



redhat<sup>®</sup>

# Troubleshooting Kubernetes apps

- troubleshooting.kubernetes.sh

Michael Hausenblas @mhausenblas

Developer Advocate, Red Hat

2019-02-08, KubeCologne

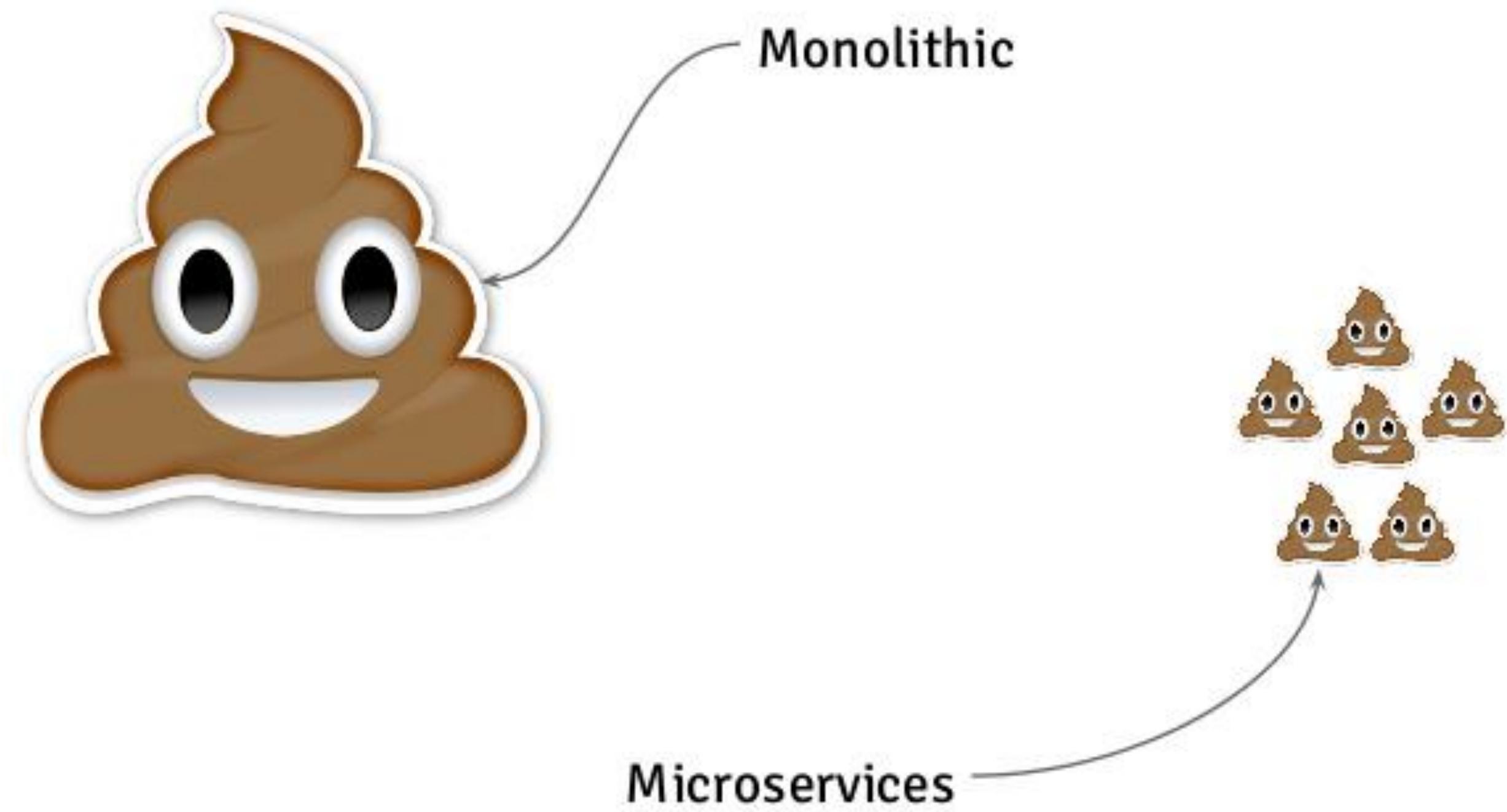
# Scope

- Focusing on prototyping, developing, and testing applications with Kubernetes from an appops perspective (tools & techniques)
- But not really (much) about ...
  - troubleshooting installation or upgrading issues
  - performance testing or optimising containerized microservices
  - SRE-style troubleshooting (check out what Googlers say **on this topic**)

# Monoliths vs. microservices



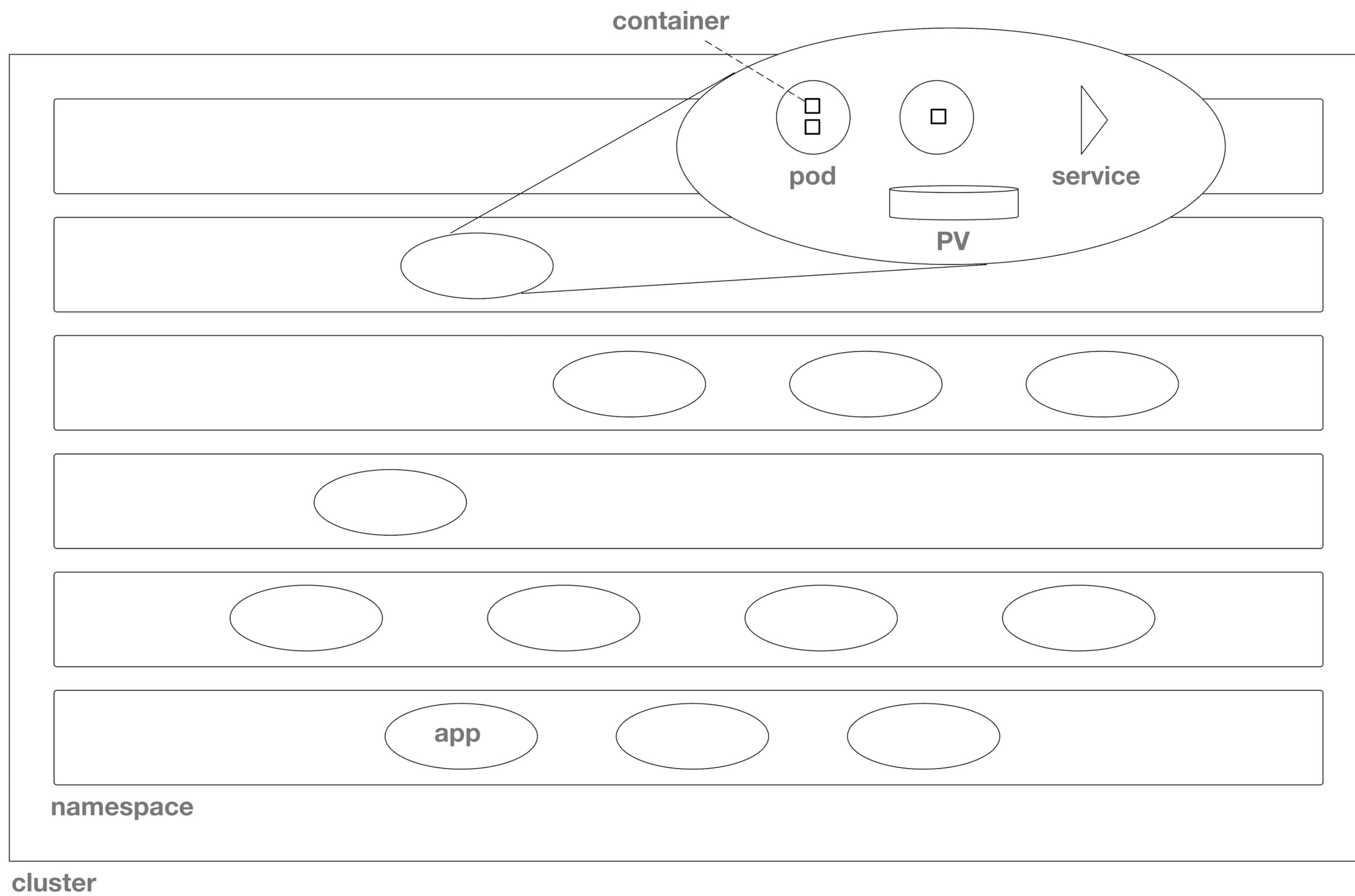
# Monolithic vs Microservices



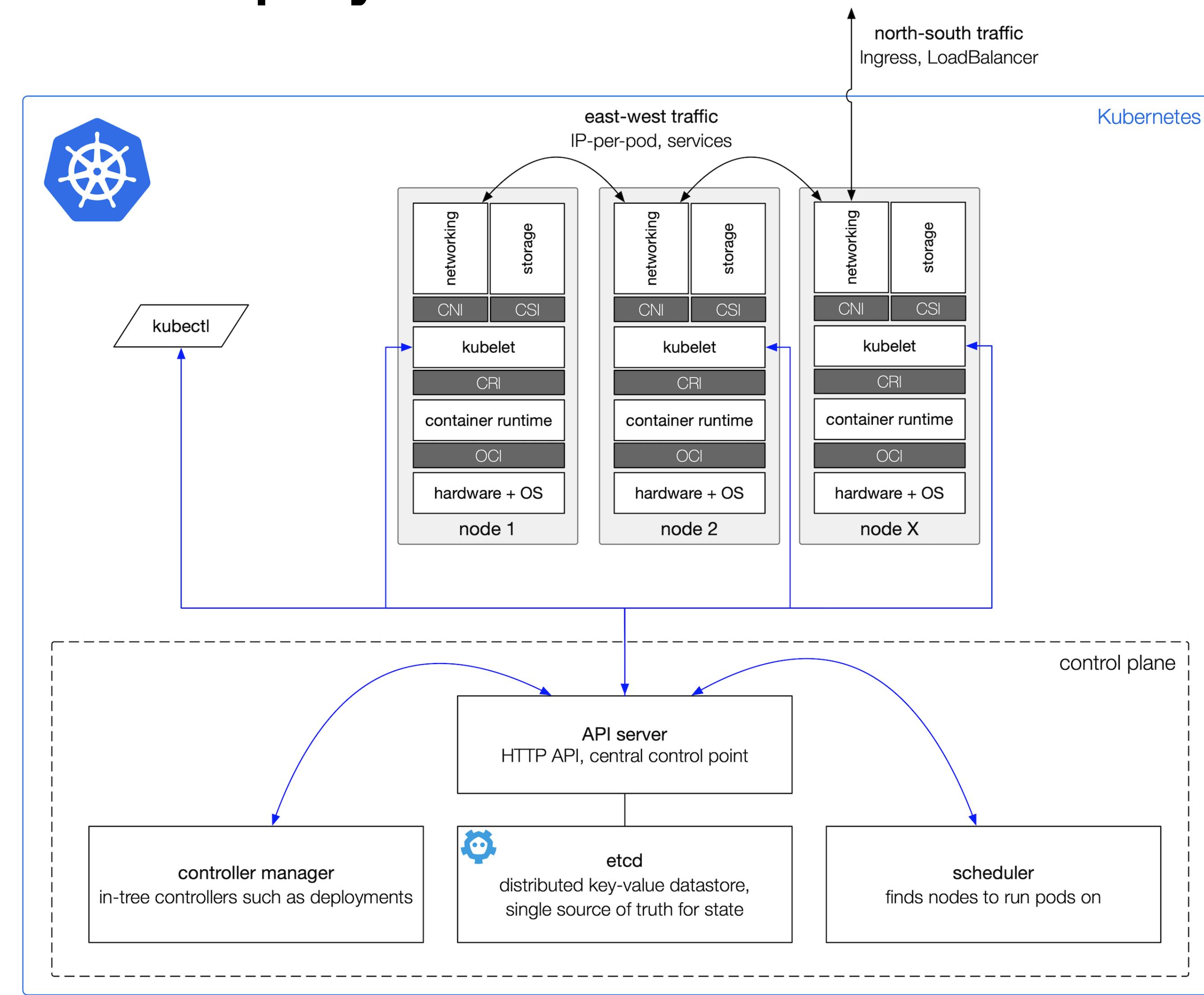
[@alvaro\\_sanchez](#)

**odobo**

# Moving parts—logical view

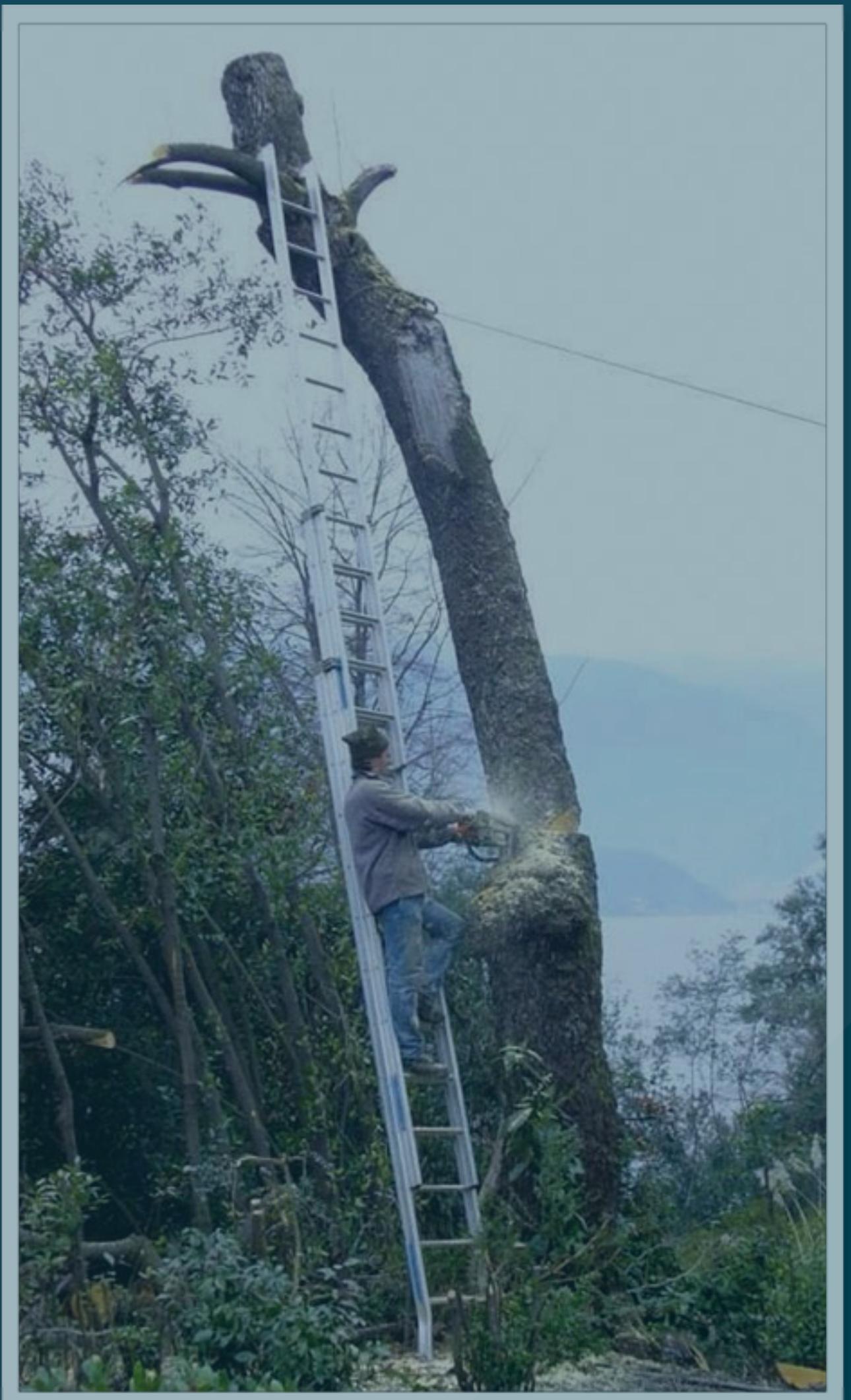


# Moving parts—physical view





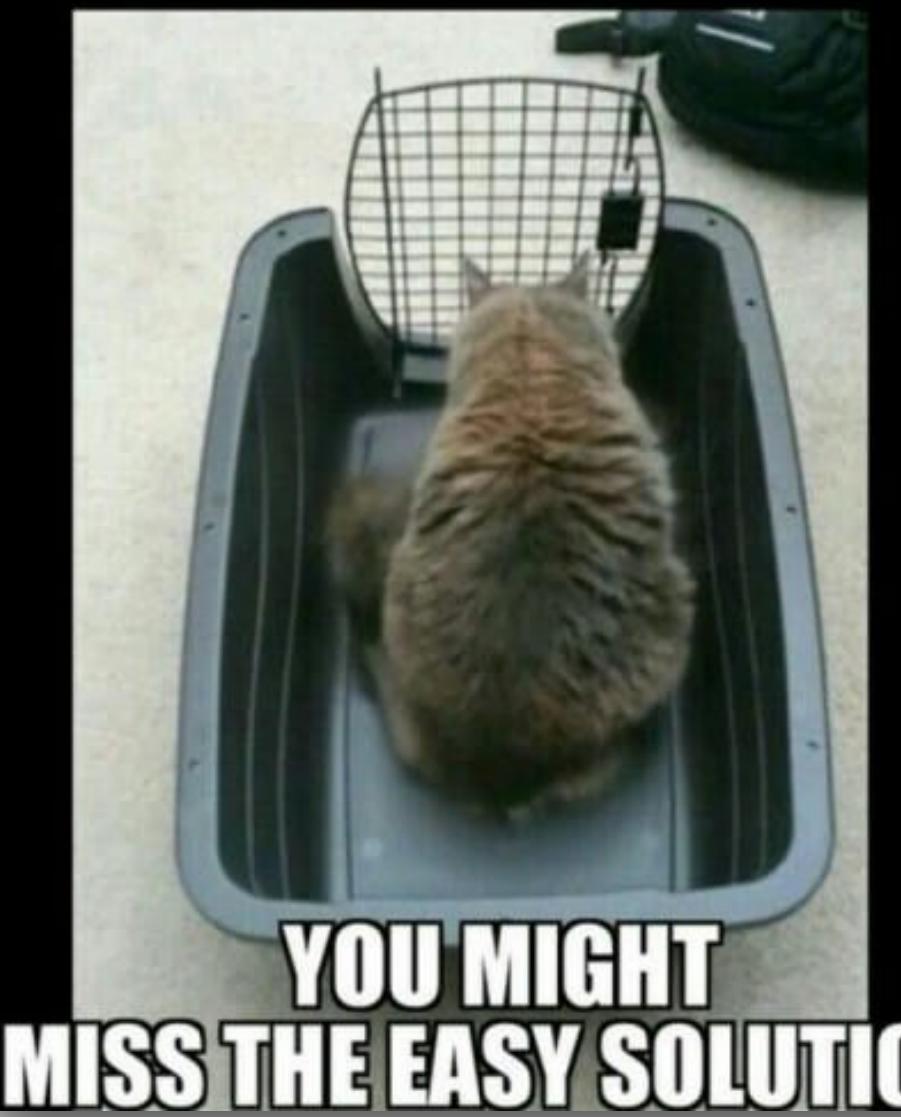
# TOP 10 failures



**IF YOU ONLY  
FOCUS ON THE PROBLEM**

# The TOP 10 list

1. invalid YAML specification
2. wrong or missing permissions
3. wrong container image
4. no access to container registry
5. supposedly long-running application exits



**YOU MIGHT  
MISS THE EASY SOLUTION**

# The TOP 10 list

6. missing/bad config or secret
7. lifecycle issues (probes fail)
8. can't reach service
9. looking at the wrong place—where is `localhost`?
10. failed mounts

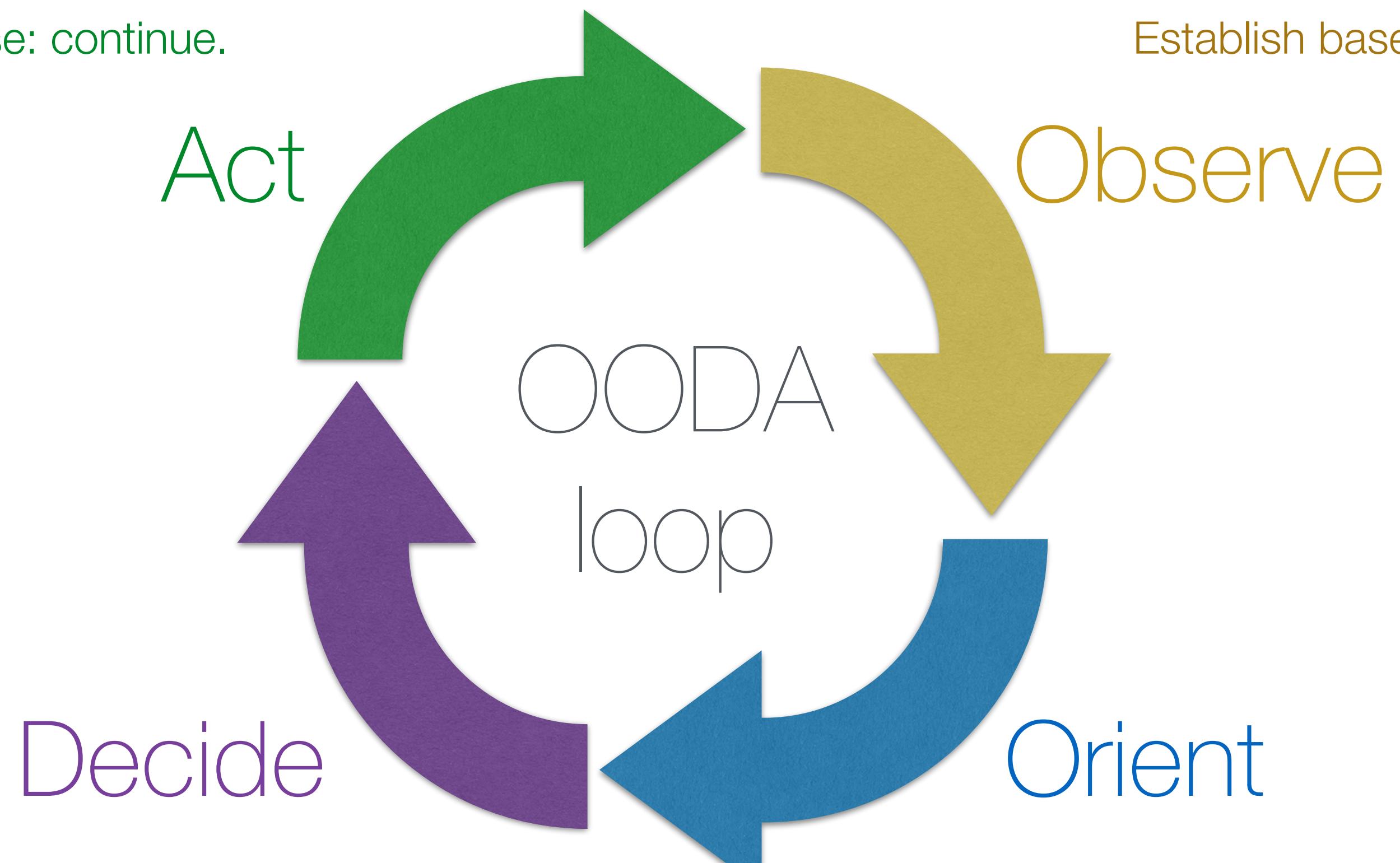


# Not just poking around ...

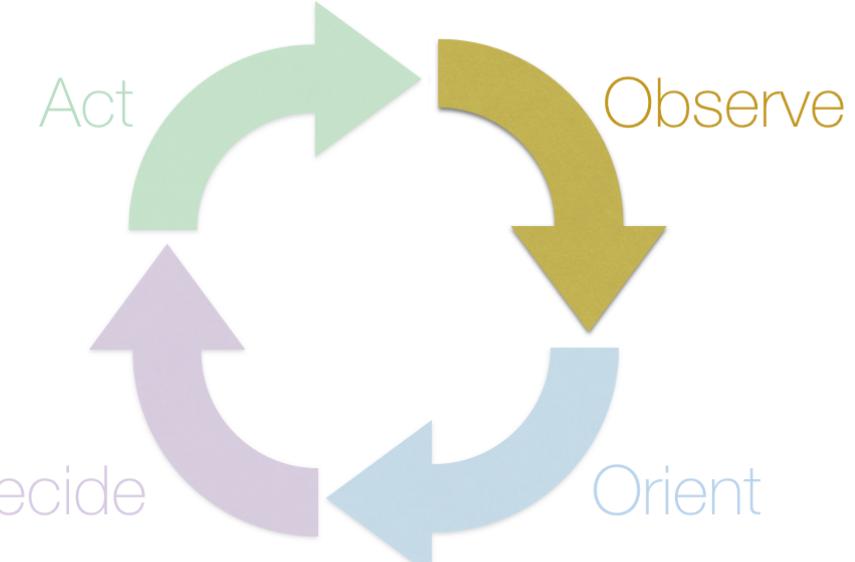


Test the hypothesis you picked.  
If confirmed: fix it, else: continue.

What's in the logs?  
Establish baseline.

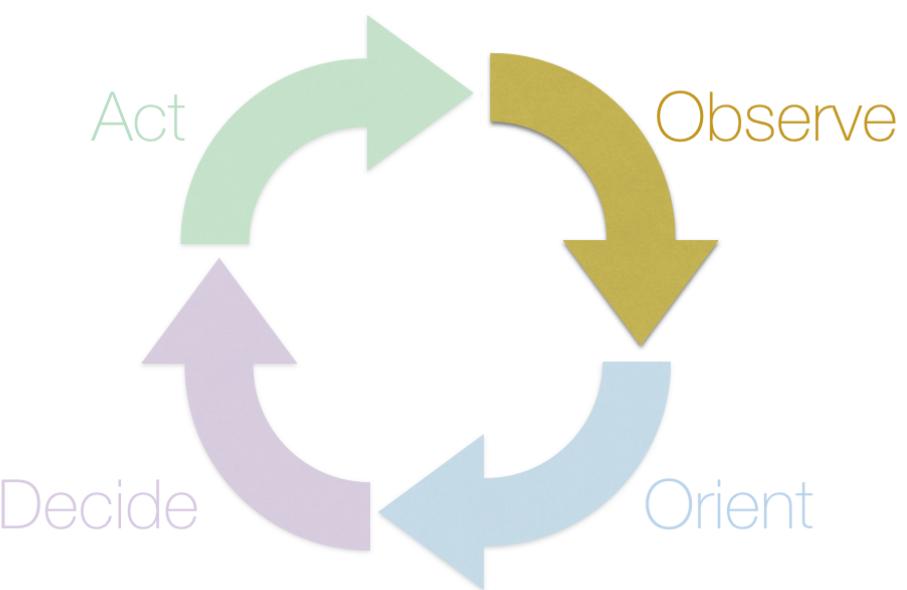


```
1 apiVersion: extensions/v1beta1
2 kind: Deployment
3 metadata:
4   name: unhappy-camper
5 spec:
6   replicas: 1
7   template:
8     metadata:
9       labels:
10      app: whatever
11   spec:
12     containers:
13       - name: shell
14         image: centos:7
15       command:
16         - sh
17         - '-c'
18         - echo "I will just print something here and then exit"
```

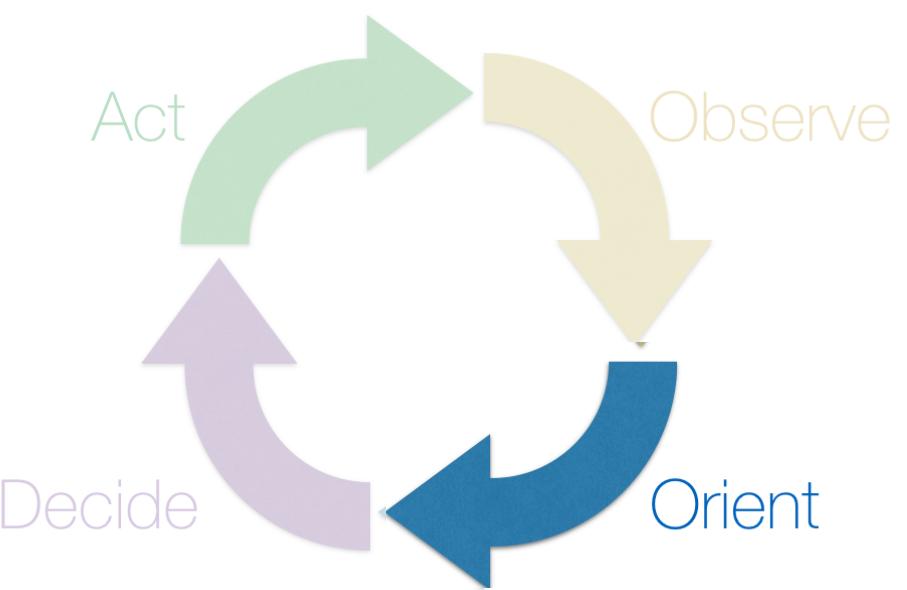


```
$ kubectl describe deploy/unhappy-camper
Name:          unhappy-camper
Namespace:     tka
CreationTimestamp:  Wed, 06 Feb 2019 11:18:18 +0000
Labels:        app=whatever
Annotations:   deployment.kubernetes.io/revision: 1
               kubectl.kubernetes.io/last-applied-configuration:
                           {"apiVersion":"extensions/v1beta1","kind":"Deployment","metadata":{"annotations":{}},
Selector:      app=whatever
Replicas:      1 desired | 1 updated | 1 total | 0 available | 1 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 1 max unavailable, 1 max surge
$ kubectl describe po/$THEPOD
Name:          unhappy-camper-7b9f8bfb4-ctp82
Namespace:     tka
Priority:      0
PriorityClassName: <none>
Node:          ip-172-31-54-2.ec2.internal
Events:
  Type    Reason     Age   From           Message
  ----  -----     --   --            --
  Normal  Scheduled  2m31s  default-scheduler  Successfully assigned tka/unhappy-camper-7b9f8bfb4-ctp82 to ip-172-31-54-2.ec2.internal
  Normal  Pulling    100s (x4 over 2m29s)  kubelet, ip-172-31-54-2.ec2.internal  pulling image "centos:7"
  Normal  Pulled    98s (x4 over 2m27s)   kubelet, ip-172-31-54-2.ec2.internal  Successfully pulled image "centos:7"
  Normal  Created    98s (x4 over 2m27s)   kubelet, ip-172-31-54-2.ec2.internal  Created container
  Normal  Started    98s (x4 over 2m27s)   kubelet, ip-172-31-54-2.ec2.internal  Started container
  Warning BackOff   83s (x6 over 2m25s)   kubelet, ip-172-31-54-2.ec2.internal  Back-off restarting failed container
  -----
  Available  True    MinimumReplicasAvailable
  Progressing True   ReplicaSetUpdated
OldReplicaSets: <none>
NewReplicaSet:  unhappy-camper-7b9f8bfb4 (1/1 replicas created)
Events:
  Type    Reason     Age   From           Message
  ----  -----     --   --            --
  Normal  ScalingReplicaSet  101s  deployment-controller  Scaled up replica set unhappy-camper-7b9f8bfb4 to 1
```

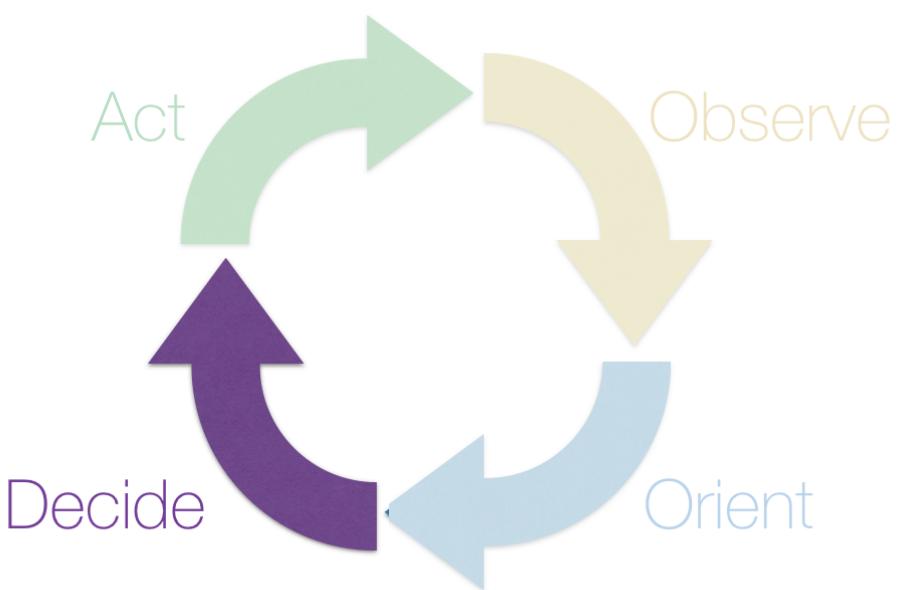
I will just print something here and then exit



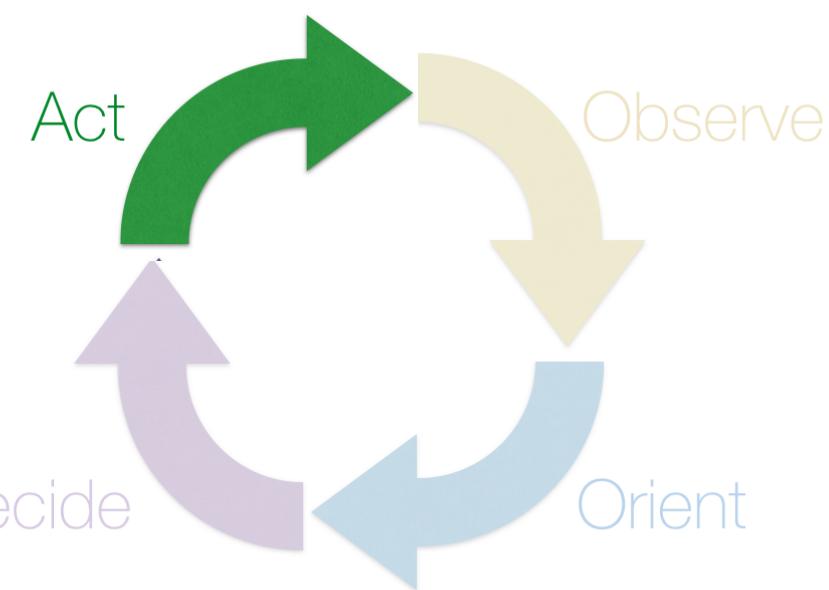
- Deployment seems OK
- Pod seems OK (image found, scheduled, launched)
- I see log output, so container is running
- Keeps crashing after launch



- Could be a resource issues (OOM, etc.)
- Could be config/data missing
- Could be an application logic/runtime error



- 1 . Could be an application logic/runtime error
- 2 . Could be a resource issues (OOM, etc.)
- 3 . Could be config/data missing



```
$ kubectl edit deploy/unhappy-camper
```

command:

```
- sh  
- '--c'  
- echo "I will just print something here and then exit" && sleep 1000
```

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
unhappy-camper-55f8488fb4-wn79x	1/1	Running	0	9s

# The How

- Using `kubectl get events`
- Using `kubectl describe`
- Using `kubectl exec`
- Using `kubectl logs` (or `kubetail`, `stern`)
- Full-blown observability approaches

WHEN A USER TAKES A PHOTO,  
THE APP SHOULD CHECK WHETHER  
THEY'RE IN A NATIONAL PARK...

SURE, EASY GIS LOOKUP.  
GIMME A FEW HOURS.

...AND CHECK WHETHER  
THE PHOTO IS OF A BIRD.

I'LL NEED A RESEARCH  
TEAM AND FIVE YEARS.

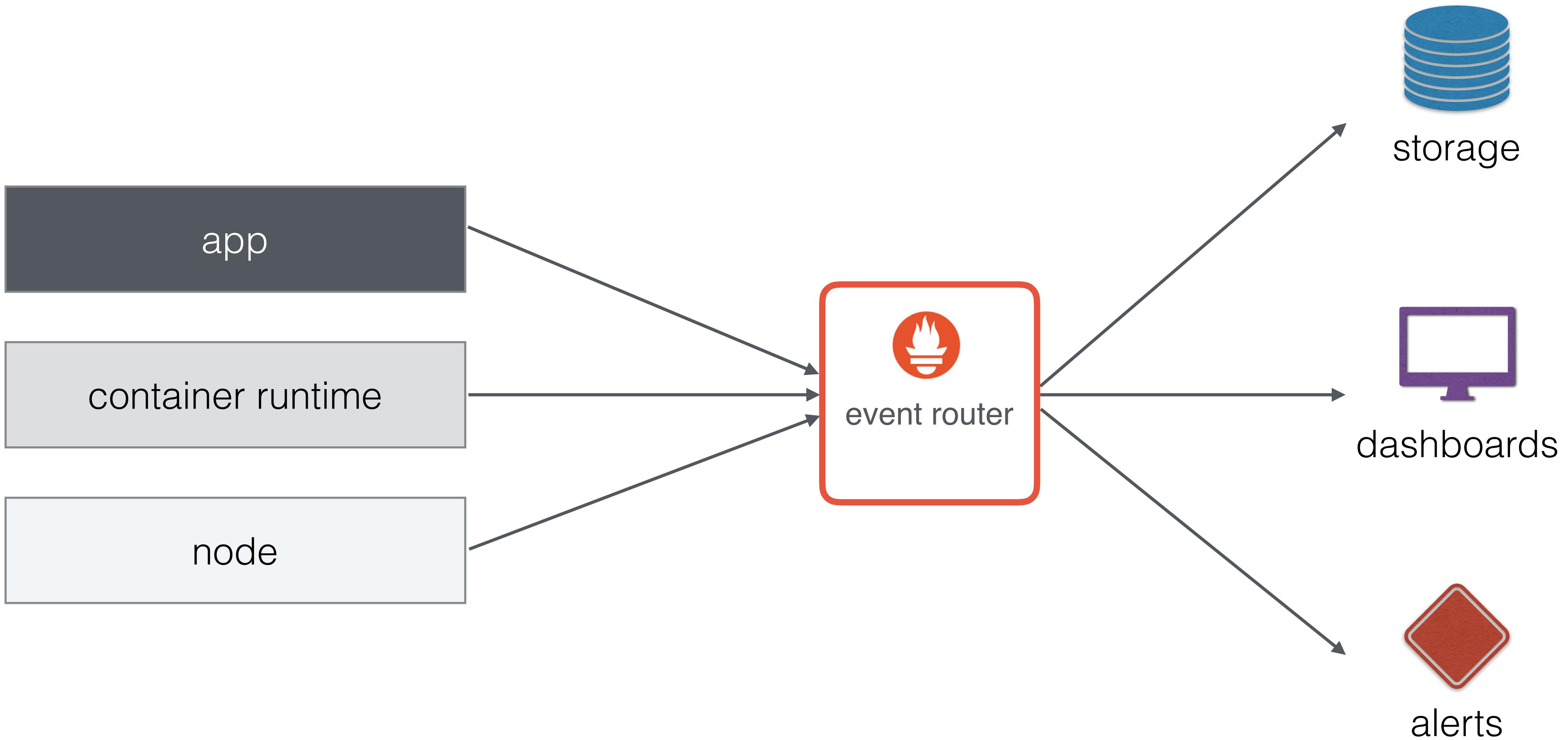


IN CS, IT CAN BE HARD TO EXPLAIN  
THE DIFFERENCE BETWEEN THE EASY  
AND THE VIRTUALLY IMPOSSIBLE.

## Observability



# Metrics



# Metrics

- Out-of-the-box low-level metrics (CPU, memory)
- Application-specific metrics (full-blown instrumentation vs service mesh-based approaches)
- Options
  - Roll your own, use the industry standards **Prometheus + Grafana**
  - Cloud provider native

**OPENSHIFT ONLINE**

dok

Search Catalog Add to Project

Overview Applications Builds Resources Storage Monitoring Catalog

Name Filter by name List by Application

APPLICATION stock-con <http://dok-stock.k8space.io>

DEPLOYMENT CONFIG stock-con, #2

CONTAINERS stock-con Image: mhauseblas/stock-con 8b670e0 22.5 MiB

Average Usage Last 15 Minutes

46 Mib Memory  
0 Cores CPU  
0.2 Kib/s Network

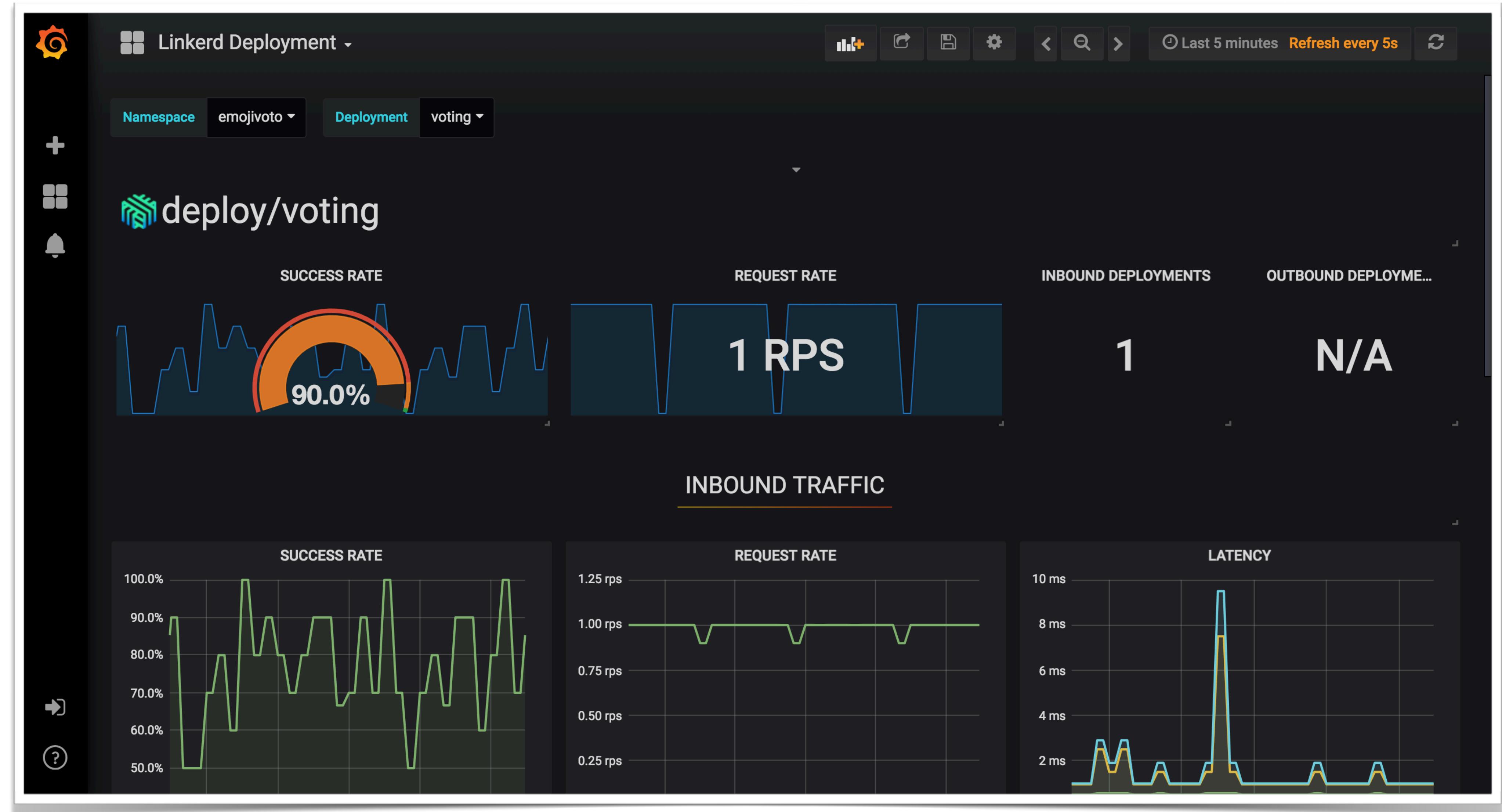
1 pod

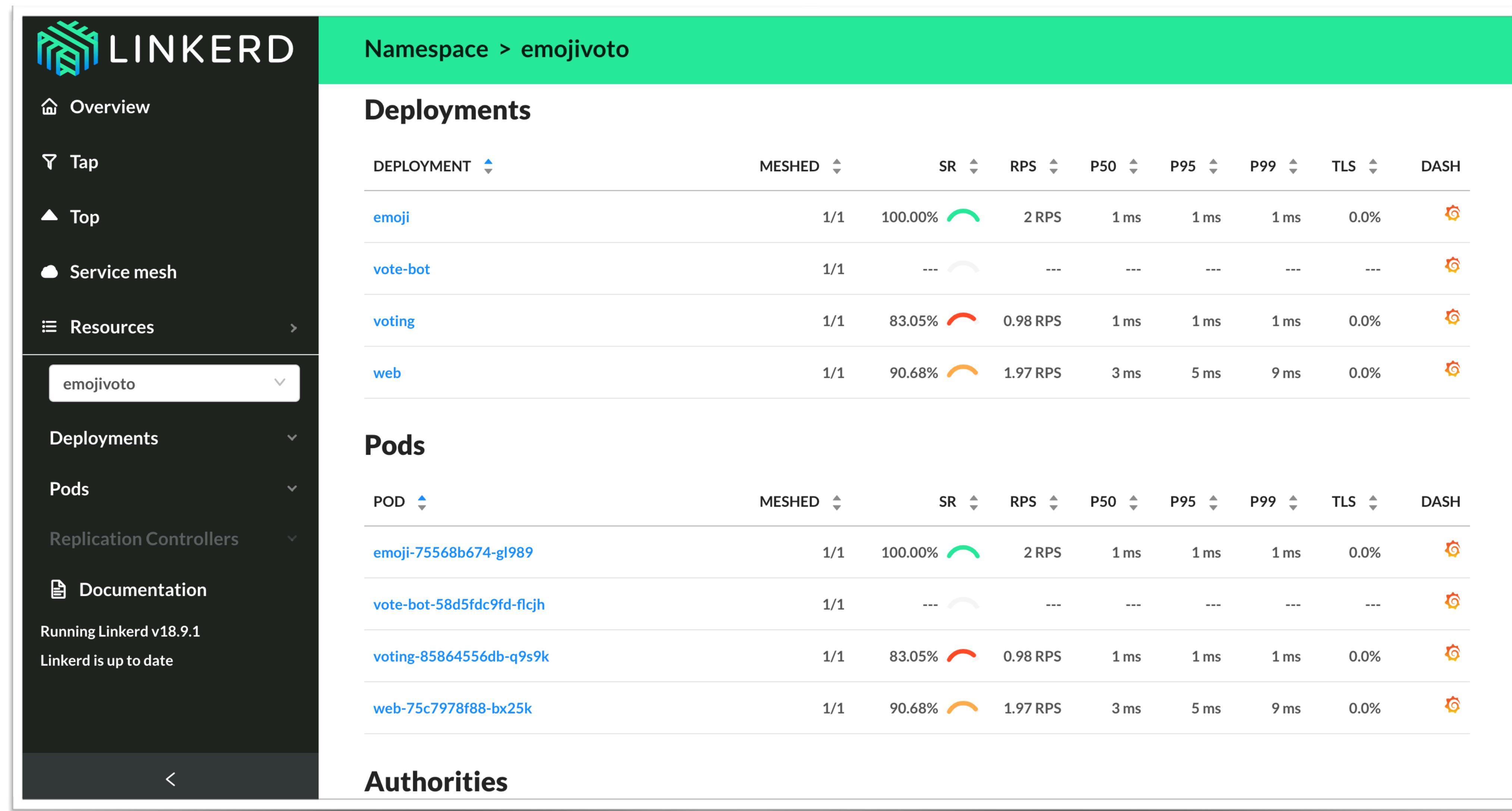
NETWORKING Service - Internal Traffic stock-con 80/TCP (80-9898) → 9898

Routes - External Traffic <http://dok-stock.k8space.io>

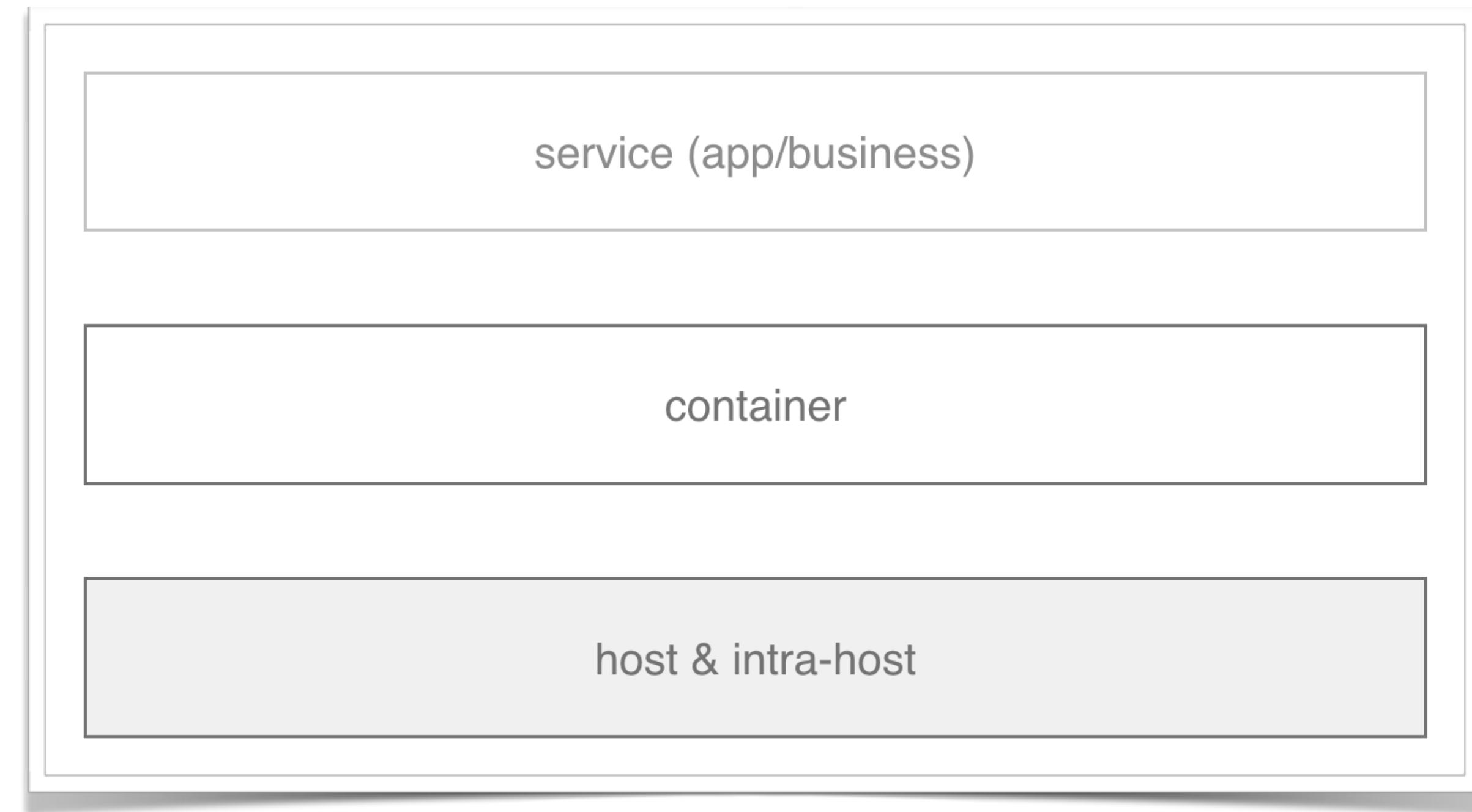
Route stock-con, target port 80-9898

This screenshot shows the OpenShift Online interface for the 'dok' project. The left sidebar contains navigation links for Overview, Applications, Builds, Resources, Storage, Monitoring, and Catalog. The main content area displays the 'stock-con' application. It includes a deployment config section for 'stock-con, #2', a container section showing 'stock-con' with an image of 'mhauseblas/stock-con 8b670e0 22.5 MiB', resource usage metrics (46 Mib Memory, 0 Cores CPU, 0.2 Kib/s Network), and a summary indicating 1 pod. Networking sections show internal traffic for 'stock-con' on port 80/TCP (80-9898) to 9898, and external traffic via a route at 'http://dok-stock.k8space.io'.



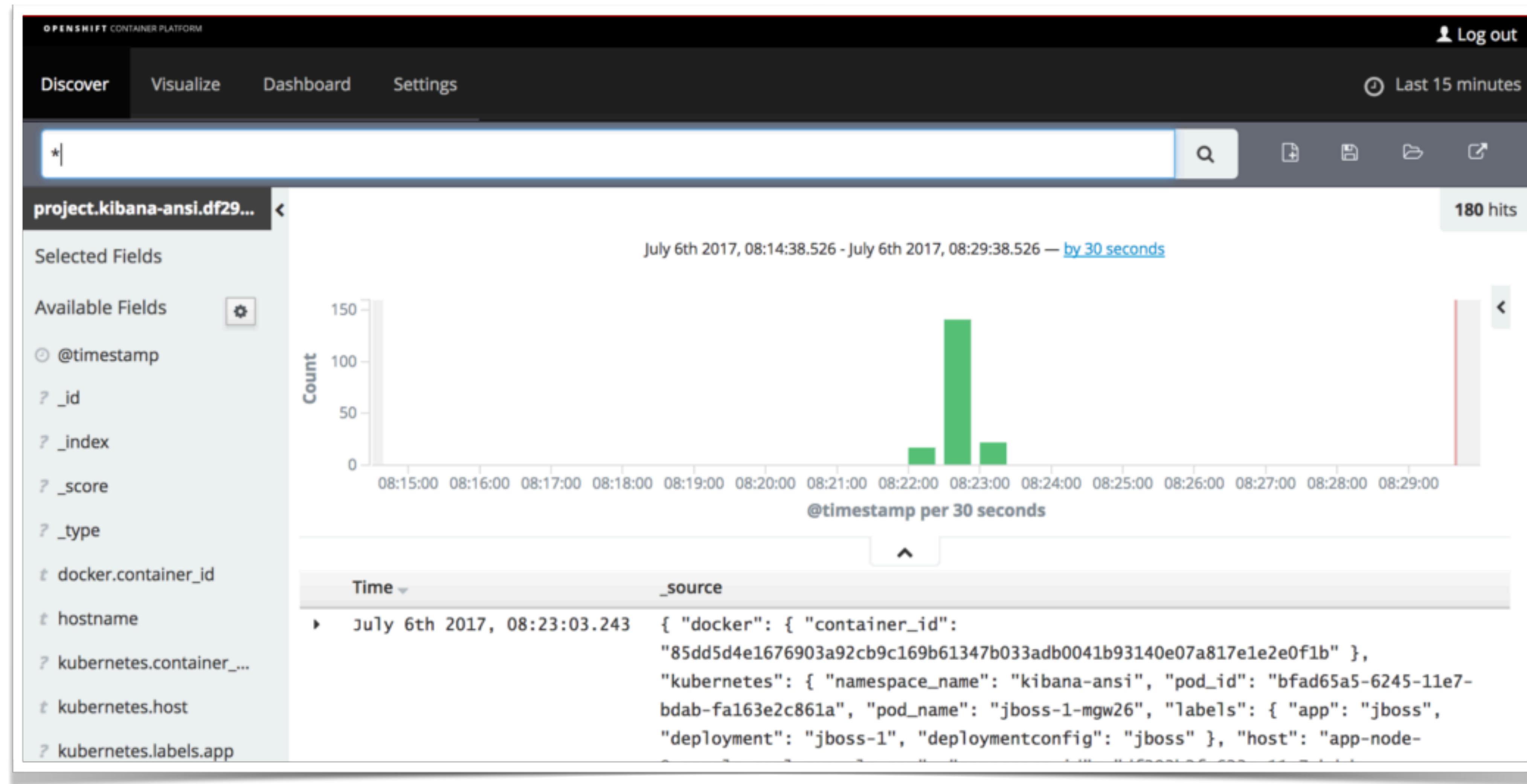


# Aggregated logs



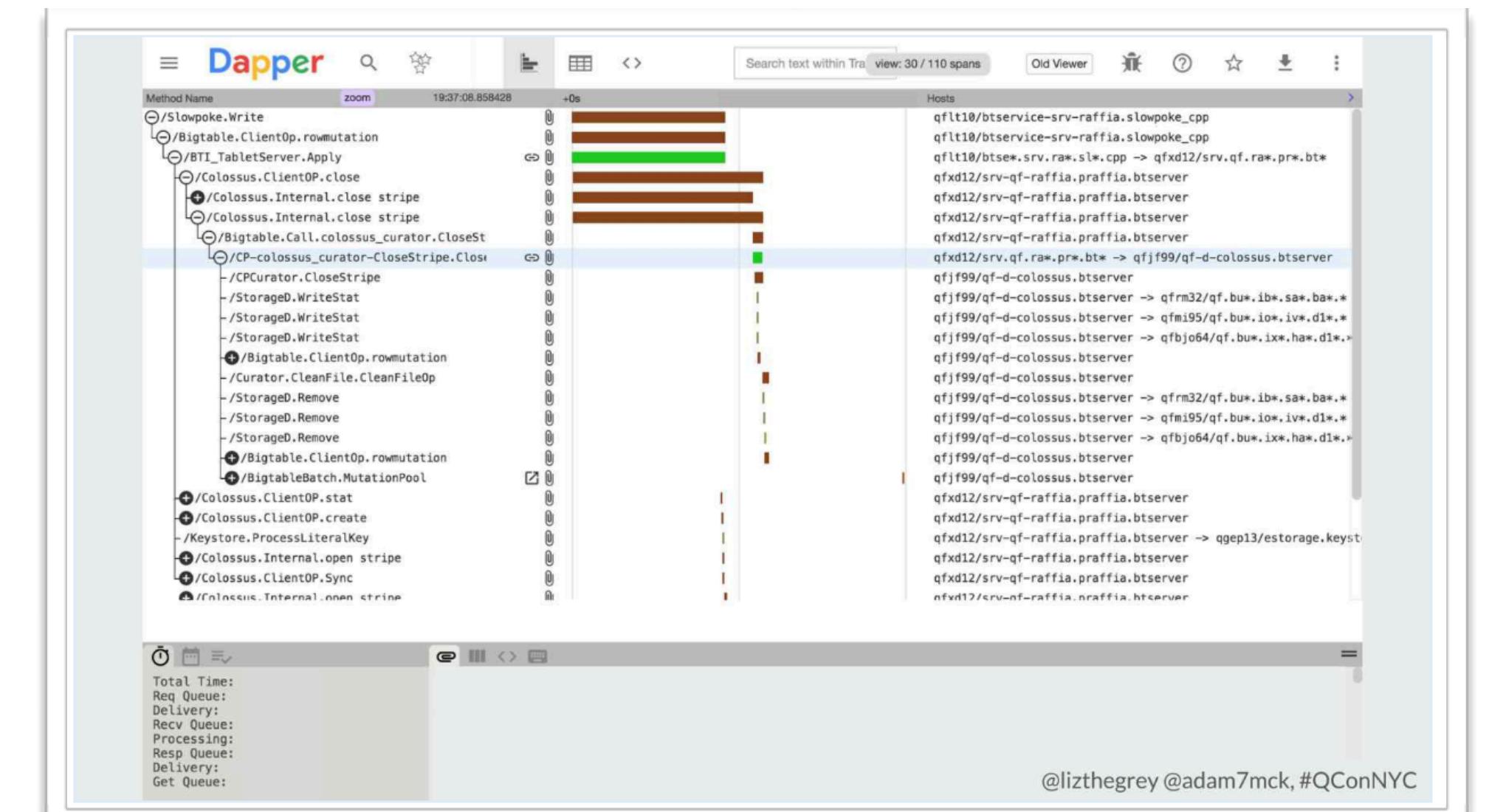
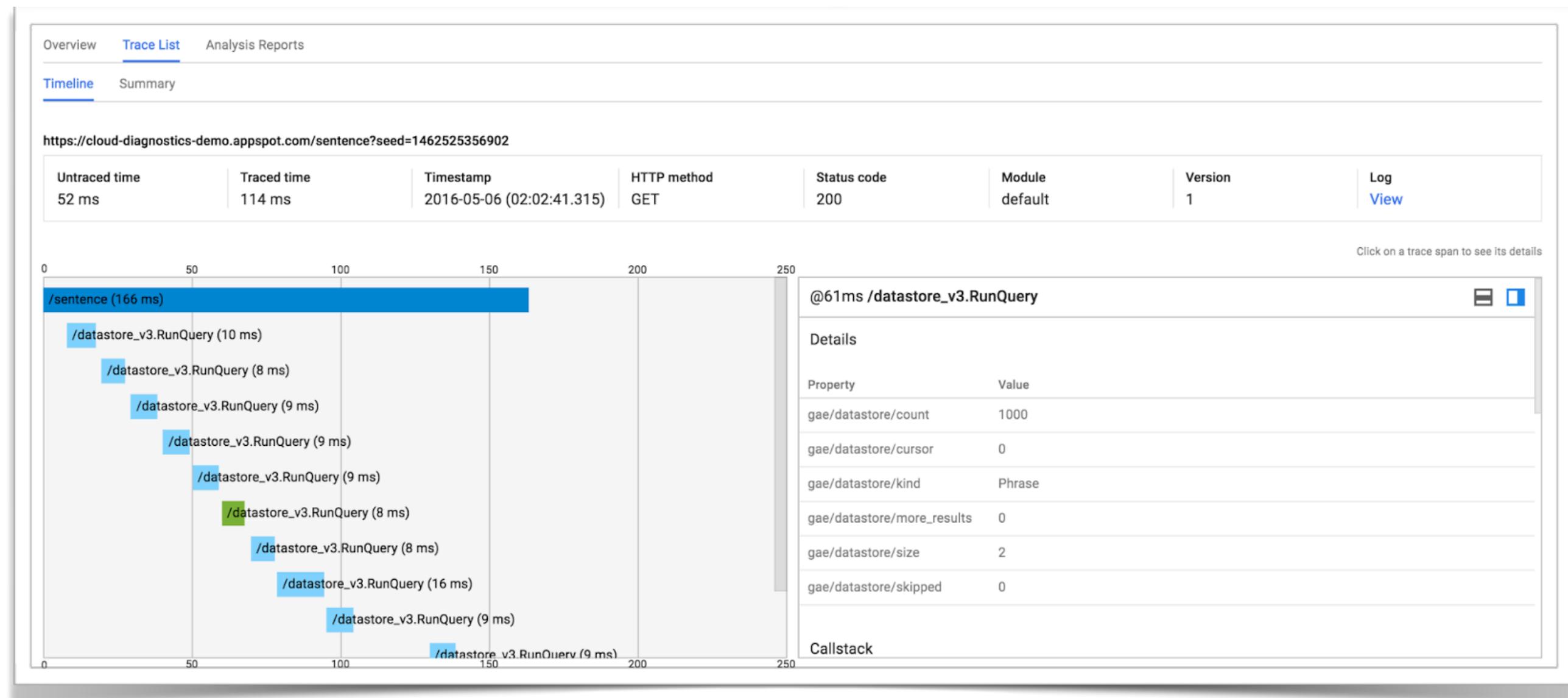
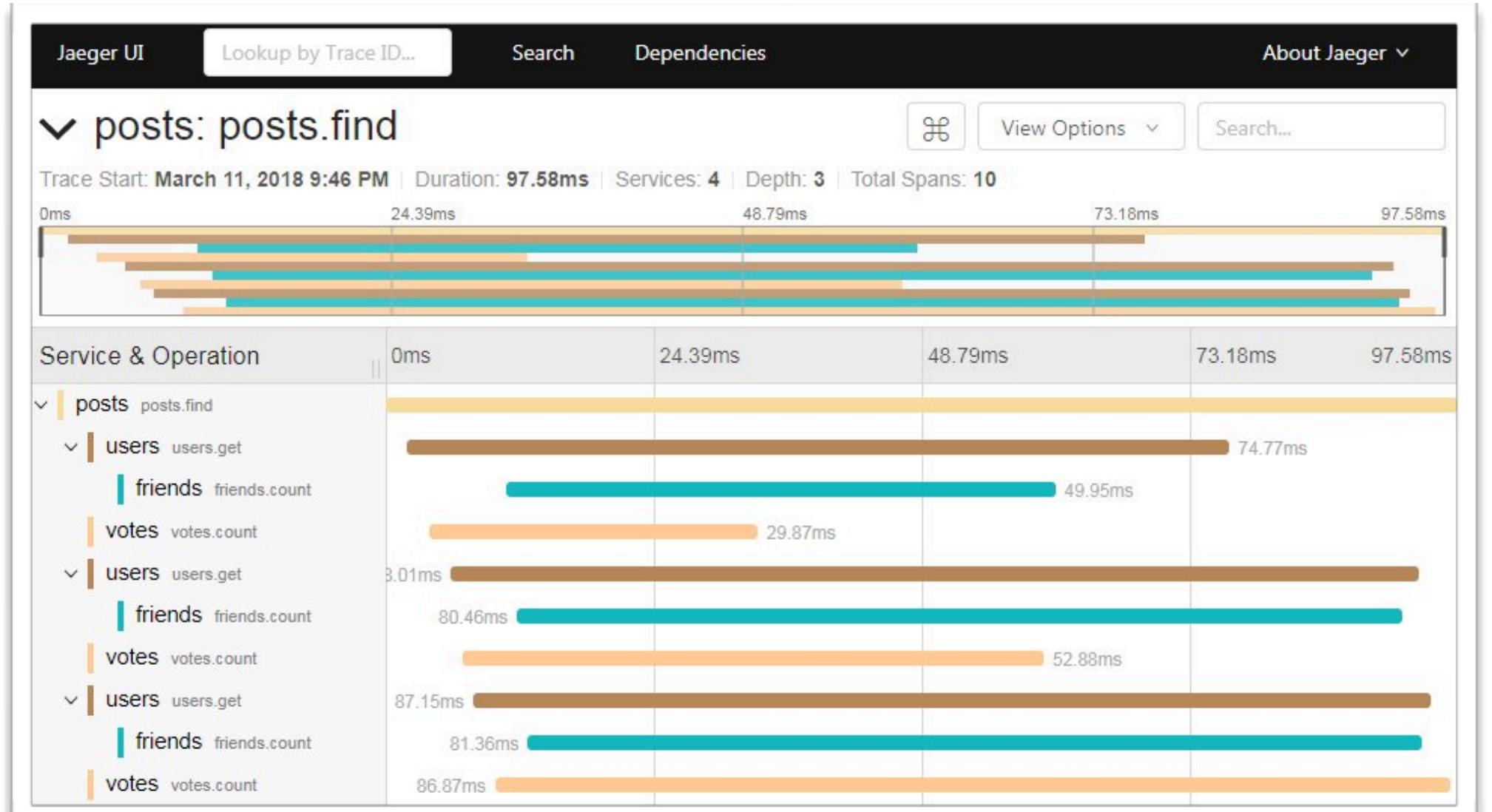
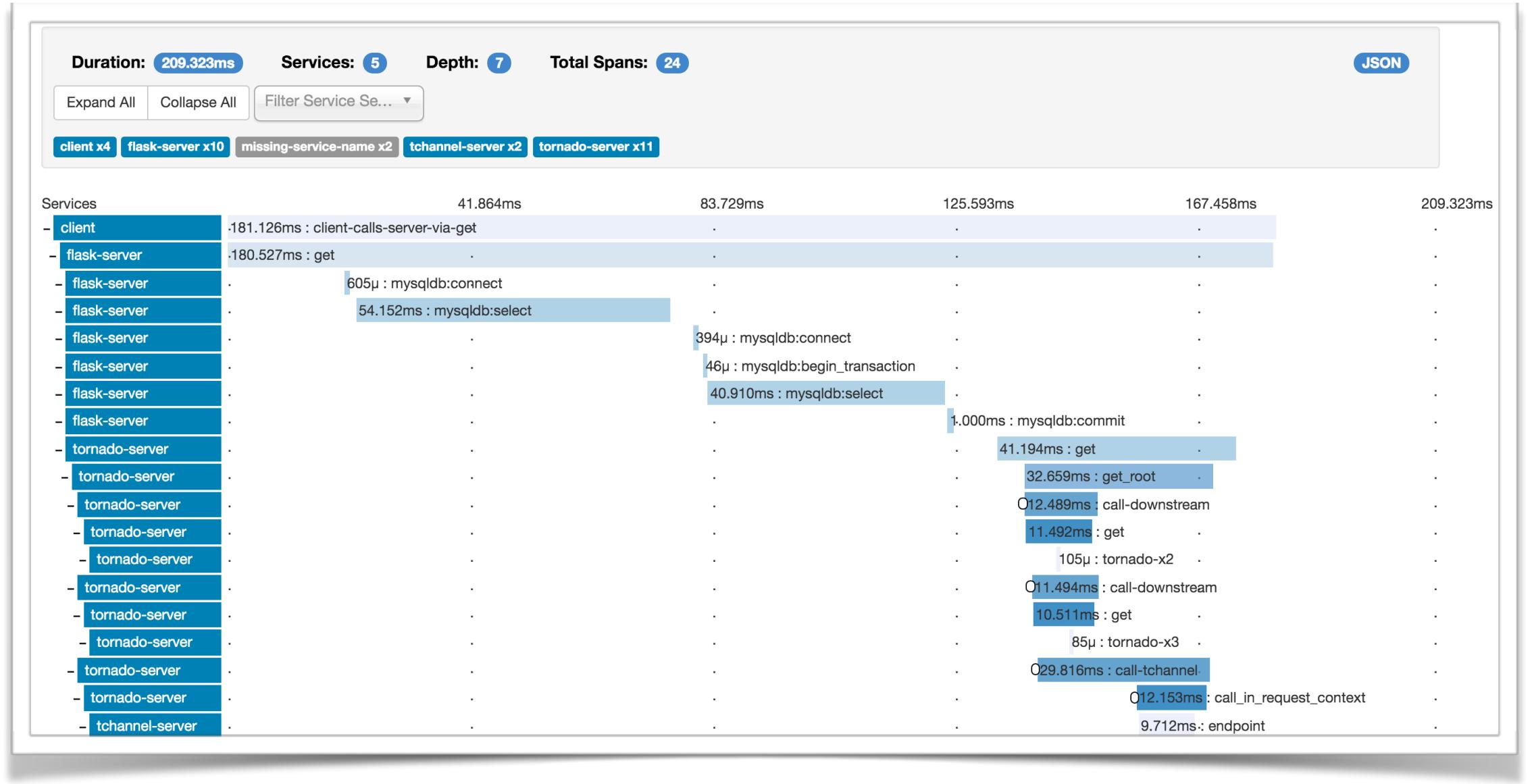
# Aggregated logs

- In app, log to `stdout` or if you can't use an adapter
- Options
  - Roll your own, use the industry standards: ELK/EFK stack
  - Cloud provider native such as CloudWatch or StackDriver



# Distributed tracing and debugging

- Roots: need to overcome limitations of “time-synced logs”
- Specifications: [OpenCensus](#) and [OpenTracing](#)
- Tooling: [Zipkin](#), [Jaeger](#), [Stackdriver](#)
- A must-have in a microservices setup
- Debugging: use [KubeSquash](#)



# Apps run in pods, and pods sometimes fail ...

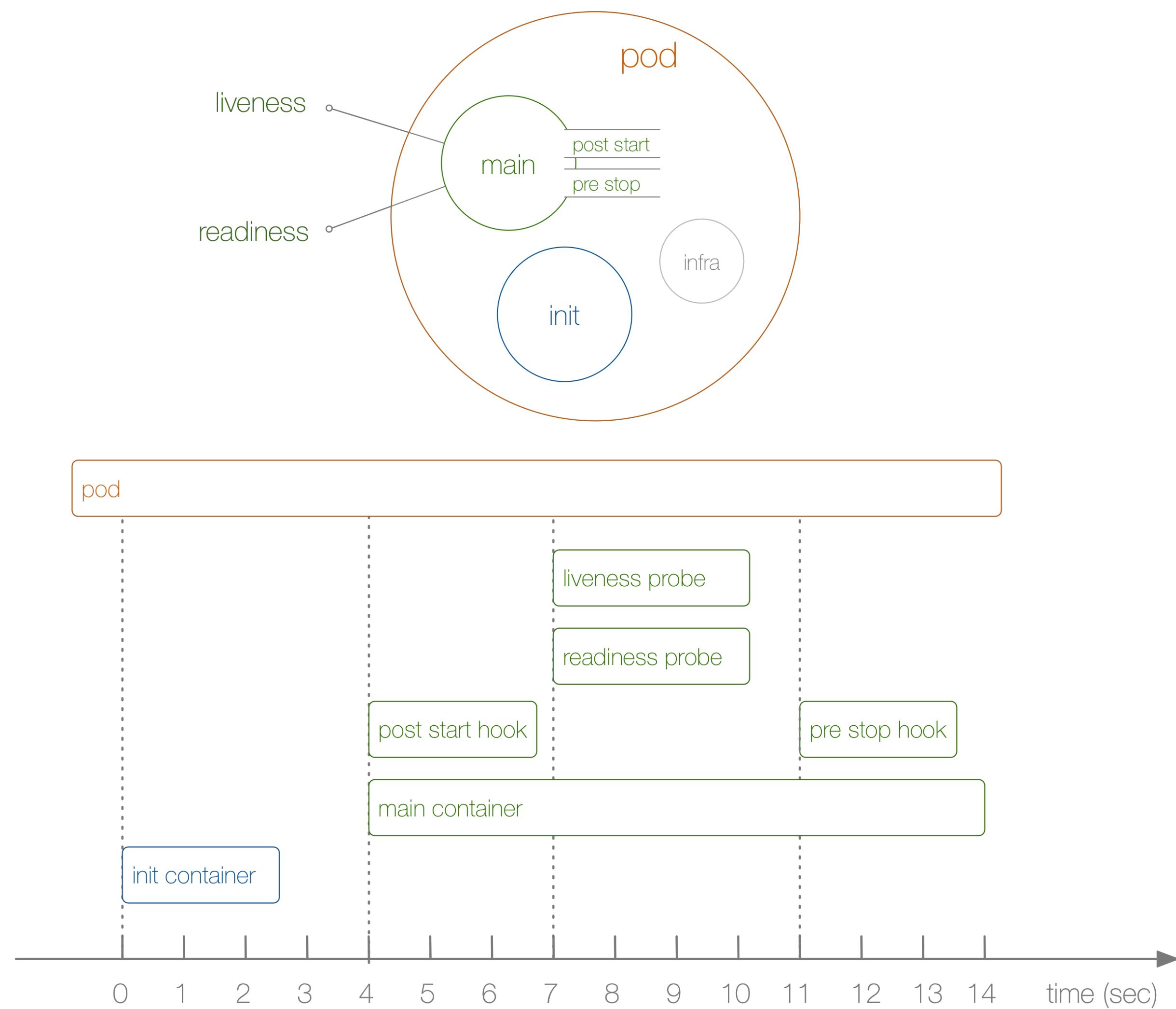


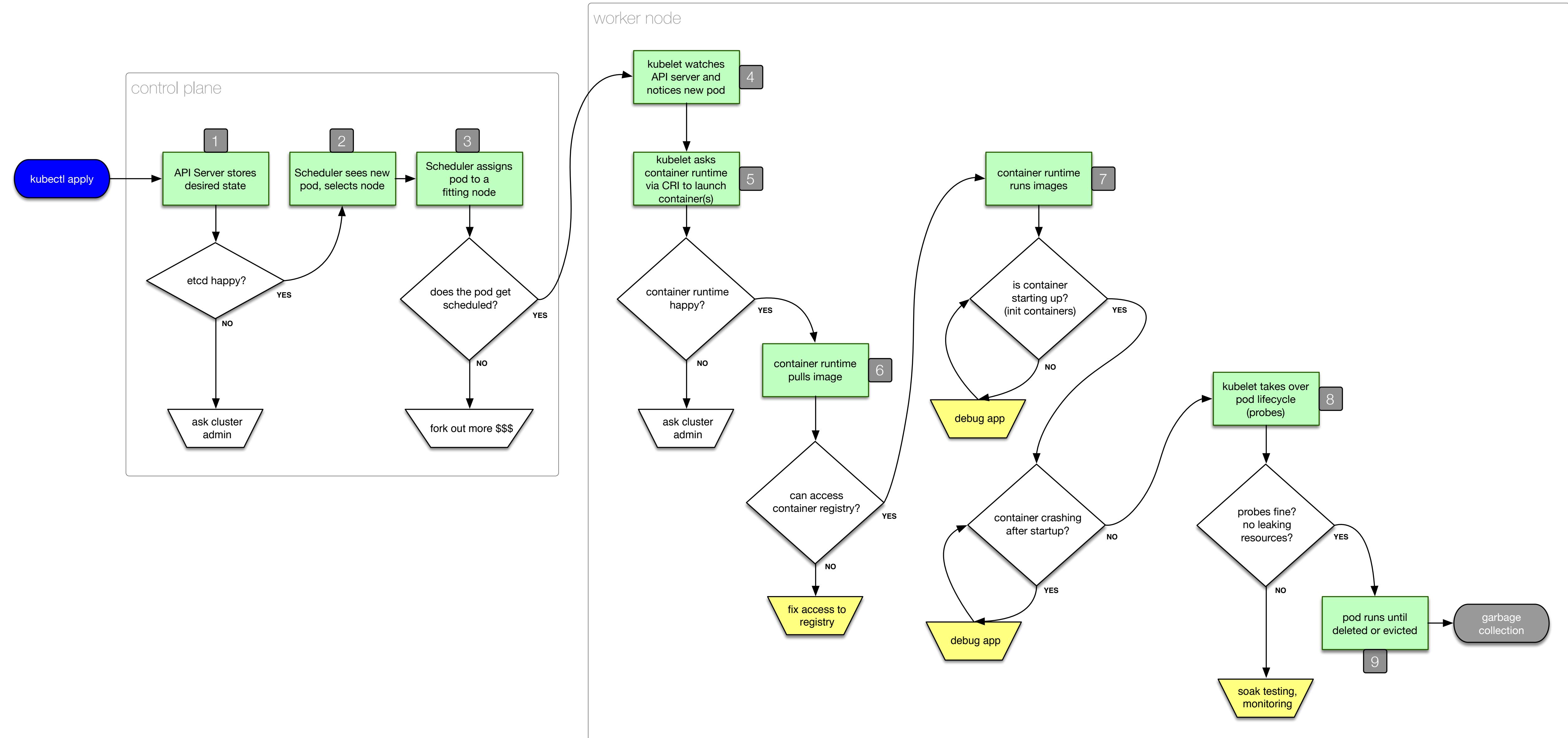
"Begin at the beginning," the King said, very gravely, "and go on till you come to the end: then stop."

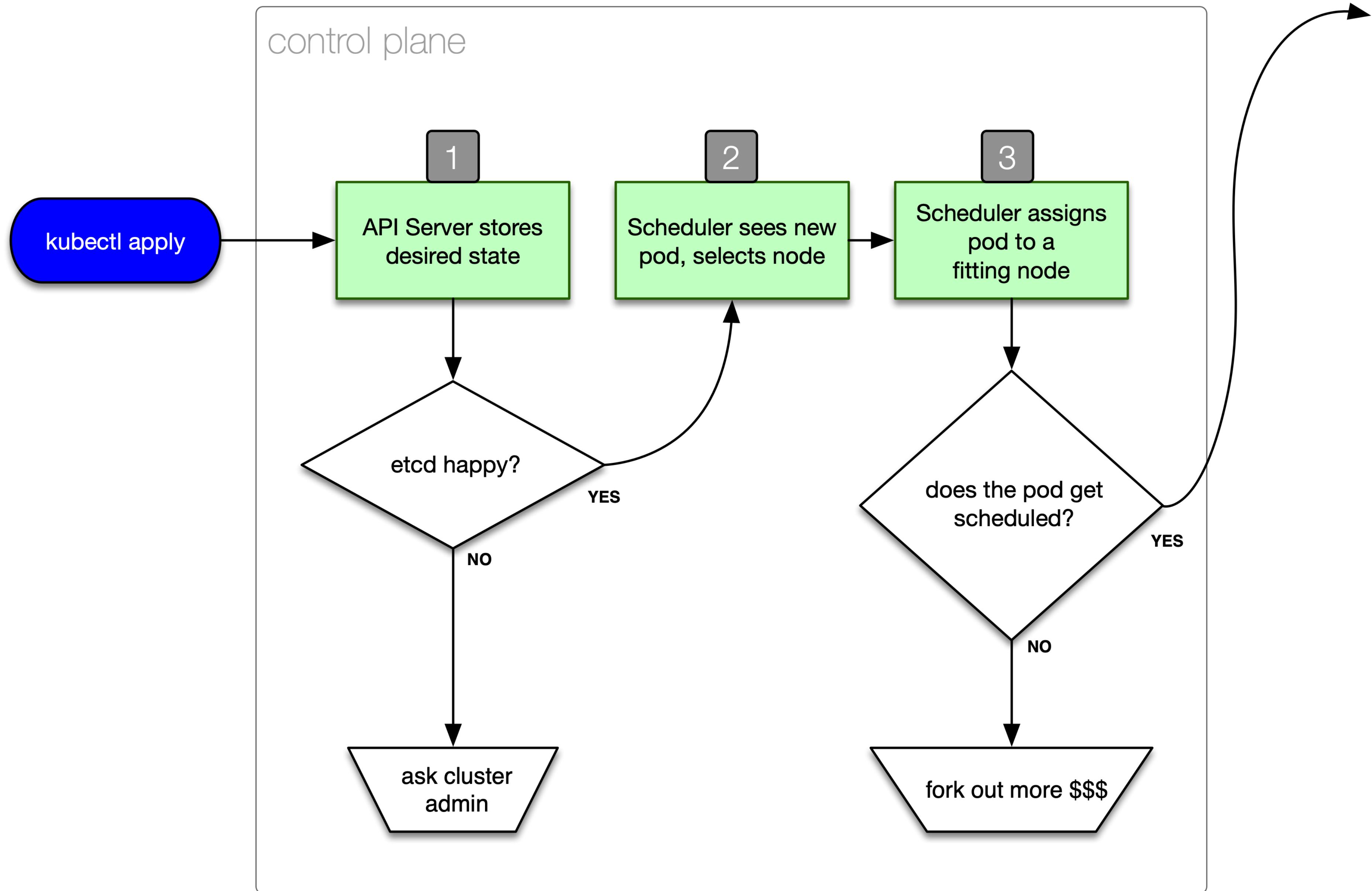
— Lewis Carroll, *Alice in Wonderland*

tags: humor

# What's (in) a pod?







## Kubernetes : error validating data: found invalid field env for v1.PodSpec;

I am using below yaml file to create the pod, kubectl command giving below error.

2 How to correct this error message?

```
apiVersion: v1
kind: Pod
metadata:
  name: command-demo
  labels:
    purpose: demonstrate-command
spec:
  containers:
  - name: command-demo-container
    image: debian
    command: ["printenv"]
    args: ["HOSTNAME", "KUBERNETES_PORT"]
  env:
  - name: MESSAGE
    value: "hello world"
    command: ["/bin/echo"]
    args: ["$(MESSAGE)"]
```

```
kubectl create -f commands.yaml
error: error validating "commands.yaml": error validating data: found invalid field en
```

asked 1 year, 9 months ago

viewed 1,781 times

active 1 year, 9 months ago

### BLOG

[Stack Overflow & InfoJobs Opportunities to Developers](#)

### FEATURED ON META

[Take the 2019 Developer Survey!](#)

[Data science time! January views to answers](#)

### HOT META POSTS

[14 Please don't share my e-mail address with Amazon without my express permission](#)

kubectl apply

invalid YAML

etcd happy?

YES

NO

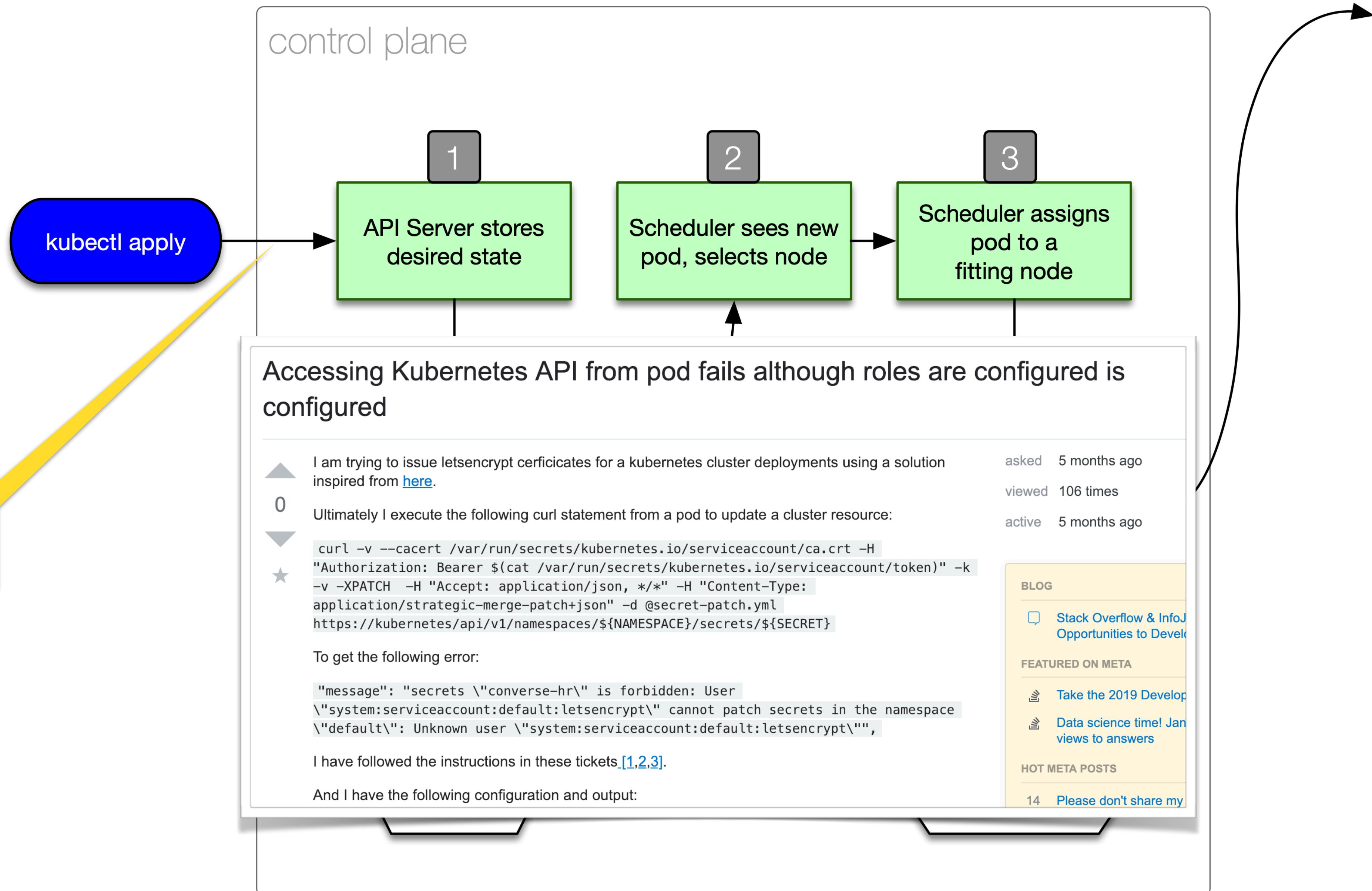
ask cluster admin

does the pod get scheduled?

YES

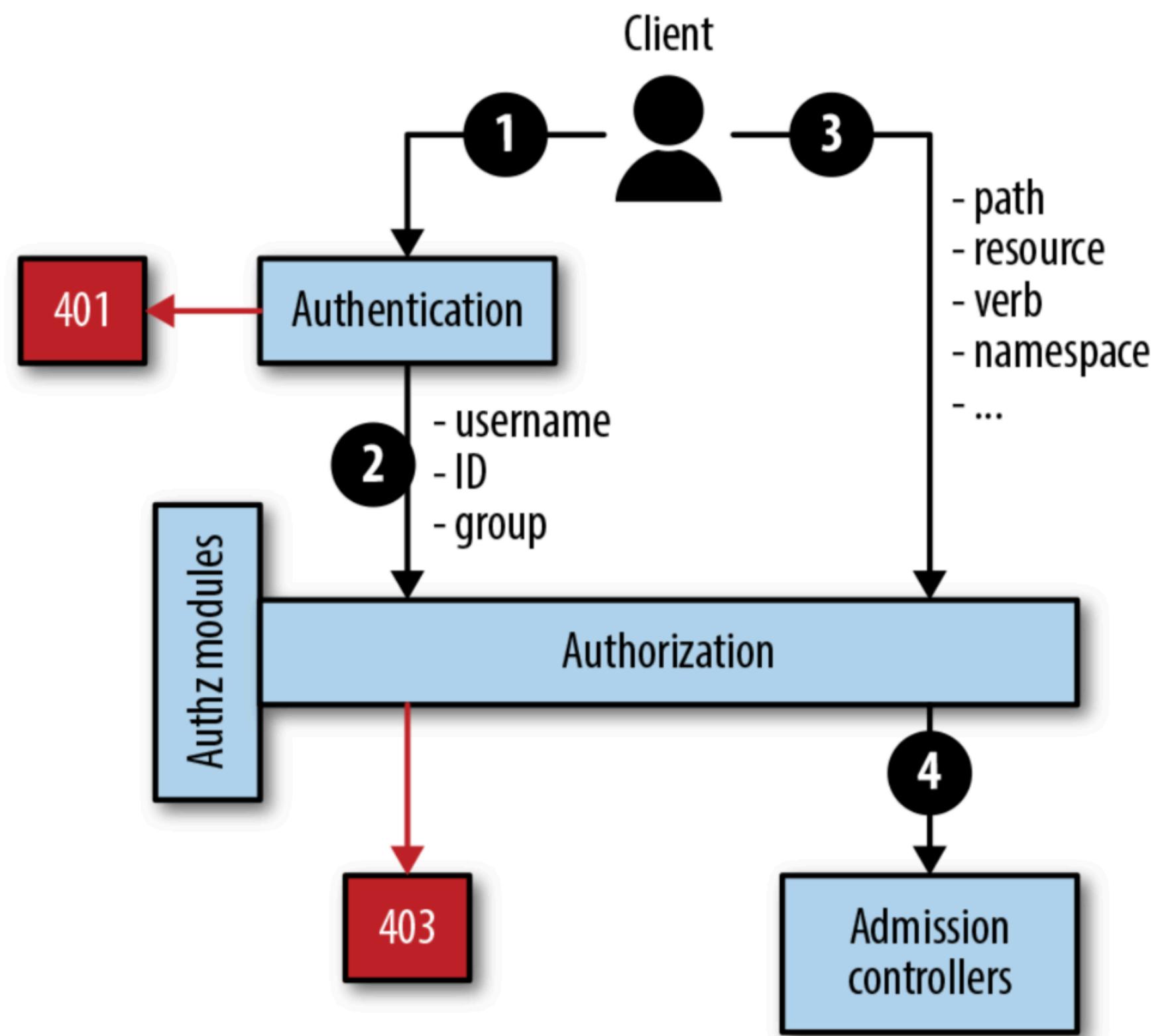
NO

fork out more \$\$\$



# Authentication & authorization

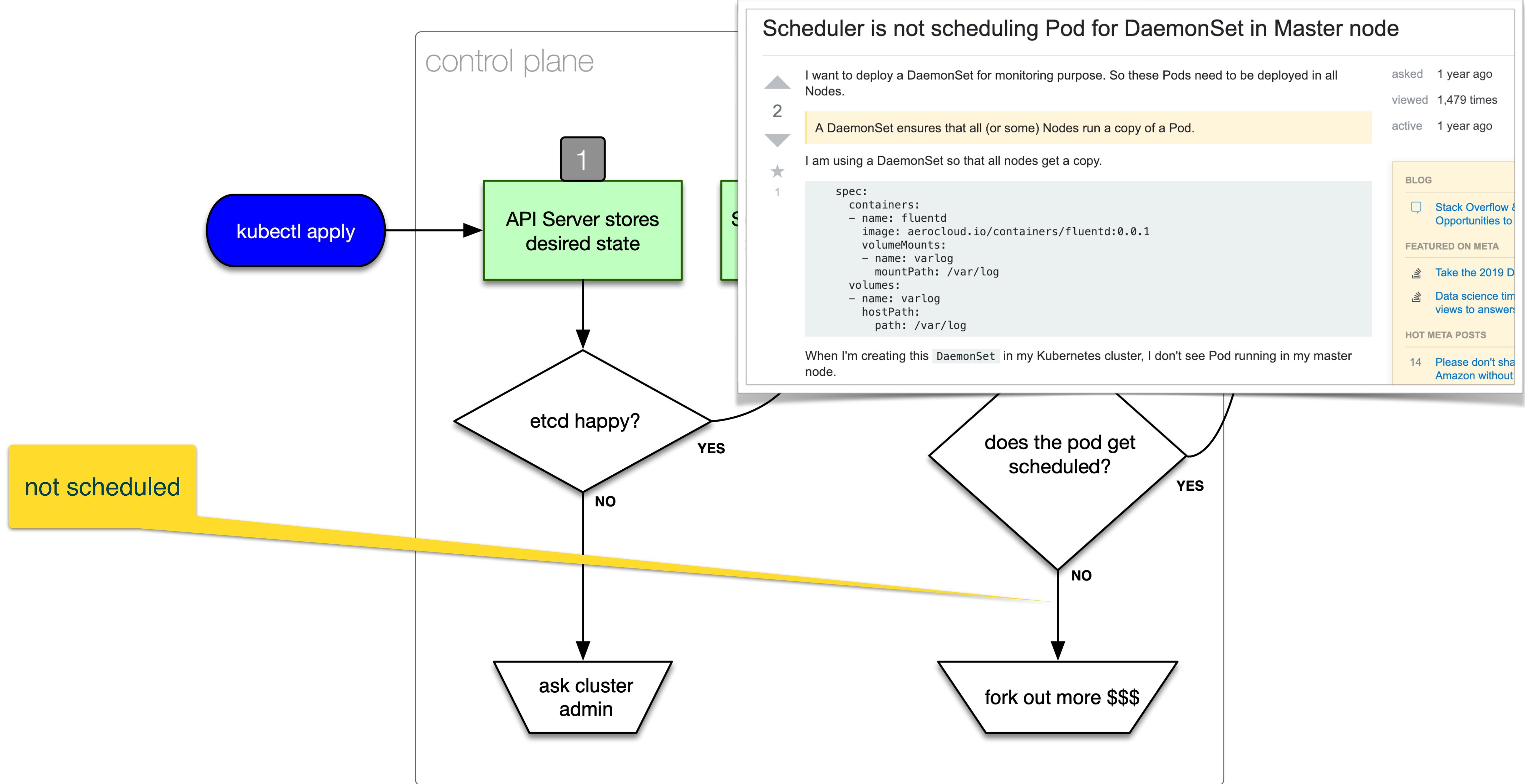
- static password/token file
- X509 client certs
- proxy+header
- OpenID Connect
- custom via Webhook

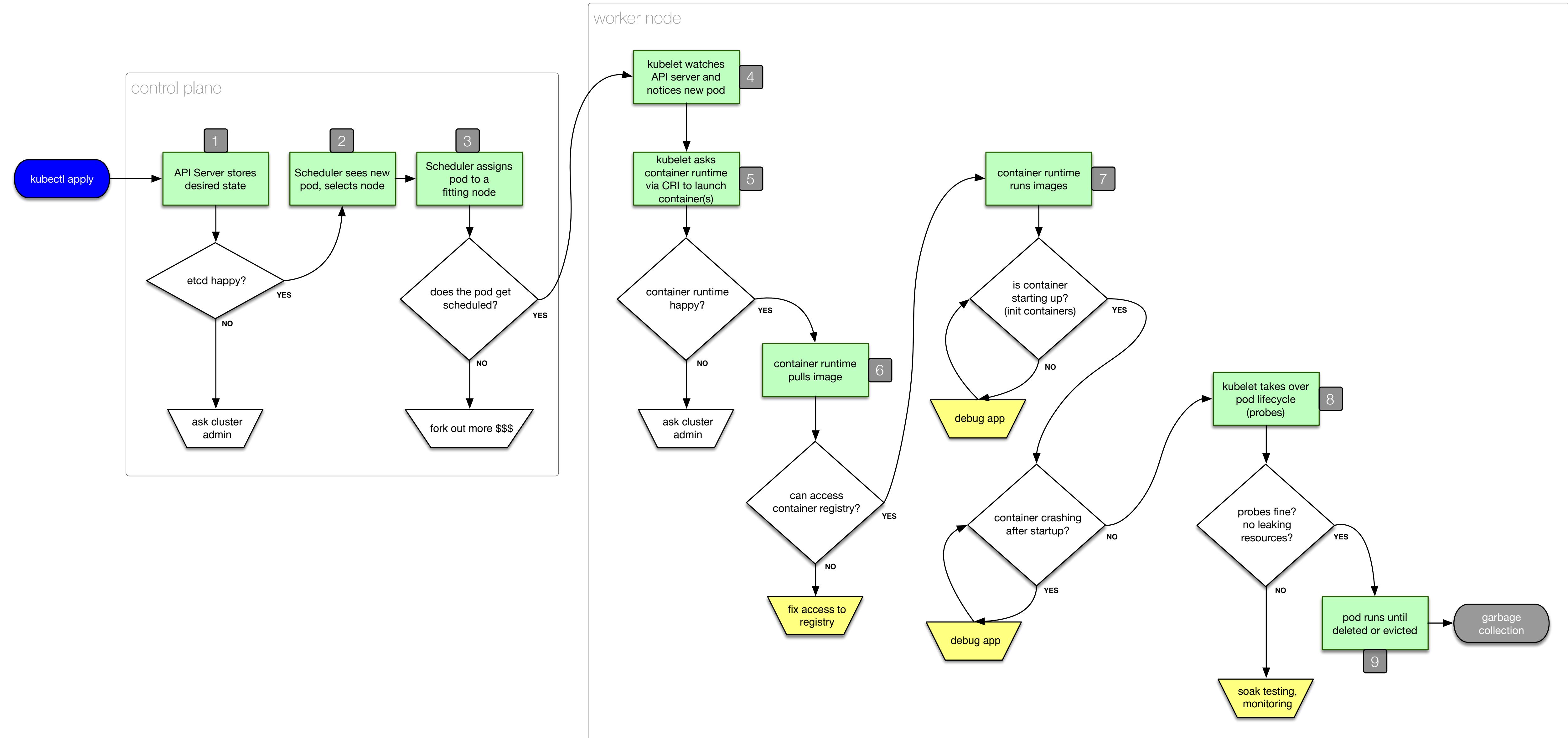


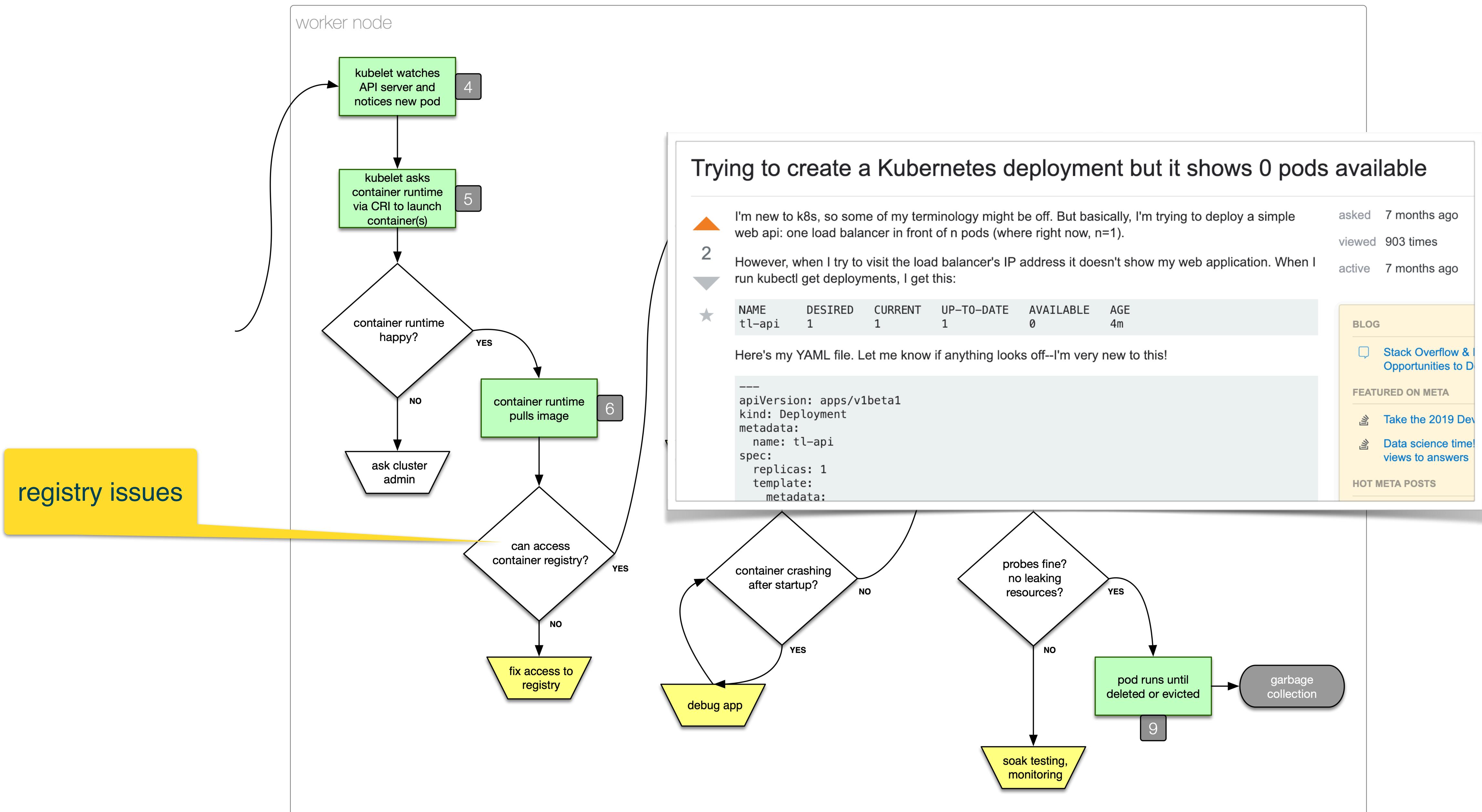
- Node (kubelet)
- ABAC (outdated)
- RBAC
- Webhook (external)

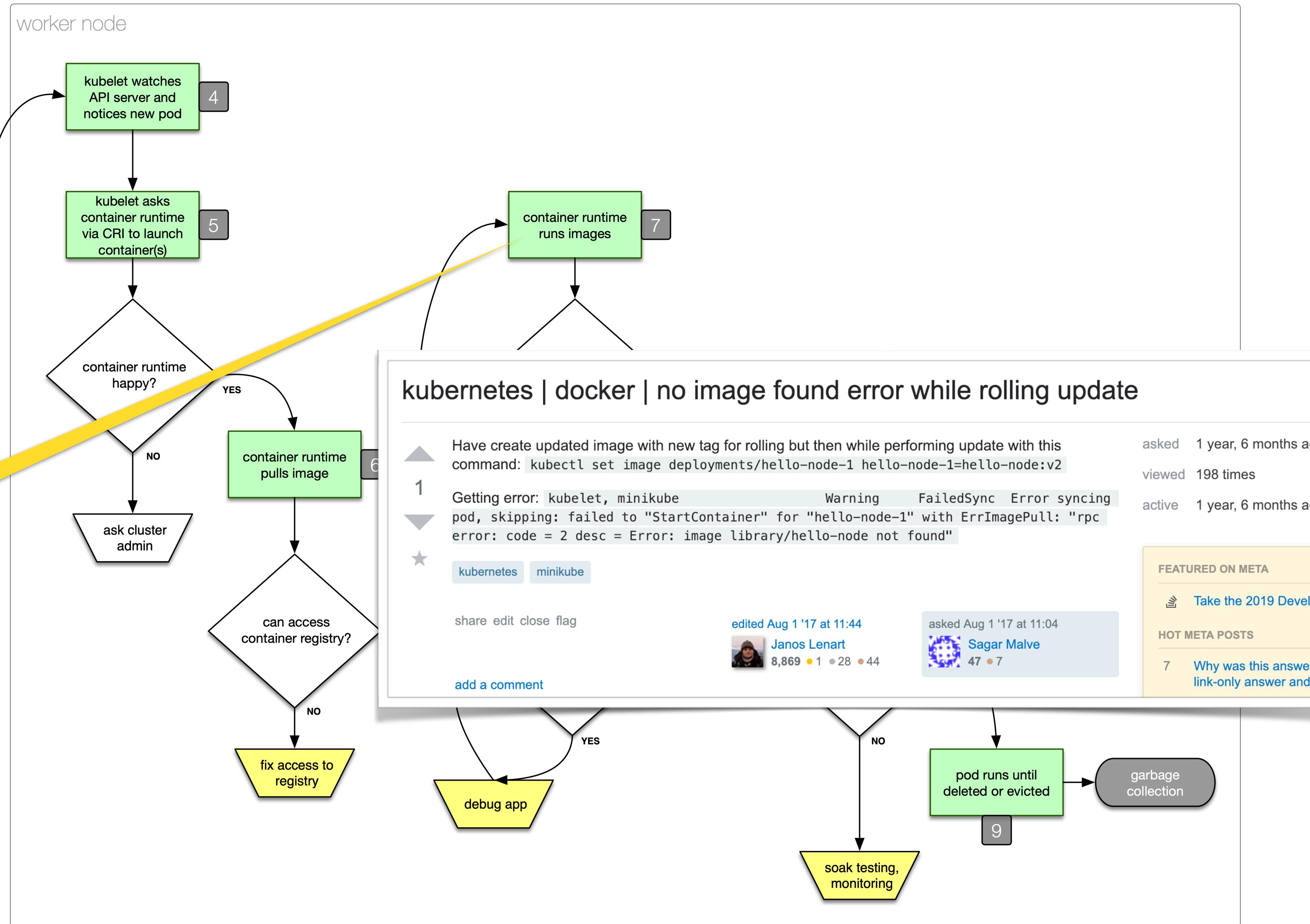
# Access control (RBAC) and policies

- Use `kubectl auth can-i` to check RBAC permissions
- Make yourself familiar with:
  - Pod Security Policies, might constrain your app too much
  - Network Policies, might be too strict for your app's communication needs
- See [kubernetes-security.info](https://kubernetes-security.info)





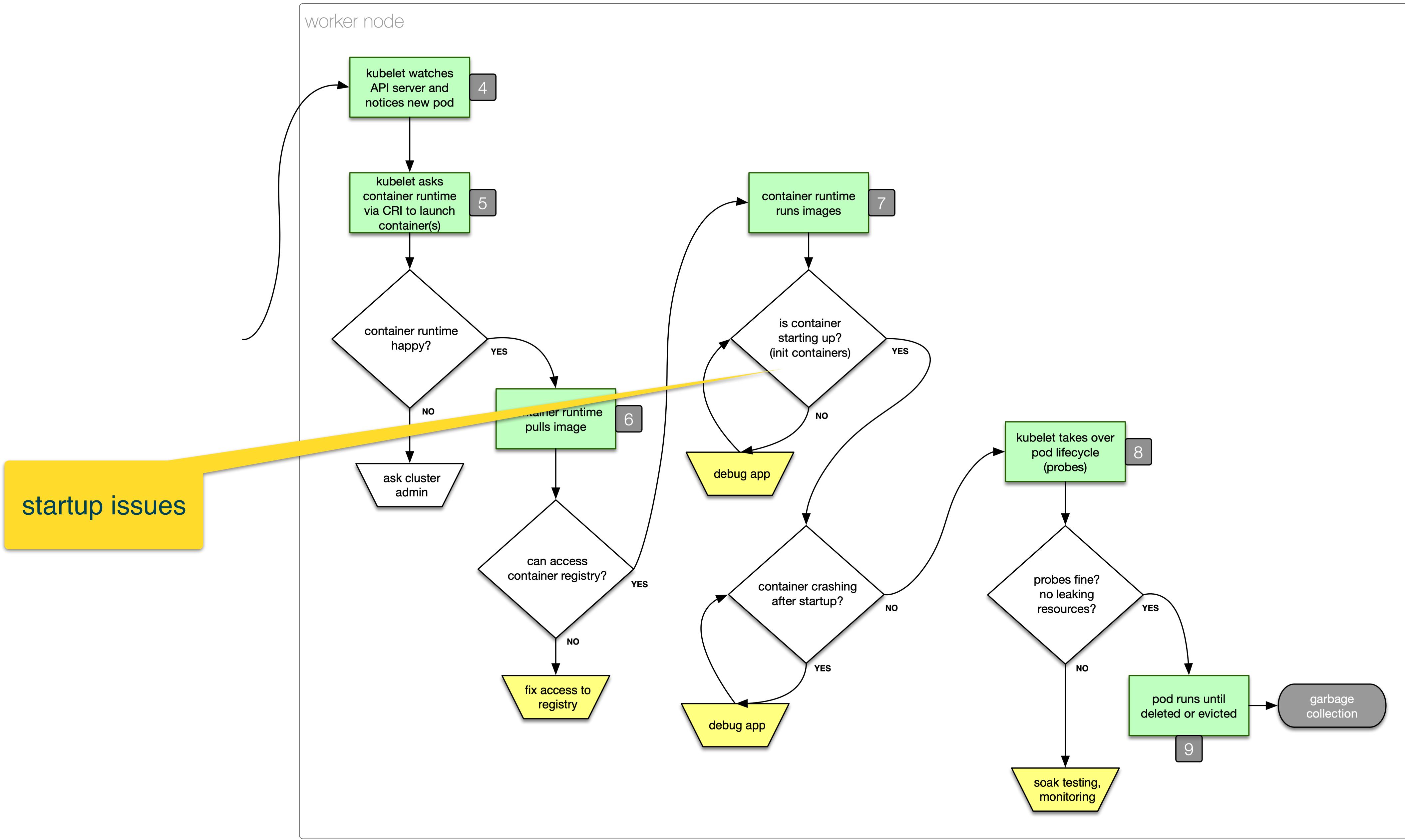


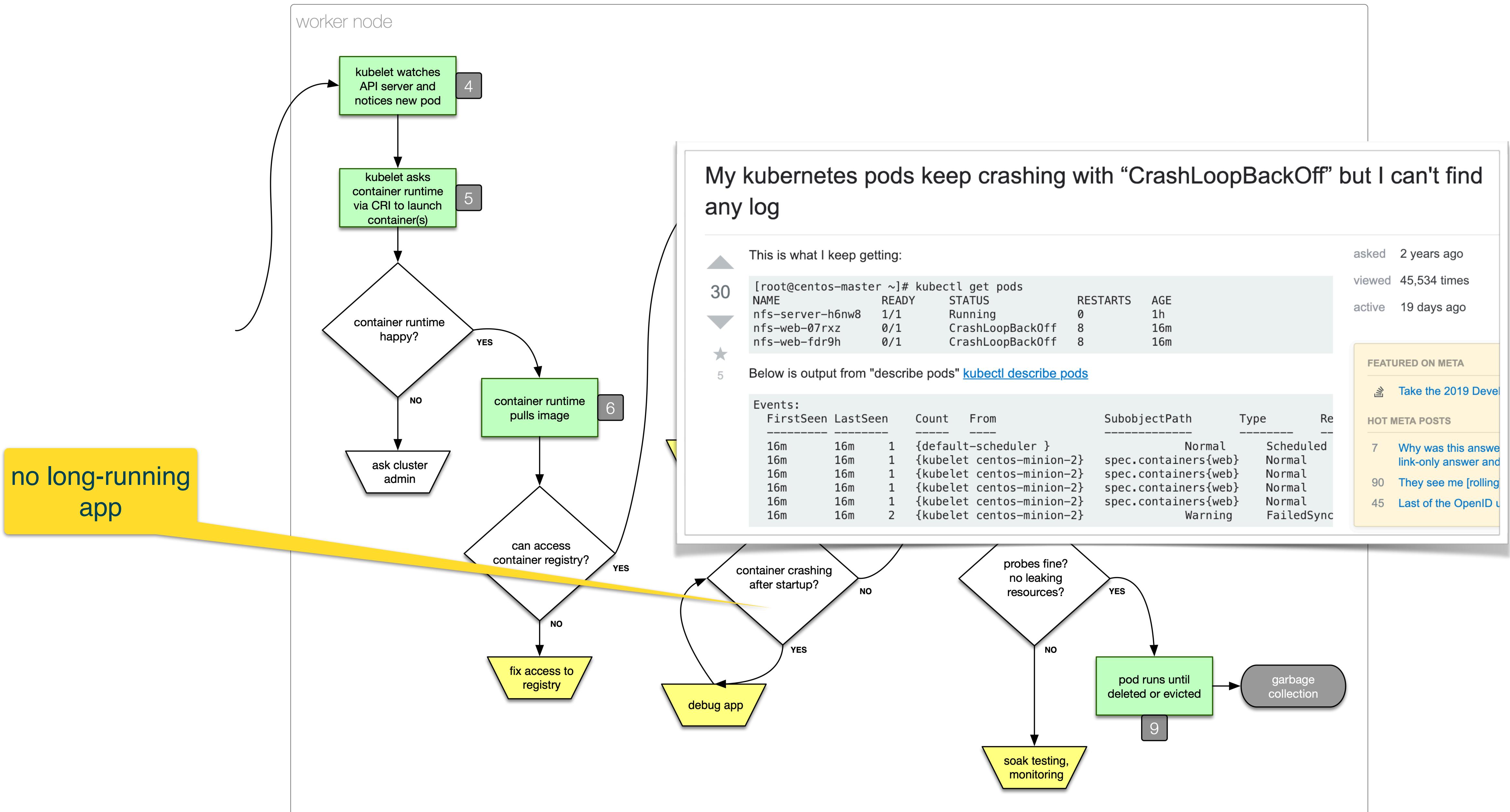


I think I'm having image issues ...



kubectl get events to the rescue?





# Dunno, just keeps crashing ...



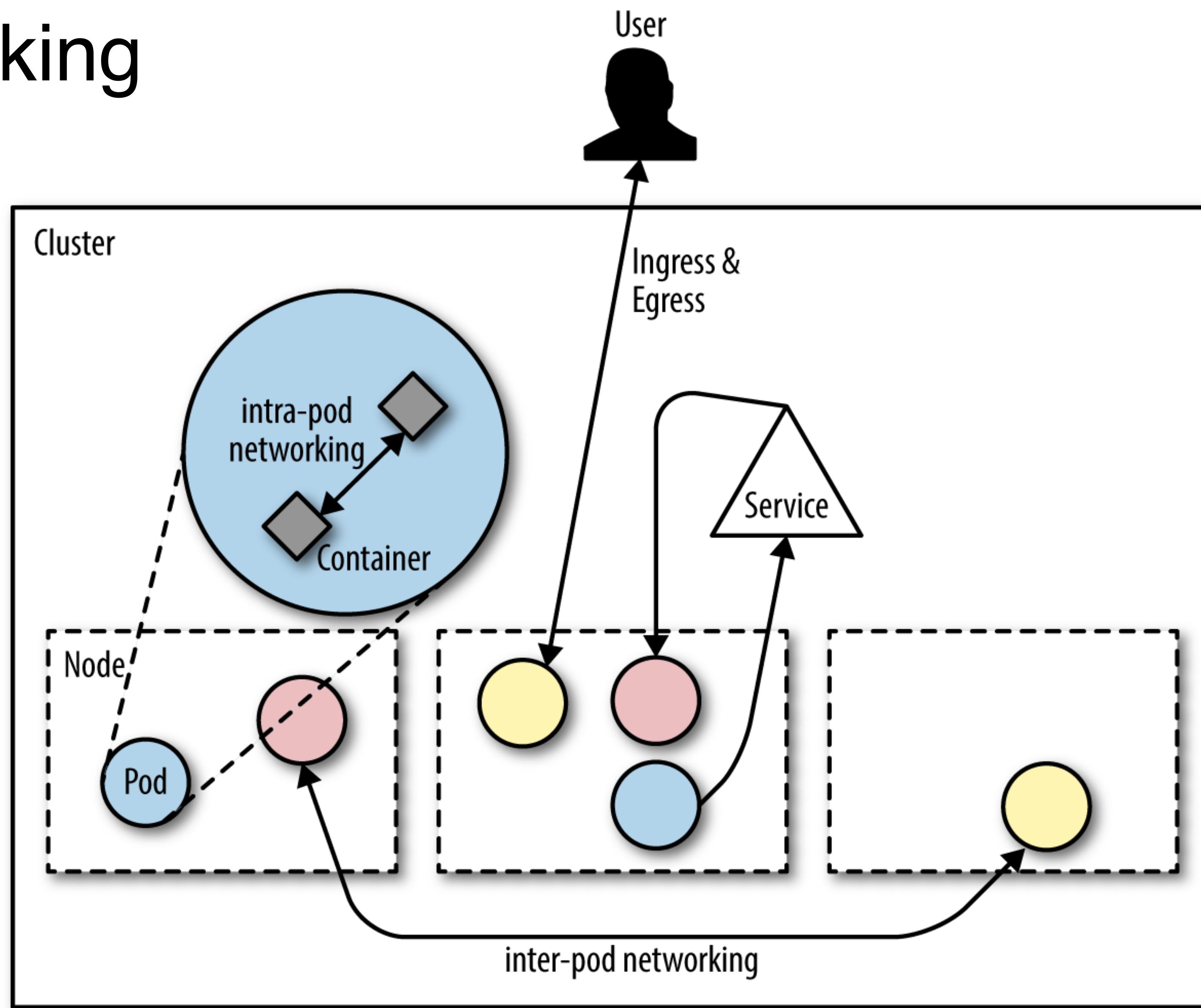
kubectl describe and exec

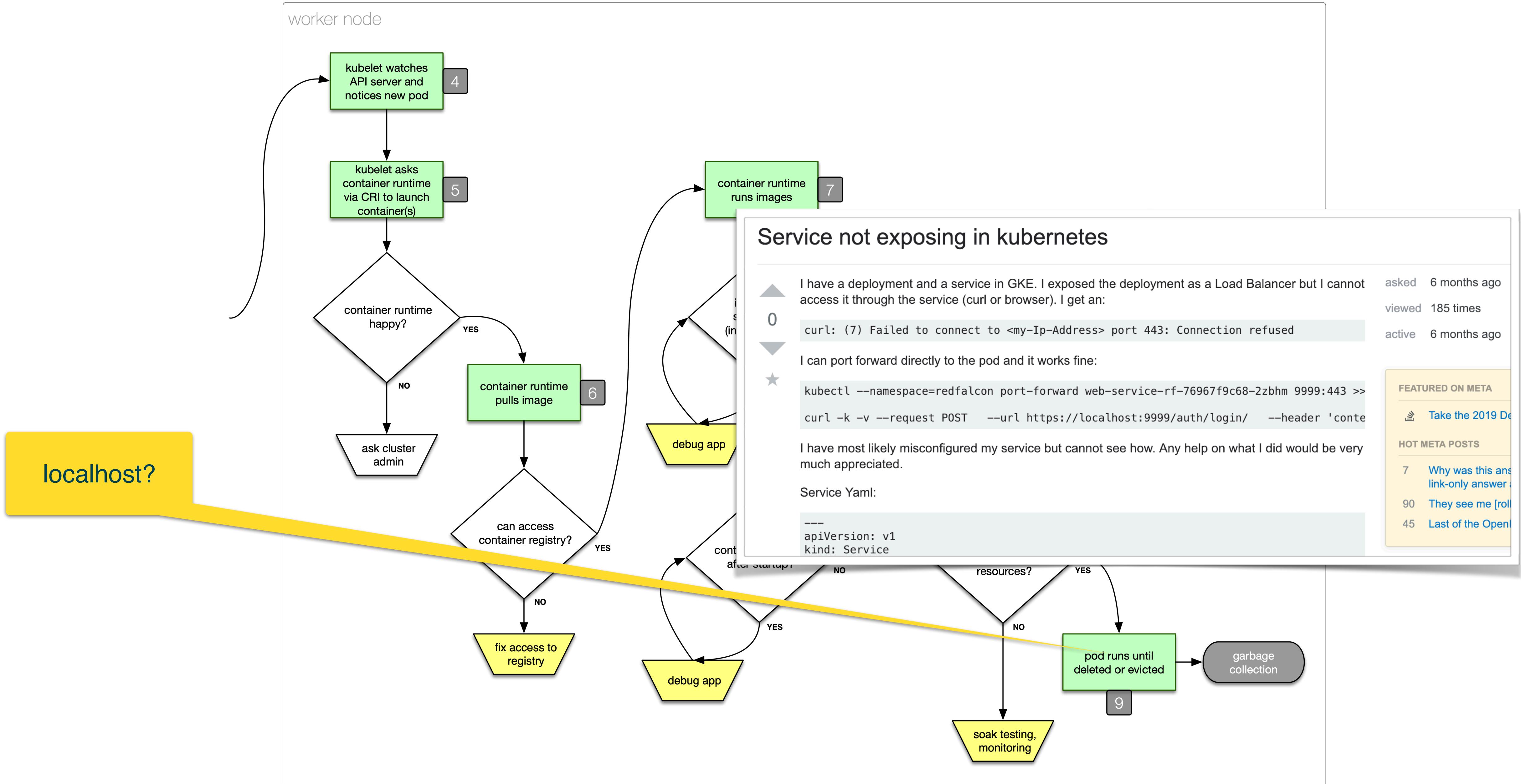
Oh my Lanta, something's wrong with the app ...



kubectl logs

# Networking

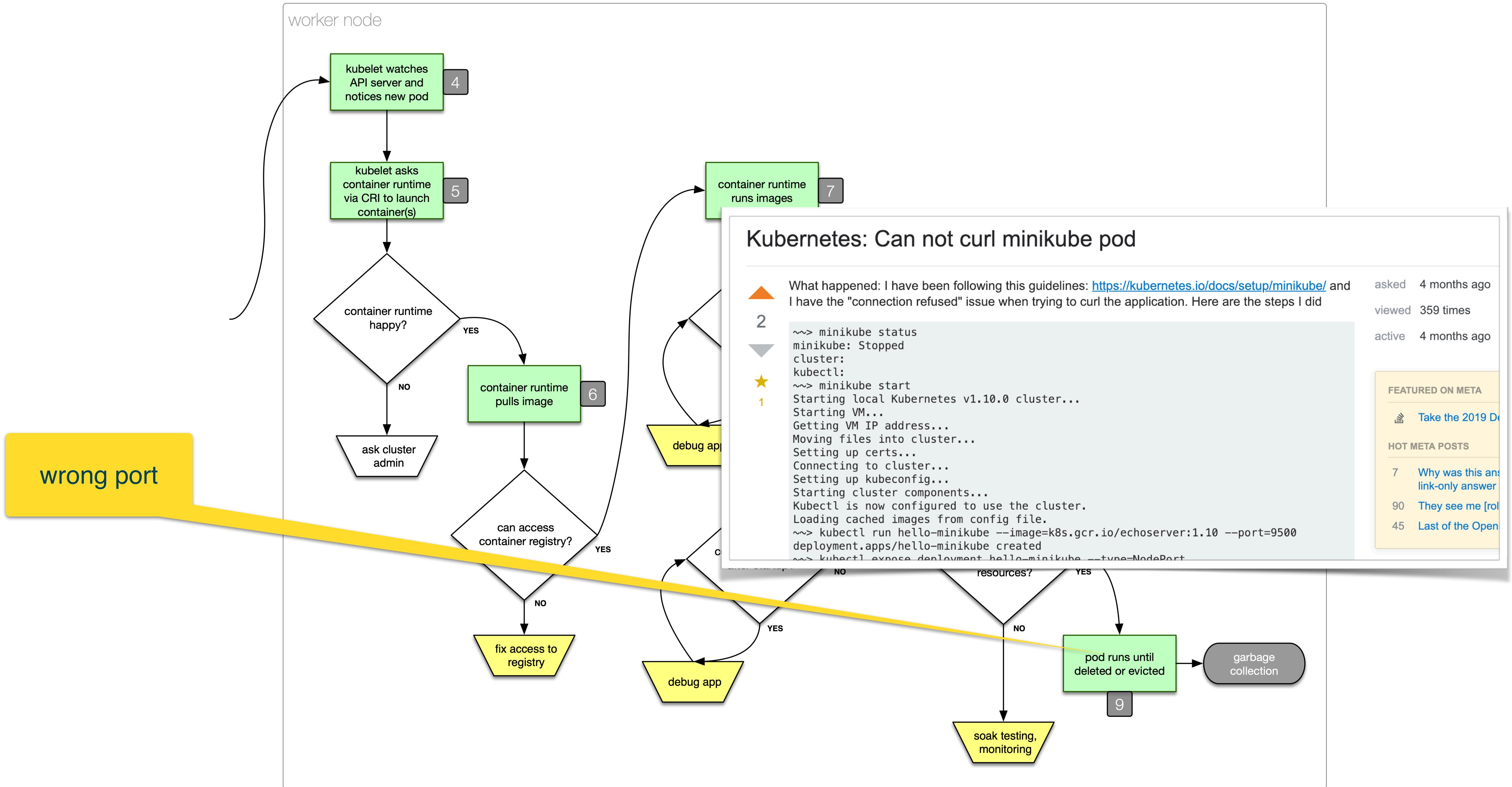




Hit me up on Twitter: @mhausenblas

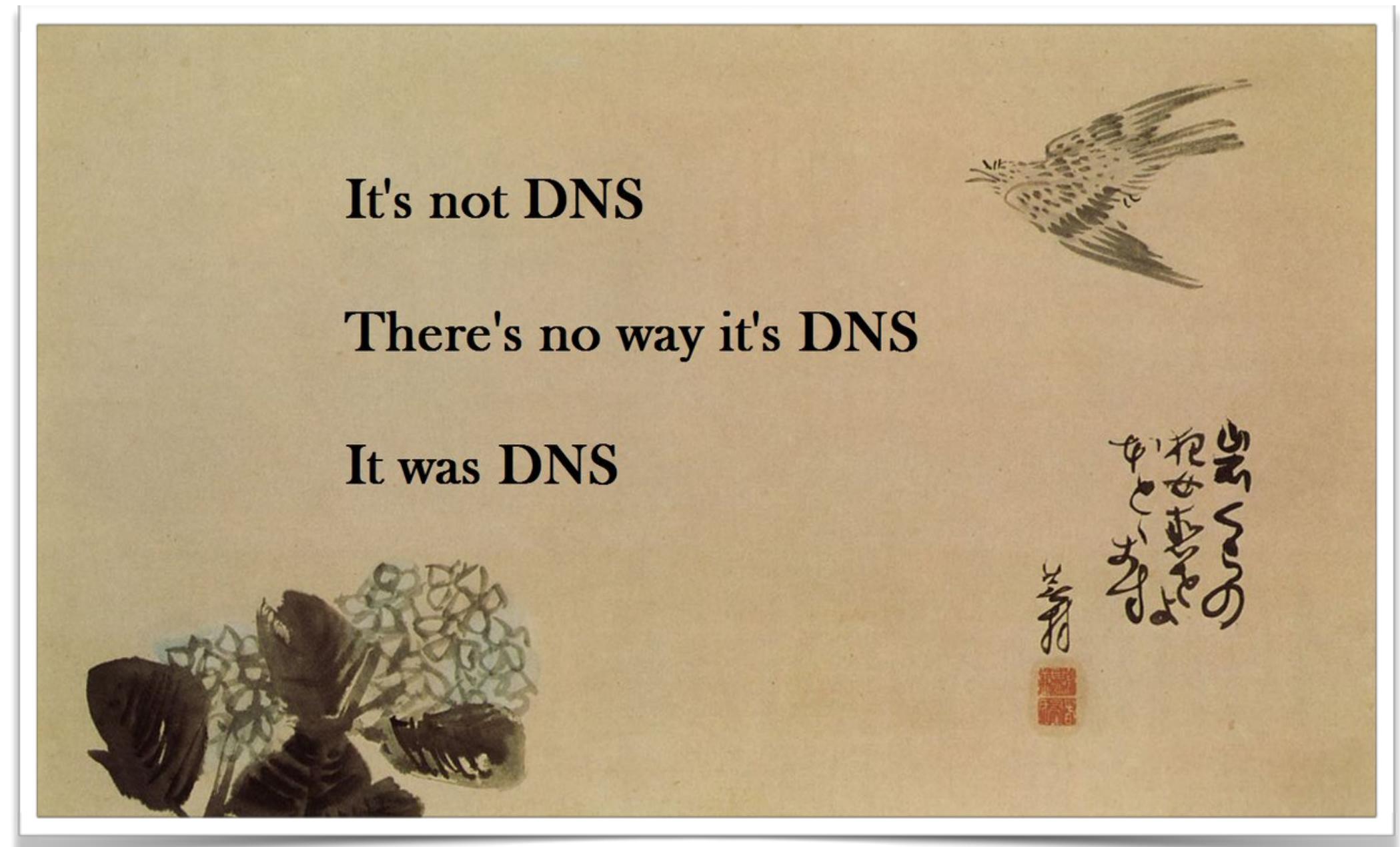
<https://stackoverflow.com/questions/51662015/service-not-exposing-in-kubernetes>



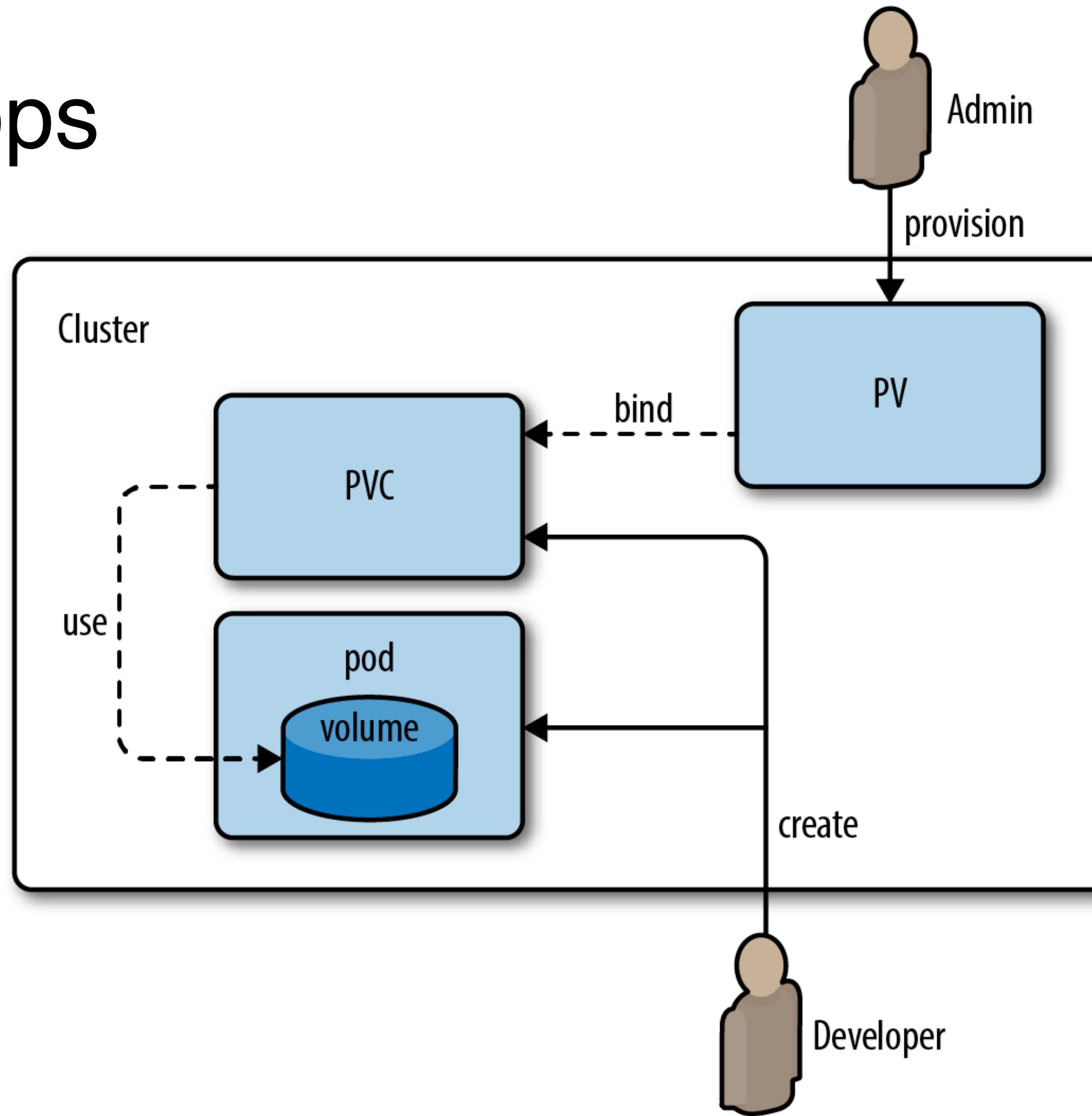


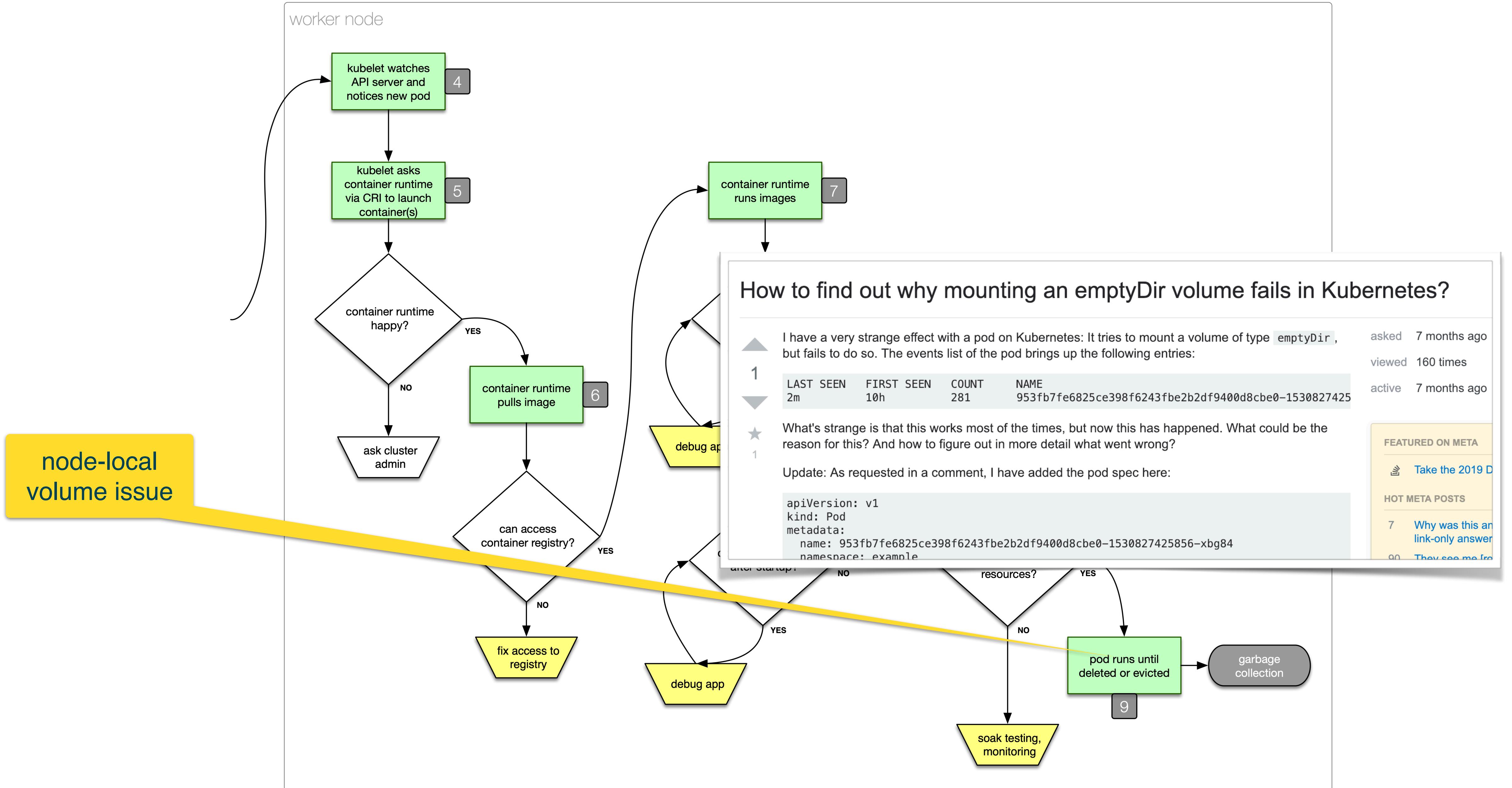
# What and how

- Container networking in Kubernetes (CNI)
- App-level or infra (CNI, DNS, etc.)?
- See [mhausenblas.info/cn-ref](http://mhausenblas.info/cn-ref)



# Stateful apps

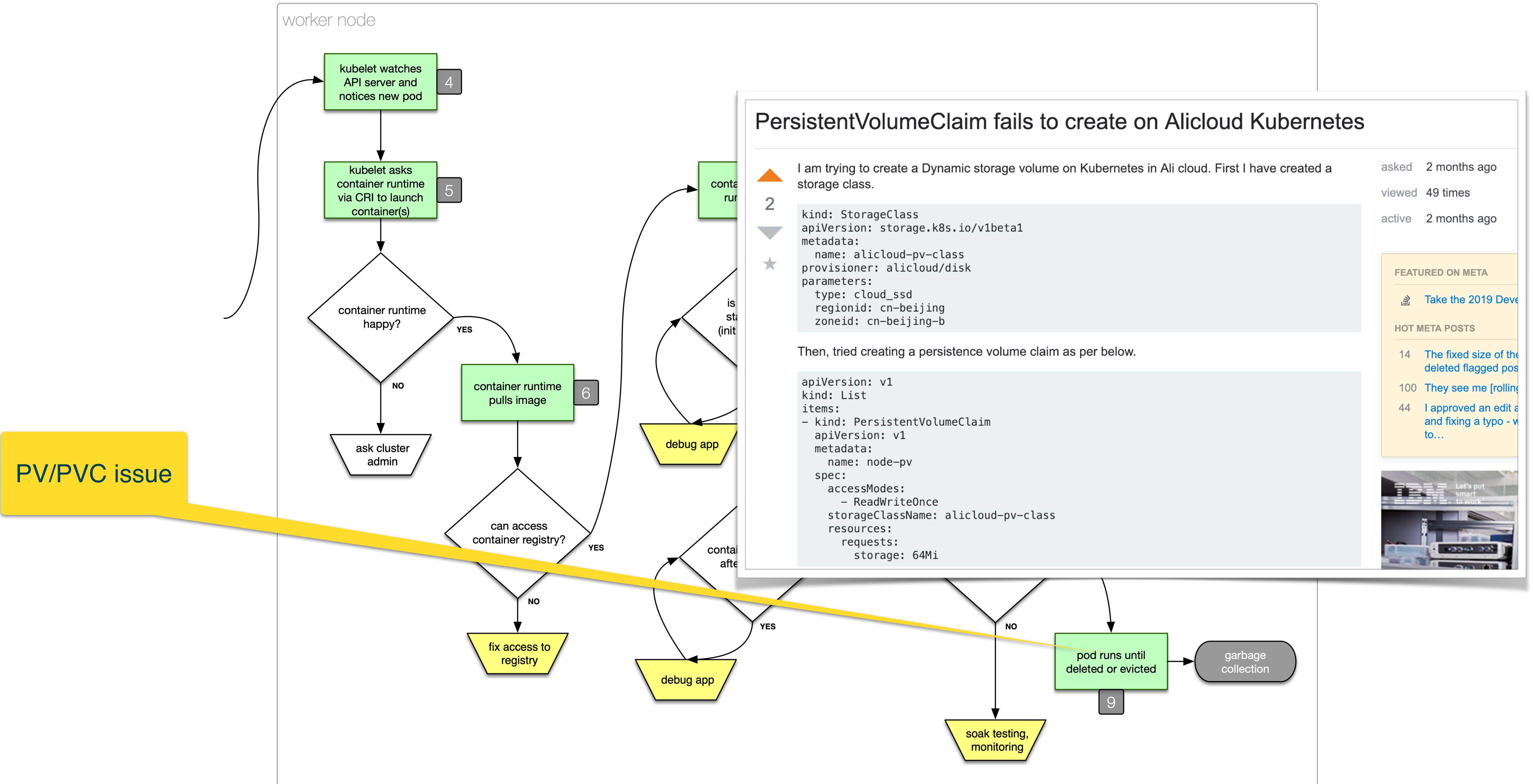




Hit me up on Twitter: @mhausenblas

<https://stackoverflow.com/questions/51206154/how-to-find-out-why-mounting-an-emptydir-volume-fails-in-kubernetes>





# What and how

- Storage in Kubernetes (CSI)
- Understand storage offerings (vendor docs!)
- Failure modes
- See [stateful.kubernetes.sh](https://stateful.kubernetes.sh)



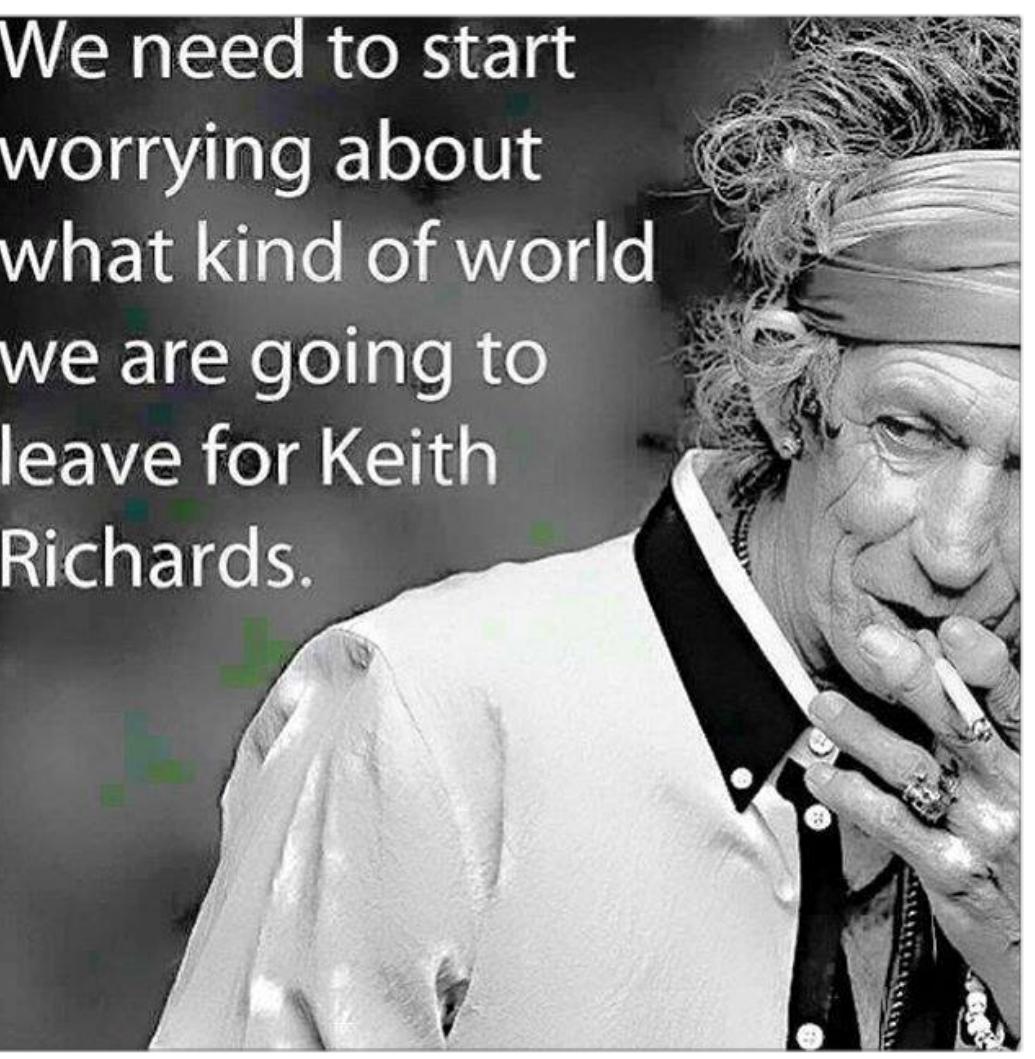
"What day is today?" asked Pooh

"It's the day we burn this motherfucker to the ground." squeaked Piglet



"My favorite day." said Pooh

# Vaccination



CODE WRITTEN IN HASKELL IS GUARANTEED TO HAVE NO SIDE EFFECTS.

...BECAUSE NO ONE WILL EVER RUN IT?



# Quizzie time!

You wrote an application server. For load-balancing purposes, where would you put a reverse proxy such as NGINX?

- A. Into the container (same Dockerfile)
- B. Into a side car container (same pod)
- C. Into a separate pod

# Proactive measures

*Architect your apps the cloud native way by ...*

- knowing and using the Kubernetes primitives (services, deployments)
- implementing retries & timeouts (in-tree or via service mesh)
- avoiding hardcoded (start-up) dependencies
- listening on `0.0.0.0` (not `127.0.0.1`)

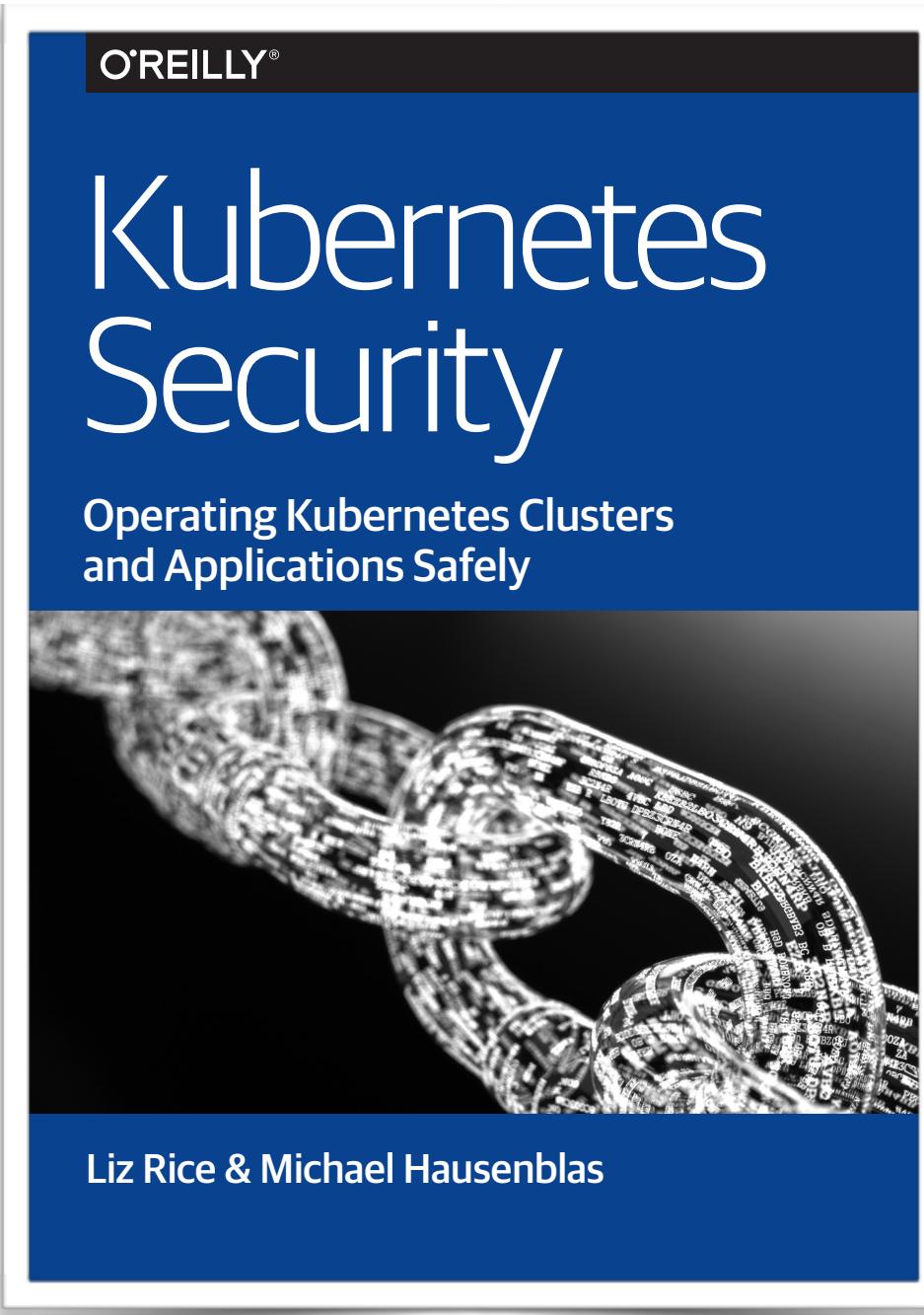
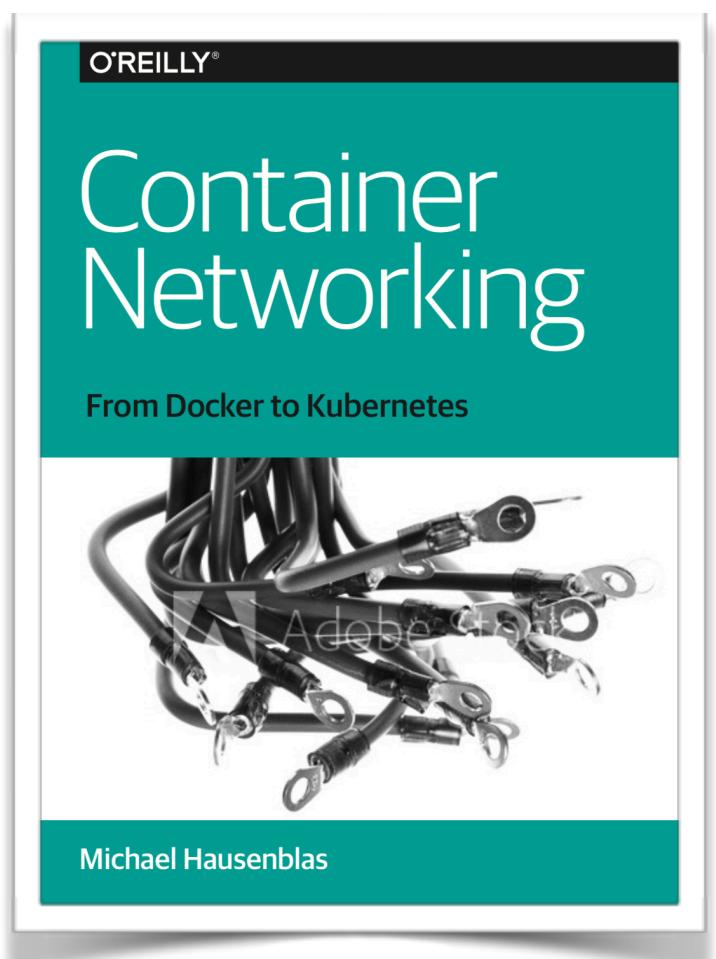
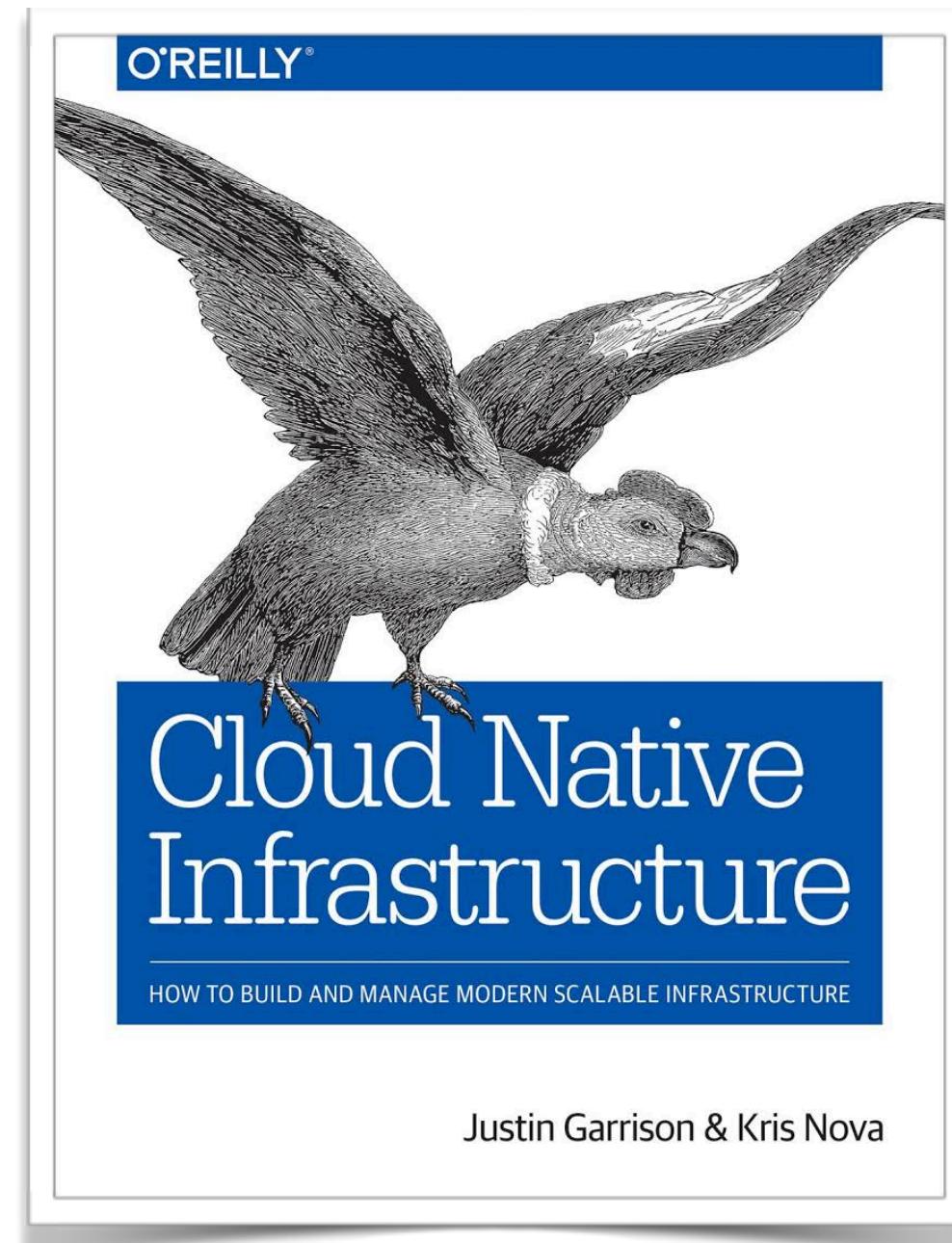
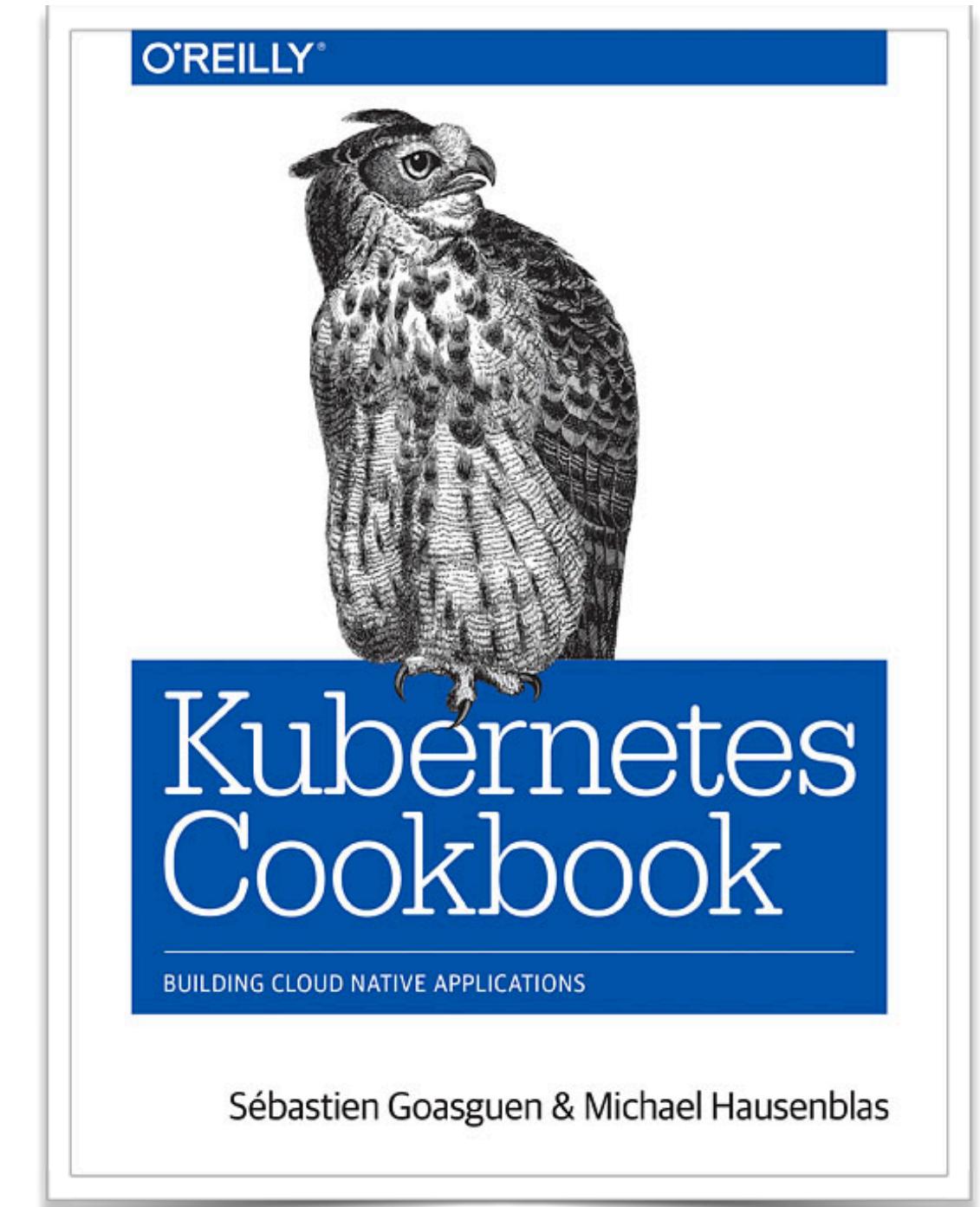
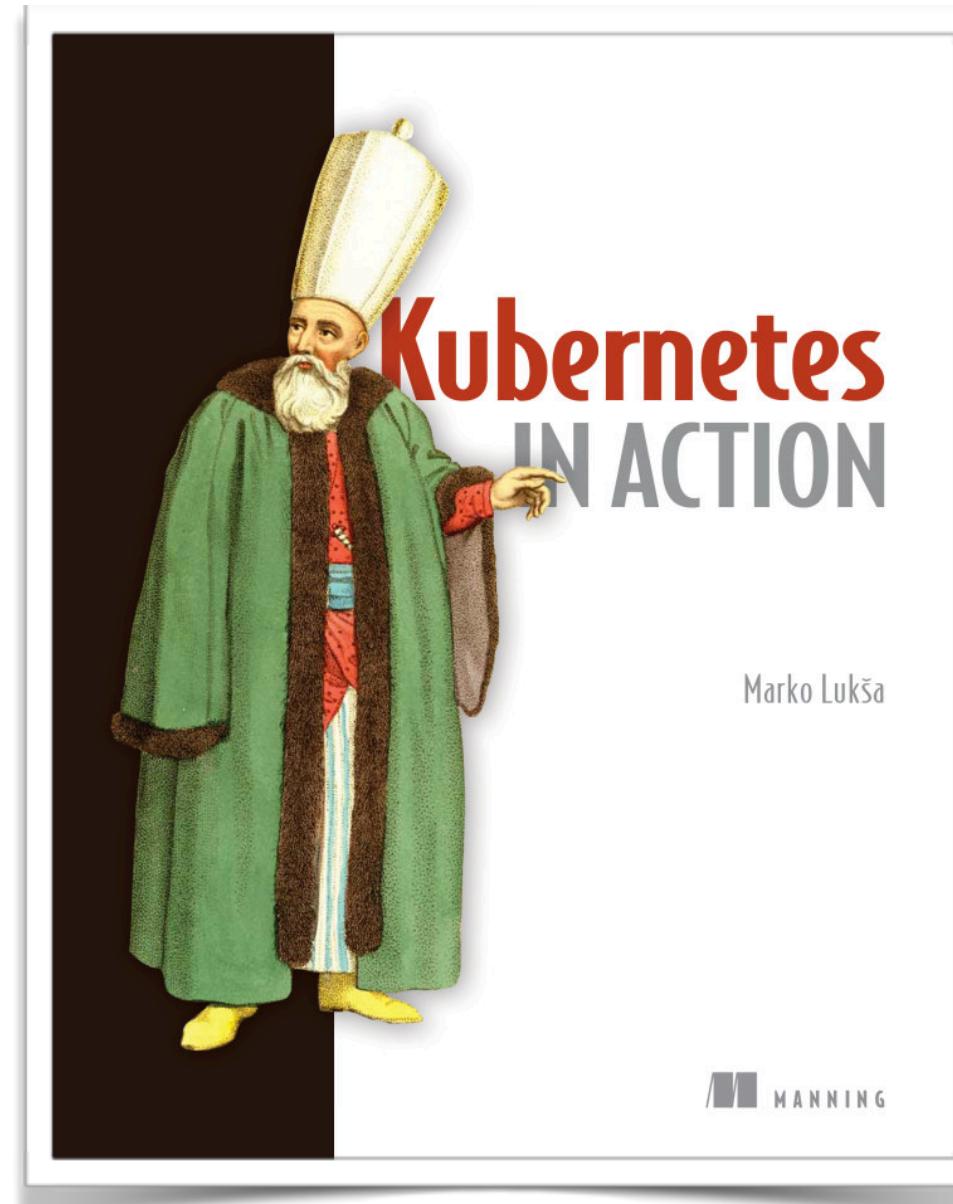
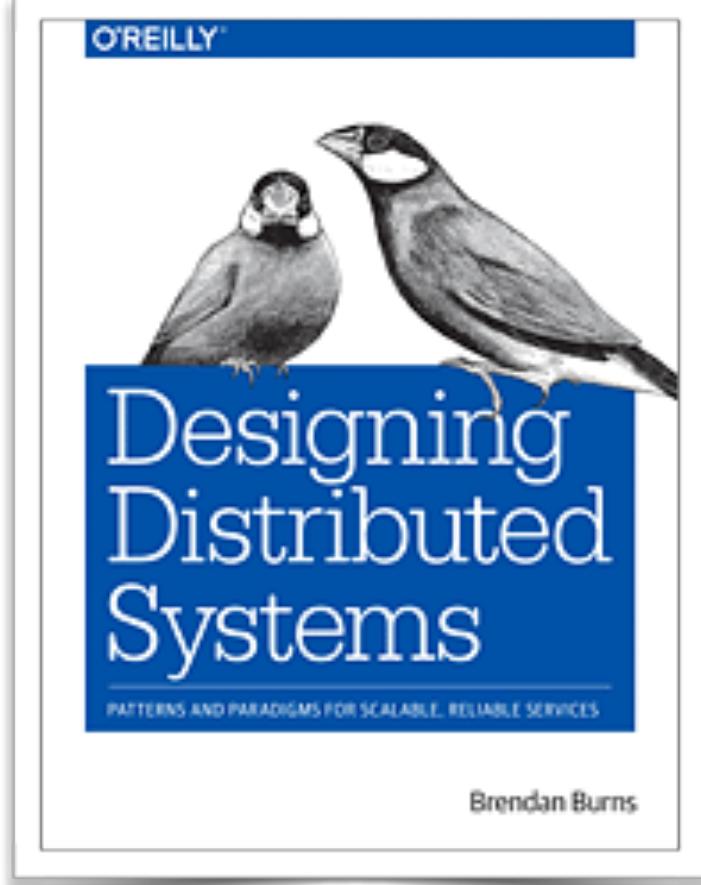
# Proactive measures

- Apply chaos engineering as long as all is well and learn from it where and how your system fails
- Provide debug tools in image, but also: footprint, security!
- Automate all the things: **Autoscaler, Brigade, Draft, Forge, Helm, knative, ksync, odo, Operators, Skaffold, watchpod**, etc.



# Resources





# Articles, slide decks, videos

- Kubernetes Troubleshooting site
- Debugging microservices - Squash vs. Telepresence
- Debugging and Troubleshooting Microservices in Kubernetes with Ray Tsang (Google)
- Troubleshooting Kubernetes Using Logs
- Debug a Go Application in Kubernetes from IDE
- Troubleshooting Kubernetes Networking Issues
- Video: CrashLoopBackoff, Pending, FailedMount and Friends: Debugging Common Kubernetes Cluster
- Video: Troubleshooting & Debugging Microservices in Kubernetes
- Slide deck: Evolution of Monitoring and Prometheus

# Articles, slide decks, videos

- 10 Most Common Reasons Kubernetes Deployments Fail: [Part 1](#) and [Part 2](#)
- Kubernetes Application Operator Basics
- Kubernetes: five steps to well-behaved apps
- Kubernetes Best Practices
- Developing on Kubernetes
- Debugging Microservices: How Google SREs Resolve Outages
- Debugging Microservices: Lessons from Google, Facebook, Lyft
- Troubleshooting Java applications on OpenShift
- Debugging Kubernetes PVCs

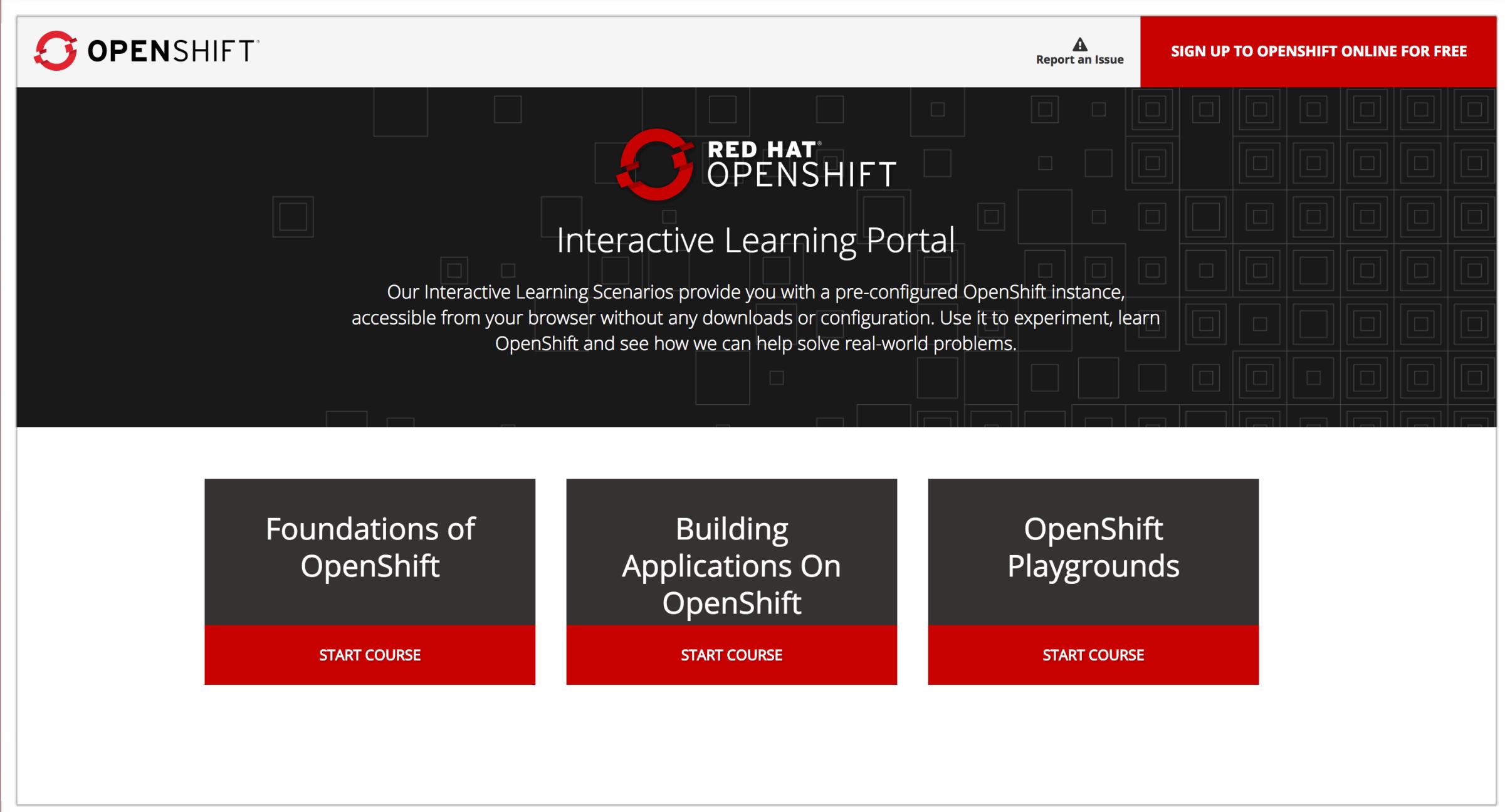
# Official Kubernetes docs

- [kubernetes.io/docs/tasks/debug-application-cluster/debug-application/](https://kubernetes.io/docs/tasks/debug-application-cluster/debug-application/)
- [kubernetes.io/docs/tasks/debug-application-cluster/](https://kubernetes.io/docs/tasks/debug-application-cluster/)
- [kubernetes.io/docs/tasks/debug-application-cluster/debug-init-containers/](https://kubernetes.io/docs/tasks/debug-application-cluster/debug-init-containers/)
- [kubernetes.io/docs/tasks/debug-application-cluster/debug-pod-replication-controller/](https://kubernetes.io/docs/tasks/debug-application-cluster/debug-pod-replication-controller/)
- [kubernetes.io/docs/tasks/debug-application-cluster/debug-service/](https://kubernetes.io/docs/tasks/debug-application-cluster/debug-service/)
- [kubernetes.io/docs/tasks/debug-application-cluster/debug-stateful-set/](https://kubernetes.io/docs/tasks/debug-application-cluster/debug-stateful-set/)
- [kubernetes.io/docs/tasks/debug-application-cluster/local-debugging/](https://kubernetes.io/docs/tasks/debug-application-cluster/local-debugging/)



redhat

# learn.openshift.com



The screenshot shows the homepage of the learn.openshift.com website. At the top, there's a navigation bar with the Red Hat logo, a search icon, and links for "Report an Issue" and "SIGN UP TO OPENSHIFT ONLINE FOR FREE". Below the header, the Red Hat OpenShift logo is displayed, followed by the text "Interactive Learning Portal". A descriptive paragraph explains that the portal provides pre-configured OpenShift instances accessible via browser. Below this, three course cards are listed: "Foundations of OpenShift", "Building Applications On OpenShift", and "OpenShift Playgrounds", each with a "START COURSE" button at the bottom.

OPENSHIFT

Report an Issue

SIGN UP TO OPENSHIFT ONLINE FOR FREE

RED HAT<sup>®</sup>  
OPENSHIFT

Interactive Learning Portal

Our Interactive Learning Scenarios provide you with a pre-configured OpenShift instance, accessible from your browser without any downloads or configuration. Use it to experiment, learn OpenShift and see how we can help solve real-world problems.

Foundations of OpenShift

Building Applications On OpenShift

OpenShift Playgrounds

START COURSE

START COURSE

START COURSE