



Rule Engines and Kubernetes DevOps

Luca Molteni

Principal Software Engineer



Rule Engines in the Cloud

- First idea was to evaluate rules in stateless interactions
- Drools works, but it's much more powerful than that

Rule Engines in the Cloud

- Applications built in the cloud from the beginning
- Problems of the applications are the problems of the cluster (and vice versa)
- To track problems, we needed something that
 - can easily write sanity rules
 - can run indefinitely with incremental data
 - can correlate different kind of data
- Drools shines in each of these problems

Case Study: Executable SOPs

SOP: standard operating procedure

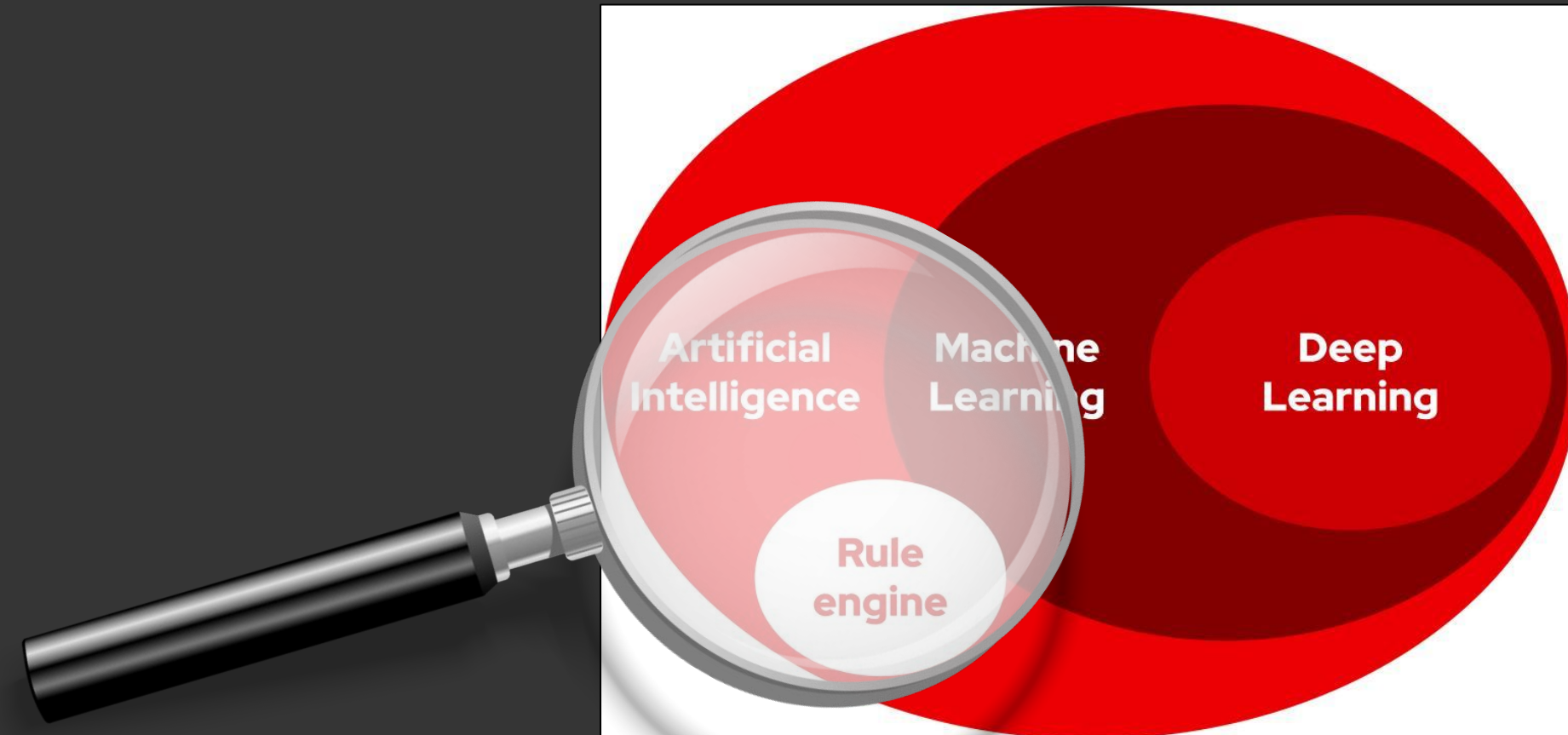
1. Simple error: experienced Devops
2. Simple error: experienced Devops at 3:00 AM
3. Complex error with custom logic
4. Complex error with custom logic at 3:00 AM

The knowledge of the whole system (infrastructure and application logic) is encoded in rules

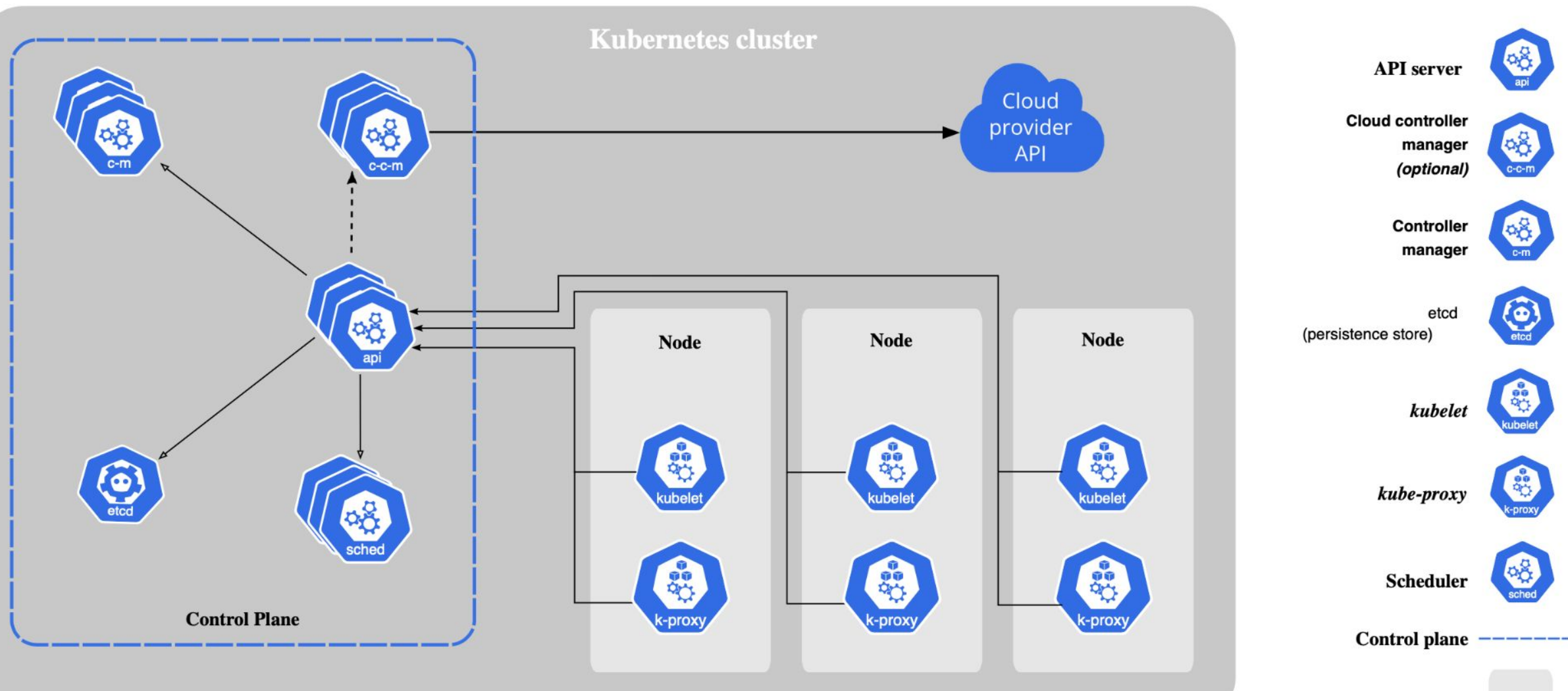
Introducing RuleOps

An exploration space for automating solutions to problematic scenarios
using Drools interactions with the Kubernetes Control Plane

Symbolic AI vs Machine Learning



Kubernetes Control Plane



Kubernetes API

- Resources
 - Pods, Deployments, Services
- Endpoints
- Custom Resources
- Java mapped type using the Kubernetes Java Client

Transactionality

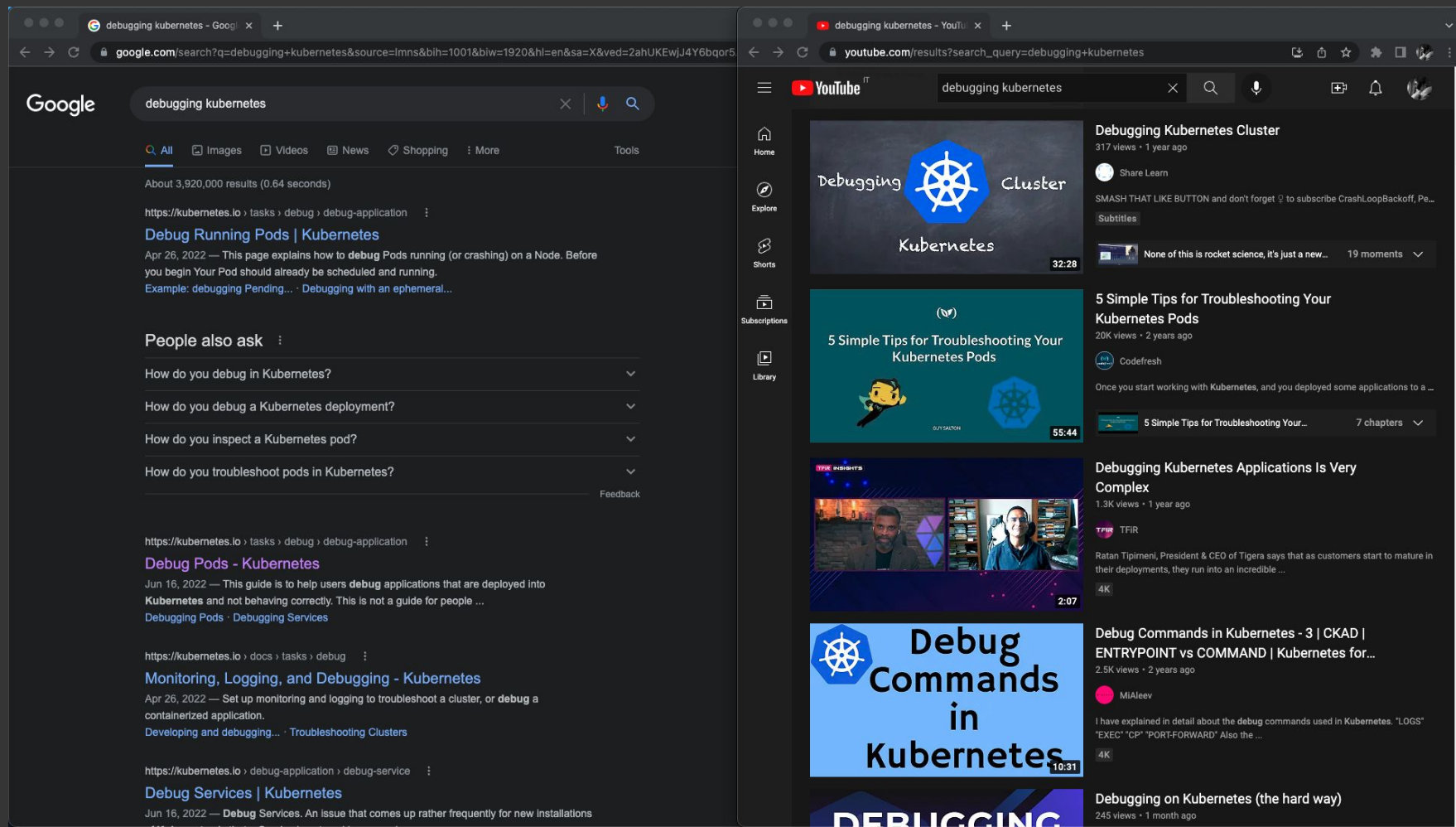
- A typical operation on the control plane involves creating multiple resources
- Having the cluster in an inconsistent state is fine, as Kubernetes will self-heal to the correct state given a correct definition of resources
- If the resources are defined wrongly, the cluster will be broken anyway

RuleOps Correlates Different Resource Types

- Database Rows
- Kubernetes Custom Resources
- Monitoring Data
- Logging

> “debugging Kubernetes”

3+ Mil results indexed, countless videos, “very complex”, “the hard way”, ...



The image displays two side-by-side browser screenshots. The left screenshot shows a Google search for "debugging kubernetes" with approximately 3,920,000 results. The top result is "Debug Running Pods | Kubernetes" from kubernet.es.io, dated April 26, 2022. Below it, a "People also ask" section lists questions like "How do you debug in Kubernetes?". The right screenshot shows YouTube search results for the same query. The top video is "Debugging Kubernetes Cluster" with 317 views. Other videos include "5 Simple Tips for Troubleshooting Your Kubernetes Pods" (20K views) and "Debug Commands in Kubernetes - 3 | CKAD | ENTRYPOINT vs COMMAND" (2.5K views).

line:



A screenshot of a Google search page for the query "debugging kubernetes flowchart". The search bar at the top shows the query and the Google logo. Below the search bar, there are tabs for "All", "Images", "Videos", "News", "Shopping", and "More". The "Images" tab is selected, displaying a grid of search results. The first row of results includes thumbnails for "cluster", "k8s", "gpu", "node", "pod", "openshift", and "container". The second row shows four thumbnails of flowcharts: "Tim McNamara on Twitter...", "troubleshooting Kubernetes deployments", "Kubernetes curated resources", and "Extending Kubernetes | Kubernetes...". The third row shows three thumbnails: "12", "External load balancer", and "kubernetes/website #19518 Flowchart for debugging cluster". The thumbnails contain various diagrams, including flowcharts, network diagrams, and Kubernetes architecture diagrams.

[illegible]

Procedural vs Declarative

Show me the code!

Demo 1:


Inconsistent Number of Pods

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: pod-demo
spec:
  hard:
    pods: "2"
```






```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: pod-quota-demo
spec:
  selector:
    matchLabels:
      purpose: quota-demo
  replicas: 3
  template:
    metadata:
      labels:
        purpose: quota-demo
    spec:
      containers:
      - name: pod-quota-demo
        image: nginx
```

 **kubernetes**

default

 Search

Workloads

Workloads N

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service N

Ingresses

Services

Config and Storage

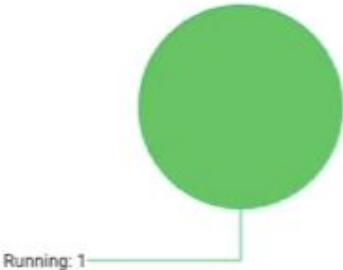
Config Maps N

Persistent Volume Claims N

Secrets N

Storage Classes


Workload Status



Running: 1

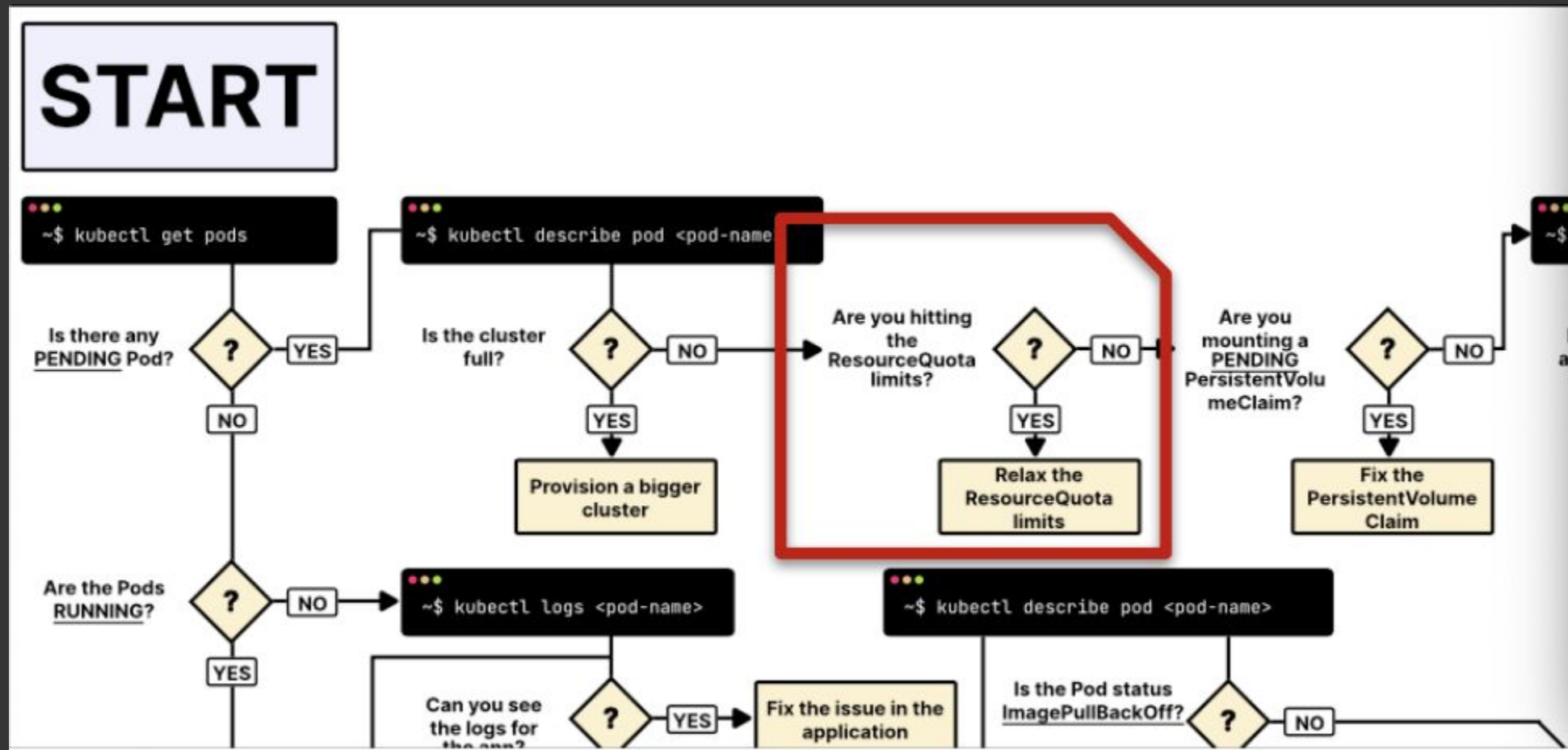
Deployments

Deployments

Name	Namespace
 pod-quota-demo	default

Pods


```
- name: Relax the ResourceQuota limits Deployment PENDING
when:
- given: Deployment
  as: $d
- given: DeploymentCondition
  having:
    - type == "Available"
    - status == "False"
    from: $d.status.conditions
- given: DeploymentCondition
  having:
    - message contains "exceeded quota"
    from: $d.status.conditions
then: |
  insert(new Advice("Relax the ResourceQuota limits", ...
```



Demo 2:

Resource Quota Hit

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: mem-cpu-demo
spec:
  hard:
    requests.cpu: "1"
    requests.memory: 1Gi
    limits.cpu: "2"
    limits.memory: 2Gi
```





```
apiVersion: v1
kind: Pod
metadata:
  name: quota-mem-cpu-demo
spec:
  containers:
    - name: quota-mem-cpu-demo-ctr
      image: nginx
      resources:
        limits:
          memory: "1Gi"
          cpu: "1"
        requests:
          memory: "600Mi"
          cpu: "500m"
```



```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  labels:
    app.kubernetes.io/version: 1.0.0-SNAPSHOT
    app.kubernetes.io/name: hello-pvdf
  name: hello-pvdf
  namespace: default
spec:
  resources:
    limits:
      cpu: 2000m
      memory: 2Gi
    requests:
      cpu: 500m
      memory: 500Mi
  volumeMounts:
    - mountPath: /mnt/data
      name: my-pvc-claim
      readOnly: false
```



Stateful Sets	
Name	Namespace
 hello-pvdf	default



```
- name: Relax the ResourceQuota limits StatefulSet PENDING
  when:
    - given: StatefulSet
      as: $s
      having:
        - spec.replicas != status.replicas
    - given: Event
      having:
        - message contains "exceeded quota"
      from: DroolsK8sClient.eventsFor($s)
  then: |
    insert(new Advice("Relax the ResourceQuota limits", ...
```

Demo 3:

Persistence Volume not set

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: task-pv-claim
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: pod-quota-demo
spec:
  selector:
    matchLabels:
      purpose: quota-demo
  replicas: 1
  template:
    metadata:
      labels:
        purpose: quota-demo
    spec:
      volumes:
      - name: task-pv-storage
        persistentVolumeClaim:
          claimName: task-pv-claim
      containers:
      - name: pod-quota-demo
        image: nginx
        ports:
        - containerPort: 80
          name: "http-server"
        volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: task-pv-storage
```

Workloads

Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses

Failed: 1

Deployments

Failed: 1

Pods

Failed: 1

Replica Sets

Deployments

```
- name: Fix the PersistentVolumeClaim Pod PENDING
when:
- given: PersistentVolumeClaim
  as: $pvc
  having:
    - status.phase == "Pending"
- given: Pod
  as: $pod
  having:
    - status.phase == "Pending"
- given: Volume
  having:
    - persistentVolumeClaim!.claimName == $pvc.metadata.name
  from: $pod.spec.volumes
then: |
  insert(new Advice("Fix the PersistentVolume", ...
```

Further development of RuleOps

CEP (Complex Events Processing)

- Correlating events of a monitoring system
- Temporal operators
 - Before, After
 - Interval ranges
- Can highlight internals of Kubernetes while used with the controllers' API

RuleOps Deployment Models

- Sidecar application
- CLI
- As a Kubernetes Controller
- As a Kubernetes plugin?

Performance / Efficiency

- Drools is fast
- Based on Drools Executable Model
- Fetch only the needed Kubernetes resource types
 - If a rule validates Pods, it will fetch only Pods from the Control Plane, therefore reducing the amount of data passing

Level trigger vs Edge Trigger

<https://medium.com/hackernoon/level-triggering-and-reconciliation-in-kubernetes-1f17fe30333d>



Matteo Mortari

Principal Sw Engineer



Luca Molteni

Principal Sw Engineer



Daniele Zonca

Sr Principal Sw Engineer, Architect

<https://github.com/kiegroup/ruleops>

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHat