



ArgoCon

NORTH AMERICA



Fix Production Rollouts on the Fly With Agentic 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

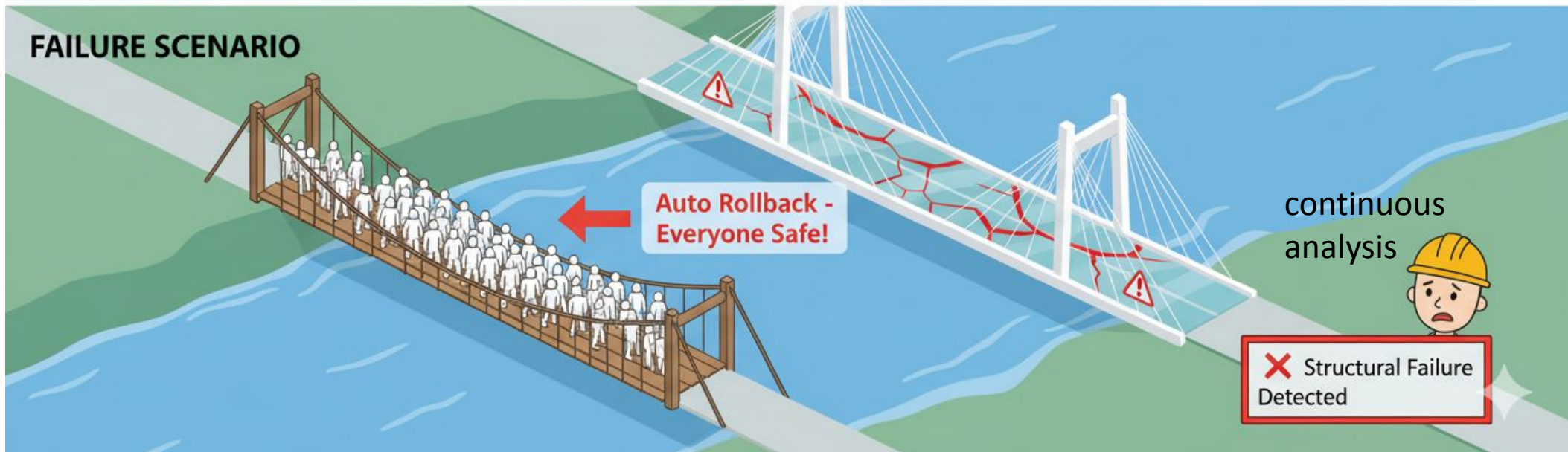
Findings: Each Template Instance should be deployed in a staged rollout.

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



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-vs
          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 }
```

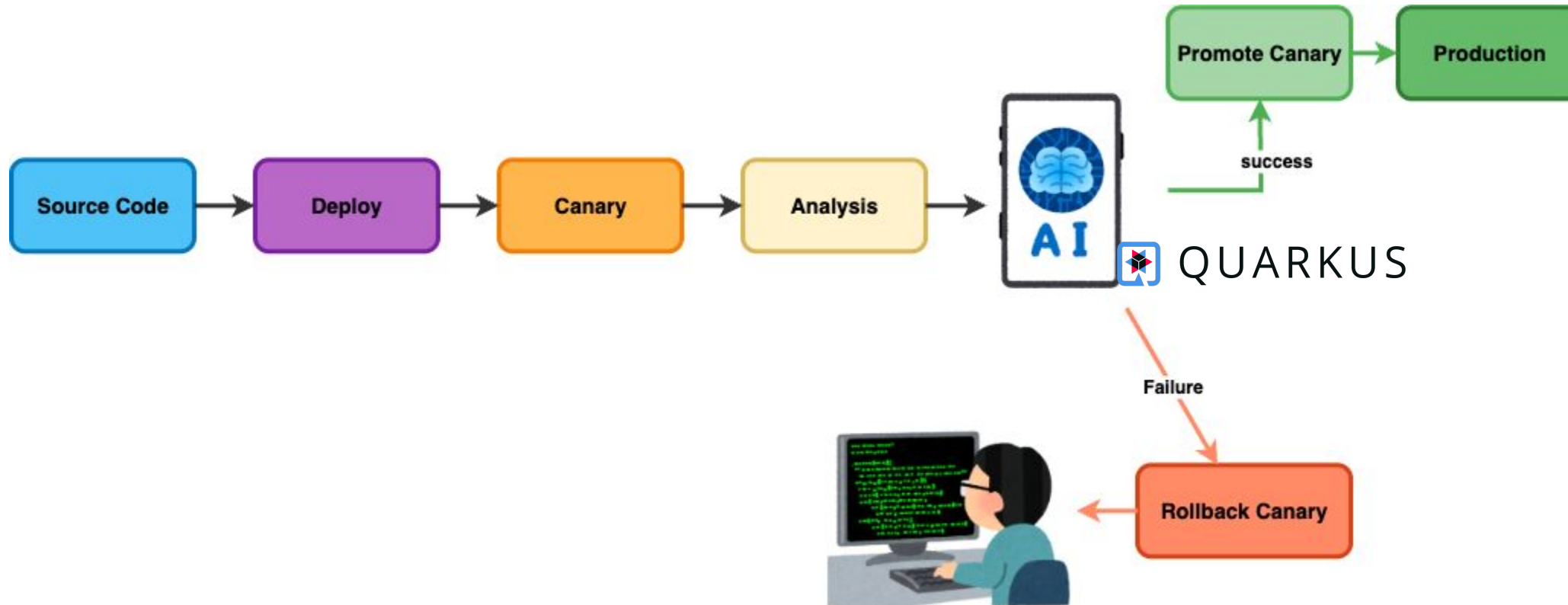
```
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: success-rate ←
spec:
  args:
  - name: service-name
  metrics:
  - name: success-rate
    interval: 10s
    successCondition: len(result) == 0 || result[0] >= 0.95
    failureLimit: 2
    provider:
      prometheus:
        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**

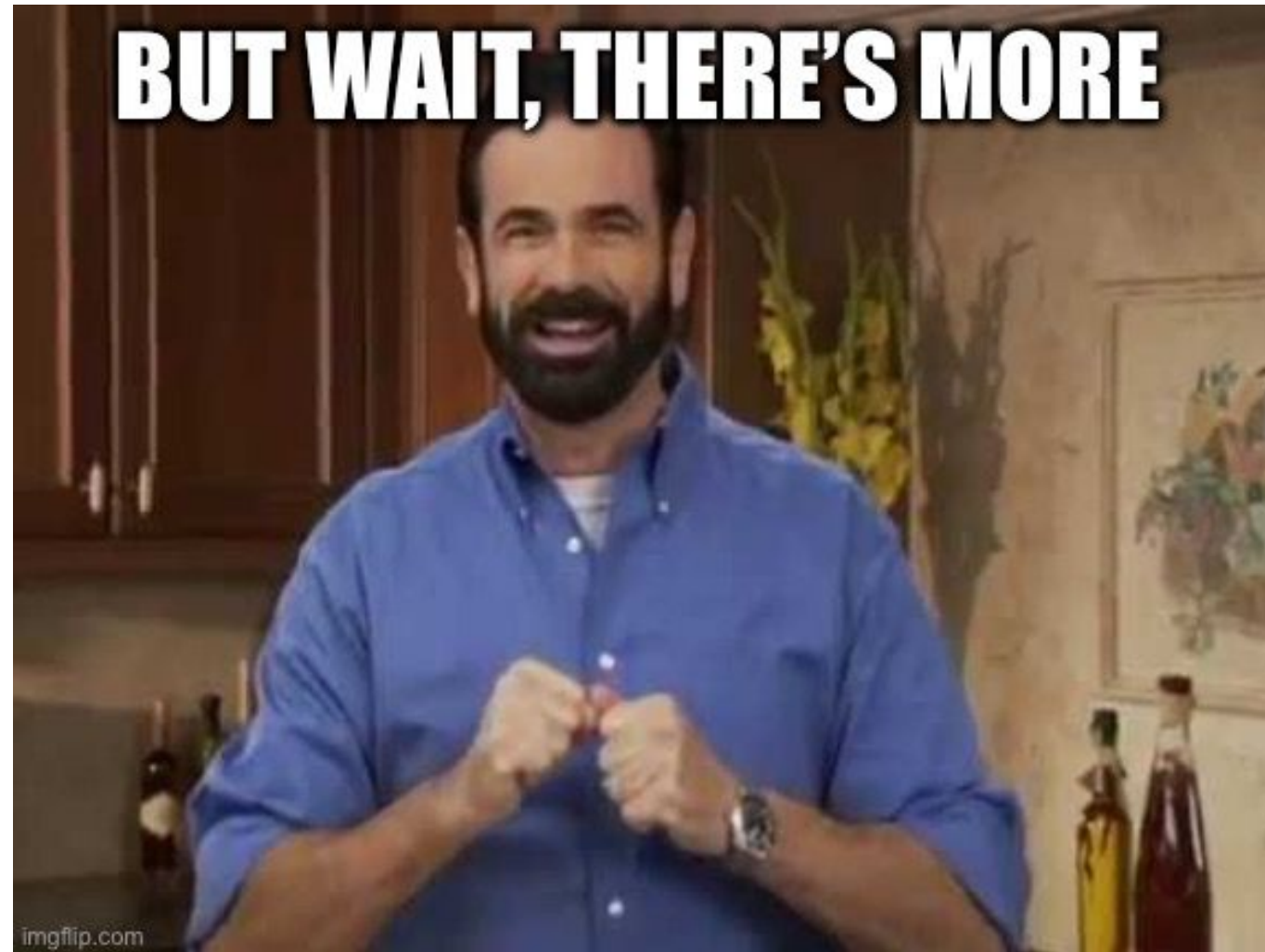


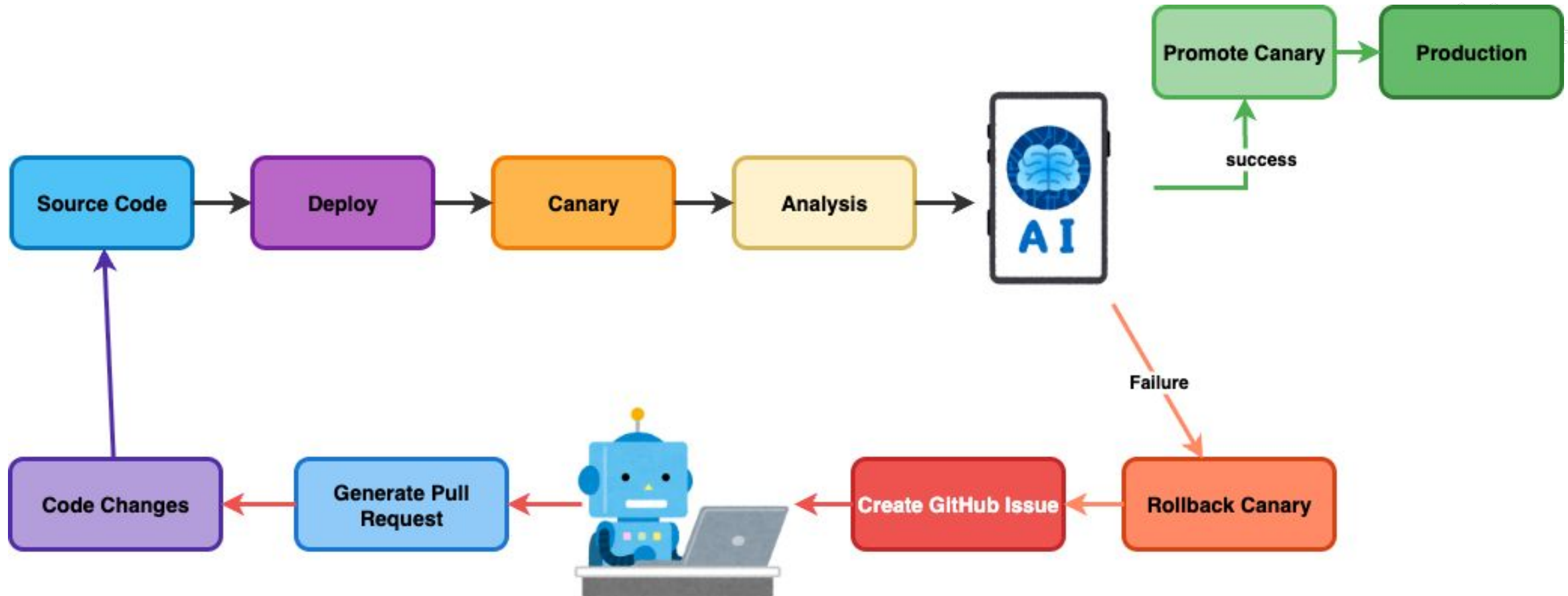
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: argo-rollouts-config
data:
  metricProviderPlugins: |-
    - name: "argoproj-labs/metric-ai"
      location: "file:///./rollouts-plugin-metric-ai/bin/metric-ai"
```

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

```
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.argo-rollouts.svc.cluster.local:8080
          analysisMode: agent
          stableLabel: role=stable
          canaryLabel: role=canary
          successCondition: result > 0.50
```





```
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.argo-rollouts.svc.cluster.local:8080
          analysisMode: agent
          stableLabel: role=stable
          canaryLabel: role=canary
          githubUrl: https://github.com/carlossg/rollouts-demo
          successCondition: result > 0.50
```

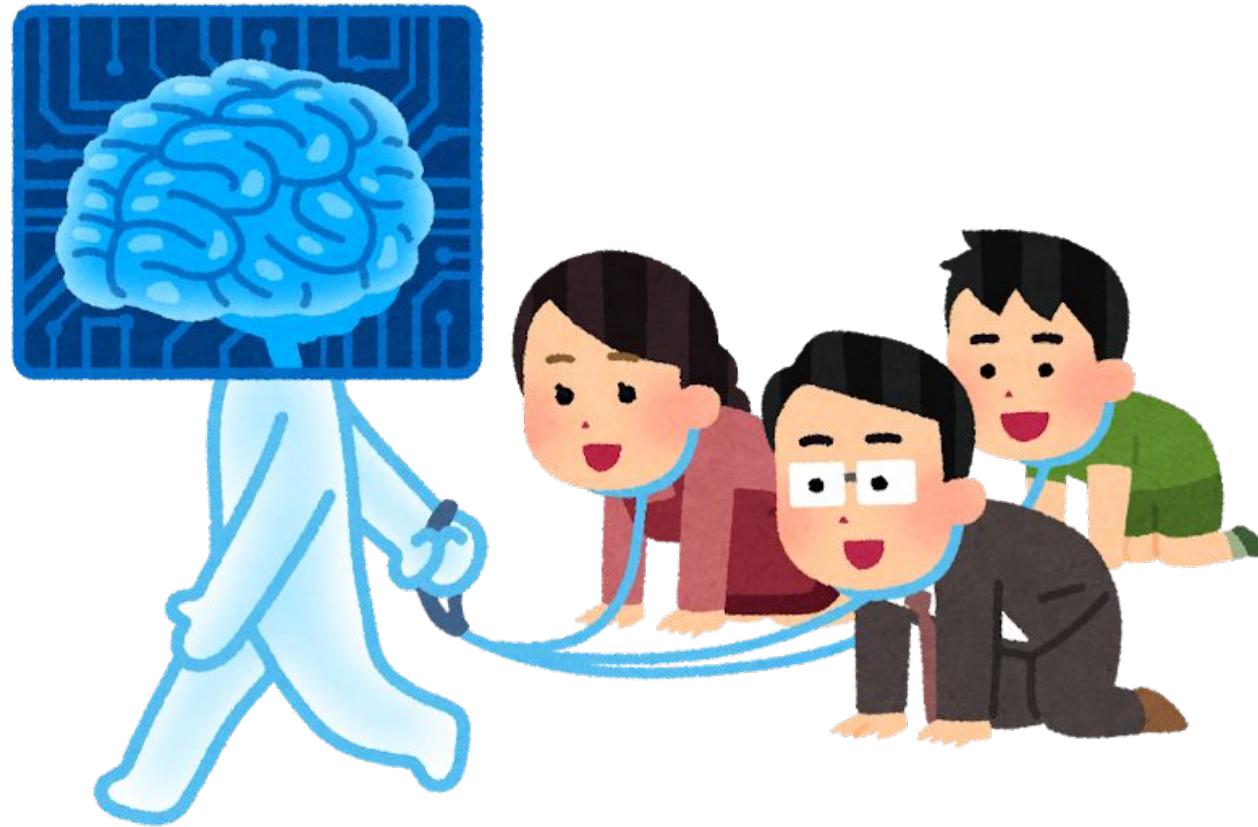
Demo

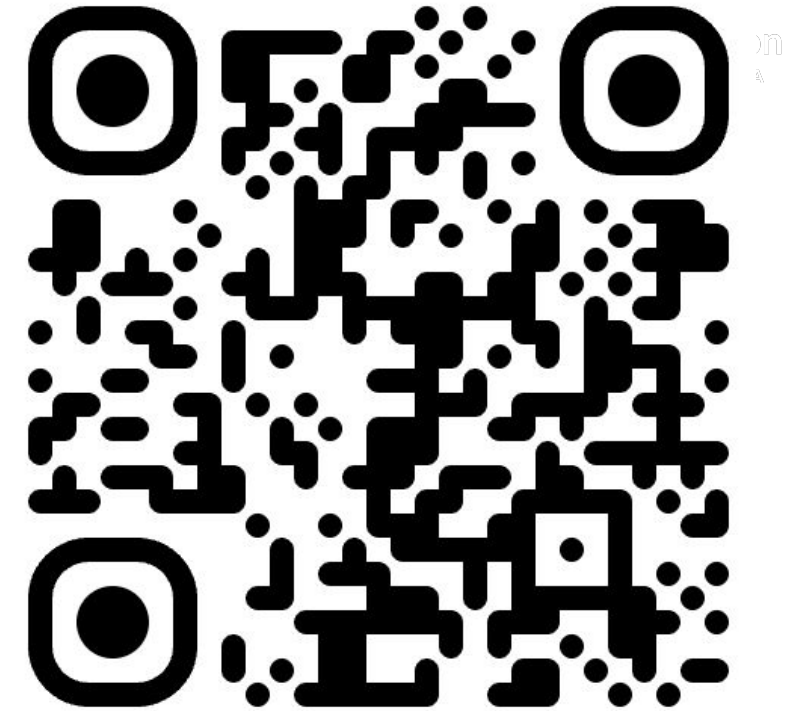
Takeaways:

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,
and even proposing fixes for the failures





github.com/argoproj-labs/rollouts-plugin-metric-ai/
github.com/carlossg/rollouts-demo
github.com/kdubois/kubernetes-agent