From Commit to Rollout

Tracing GitOps with Argo CD and OpenTelemetry



Antonio Jimenez Martinez

Tech Lead Software Engineer, Cisco ThousandEyes

ArgoCon at KubeCon + CloudNativeCon North America 2026

Agenda

01 The GitOps observability gap

02 Demo

03 Traces and causal handoffs

04 Upstream issues and contributing

05 Takeaways

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

The GitOps observability gap

CI passed. Sync passed. The release failed.

Which controller decision explains why?

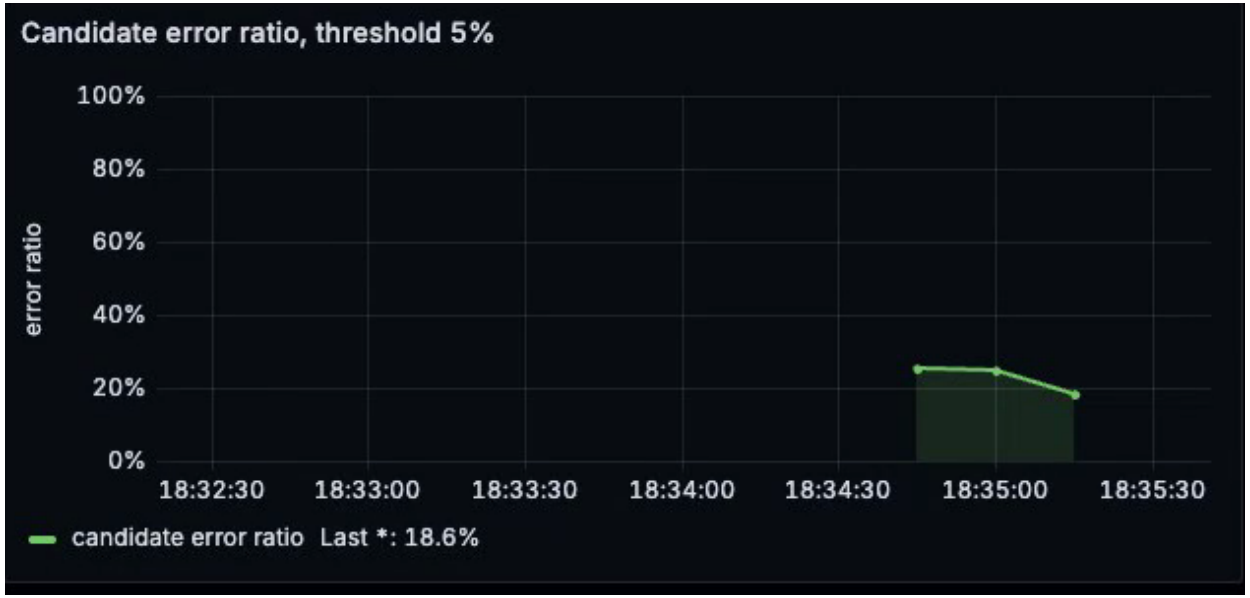
checkout v1.3.0

CI passed

Argo CD sync passed

Analysis failed

What stopped promotion?



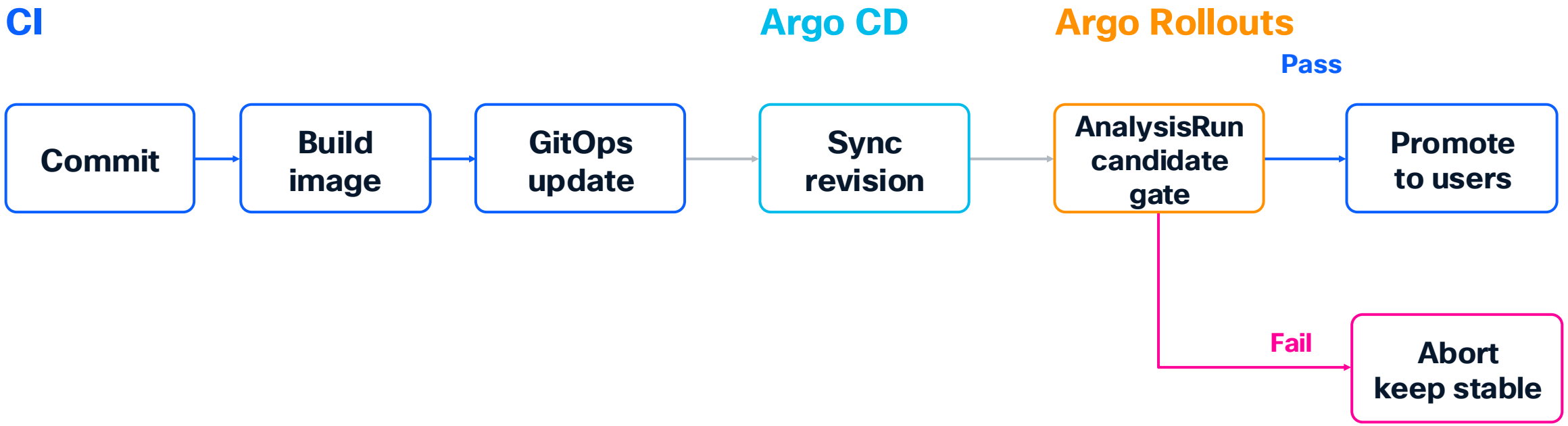
Runtime evidence can reject a synchronized release.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

CI success is only the first outcome

Build, sync, and runtime analysis answer different questions.



Failed-release example

CI: PASSED

Built and updated Git

Argo CD: SYNCED

Applied the requested state

Rollout: FAILED

Candidate rejected by analysis

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Controllers react at different times

Git changes, resource updates, and analysis results trigger controller work.

Git	→	A new desired state is committed.
Argo CD	→	Detects the change and synchronizes resources.
Kubernetes	→	Updates objects and reports their status.
Rollouts	→	Checks the candidate and reacts to analysis results.

Each controller processes events on its own schedule.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Demo

Delivery, traces, and analysis metrics

Instrumentation creates the relationships. The Collector transports the traces.

Desired state and workload delivery



Argo CD applies the desired Rollout resource in Kubernetes.

OpenTelemetry traces



Prometheus metrics for the analysis decision



© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Demo

One service. Three deployment attempts.

Success	Failure	Recovery
v1.2.0 reaches 100%	v1.3.0 stops at analysis	new v1.2.0 GitOps attempt

Recorded demo using instrumented Argo CD and Argo Rollouts forks.

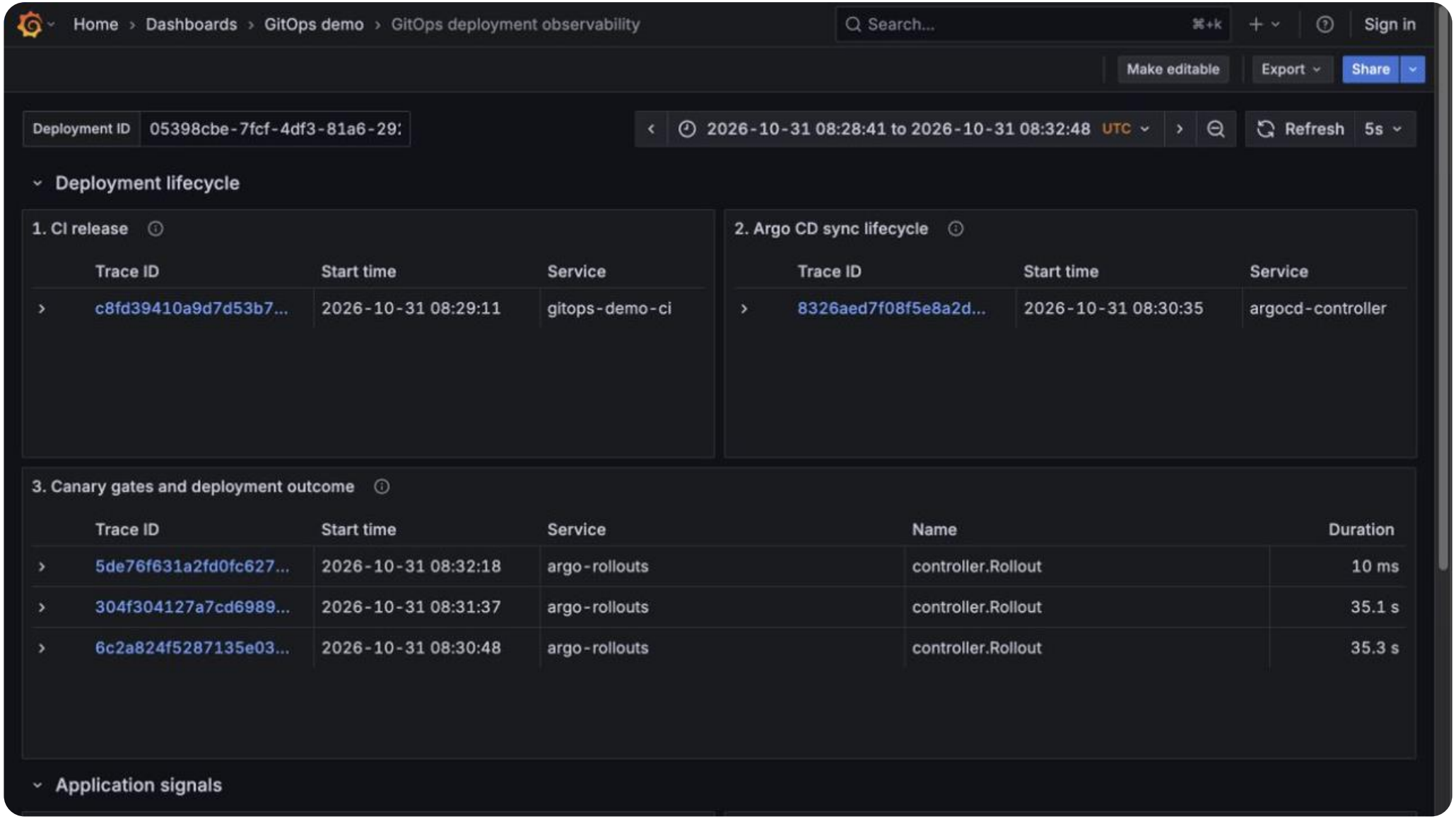
Filter each view by the selected release’s deployment.id

Follow that release across CI, controller decisions, and application signals.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

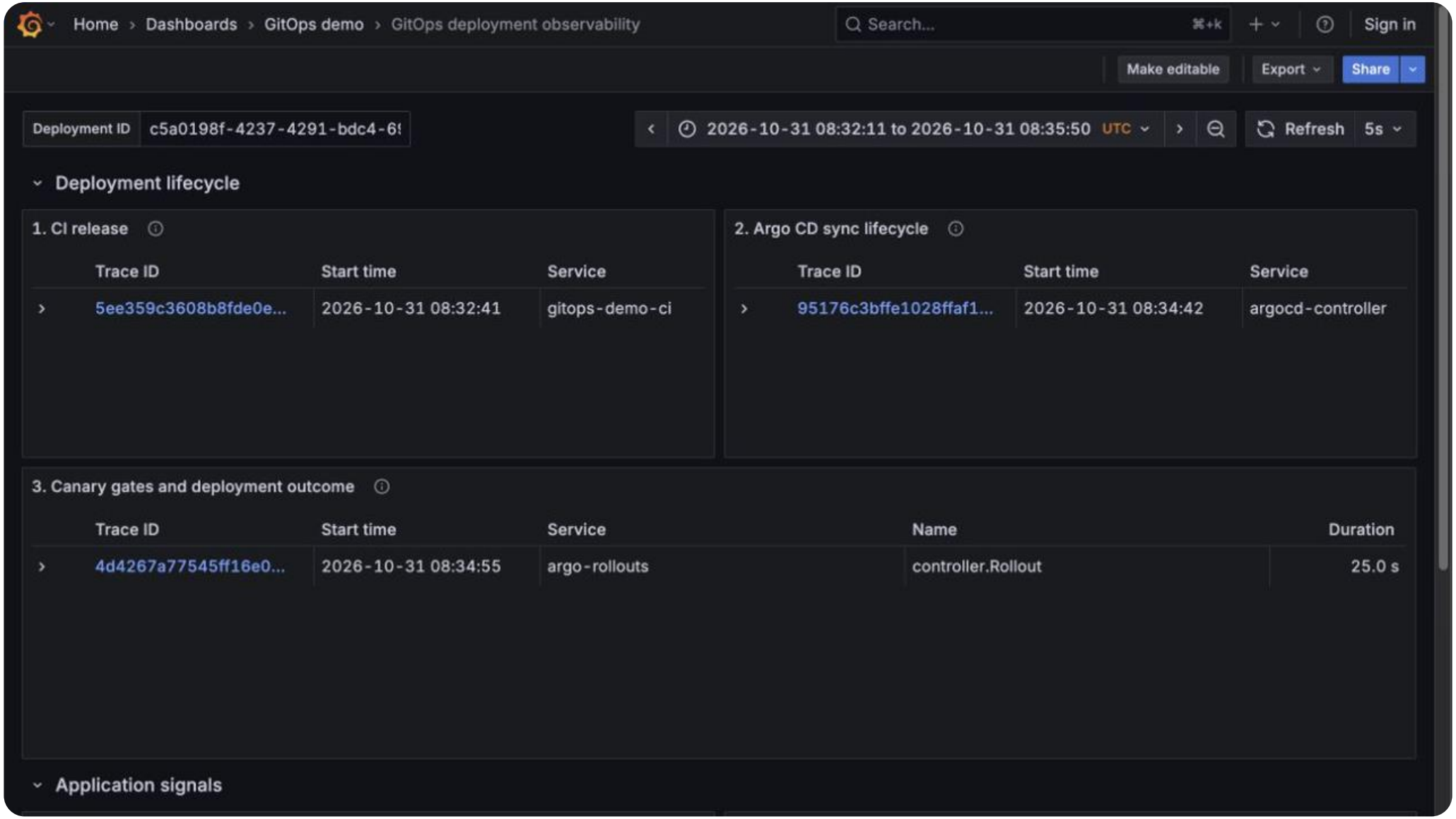
Success: promotion reaches 100%



© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

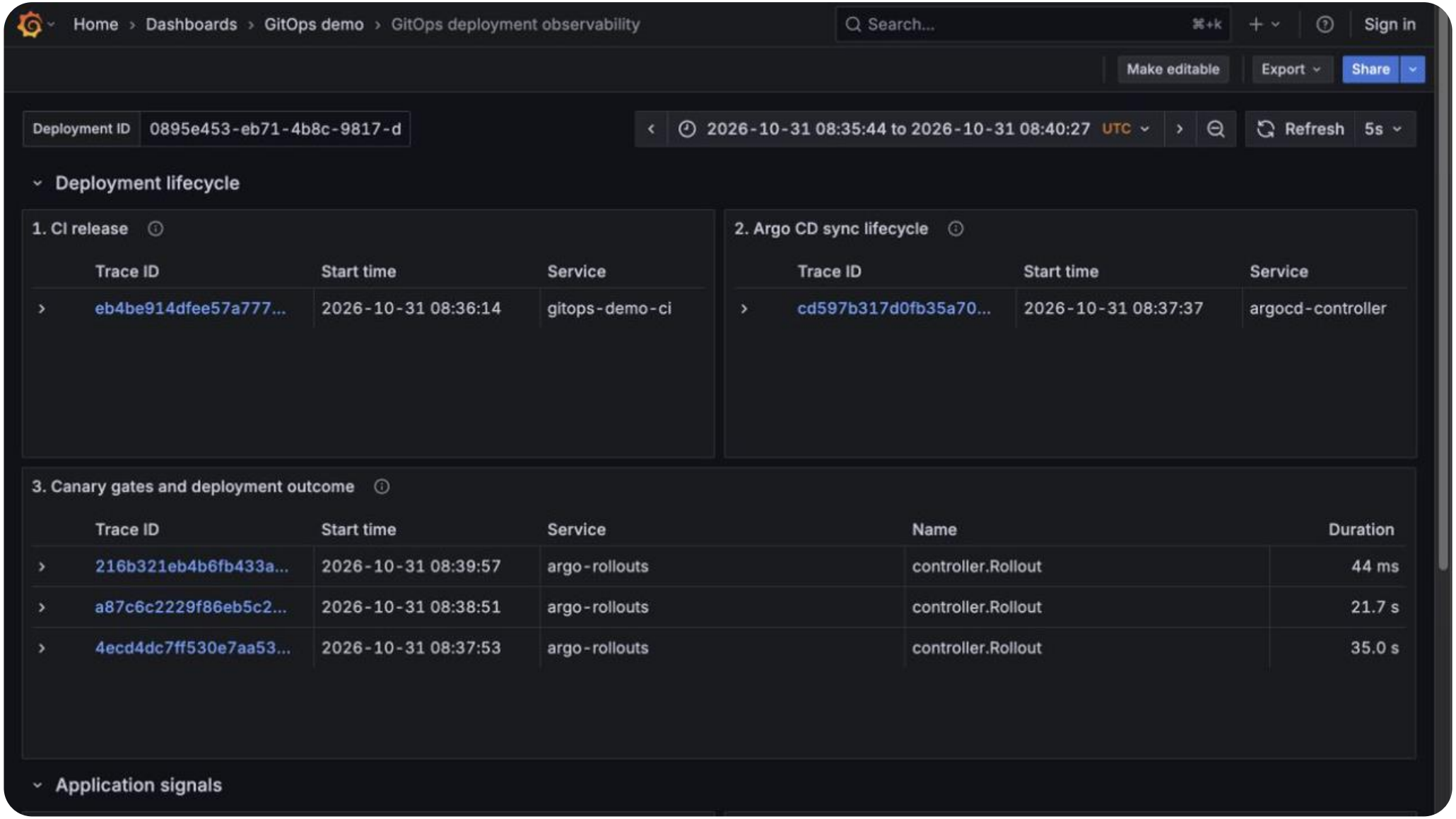
Failure: analysis stops promotion



© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Recovery: rollback is a new release

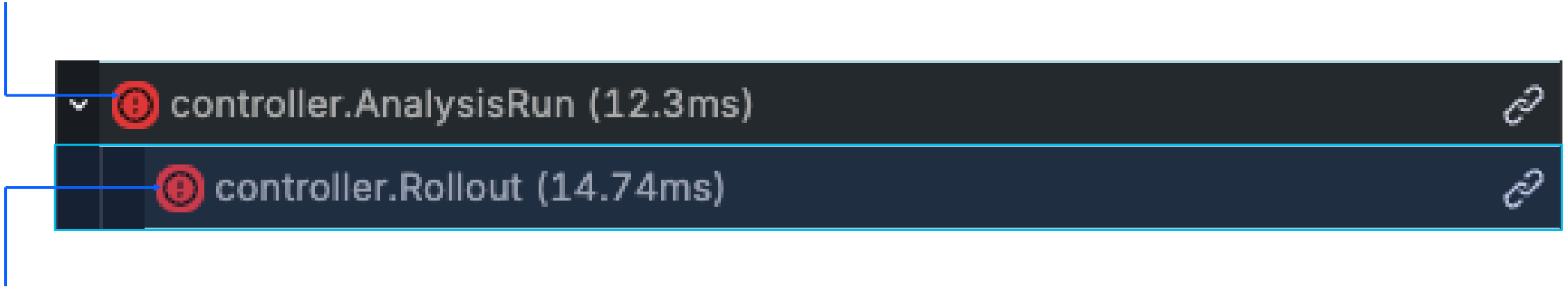


© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

The failed analysis stopped promotion.

Analysis rejects the candidate



Rollout processes the result

This validated handoff keeps the two spans in the same trace.

argo.rollouts.reconciliation.cause = analysis_run_result

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Candidate errors exceed the rejection threshold

27.4%

Captured candidate-error peak

>

5%

Analysis rejection threshold



The analysis checks errors in the new release, not all traffic.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

The release failed. The controller did its job.

The candidate failed analysis.

Rollouts processed the result and stopped promotion.

Why did it run?	What changed?	How is it marked?
An analysis result arrived.	Promotion stopped.	ERROR.

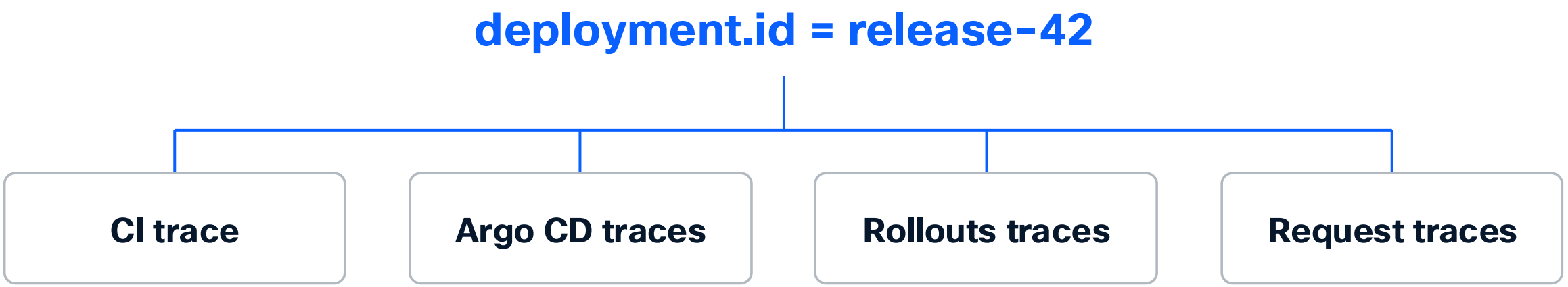
Prototype policy: ERROR marks an aborted rollout, not a controller crash.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Connecting traces across controller actions

One deployment ID, multiple trace IDs



Shared identity groups the release. It does not merge trace IDs.

Deployment ID

finds the release.

Span links

identify its CI origin.

Parent context

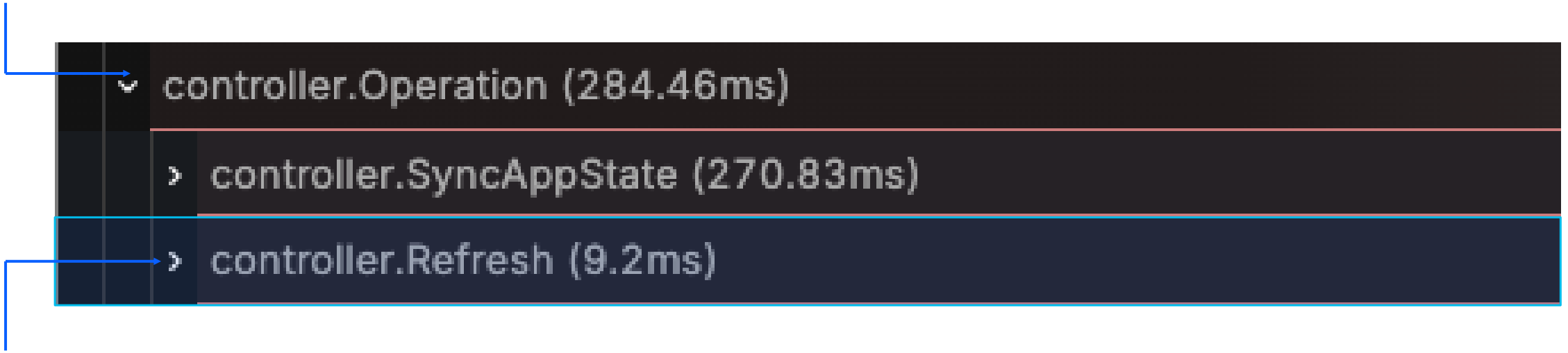
connects a validated handoff.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

After sync, Argo CD checks the Application again

1. Sync Operation completes



2. Refresh rechecks the Application

The validated handoff keeps this Refresh in the Operation's trace.

argocd.reconciliation.cause = operation_completed

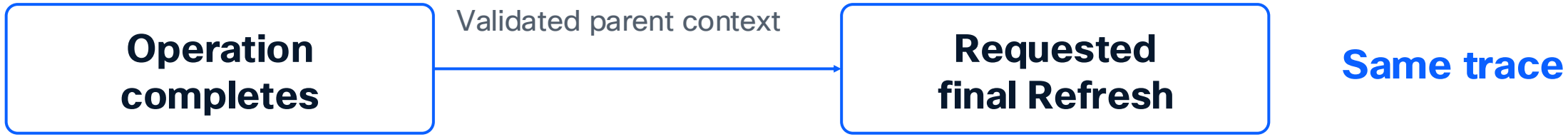
© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Connecting controller actions

Which earlier action triggered this work?

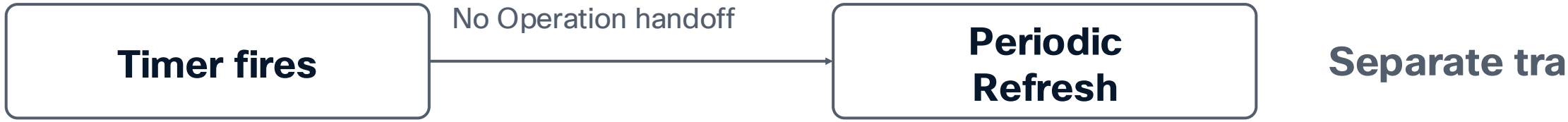
One action requests the next



Same trace

The Operation may already have ended.

An independent trigger starts separate work



Separate trace

Same deployment ID or nearby timing does not establish the trigger.

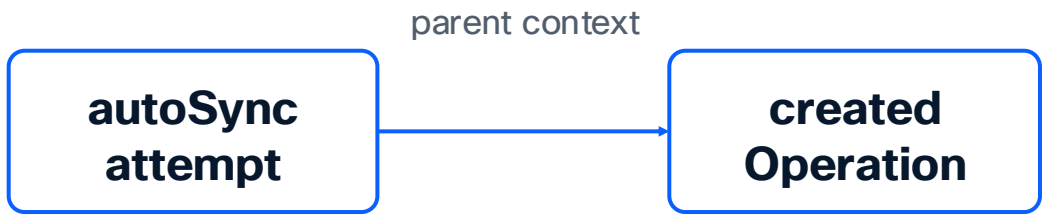
© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

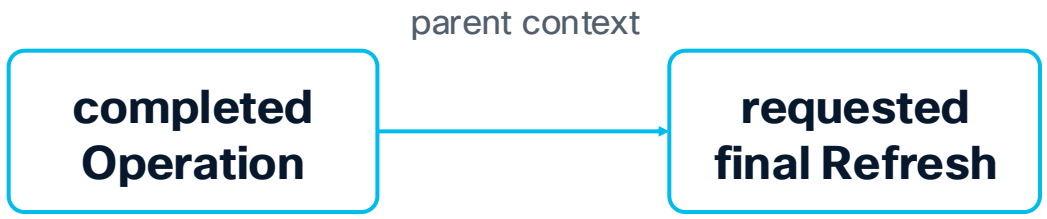
Four validated controller handoffs

Parent context joins only the controller transitions that the implementation checks.

Argo CD

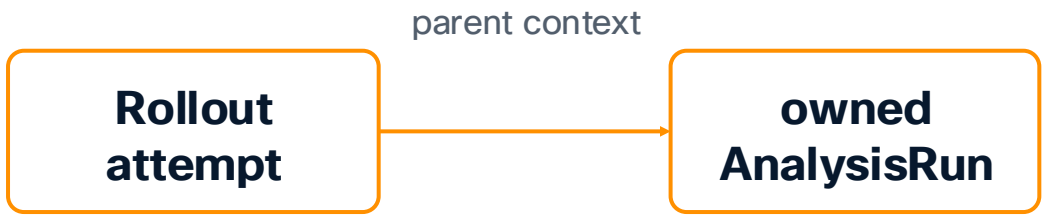


Creator context stored on the Operation

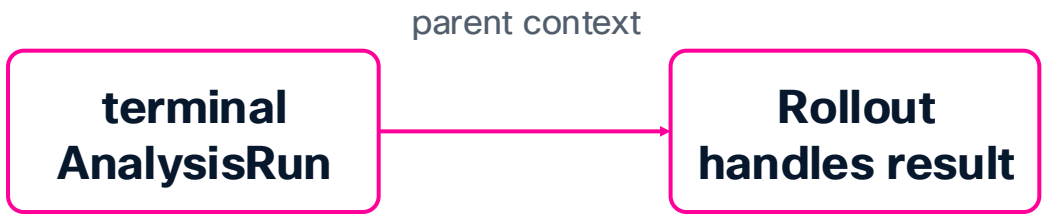


Bounded, one-use completion context

Argo Rollouts



Creator context on the owned AnalysisRun



Context for the current terminal result

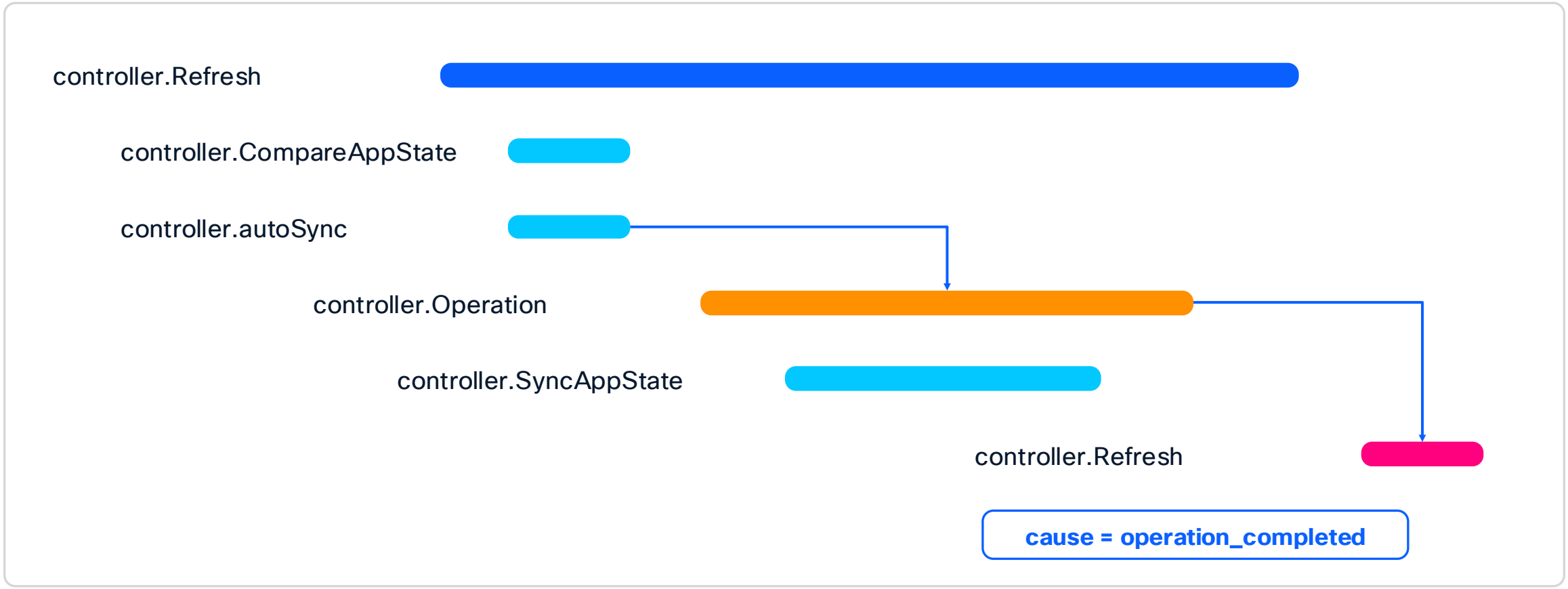
Other attempts start separate traces. The deployment ID keeps them searchable together.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Argo CD lifecycle trace

The release-driven path keeps comparison, auto-sync, Operation, and its exact final Refresh together.



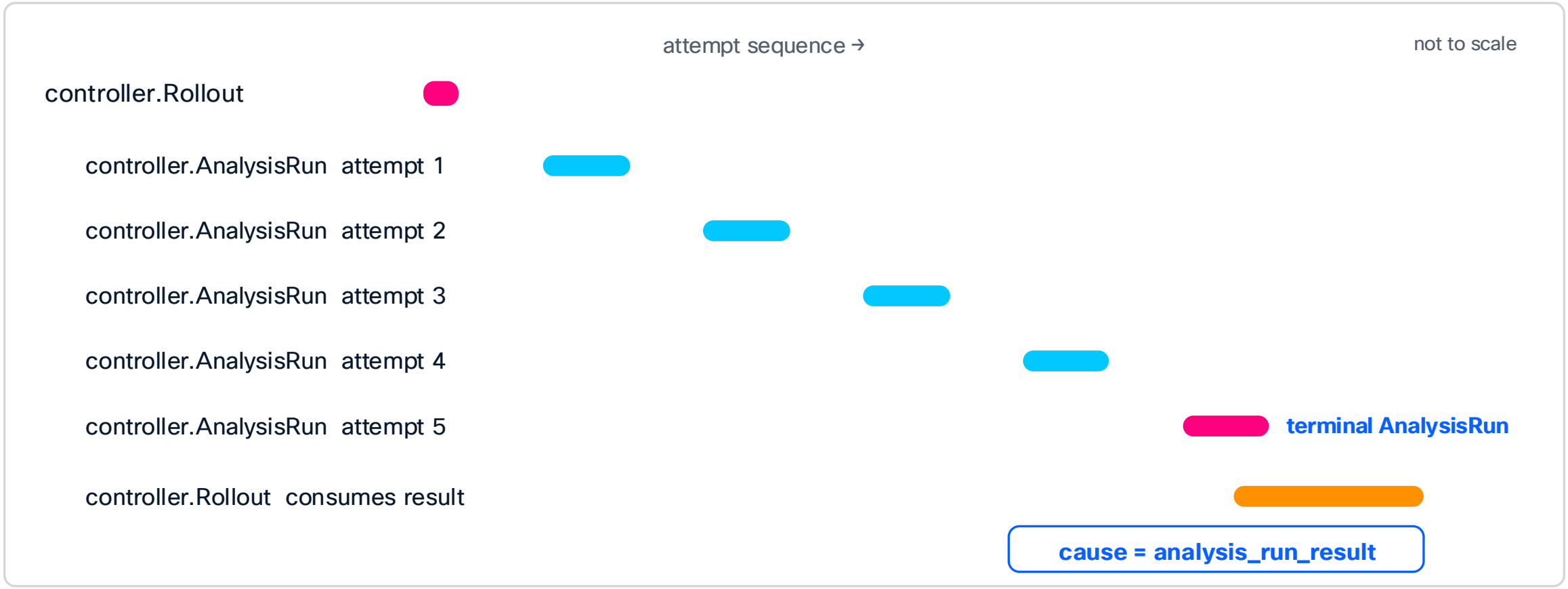
Other periodic or manual Refresh attempts remain independent roots.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Argo Rollouts gate trace

One canary gate groups the created AnalysisRun attempts and the Rollout that consumes the terminal result.



Short attempts, with waiting gaps

The persisted carrier survives a Rollouts controller restart.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Upstream path

Recorded demo using instrumented Argo CD and Argo Rollouts forks.

Three issues define the upstream path

Recorded demo using instrumented Argo CD and Argo Rollouts forks.

01 **Argo CD #29567**

Adds final outcome metadata to Argo CD controller spans.

https://github.com/argoproj/argo-cd/issues/29567

02 **Argo Rollouts #5085**

Adds OpenTelemetry tracing to the Rollouts controller.

https://github.com/argoproj/argo-rollouts/issues/5085

03 **Argo Rollouts #5105**

Traces rollout decisions and Rollout/AnalysisRun handoffs.

https://github.com/argoproj/argo-rollouts/issues/5105

Open upstream proposals. Demo correlation vocabulary remains experimental.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Takeaways

Three tracing rules

Use deployment.id to find the release story.

Share a trace only when one action triggers the next.

Check CI, Argo CD sync, and Argo Rollout release separately.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Questions



Controller forks

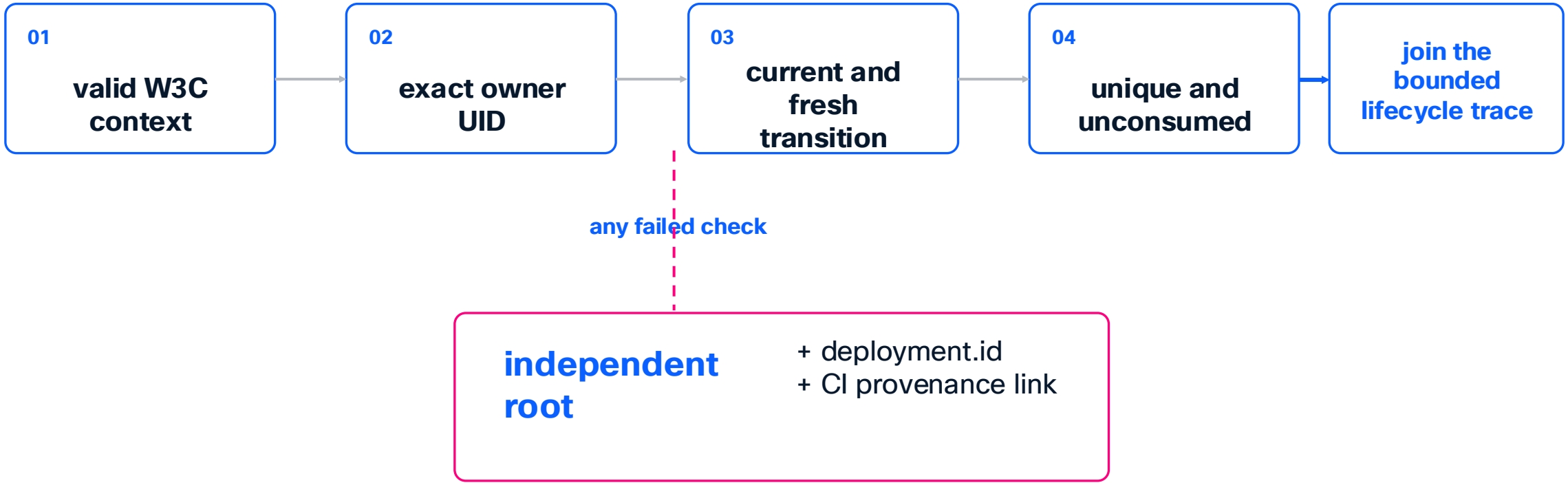
Argo CD and Argo Rollouts


CISCO

Appendix

Ambiguity creates a new root

The instrumentation fails closed instead of inventing a causal edge.



Missing, stale, malformed, owner-mismatched, already-consumed, or ambiguous context never becomes a parent.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Git carries the release envelope

Two ordered commits make the workload revision immutable before Argo CD follows it.



The Git boundary

traceparent becomes a span link, not a remote parent.
deployment.id stays the release-wide query key.

Release envelope

observability.demo/
deployment-id
observability.demo/
traceparent
observability.demo/
source-revision
observability.demo/
rollback-of

Allowlisted. Versioned. No secrets.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

The Collector receives and exports the spans

Instrumentation creates the relationships. The Collector transports them.



Actual traces pipeline

Receives spans, applies processors, then exports.

```
service:
  pipelines:
    traces:
      receivers: [otlp]
      processors: [memory_limiter, batch]
      exporters: [otlp/tempo, debug]
```

Prometheus supplies the analysis metrics separately.

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential

Annotations become span metadata

Kubernetes annotations

```
metadata:
  annotations:
    observability.demo/deployment-id: release-42
    observability.demo/traceparent: <CI context>
    observability.demo/source-revision: <SHA>
```


Controller span

deployment.id = release-42

vcs.ref.head.revision
= <SHA>

Span link to the CI span

At the Git boundary, CI context becomes a link, not a parent.

deployment.id: OTel Development | observability.demo/*: project-specific

© 2026 Cisco and/or its affiliates. All rights reserved.

Cisco Confidential