ArgoCon
EUROPE

Beyond Argo Rollouts: Boosting Developer Experience With Intelligent AIOps

Carlos Sanchez, Adobe & Kevin Dubois, IBM

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

6. Template Instances should have staged deployment

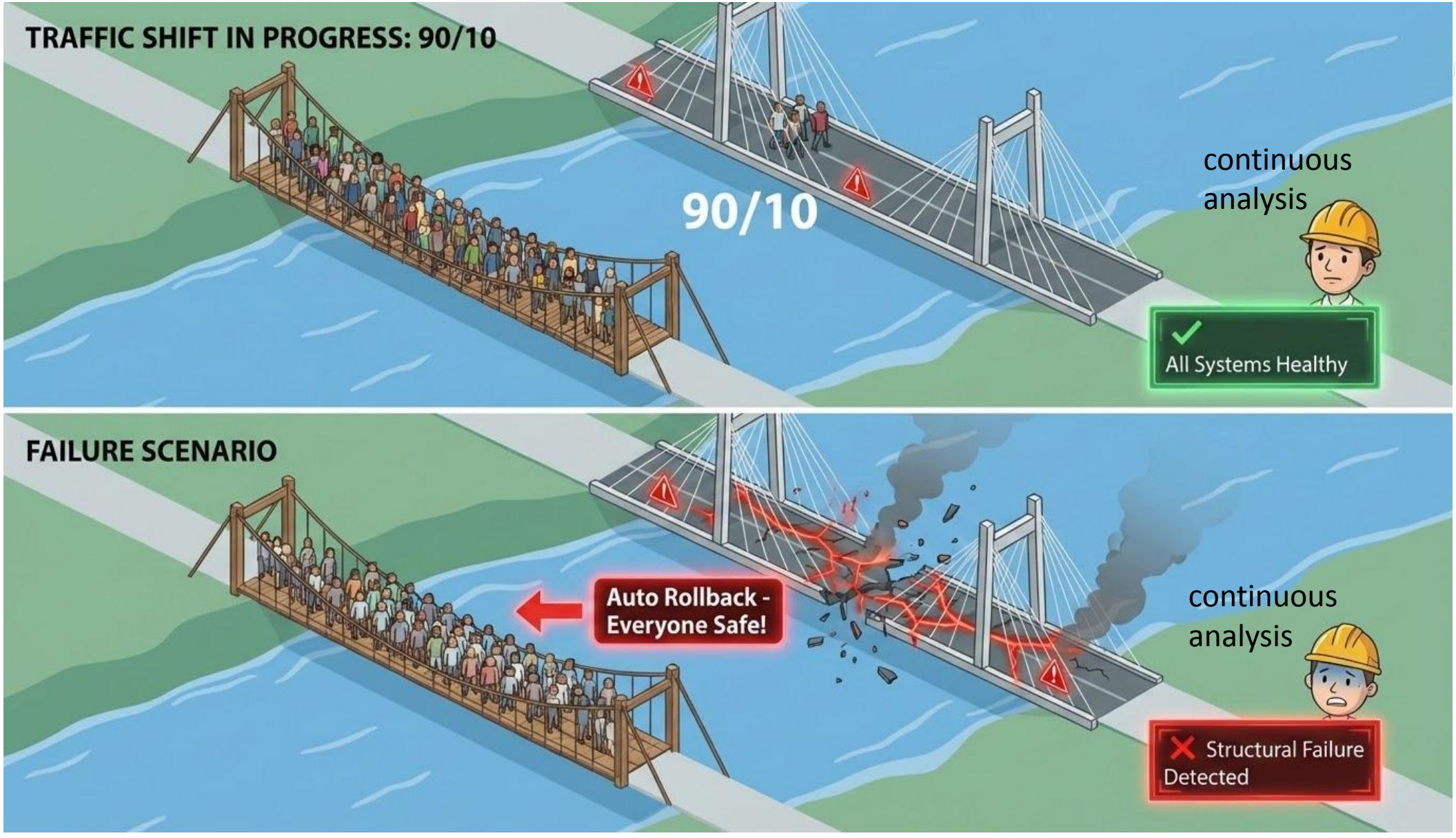
Findings: Each Template Instance should be deployed in a staged rollout.

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 with Argo Rollouts

Progressive Delivery with Argo Rollouts



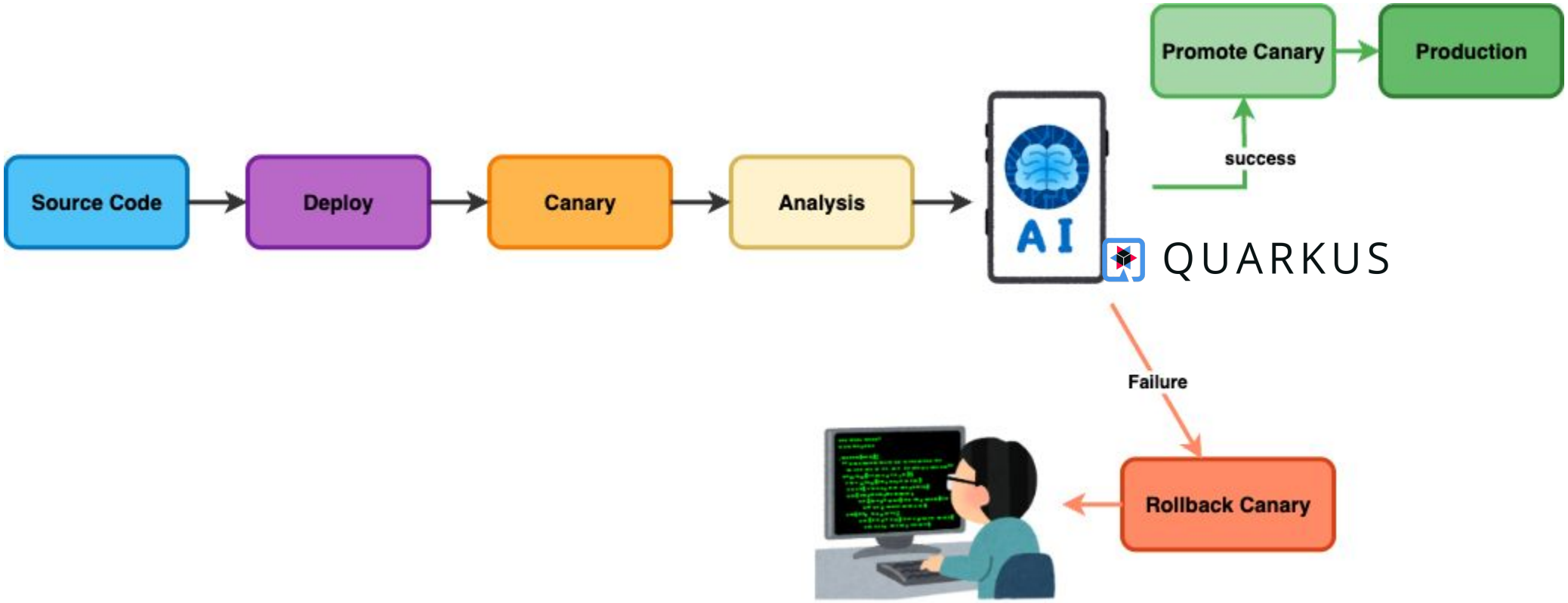
Metrics Based Rollouts

```
 strategy:
  canary:
    analysis:
      args:
      - name: service-name
        value:
rollouts-demo-canary.canary.svc.cluster.local
      templates:
        - templateName: success-rate
    canaryService: rollouts-demo-canary
    stableService: rollouts-demo-stable
    trafficRouting:
      istio:
        virtualService:
          name: rollout-vsvc
          routes:
          - primary
    steps:
    - setWeight: 30
    - pause: { duration: 20s }
    - setWeight: 40
    - pause: { duration: 10s }
    - setWeight: 60
    - pause: { duration: 10s }
    - setWeight: 80
    - pause: { duration: 5s }
    - setWeight: 90
    - pause: { duration: 5s }
    - setWeight: 100
    - pause: { duration: 5s }
```

```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

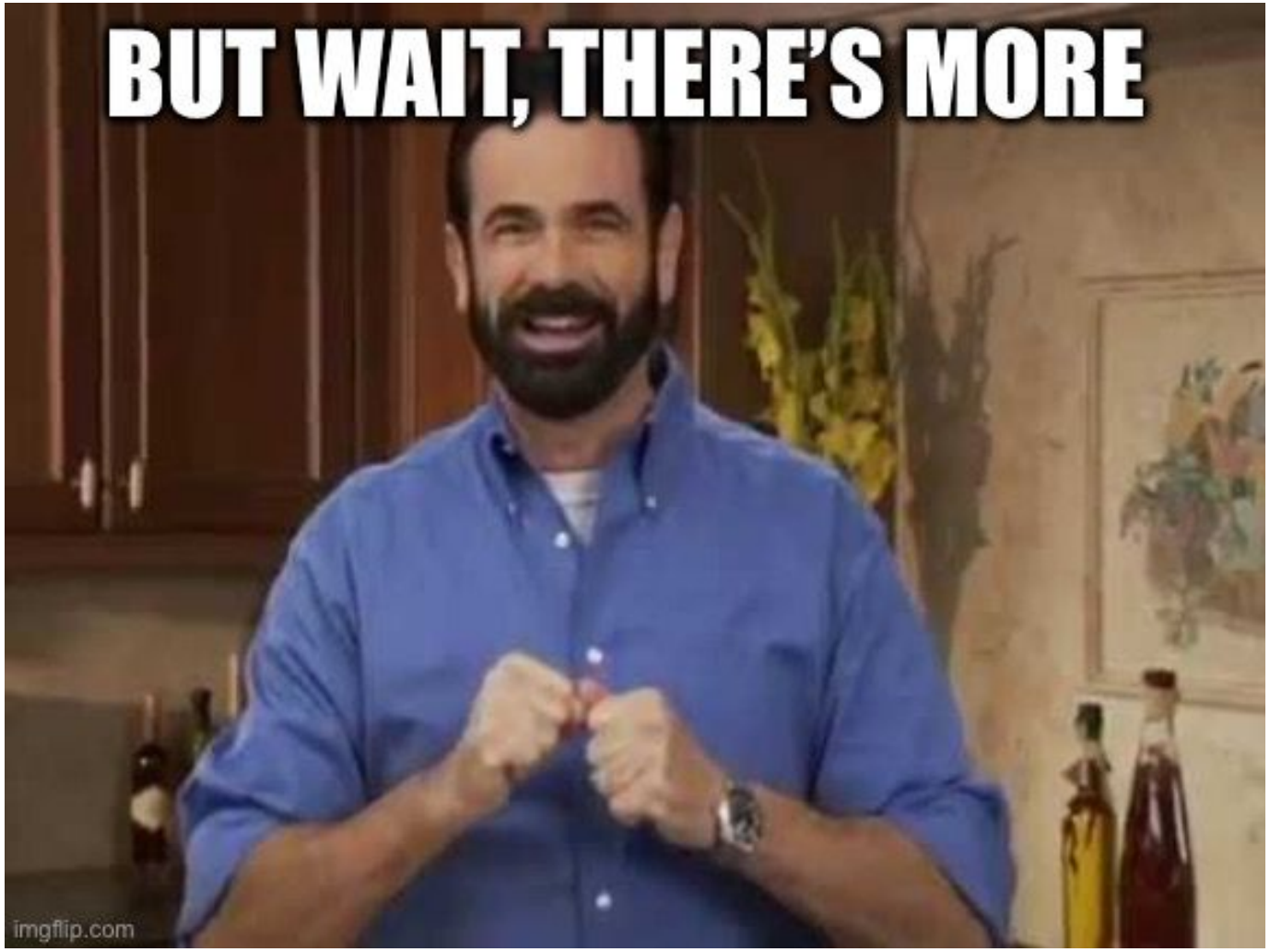
Use AI to analyze failures

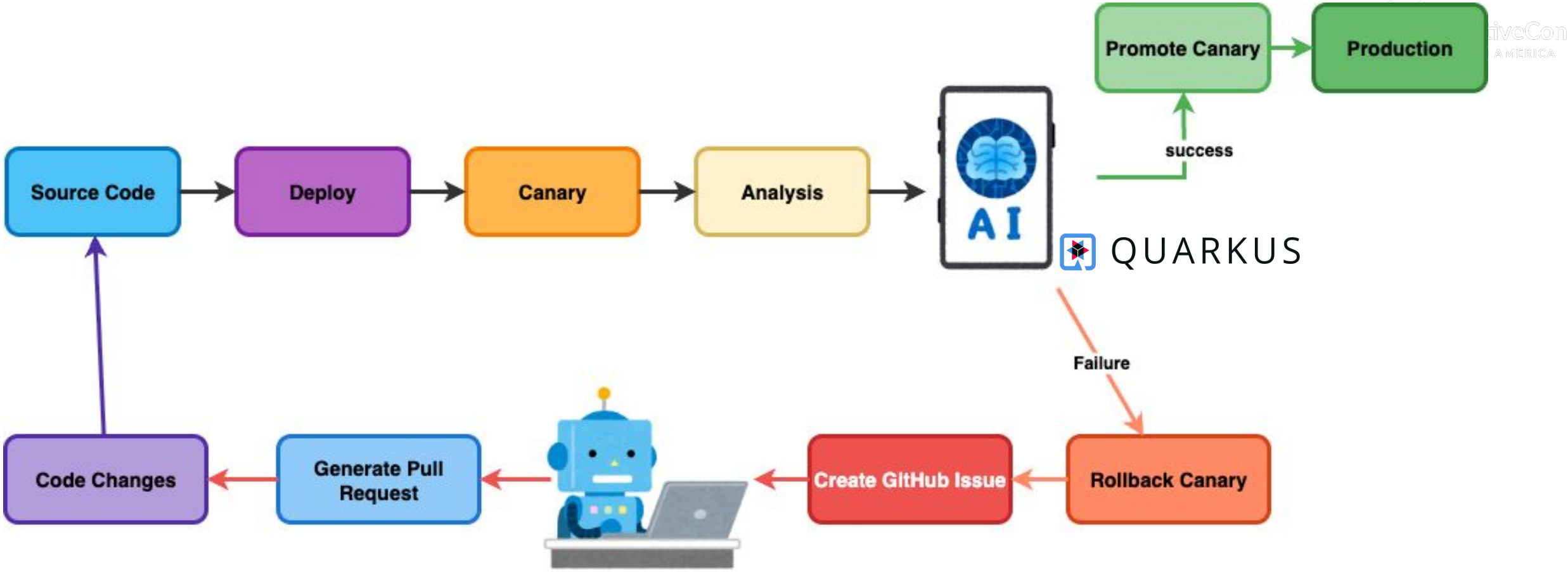

Source Code
Deploy
Canary
Analysis
AI
QUARKUS
Promote Canary
Production
success
Failure
Rollback Canary

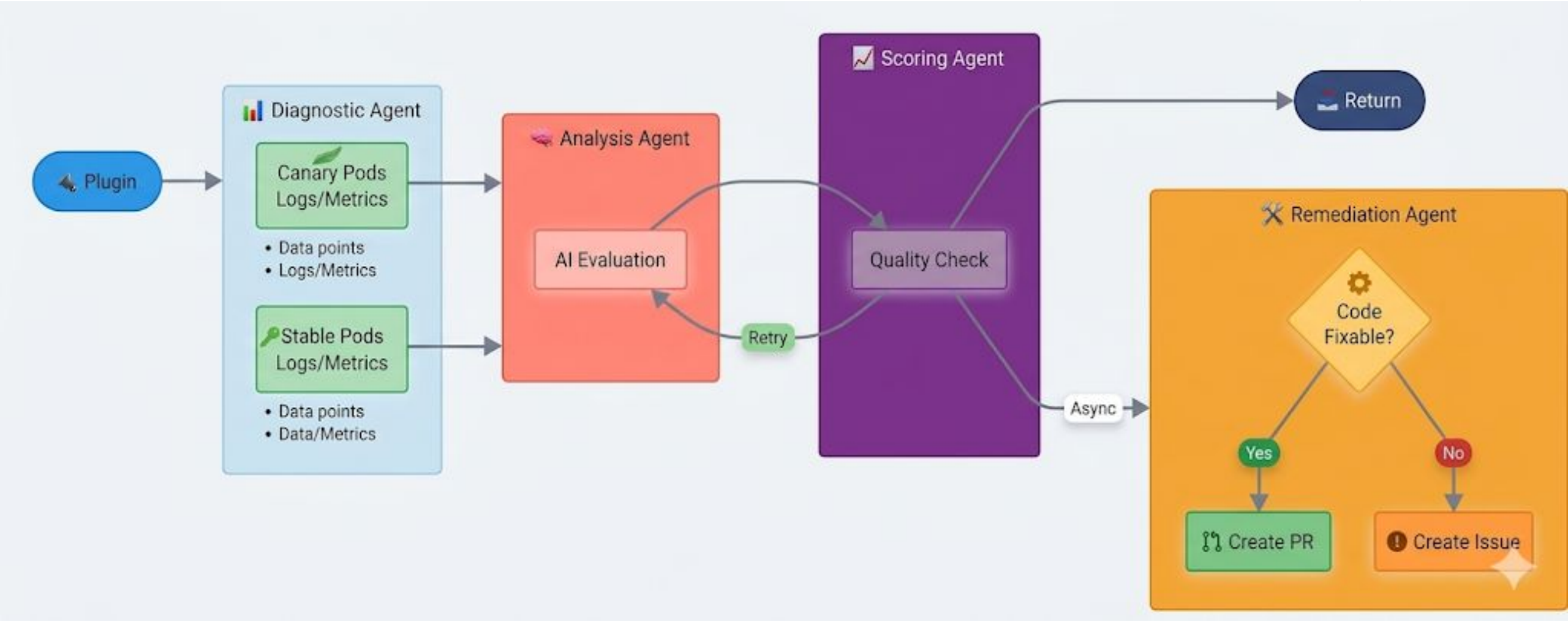
```
apiVersion: argoproj.io/v1alpha1
kind: RolloutManager
metadata:
  name: argo-rollouts
spec:
  plugins:
    metric:
      - name: argoproj-labs/metric-ai
        location:
https://github.com/argoproj-labs/rollouts-plugin-metric-ai/releases/
download/v0.0.1/rollouts-plugin-metric-ai-linux-amd64
```

```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


Source Code
Deploy
Canary
Analysis
AI
Promote Canary
Production
success
QUARKUS
Failure
Rollback Canary
Create GitHub Issue
Generate Pull Request
Code Changes


Plugin
Diagnostic Agent
Canary Pods Logs/Metrics
• Data points
• Logs/Metrics
Stable Pods Logs/Metrics
• Data points
• Data/Metrics
Analysis Agent
AI Evaluation
Retry
Scoring Agent
Quality Check
Return
Async
Remediation Agent
Code Fixable?
Yes
No
Create PR
Create 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
```

ArgoCon
EUROPE

KnativeCon
NORTH AMERICA

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

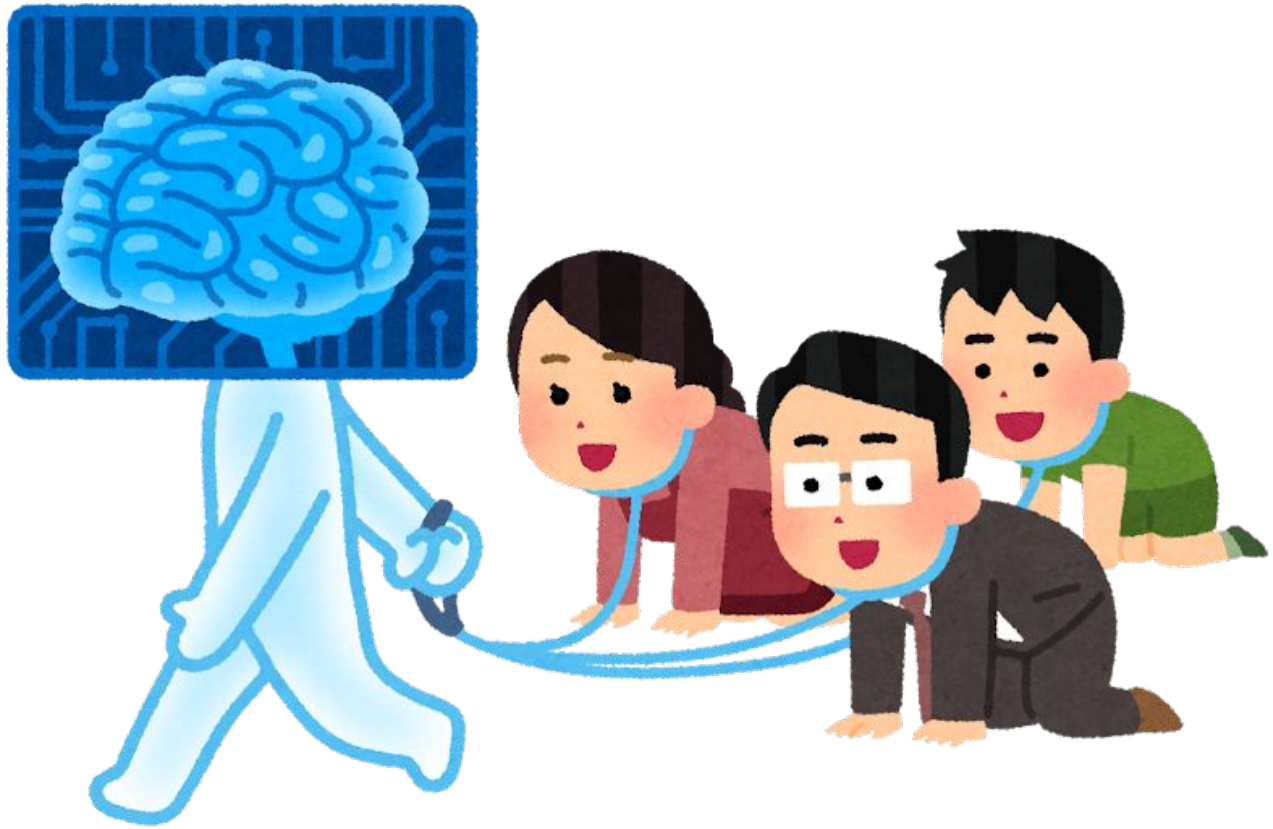
(could’ve used Serverless MCP, external code assistant for PR creation, async remote agents, etc.)

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



ArgoCon EUROPE

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

carlossg
kevindubois
csanchez.org
kevindubois.com
carlossg
kdubois