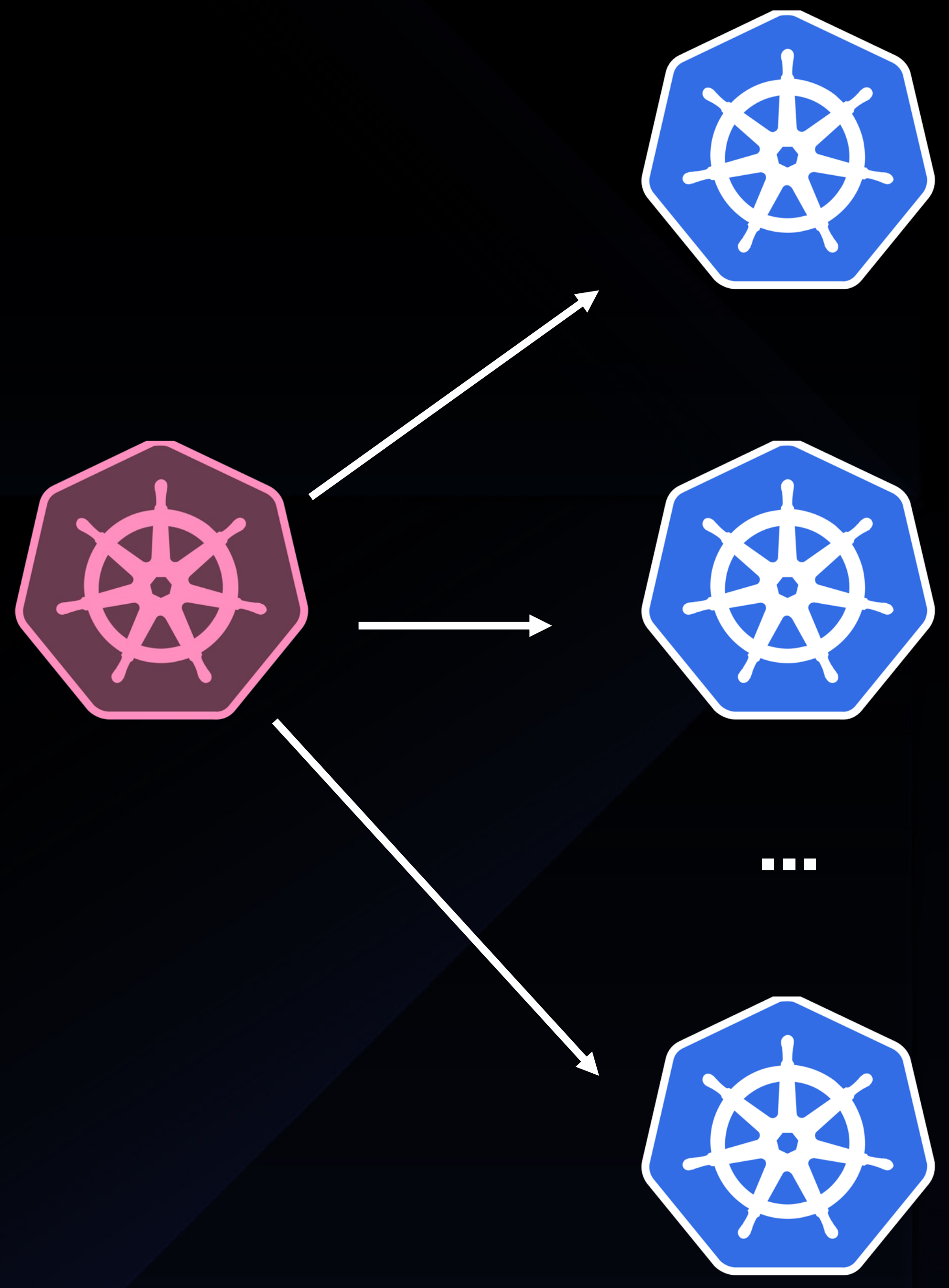
How to Not Destroy Your Kubernetes Clusters

Qian Ding

ANT GROUP

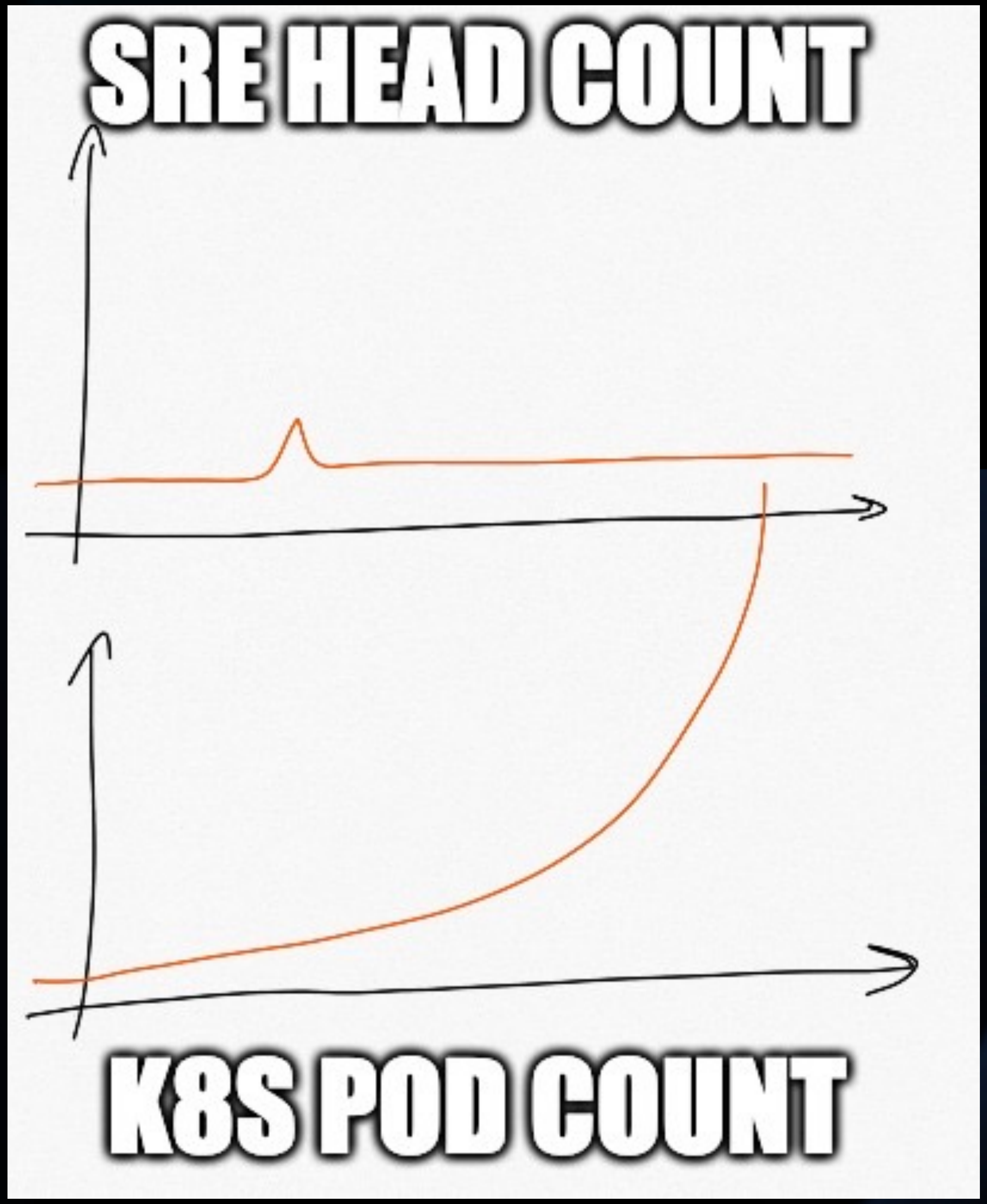
SRE CON ASIA PACIFIC

Background



Kube-on-Kube

#K8S Dev	30
#K8S SRE	10
#Clusters	100+
#Nodes	100K
#Pods	3M
Max (#nodes/cluster)	17K
Max (#pods/cluster)	620K



🤦‍♂️

Greatest Hits*

☸️ 01 The powerful operator

🏁 02 The paradoxical finalizer

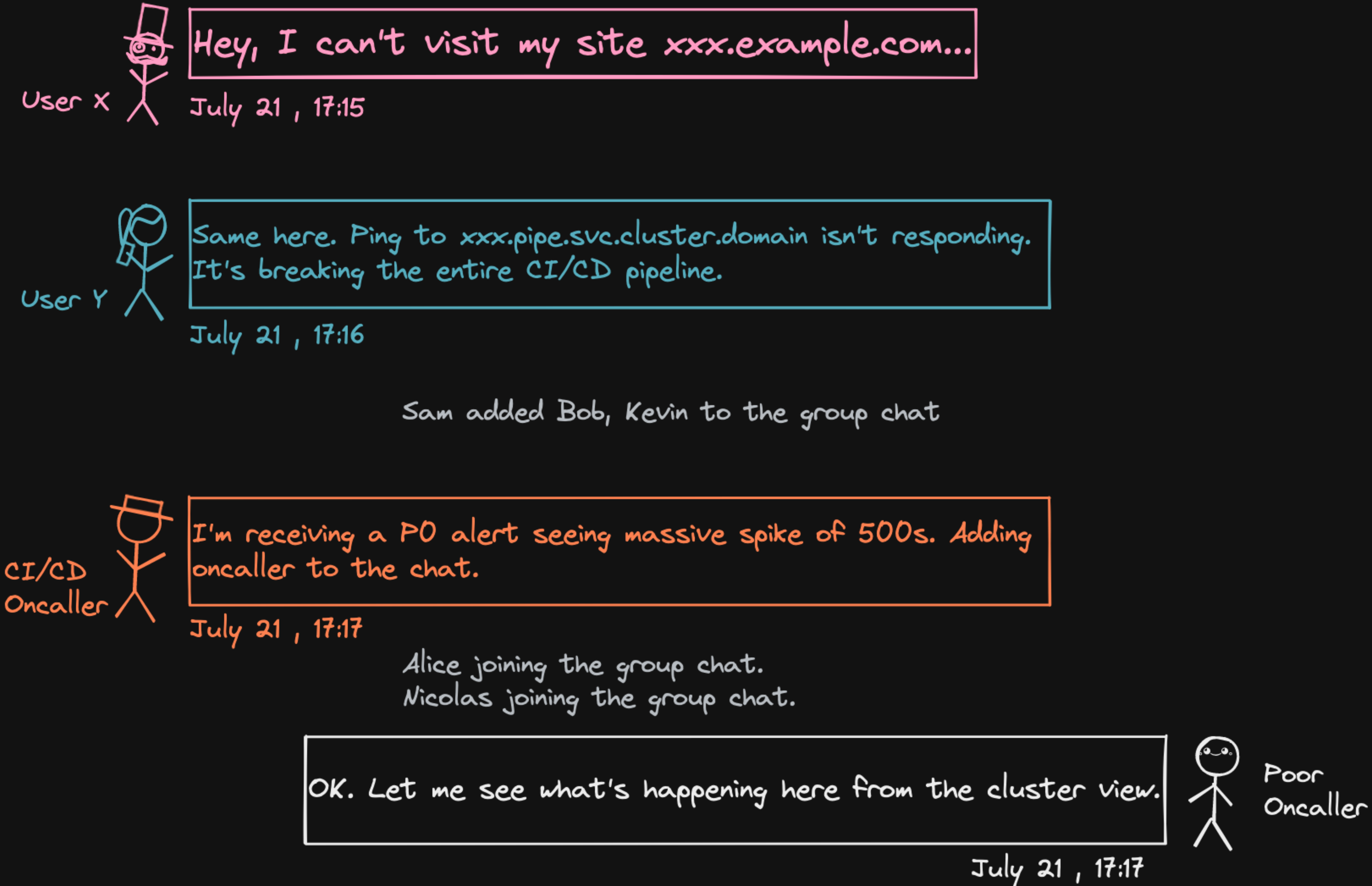
🪝 03 The evil webhook

*This list is based on our postmortem database and ranked by the number of internal reads.

☸️ 01

The powerful operator

Let’s NOT make too many operators...



Hey, I can't visit my site xxx.example.com...

July 21 , 17:15

Same here. Ping to xxx.pipe.svc.cluster.domain isn't responding. It's breaking the entire CI/CD pipeline.

July 21 , 17:16

Sam added Bob, Kevin to the group chat

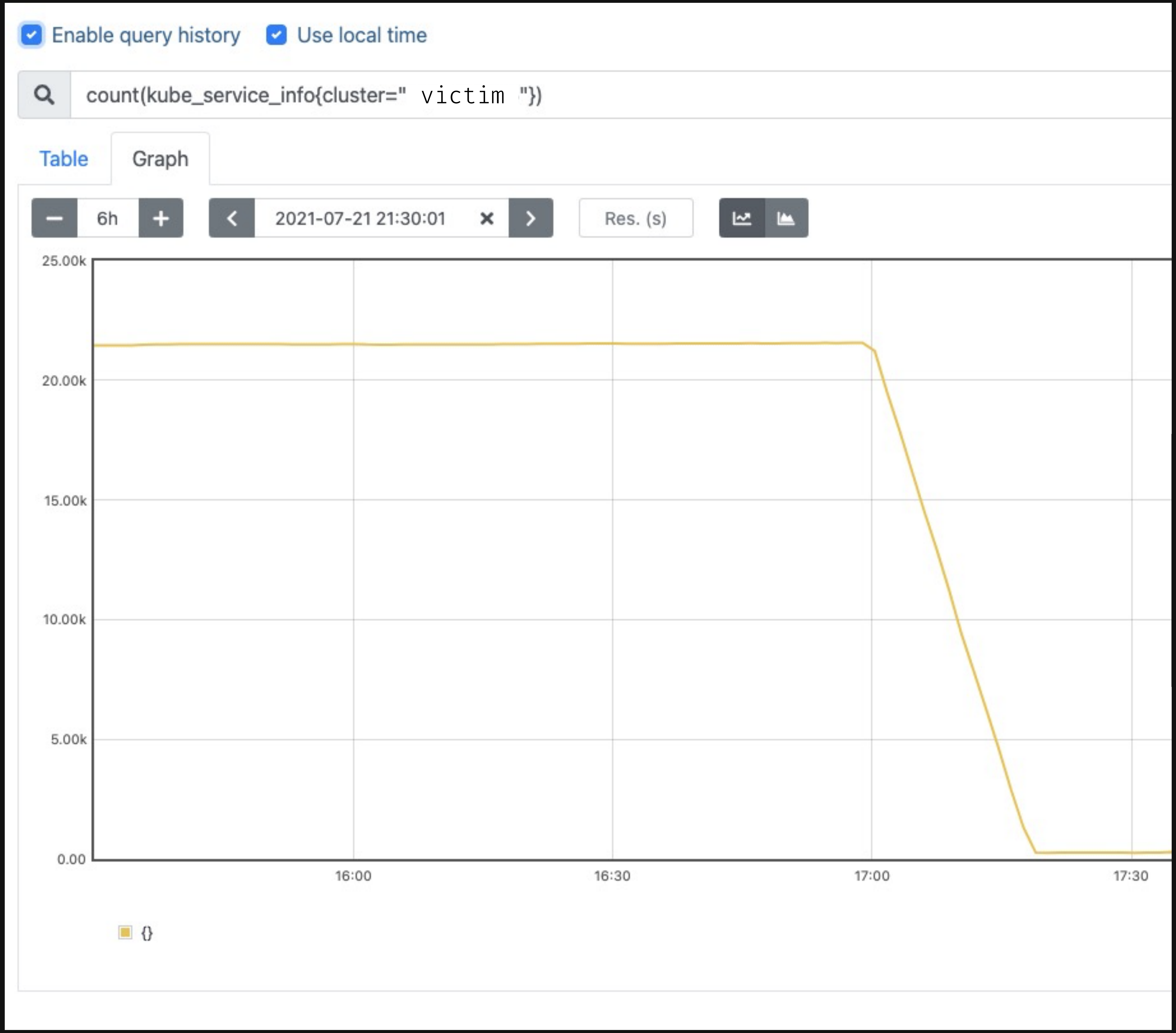
I'm receiving a P0 alert seeing massive spike of 500s. Adding oncaller to the chat.

July 21 , 17:17

Alice joining the group chat.
Nicolas joining the group chat.

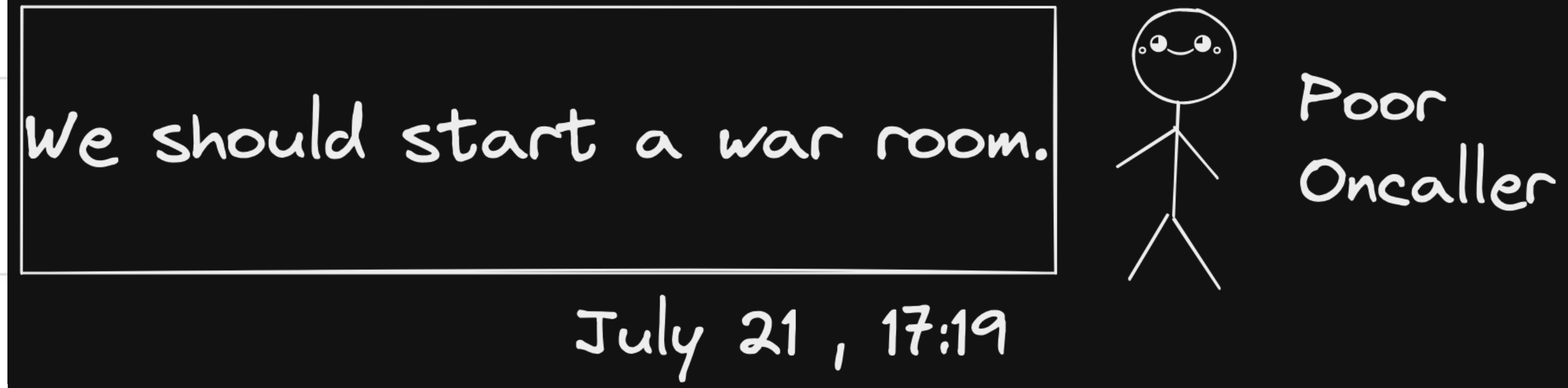
OK. Let me see what's happening here from the cluster view.

July 21 , 17:17



```
count(kube_service_info{cluster=" victim "})
```

Number of service objects in the victim cluster



Victim cluster property
Size:2K nodes, 30K pods
Usage: CI/CD, dev production
Status: Deprecating, ETA 2023

17:20 War room declared

17:30 Locate the offending operator

- **Audit logs indicated all svc were deleted from a single IP.**
- **It was a custom load balancer operator.**

17:20 War room declared

17:30 Locate the offending operator

17:32 Another cluster screamed due to high traffic load

17:35 More users came in and complained

- **No change to the operator in the past 90 days.**

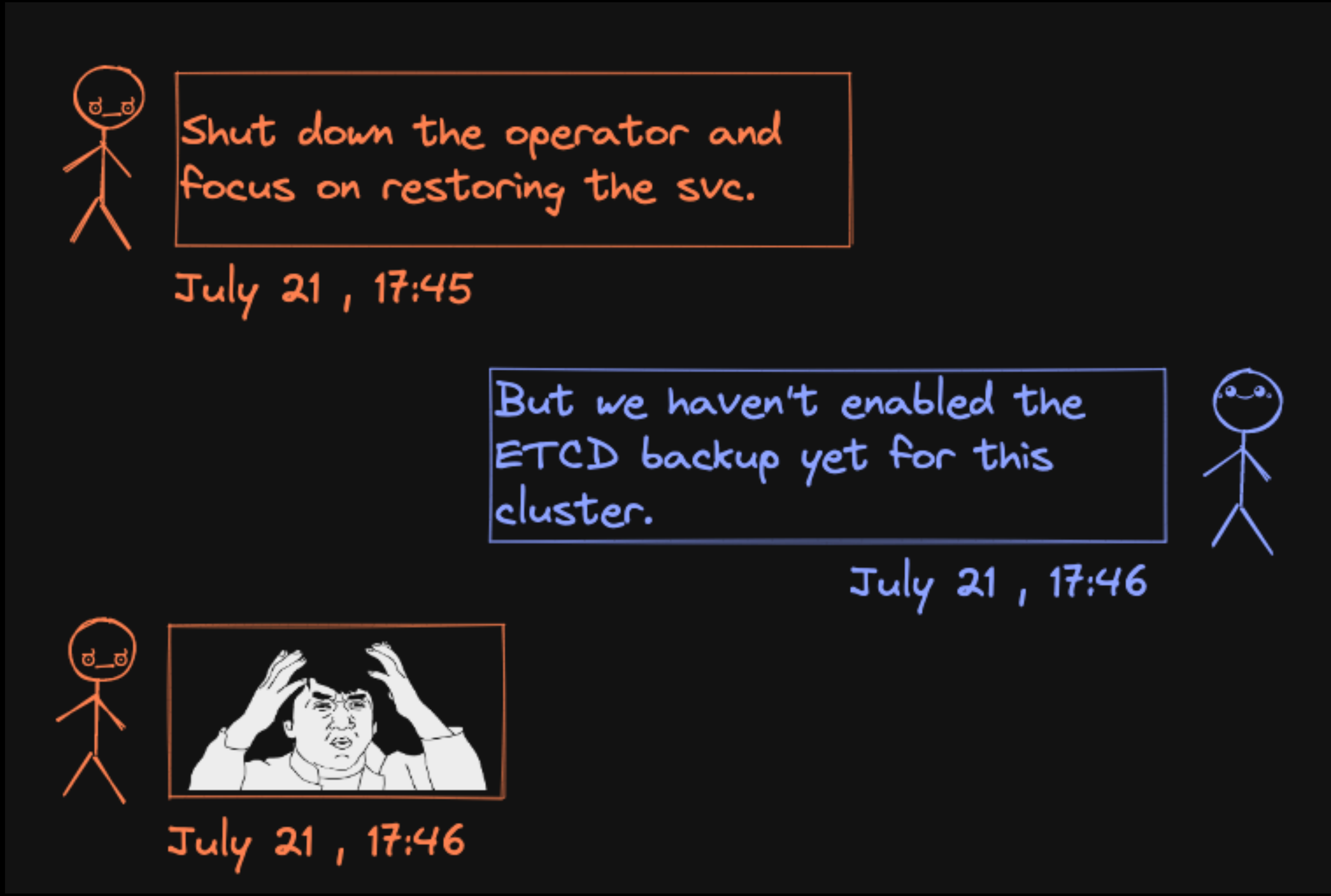
17:20 War room declared

17:30 Locate the offending operator

17:32 Another cluster screamed due to high traffic load

17:35 More users came in and complained

17:45 Shut down the operator


Shut down the operator and focus on restoring the svc.

July 21 , 17:45

But we haven't enabled the ETCD backup yet for this cluster.

July 21 , 17:46

July 21 , 17:46

Victim cluster property
Size:2K nodes, 30K pods
Usage: CI/CD, dev production
Status: Deprecating, ETA 2023

17:20 War room declared

17:30 Locate the offending operator

17:32 Another cluster screamed due to high traffic load

17:35 More users came in and complained

17:45 Shut down the operator

Service restoring

22:25 All services fully recovered

- **Collect svc from audit logs and other monitoring data**
- **Encourage users to self-restore**

17:20 War room declared

17:30 Locate the offending operator

17:32 Another cluster screamed due to high traffic load

17:35 More users came in and complained

17:45 Shut down the operator

18:10 Alert:TooManySLOFailures PodCreation

- **Second Wave: Stop creating pods**

22:25 All services fully recovered

17:20 War room declared

17:30 Locate the offending operator

17:32 Another cluster screamed due to high traffic load

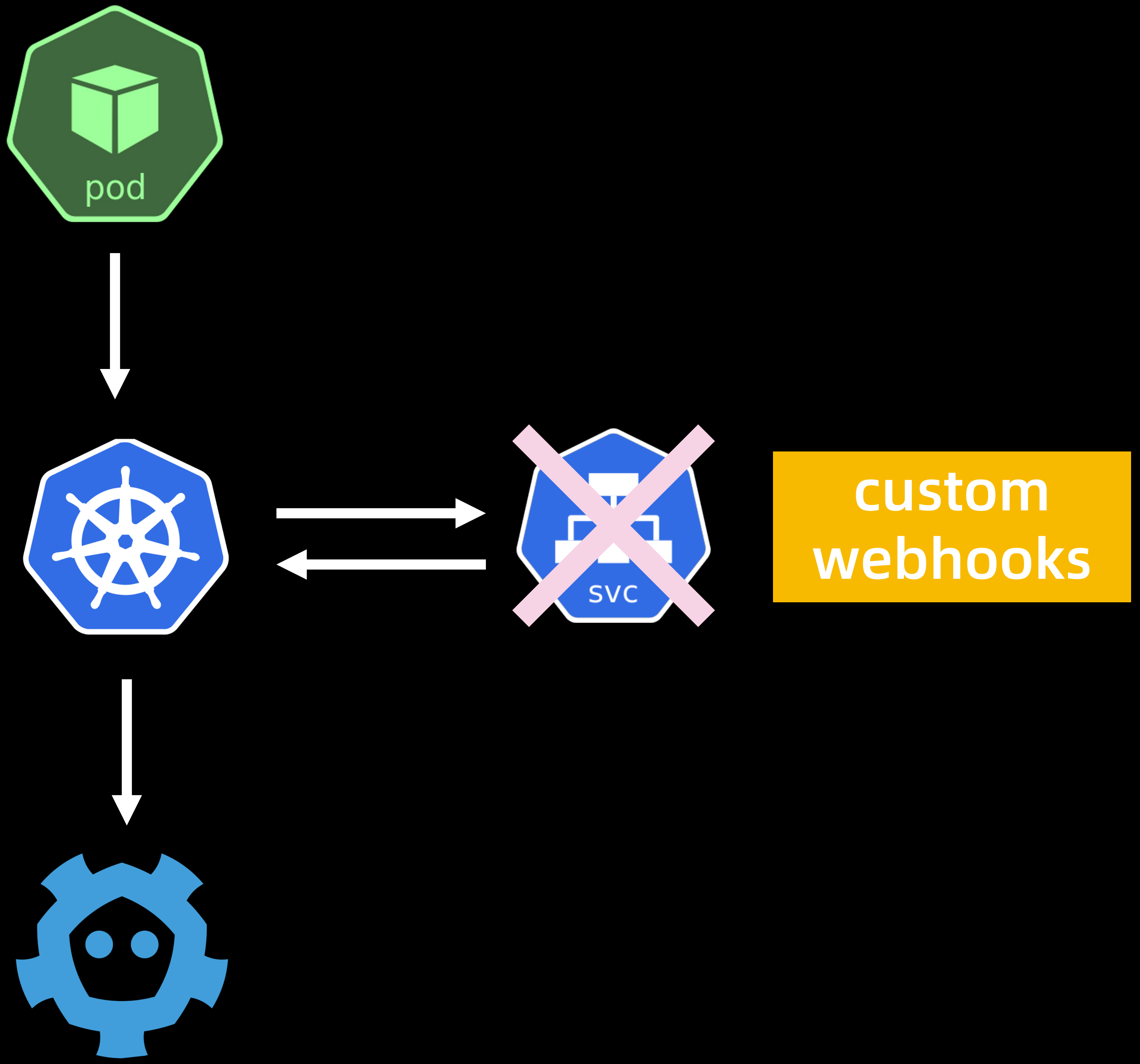
17:35 More users came in and complained

17:45 Shut down the operator

18:10 Alert:TooManySLOFailuresOnPodCreation

19:16 Pinpoint the missing service for pod creation

22:25 All services fully recovered



17:20 War room declared

17:30 Locate the offending operator

17:32 Another cluster screamed due to high traffic load

17:35 More users came in and complained

17:45 Shut down the operator

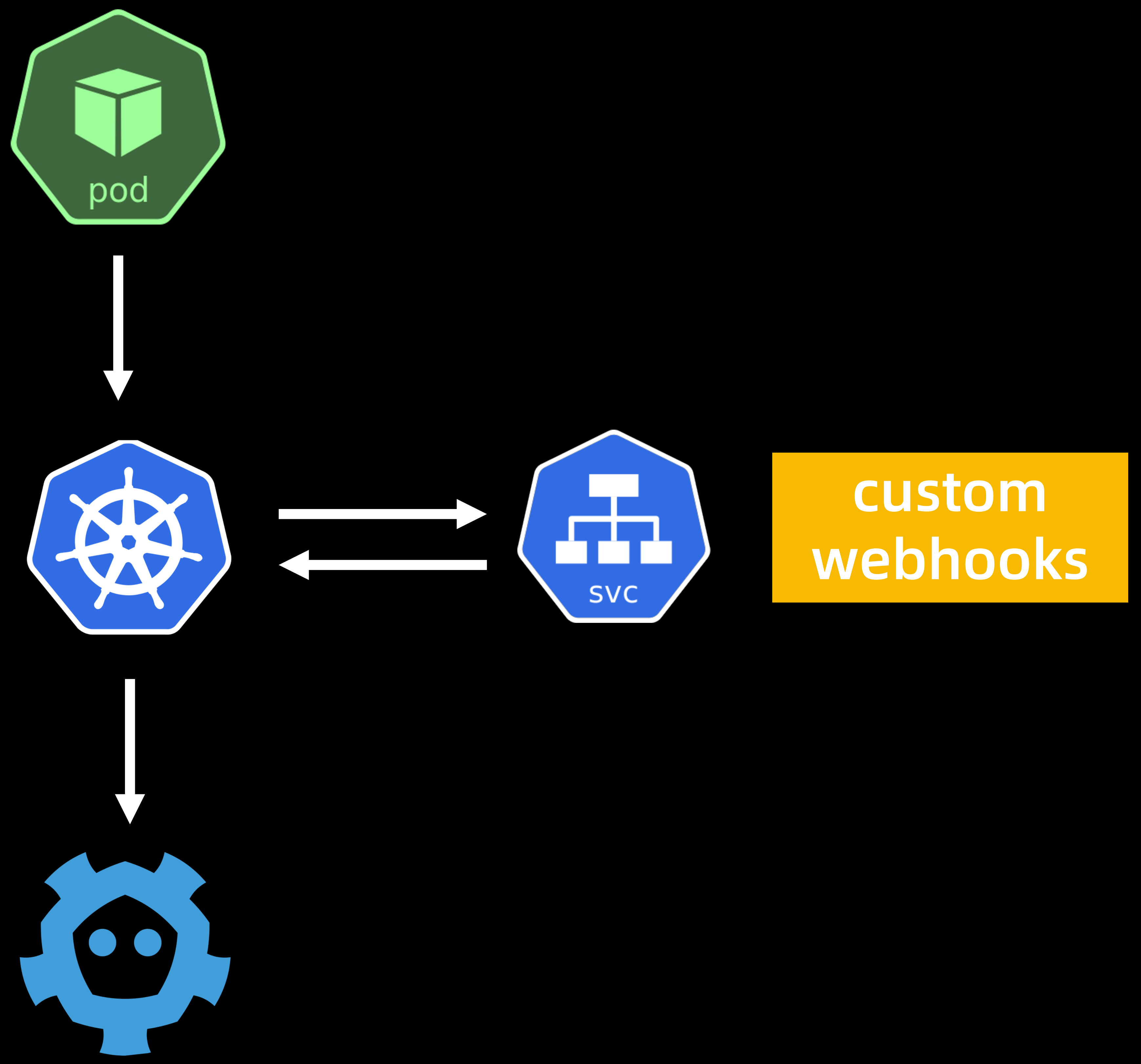
18:10 Alert:TooManySLOFailuresOnPodCreation

19:16 Pinpoint the missing service for pod creation

19:20 Fix all critical webhook svcs

Human errors by service
owners delayed the recovery

22:25 All services fully recovered



17:20 War room declared

17:30 Locate the offending operator

17:32 Another cluster screamed due to high traffic load

17:35 More users came in and complained

17:45 Shut down the operator

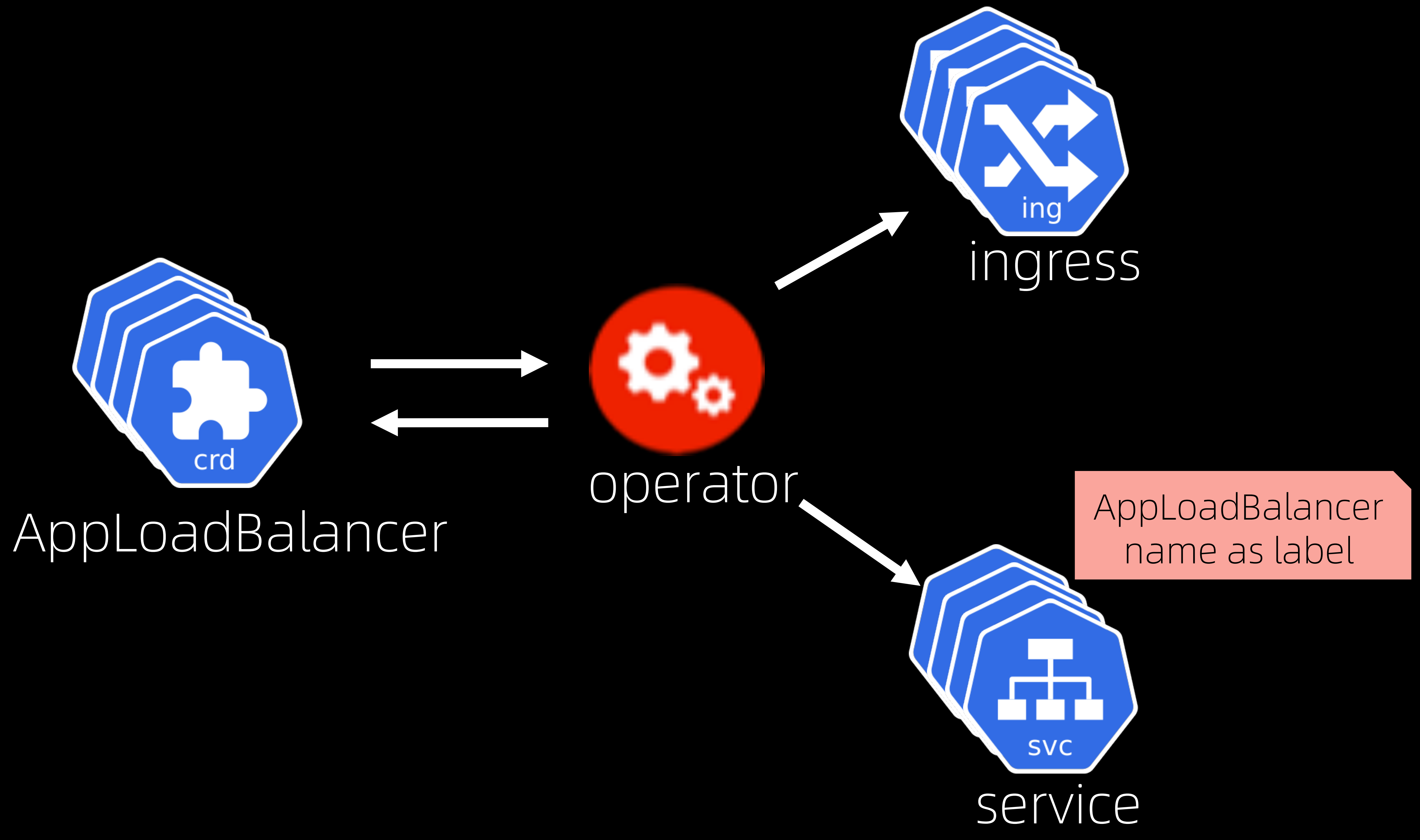
18:10 Alert:TooManySLOFailuresOnPodCreation

19:16 Pinpoint the missing service for pod creation

19:20 Fix all critical webhook svcs

19:37 Root cause understood

22:25 All services fully recovered



17:20 War room declared

17:30 Locate the offending operator

17:32 Another cluster screamed due to high traffic load

17:35 More users came in and complained

17:45 Shut down the operator

18:10 Alert:TooManySLOFailuresOnPodCreation

19:16 Pinpoint the missing service for pod creation

19:20 Fix all critical webhook svcs

19:37 Root cause understood

22:25 All services fully recovered

23:57 Operator code fixed

```
// apimachinery/pkg/labels/selector.go
// SelectorFromSet returns a Selector which will match
// exactly the given Set. A nil and empty Sets are
// considered equivalent to Everything().

if len(labelKey) > qualifiedNameMaxLength
    return "A nil Selector Set"
```

Operator selected ALL services from a nil selector and deleted them

operator

```
// from standard k8s pkg.
qualifiedNameMaxLength = 63
```

Victim cluster

```
// from our custom k8s pkg.
qualifiedNameMaxLength = 127
```

17:00 - A test engineer accidentally created an AppLoadBalancer with the name length > 63 and deleted it.

Lesson Learned

Observability
Audit log is important.

Operator Development
Audit the permission and scope.
Use shared client library.

Data Integrity
Backup! Backup! Backup!

Precaution
Rate limiting on risky operations like mutation / deletion.
Alert on abnormal cluster-level behaviors.

🏁 02

The paradoxical finalizer

Let’s NOT create any dependency loop.

🔥🔥🔥 03:51:00AM Alert:TooManyPodCreationFailures at cluster X

🔥🔥🔥 03:51:15AM Alert:TooManyPodDeletionFailures at cluster Y

🔥🔥🔥 03:51:20AM Alert:TooManyPodCreationFailures at cluster Z

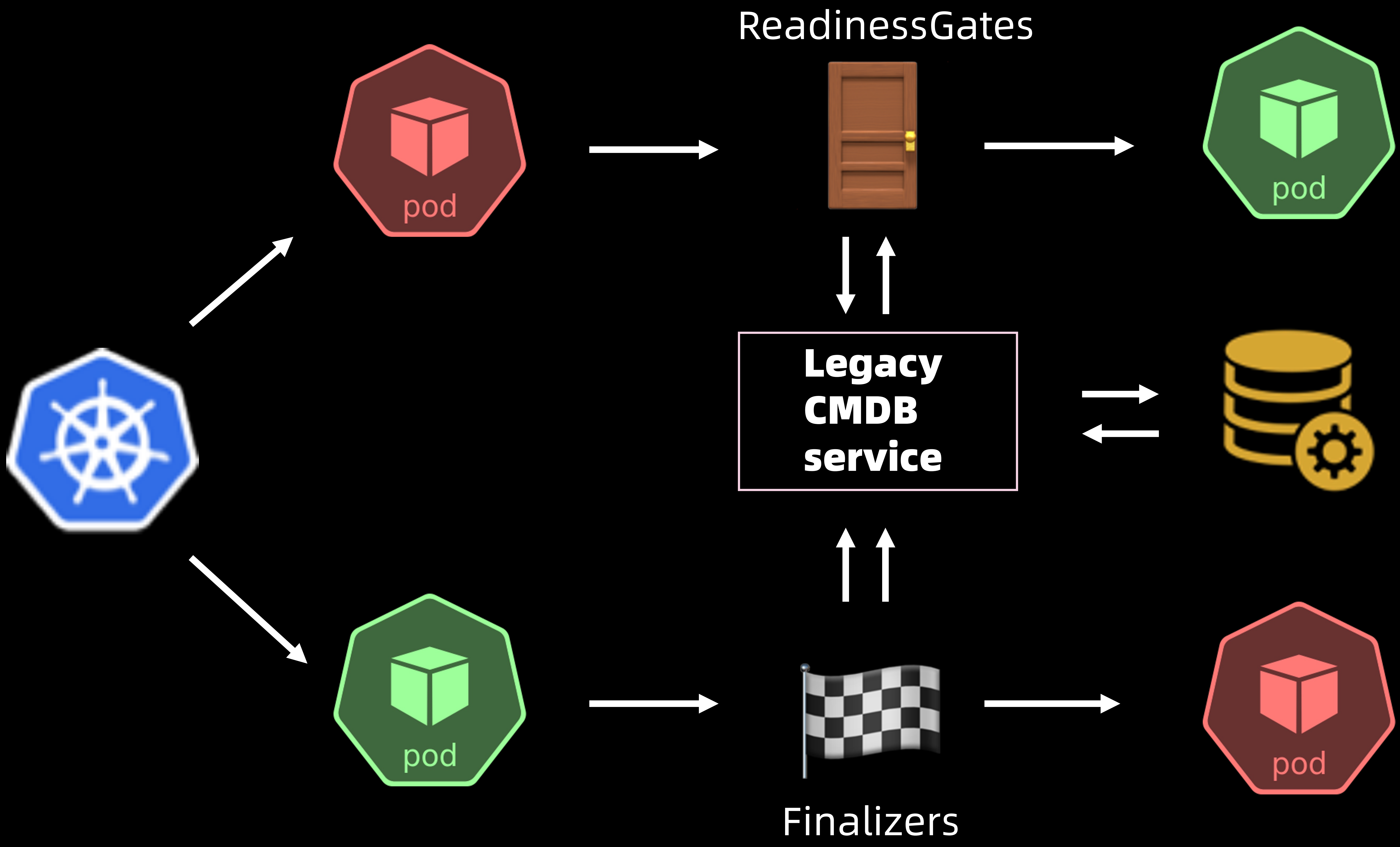
...

🔥🔥🔥 03:52:30AM Alert:TooManyPodDeletionFailures at cluster T

Possible Guess

It’s 4AM, so...

Rollout	High Traffic Load	Global Dependencies
✗	✗	?



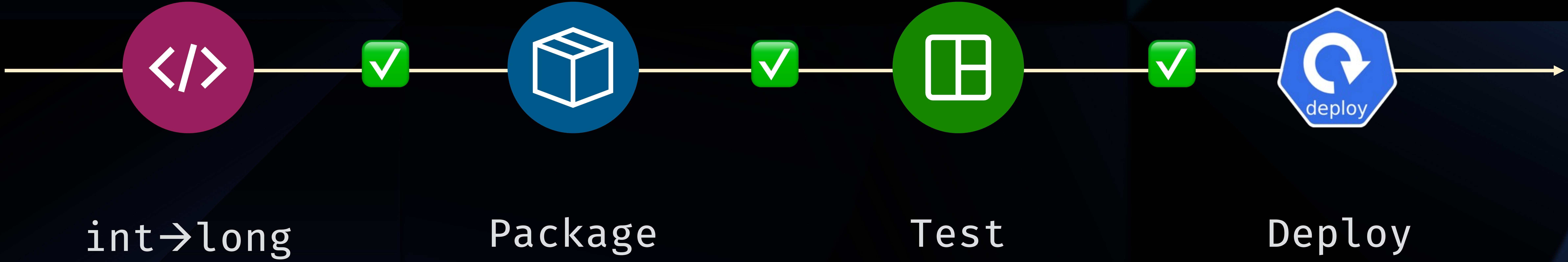
Global Dependencies

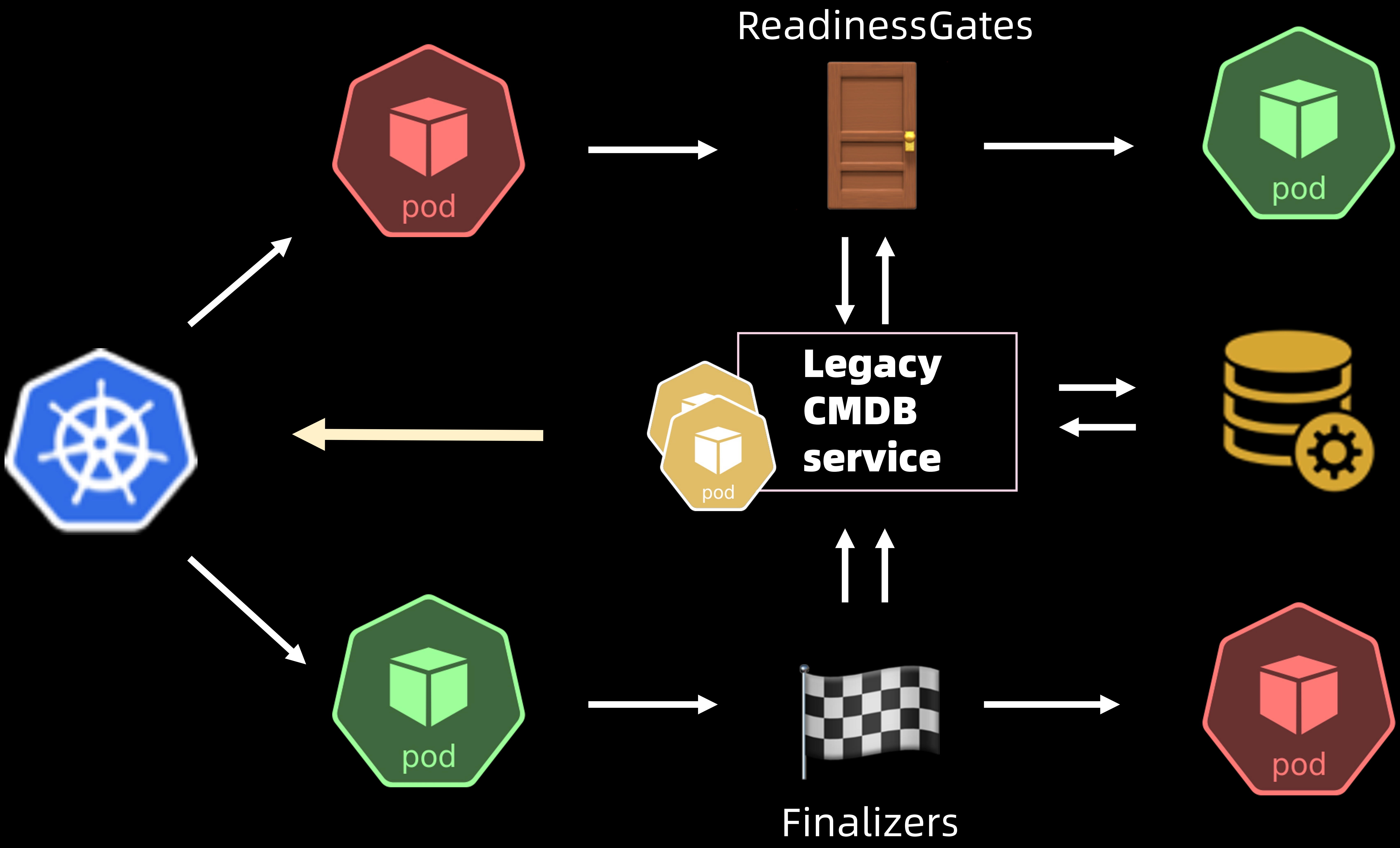
It’s called legacy for a reason

```
SQLDataException: ‘2.157132229E9’ in
column ‘1’ is outside valid range for the
datatype INTEGER.
```

Mitigation

Ideally...

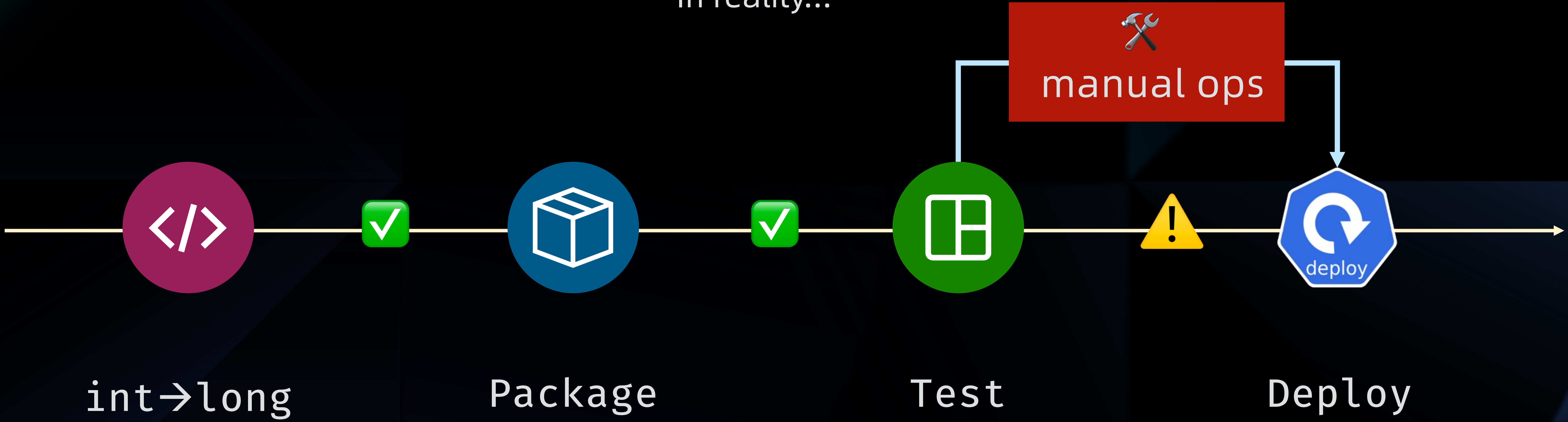




Dependency Loop

Mitigation

In reality...



Lesson Learned

Dependency Loops

Remove global dependency

Legacy Systems

Examine by chaos attack

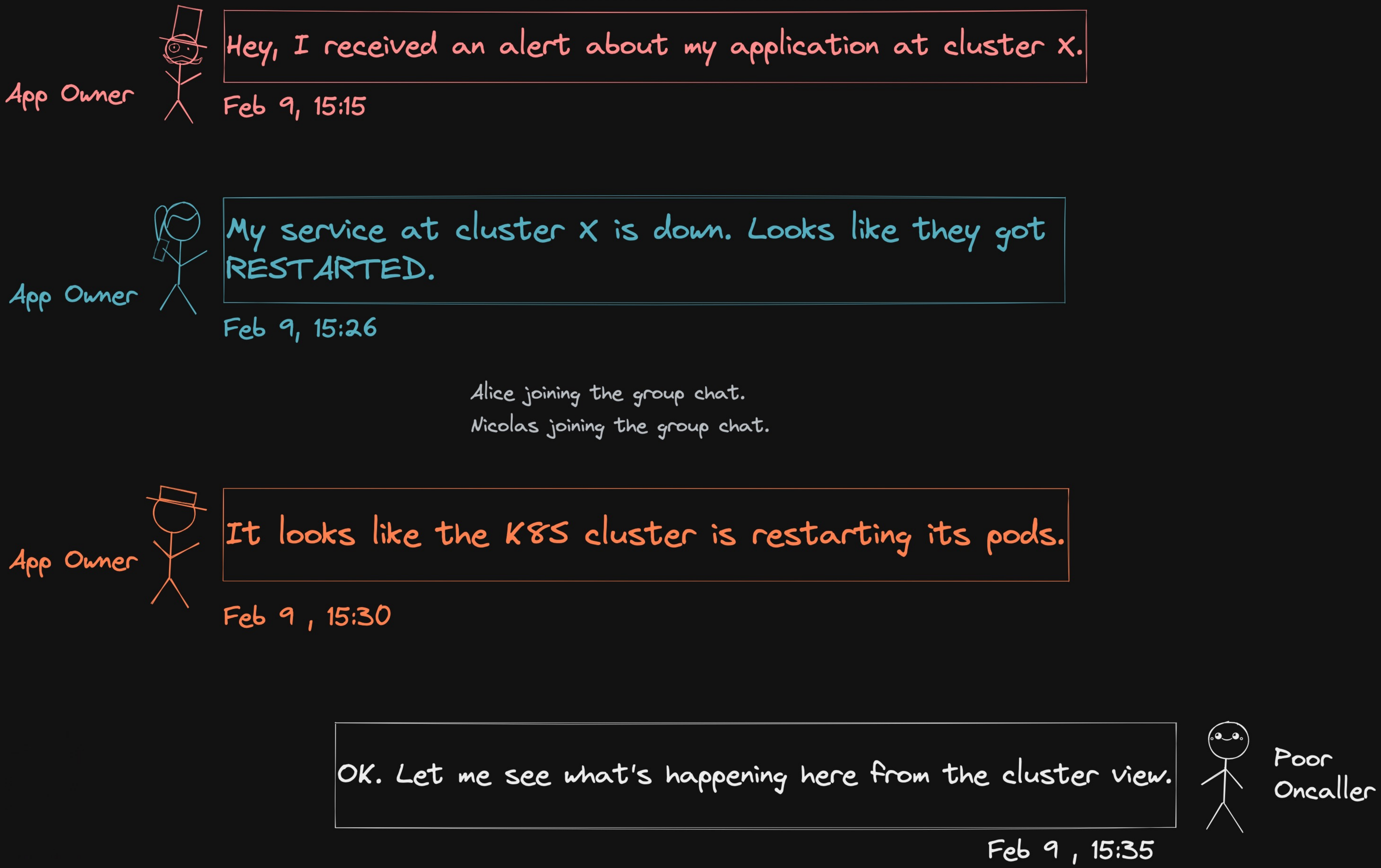
Automation

Practice manual operations

🪝03

The evil webhook

How would you do a canary rollout for a webhook?



App Owner

Hey, I received an alert about my application at cluster X.

Feb 9, 15:15

App Owner

My service at cluster X is down. Looks like they got RESTARTED.

Feb 9, 15:26

Alice joining the group chat.

Nicolas joining the group chat.

App Owner

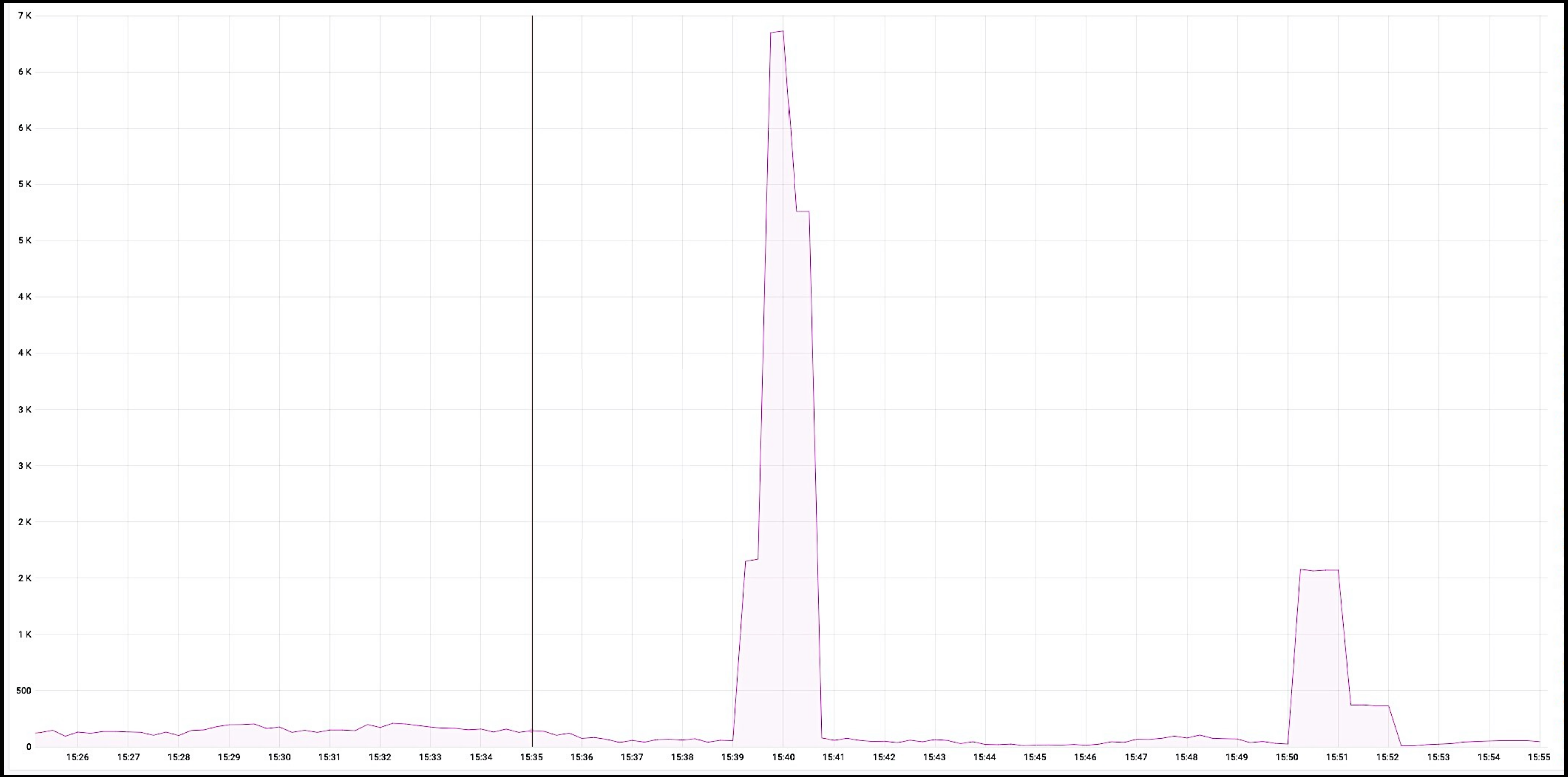
It looks like the K8S cluster is restarting its pods.

Feb 9 , 15:30

OK. Let me see what's happening here from the cluster view.

Poor Oncaller

Feb 9 , 15:35



Number of restarted PODS in the victim cluster

Victim cluster property
Size:1K nodes, 23K pods
Usage: canary, 1% production
Status: Serving

Given the following conditions:

- Pods belonged to multiple owners with no obvious correlation
- DeamonSet, Deployment, StatefulSet pods were all affected
- Not sure if another wave was coming in

Victim cluster property
Size:1K nodes, 23K pods
Usage: canary, 1% production
Status: Serving

The Pod Spec Change

from a dynamic mutating admission webhook

```
"securityContext”: {
    ...
    “capabilities”: {},



    ...
}
```

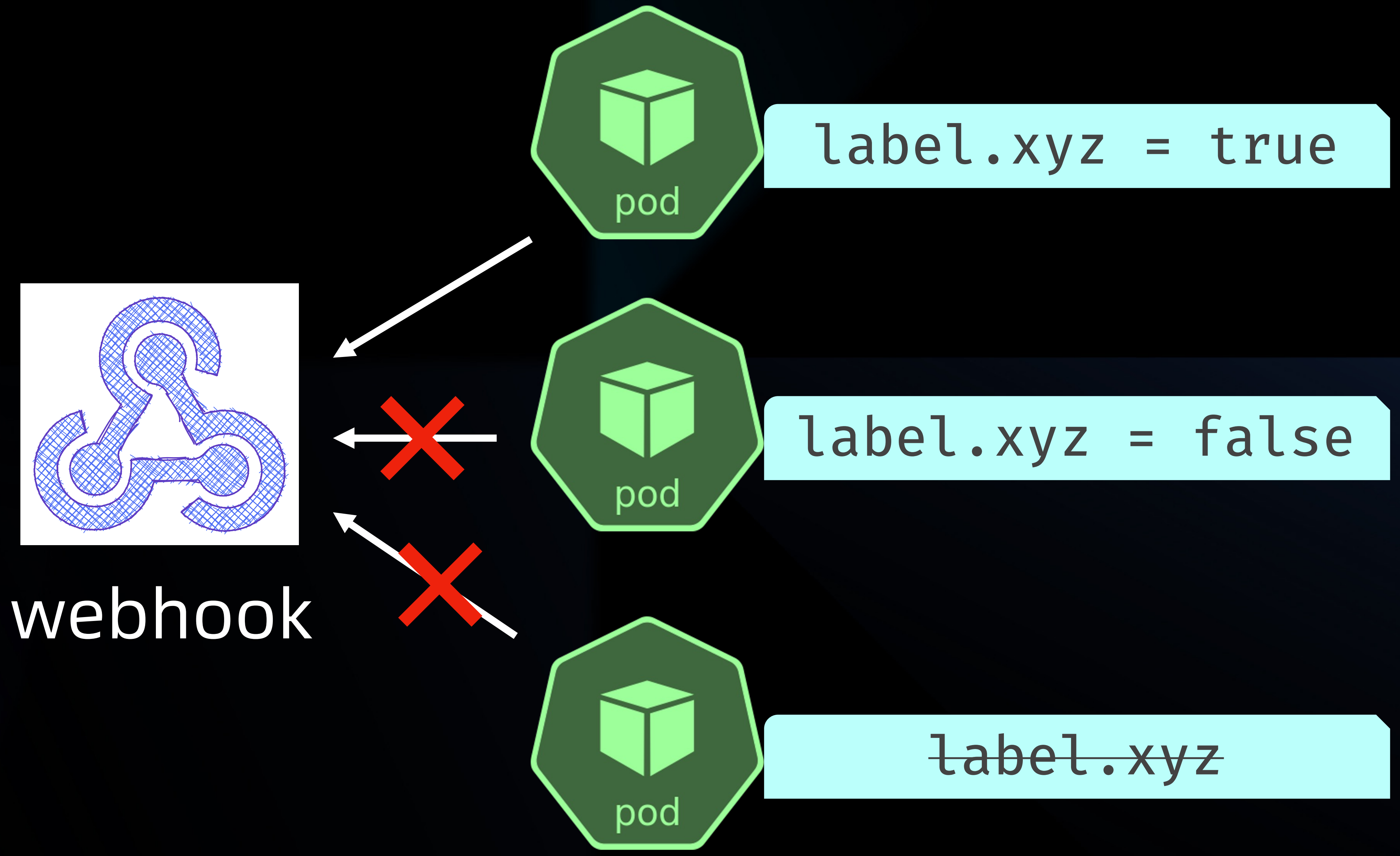
old

```
"securityContext”: {
    ...
    “capabilities”: {
        “drop”:[
            “SYS_MODULE”,
            ”DAC_READ_SEARCH”
        ]
    },
    ...
}
```

new

The Webhook: Previously

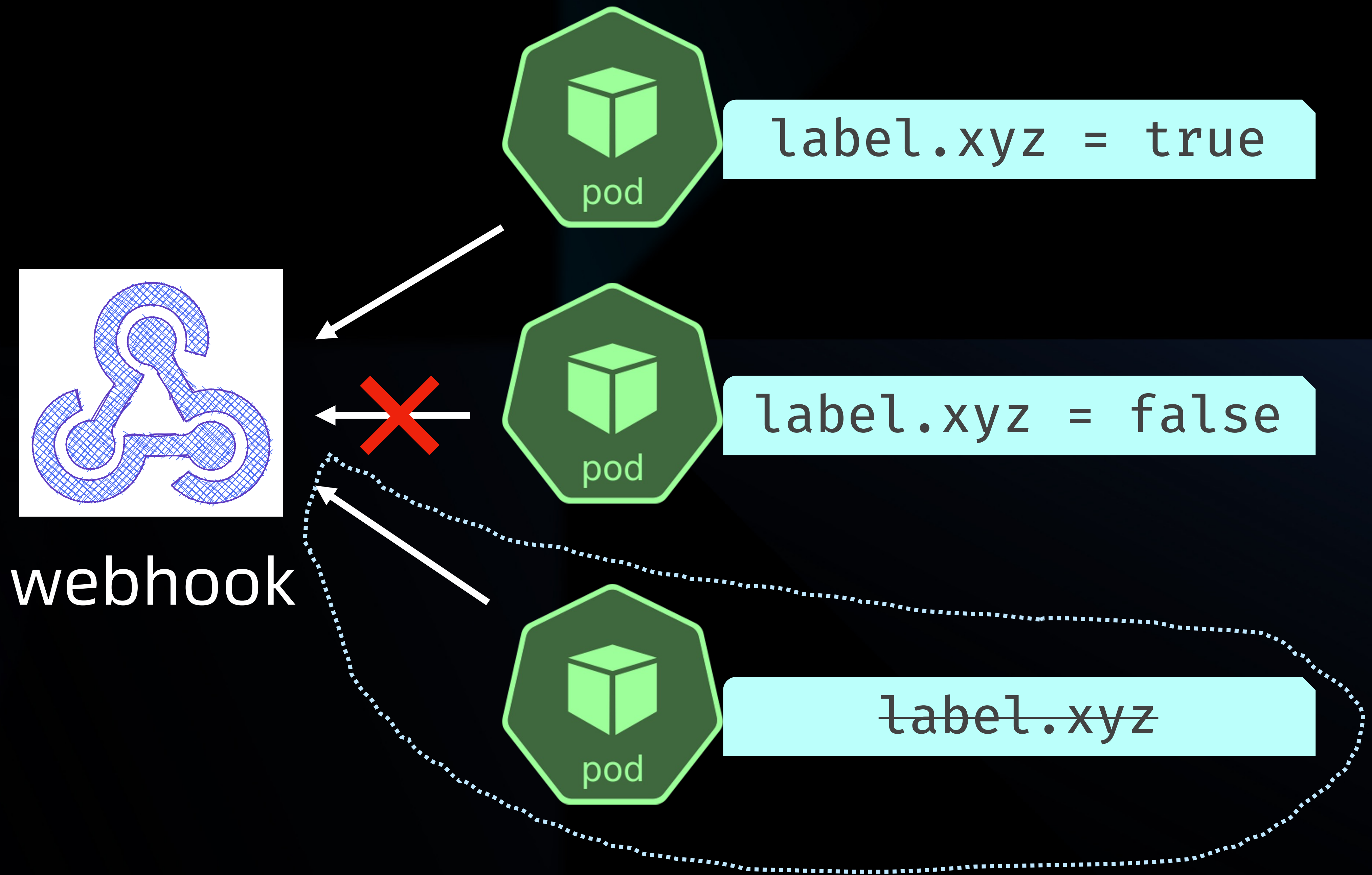
```
apiVersion: admissionregistration.k8s.io/v1
kind: MutatingWebhookConfiguration
…
 rules:
  - apiGroups:
    - ""
    apiVersions:
    - v1
    operations:
    - CREATE
    - UPDATE
    resources:
    - pods
    scope: ‘*’
```



Mutate on CREATE / UPDATE events for pods IFF dedicated label xyz = true.

The Webhook: Buggy

```
apiVersion: admissionregistration.k8s.io/v1
kind: MutatingWebhookConfiguration
 …
 rules:
  - apiGroups:
    - ""
    apiVersions:
    - v1
    operations:
    - CREATE
    - UPDATE
    resources:
    - pods
    scope: ‘*’
```



Mutate on CREATE / UPDATE events for pods IFF dedicated label xyz = true.

Don’t mutate on CREATE / UPDATE events for pods IFF dedicated label xyz = false.

Lesson Learned

Progressive Rollout
Set blast radius for webhooks explicitly.

Change Management
Restrict the usage of dynamic mutating webhooks.

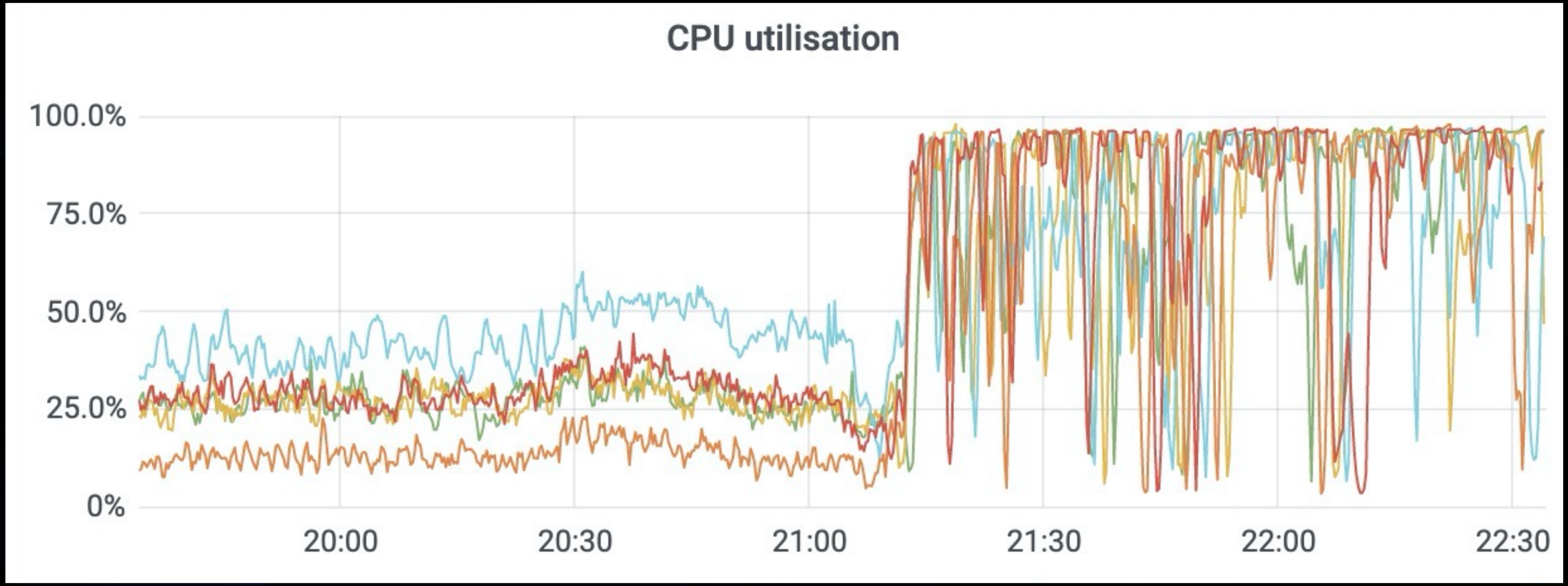
Audit log
Organize monitoring data to facilitate debugging.

Team Collaboration
Educate and communicate in the same language.

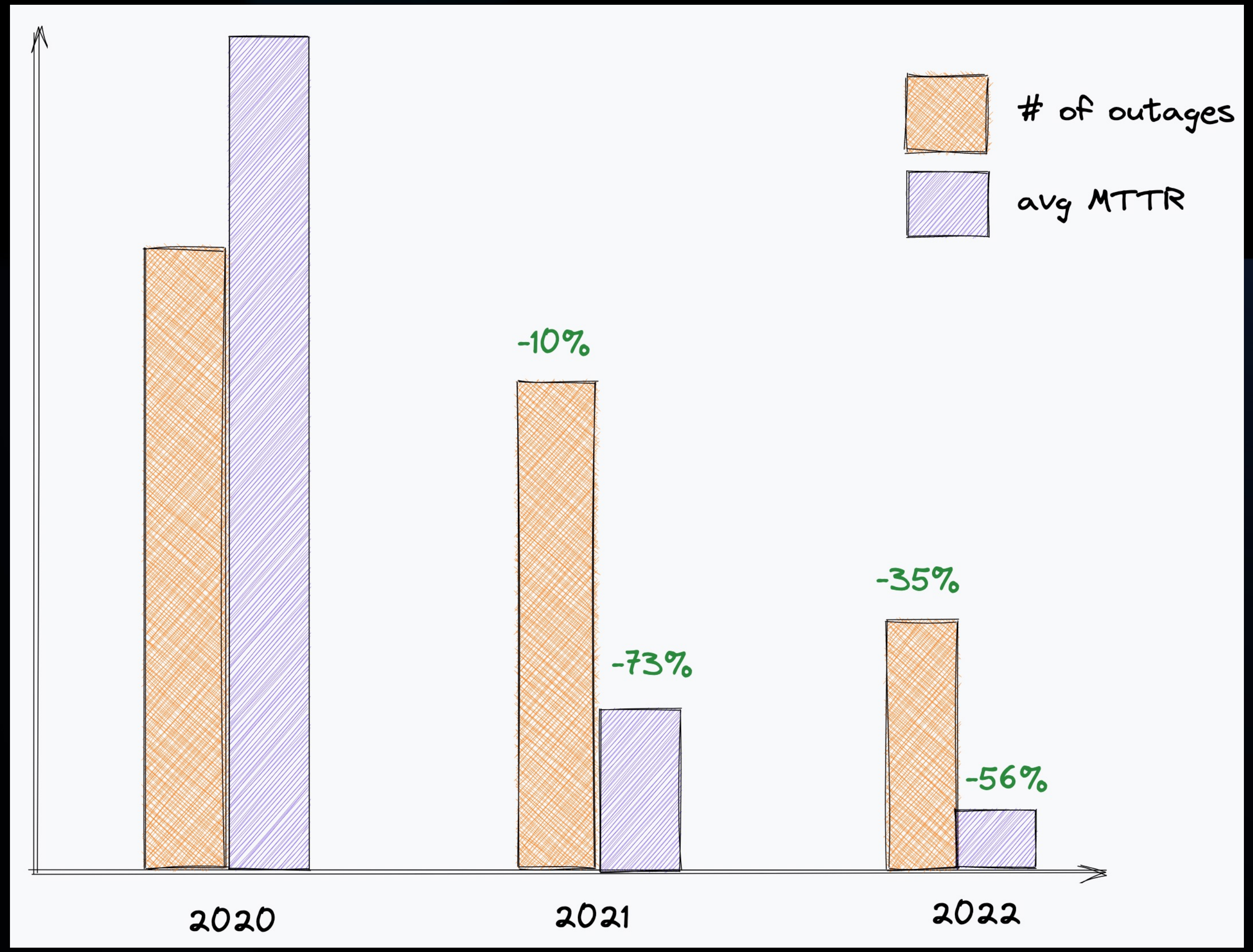
Summary

It’s already too complicated...

Trends



kube-apiserver performance
during our 1.16 -> 1.18 upgrade



# of outages and average MTTR

Key Takeaways

Observability is still the key: #audit #log #traces

Change management is hard: #scopes #permissions #dependencies

Large-scale clusters have different implication: #integrity #redundancy

Communication and education: #manual ops #incident management

SRE CON ASIA PACIFIC

Thanks