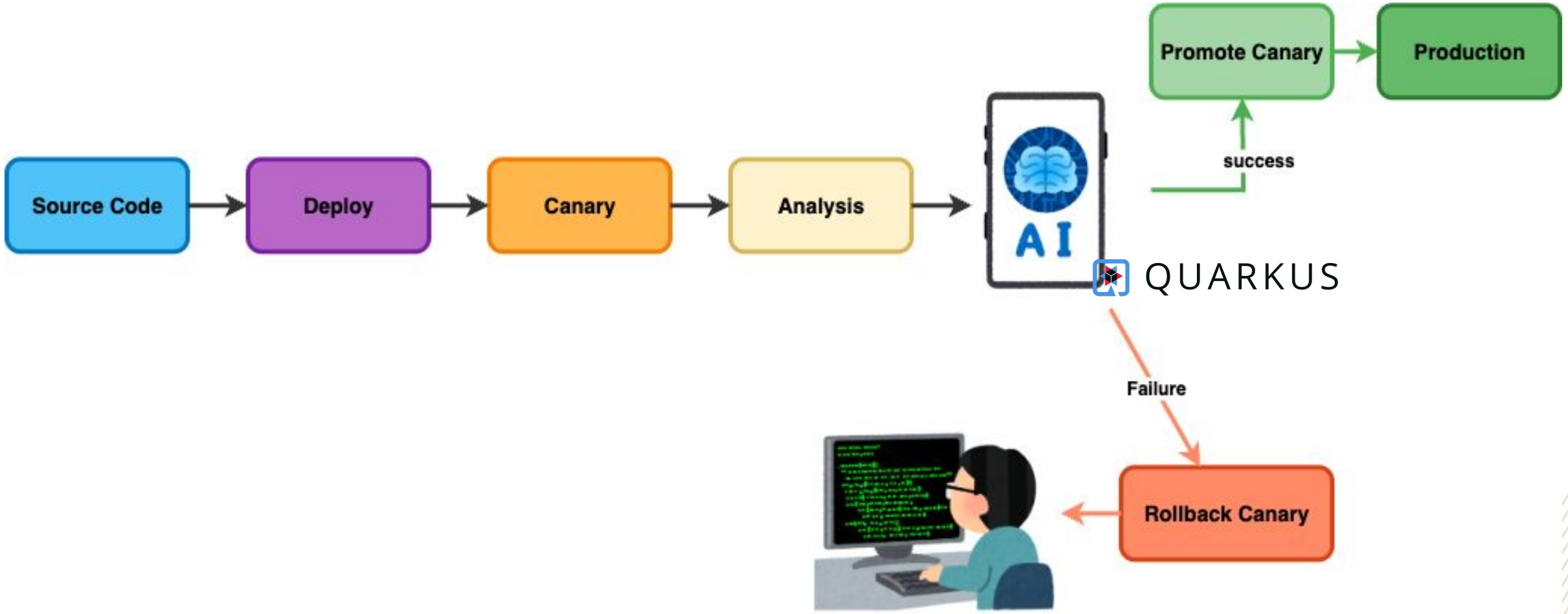

HEY KID
kevindubois
WANNA SEE SOME YAML?
imgflip.com

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
name: success-rate
spec:
args:
- name: service-name
metrics:
- name: success-rate
  interval: 10s
  successCondition: len(result) == 0 || result[0] >= 0.95
  failureLimit: 2
  provider:
    prometheus:
      address:
https://internal:demo@prometheus.istio-system.svc.cluster
.local:9090
      query: |
        sum(irate(istio_requests_total{
          reporter="source",
          destination_service=~"{{args.service-name}}",
          response_code!~"5.*"}[30s])
        )
```

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
name: success-rate
spec:
args:
- name: service-name
metrics:
- name: success-rate
  interval: 10s
  successCondition: len(result) == 0 || result[0] >= 0.95
  failureLimit: 2
  provider:
    prometheus:
      address:
https://internal:demo@prometheus.istio-system.svc.cluster
.local:9090
      query: |
        sum(irate(istio_requests_total{
          reporter="source",
          destination_service=~"{{args.service-name}}",
          response_code!~"5.*"}[30s])
        )
```

Write custom PromQL

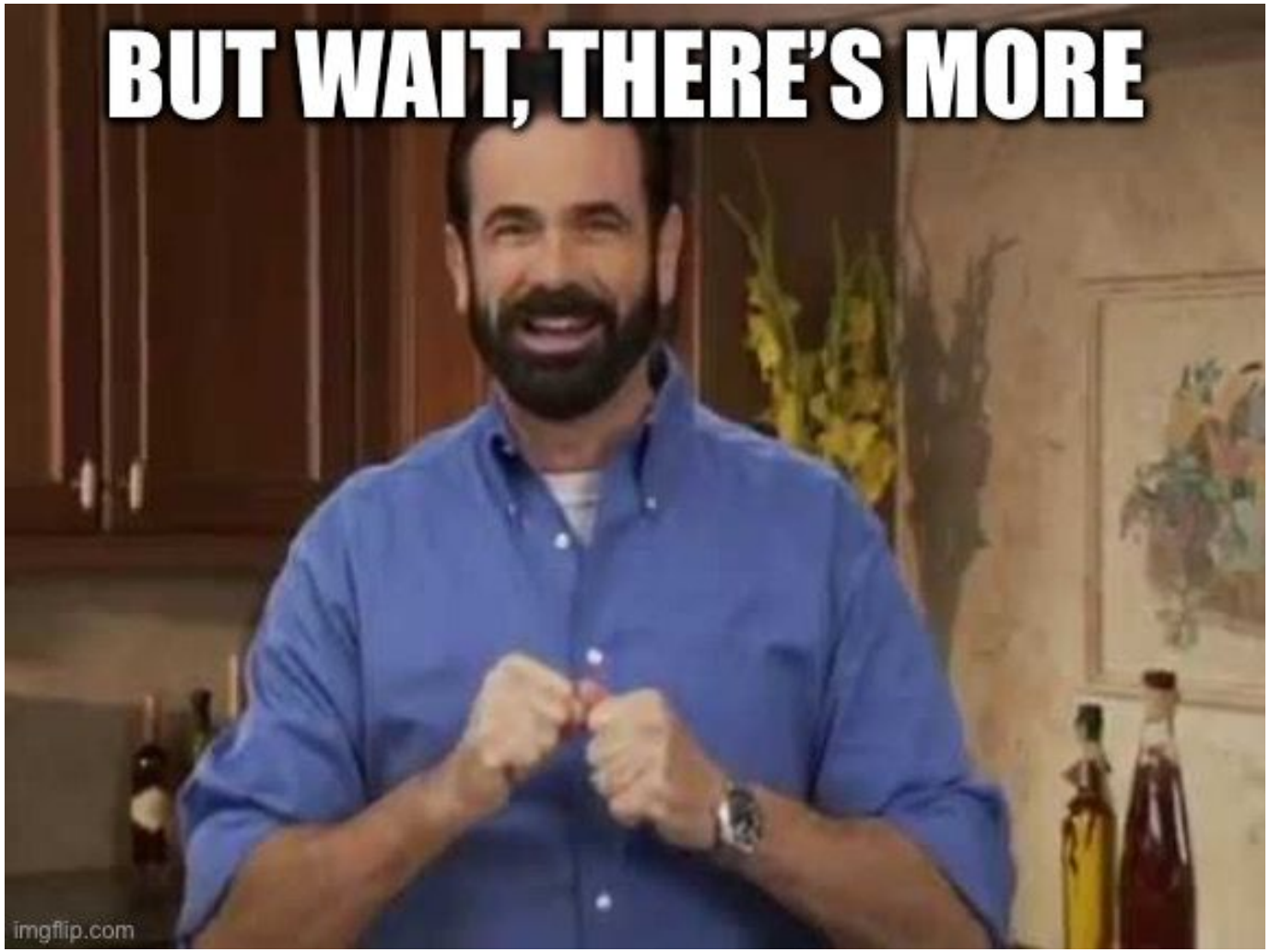
Use AI to analyze failures

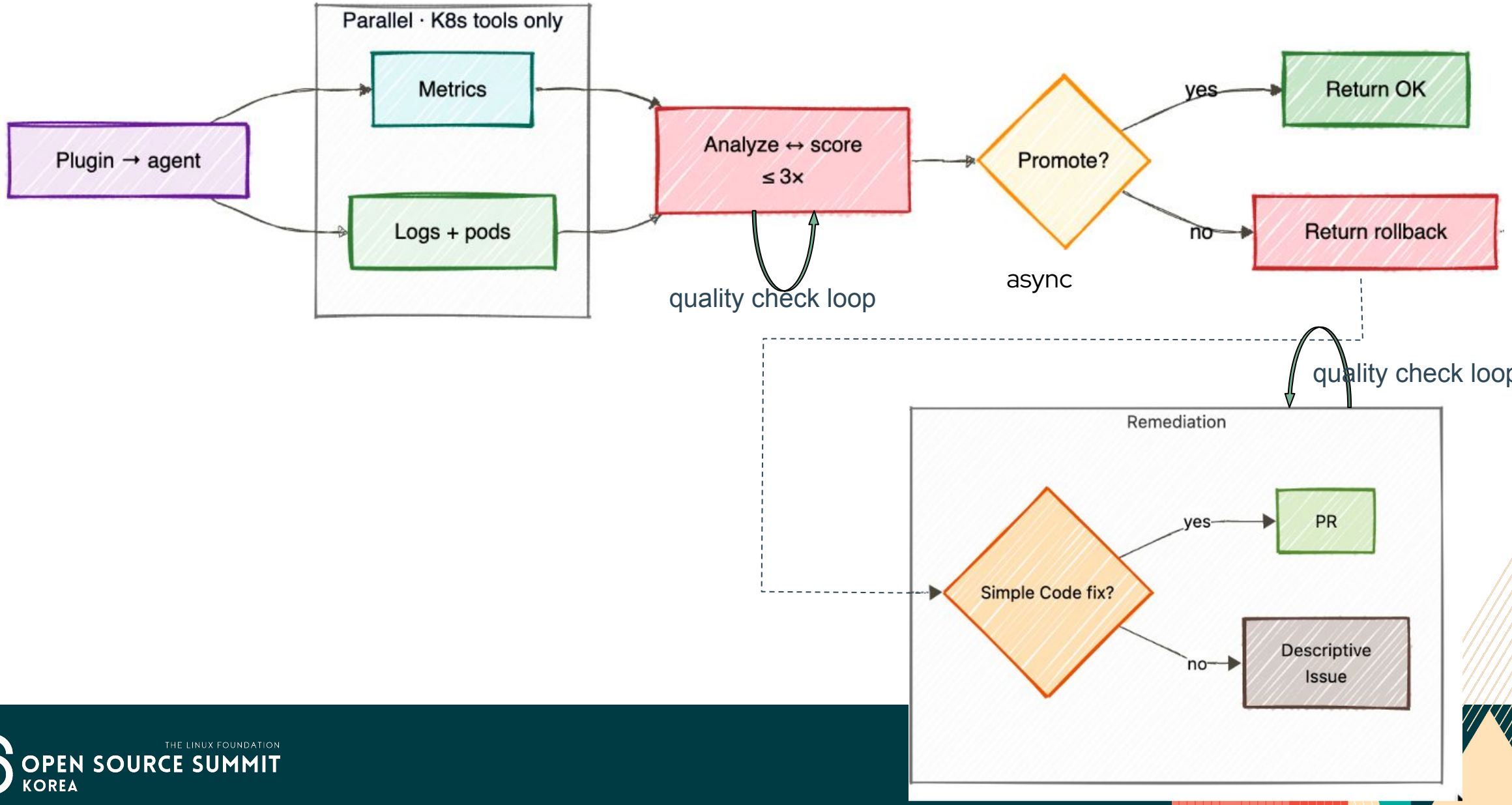

Source Code
Deploy
Canary
Analysis
AI
QUARKUS
Promote Canary
Production
success
Failure
Rollback Canary

```
apiVersion: argoproj.io/v1alpha1
kind: RolloutManager
metadata:
  name: argo-rollouts
spec:
  plugins:
    metric:
      - name: argoproj-labs/metric-ai
        location:
https://github.com/argoproj-labs/rollouts-plugin-metric-ai/releases/
download/v0.0.1/rollouts-plugin-metric-ai-linux-amd64
```

```yaml
canary:
  analysis:
    startingStep: 1
    templates:
    - templateName: canary-analysis-ai-agent
  canaryMetadata:
    labels:
      role: canary
  stableMetadata:
    labels:
      role: stable
```

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
 name: canary-analysis-ai-agent
spec:
  metrics:
  - interval: 10s
    name: success-rate
    provider:
      plugin:
        argoproj-labs/metric-ai:
          agentUrl: http://kubernetes-agent:8080
          stableLabel: role=stable
          canaryLabel: role=canary
          extraPrompt: ignore aesthetic changes
    successCondition: result > 0.50
```


BUT WAIT, THERE'S MORE
imgflip.com


Parallel · K8s tools only
Metrics
Logs + pods
Plugin → agent
Analyze ↔ score
≤ 3x
quality check loop
Promote?
yes
Return OK
no
Return rollback
async
quality check loop
Remediation
Simple Code fix?
yes
PR
no
Descriptive
Issue

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
 name: canary-analysis-ai-agent
spec:
  metrics:
  - interval: 10s
    name: success-rate
    provider:
      plugin:
        argoproj-labs/metric-ai:
          agentUrl: http://kubernetes-agent:8080
          stableLabel: role=stable
          canaryLabel: role=canary
          extraPrompt: ignore aesthetic changes
          githubUrl: …github.com/kdubois/argo-rollouts-quarkus-demo
```

Demo

Lessons learned:

Performance was an interesting challenge:

went from one AI service to agentic system: parallel & async agents

LLM choice + “context engineering” + tool calling

especially for PR creation

Complexity vs portability

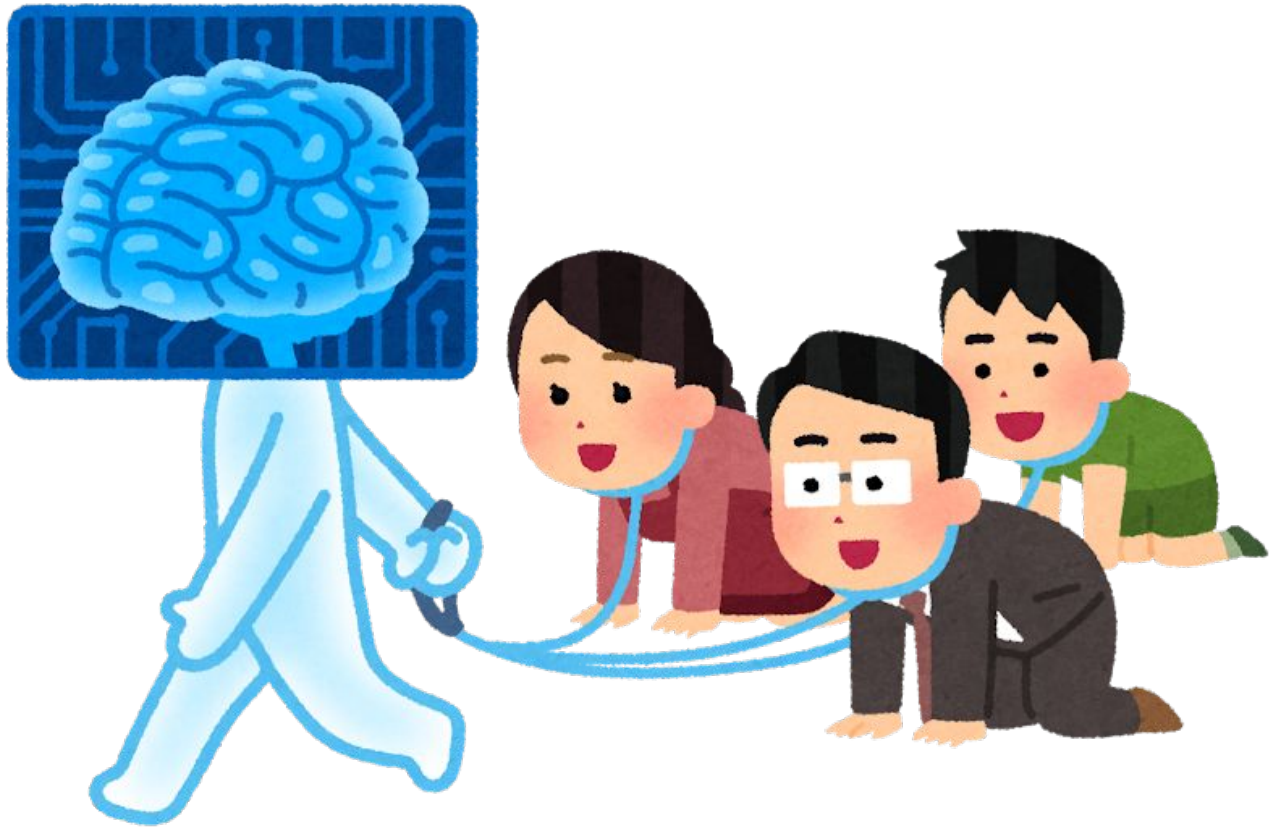
(could’ve used Serverless MCP, external code assistant for PR creation, async remote agents, etc.)

Takeaways:

Rolling out changes to **all** users **at once** is risky

Canary rollouts and feature flags are safer

AI Agents can automate the loop by analyzing metrics and logs, *and* even proposing fixes for the failures



kevindubois

github.com/argoproj-labs/rollouts-plugin-metric-ai/
github.com/kdubois/argo-rollouts-quarkus-demo
github.com/kdubois/kubernetes-aiops-agent
github.com/kdubois/progressive-delivery
https://quarkus.io/quarkus-workshop-langchain4j/

Self-Healing Rollouts: Automating Production Fixes with Agentic AI

feedback please :)

kevindubois kevindubois.com kdubois