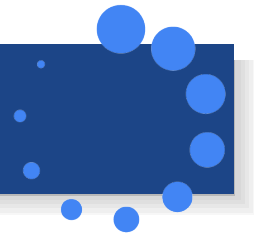
Rightsize your Pods with Vertical Pod Autoscaling

Beata Skiba <bskiba@google.com>
Software Engineer @ Google, working on Kubernetes Autoscaling

Life of a resource request



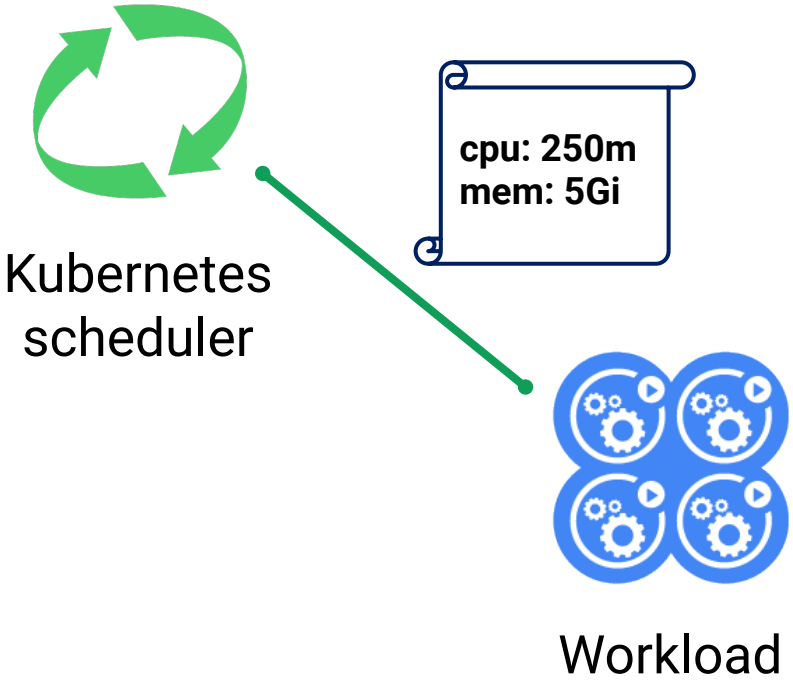
Deploying is easy, right?

```
apiVersion: apps/v1
kind: Deployment
...
template:
  metadata:
    labels:
      app: worker
  spec:
    containers:
    - name: worker
      resources:
        requests:
          memory: "100M"
          cpu: "250m"
```

Well...

```
apiVersion: apps/v1
kind: Deployment
...
template:
  metadata:
    labels:
      app: worker
  spec:
    containers:
    - name: worker
      resources:
        requests:
          memory: ??
          cpu: ??
```

Resource request is a contract



Requests in action: Case 1

Will it fit?

Kubernetes scheduler

Pods

cpu: 200m

cpu: 600m

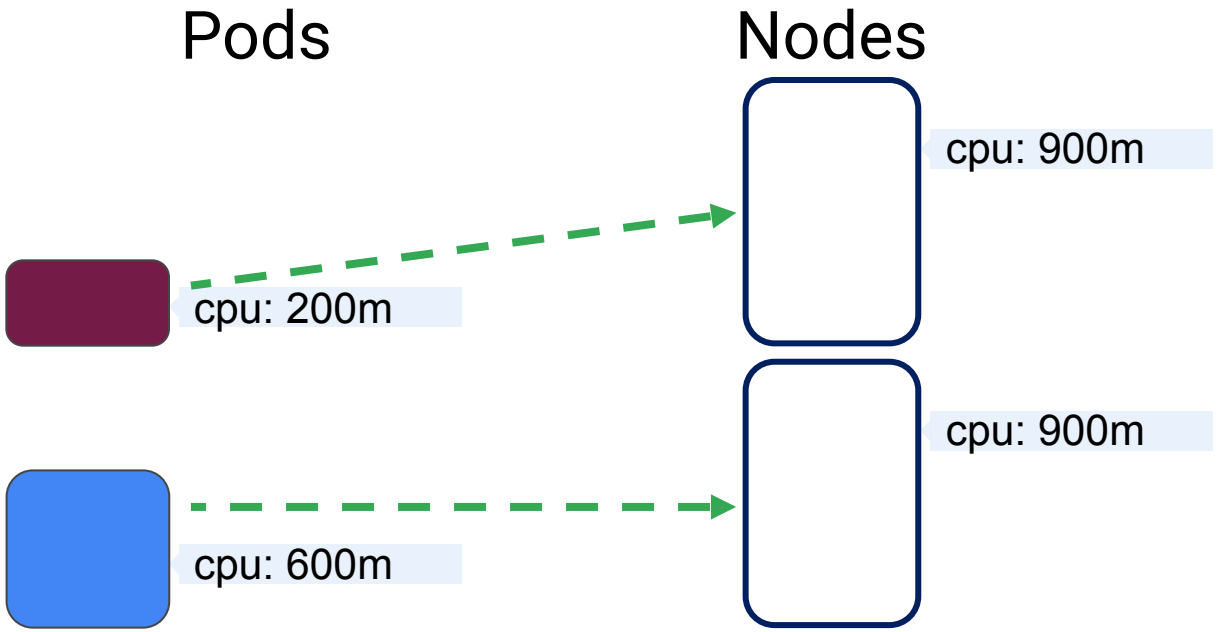
Nodes

cpu: 900m

cpu: 900m

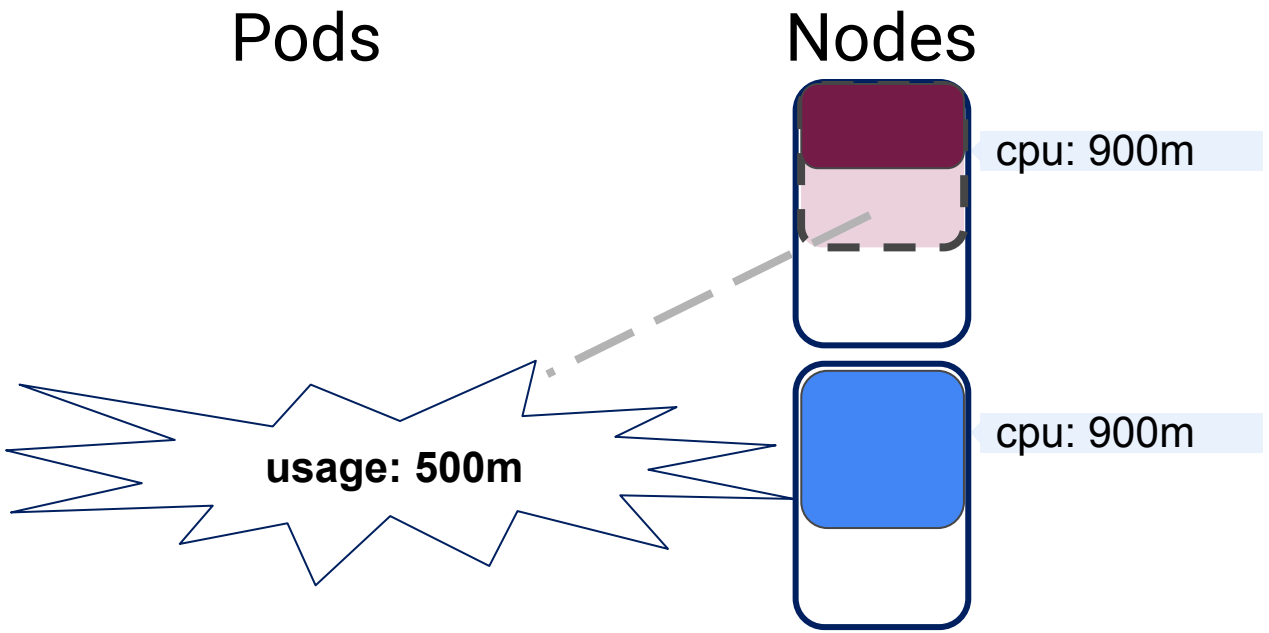
Will it fit?

Kubernetes scheduler



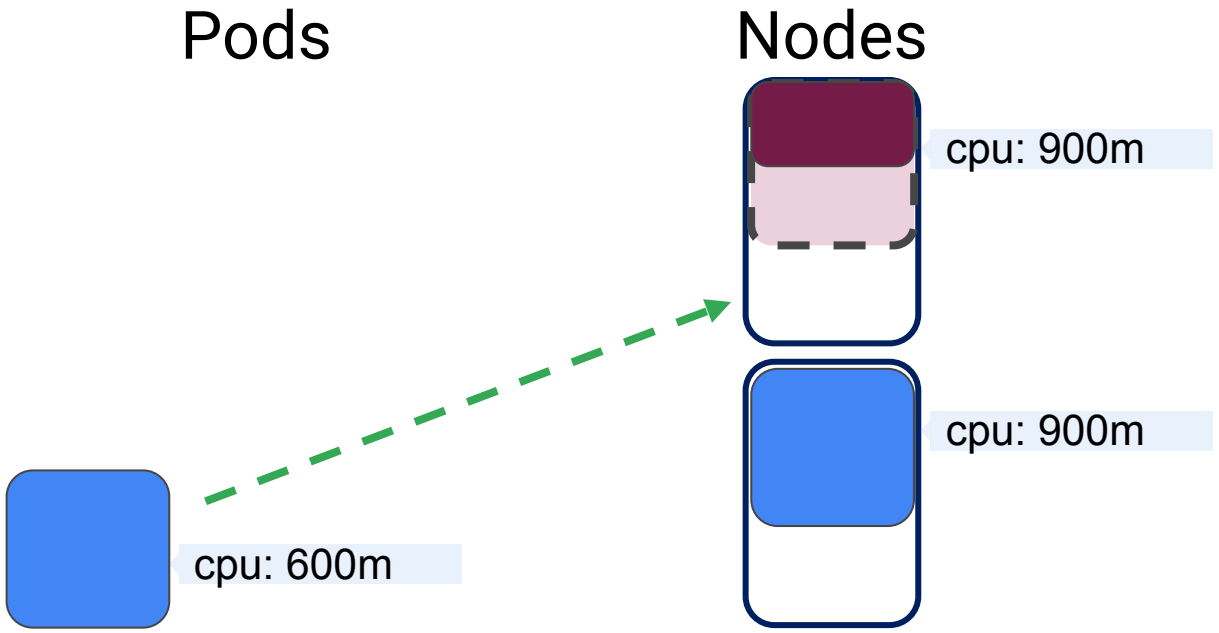
Will it fit?

Kubernetes scheduler



Will it fit?

Kubernetes scheduler

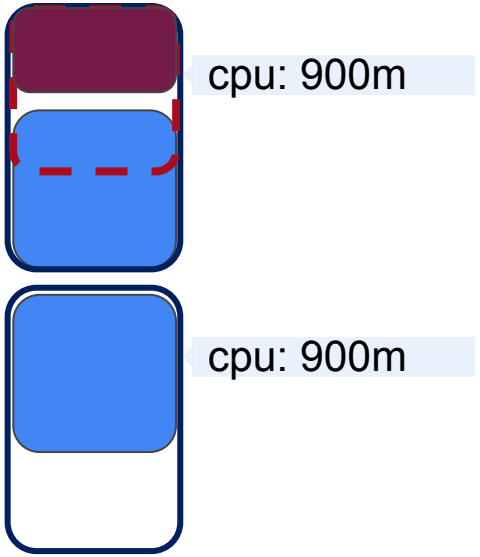


Will it fit?

Pods

Nodes

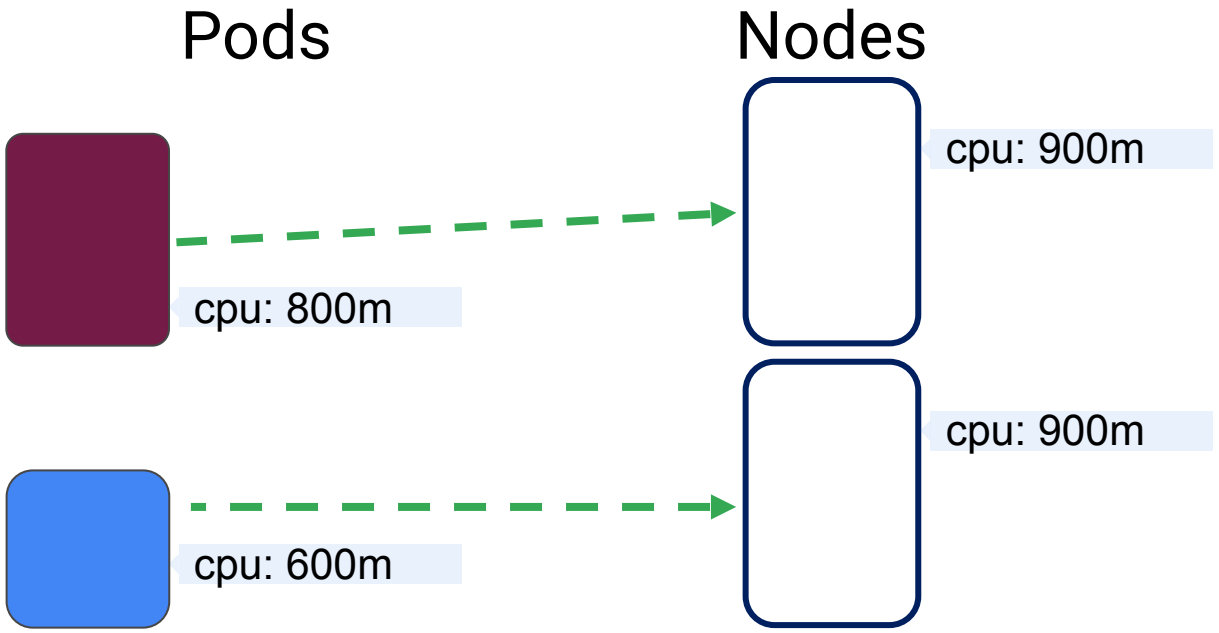
Kubernetes scheduler



Requests in action: Case 2

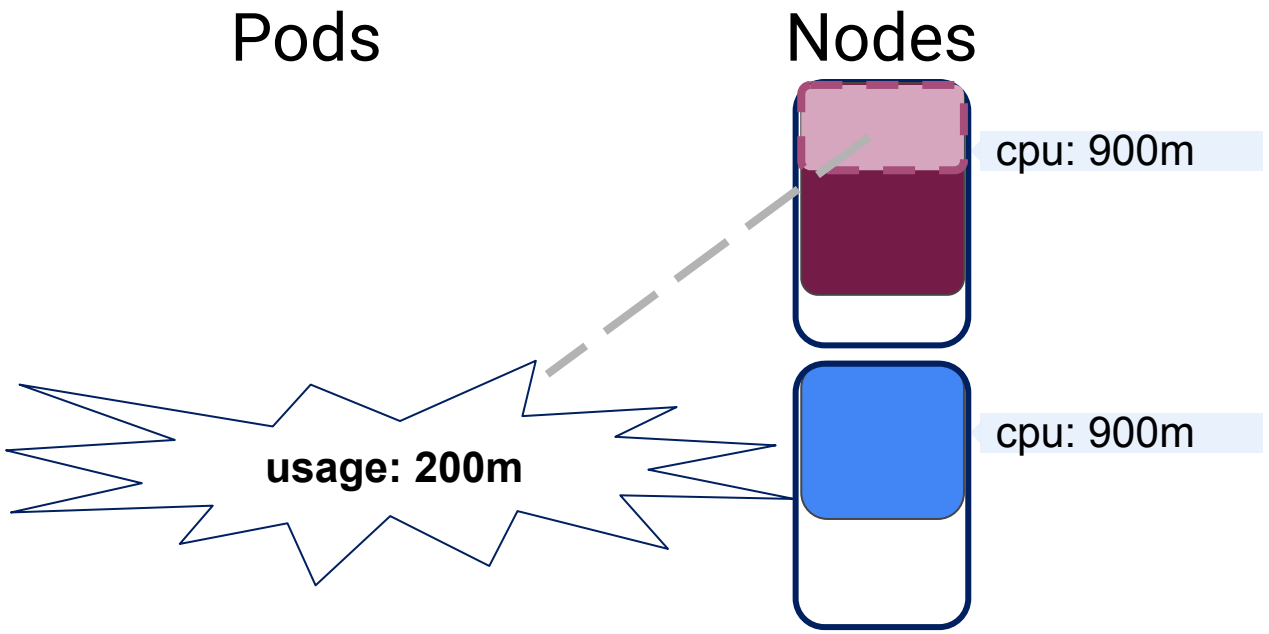
Will it fit?

Kubernetes scheduler



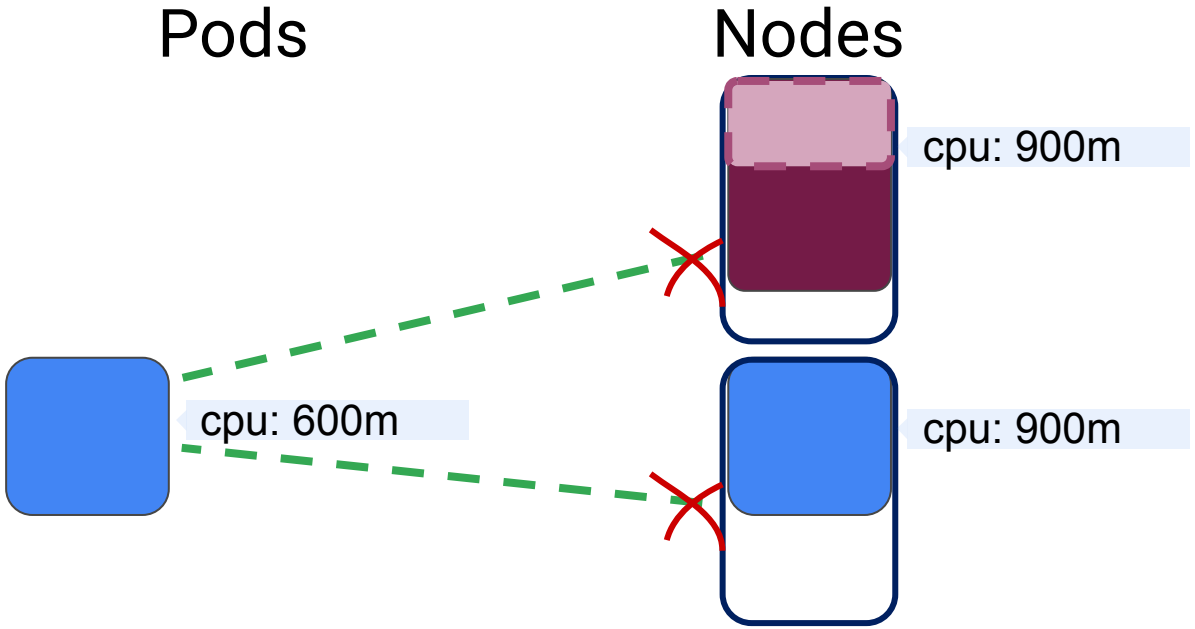
Will it fit?

Kubernetes scheduler



Will it fit?

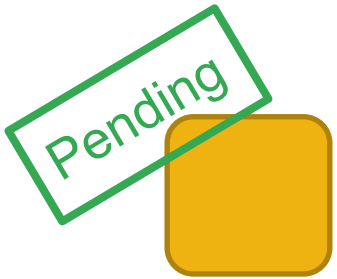
Kubernetes scheduler



Will it fit?

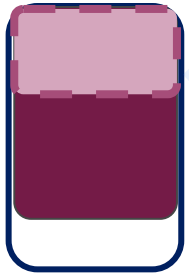
Kubernetes scheduler

Pods

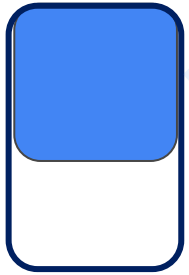


cpu: 600m

Nodes



cpu: 900m



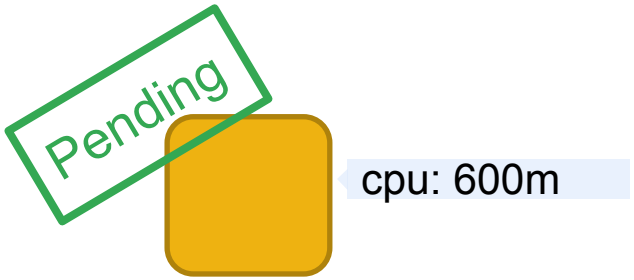
cpu: 900m

Will it fit?

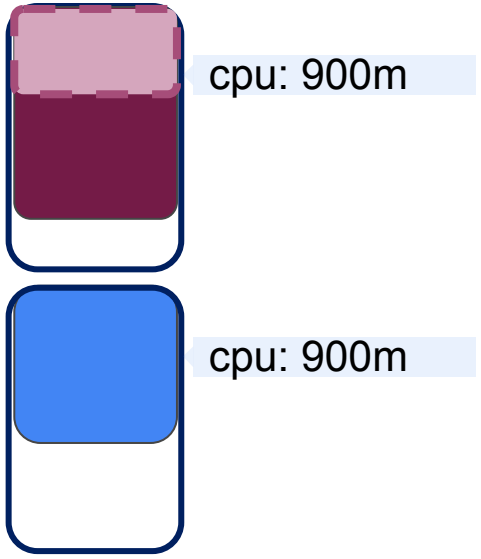
Kubernetes scheduler

Cluster Autoscaler

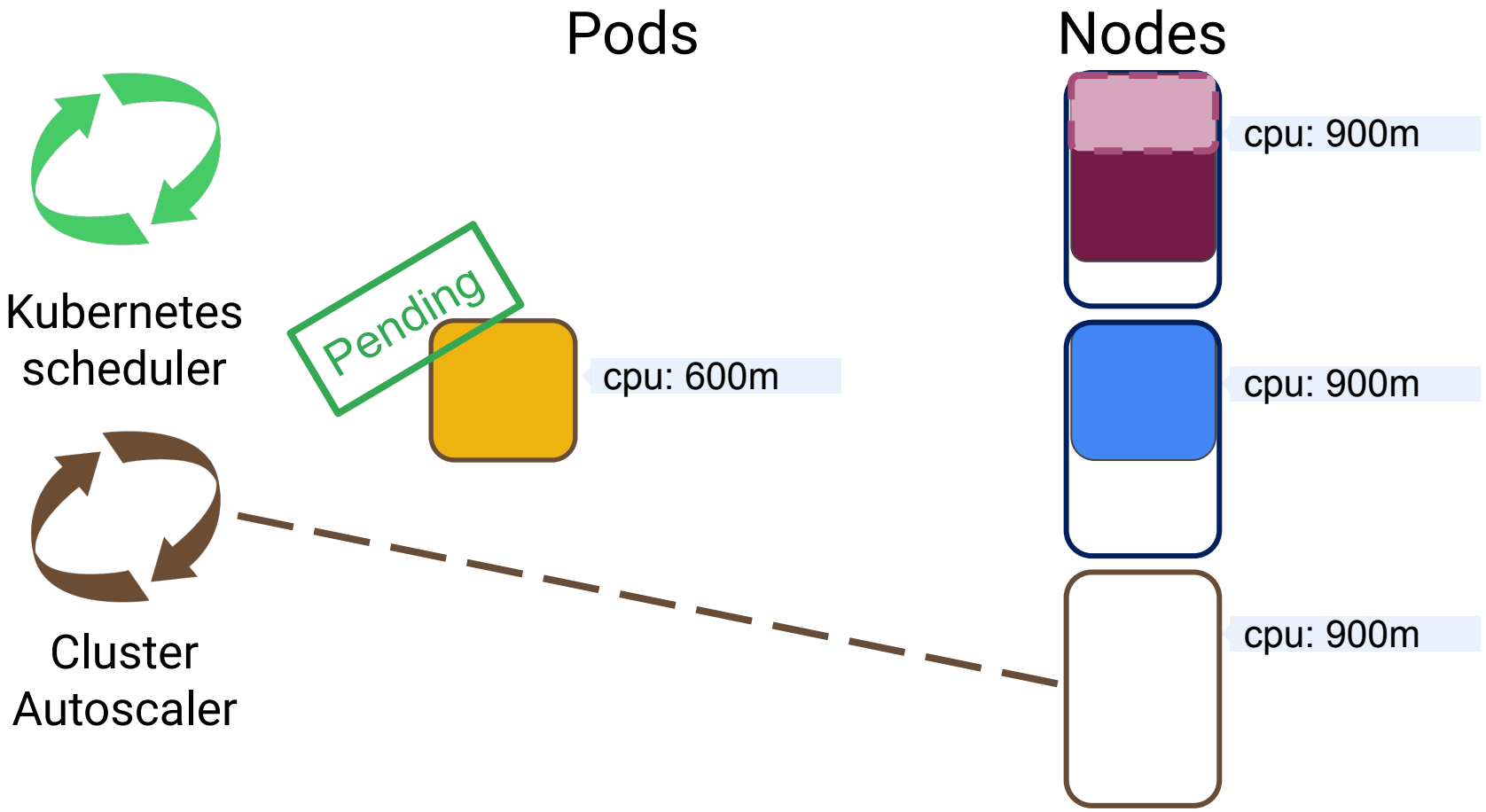
Pods



Nodes



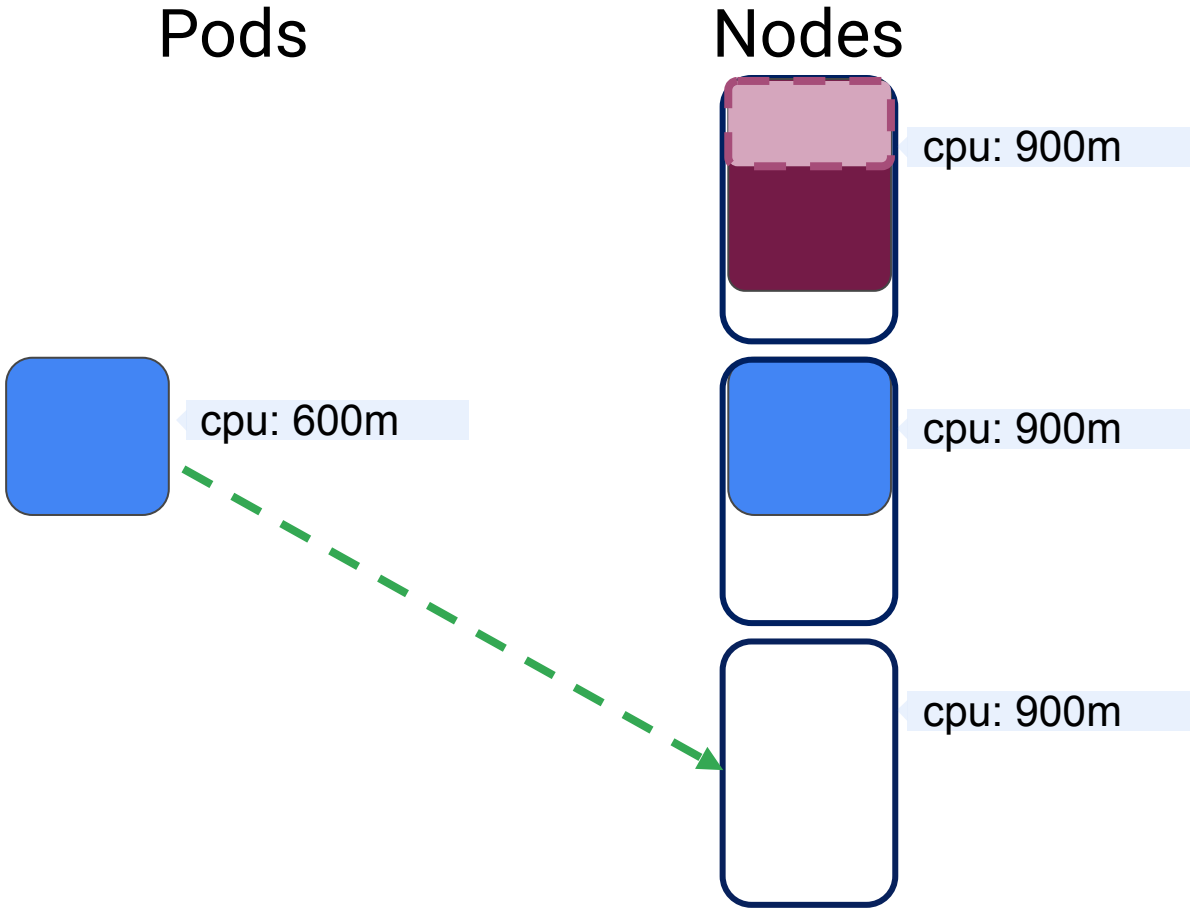
Will it fit?


Pods
Nodes
Kubernetes scheduler
Pending
cpu: 600m
cpu: 900m
cpu: 900m
cpu: 900m
Cluster Autoscaler

Will it fit?

Kubernetes scheduler

Cluster Autoscaler



HELP! My cluster is on fire

Too little

- CPU starvation
- Out Of Memory errors
- Workload eviction

Too much

- Resource waste

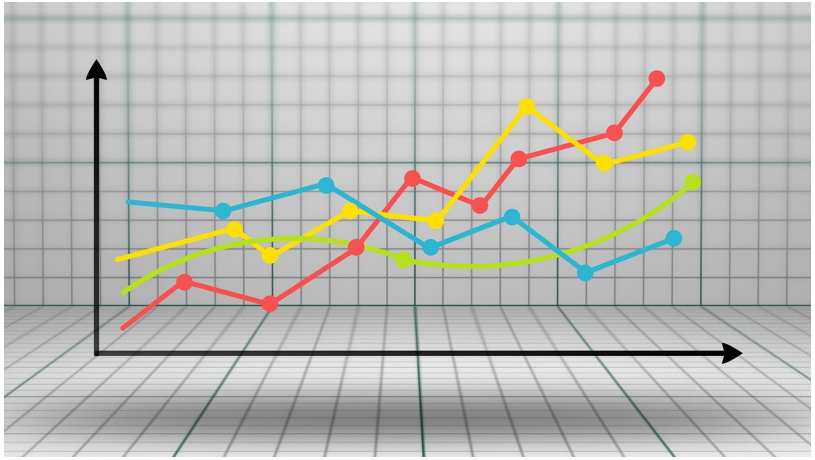
How to get it right?

- ? Canarying
- ? Trial and error
- ? Tea leaves

But....

Where to get the numbers from?

- Time, effort & resources
- Canary may differ from prod
- Resource needs will change over time



VPA to the rescue

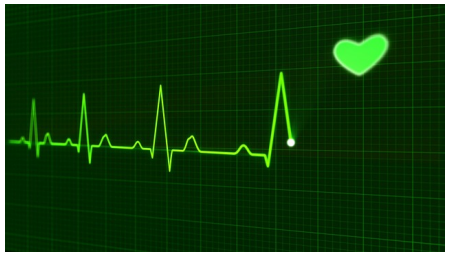
VPA to the rescue

Observes usage

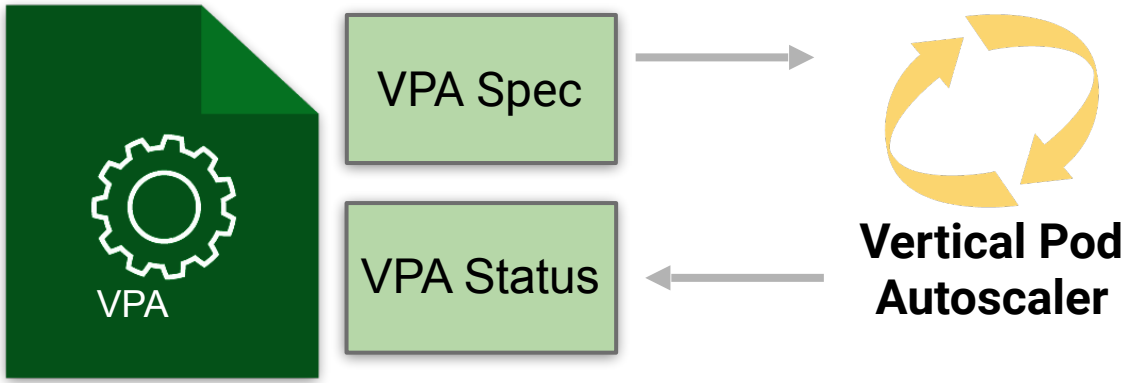
Recommends resources

Updates resources

EXPERIMENTAL



It's VPA objects all the way down



VPA Object

What the Spec?

```yaml
apiVersion: "autoscaling.k8s.io/v1beta1"
kind: VerticalPodAutoscaler
metadata:
  name: worker-vpa
spec:
  selector:
    matchLabels:
      app: worker
  updatePolicy:
    updateMode: "Off"
  resourcePolicy:
    containerPolicies:
    - containerName: "*"
      maxAllowed:
        memory: 5Gi
```

Scale WHAT?

```yaml
apiVersion: "autoscaling.k8s.io/v1beta1"
kind: VerticalPodAutoscaler
metadata:
  name: worker-vpa
spec:
  selector:
    matchLabels:
      app: worker
  updatePolicy:
    updateMode: "Off"
  resourcePolicy:
    containerPolicies:
    - containerName: "*"
      maxAllowed:
        memory: 5Gi
```

Scale HOW?

```yaml
apiVersion: "autoscaling.k8s.io/v1beta1"
kind: VerticalPodAutoscaler
metadata:
  name: worker-vpa
spec:
  selector:
    matchLabels:
      app: worker
  updatePolicy:
    updateMode: "Off"
  resourcePolicy:
    containerPolicies:
    - containerName: "*"
      maxAllowed:
        memory: 5Gi
```

Off
Initial
Auto

EXPERIMENTAL

Scale HOW?

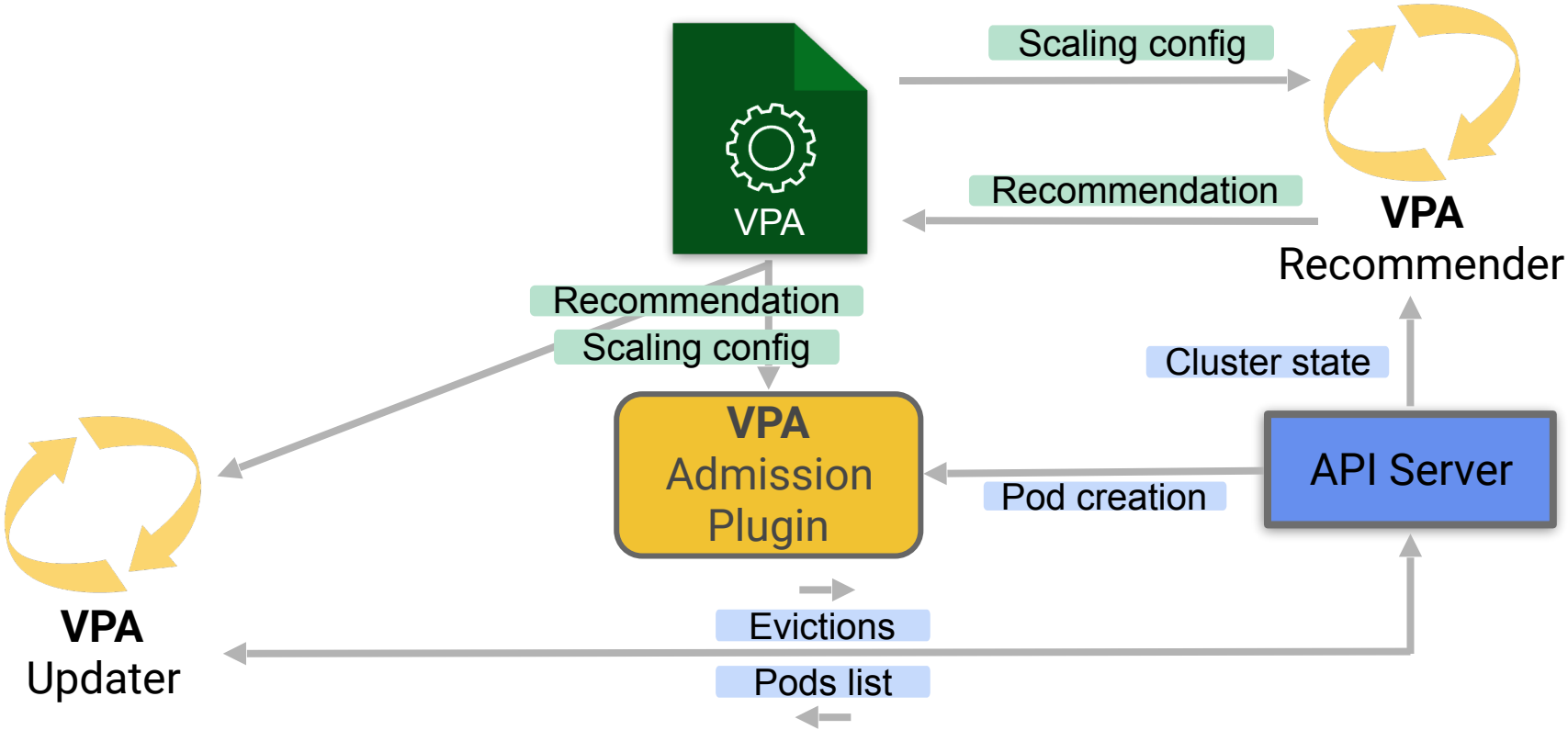
```yaml
apiVersion: "autoscaling.k8s.io/v1beta1"
kind: VerticalPodAutoscaler
metadata:
  name: worker-vpa
spec:
  selector:
    matchLabels:
      app: worker
  updatePolicy:
    updateMode: "Off"
  resourcePolicy:
    containerPolicies:
    - containerName: "*"
      maxAllowed:
        memory: 5Gi
```

VPA status

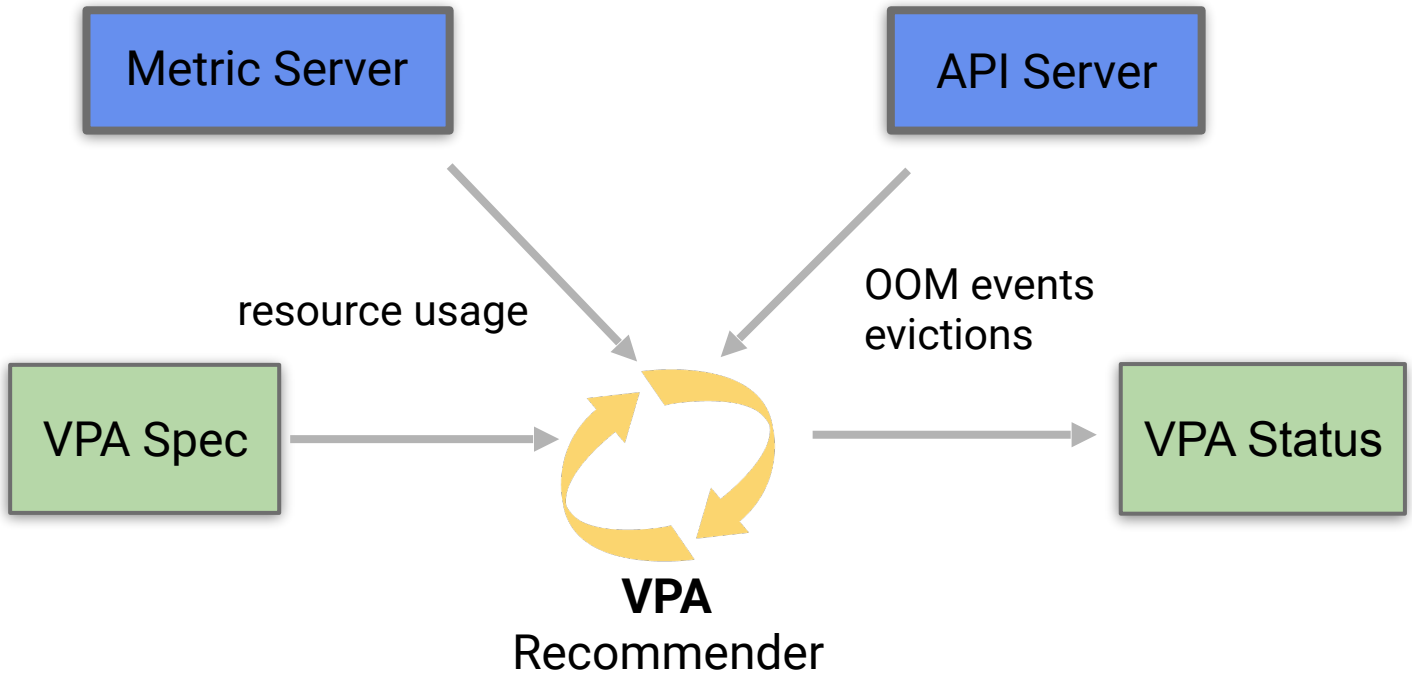
```
status:
  recommendation:
    containerRecomendations:
    - containerName: worker
      lowerBound:
        cpu: 584m
        memory: 218M
      target:
        cpu: 587m
        memory: 262M
      upperBound:
        cpu: 2088m
        memory: 387M
```

VPA under the hood

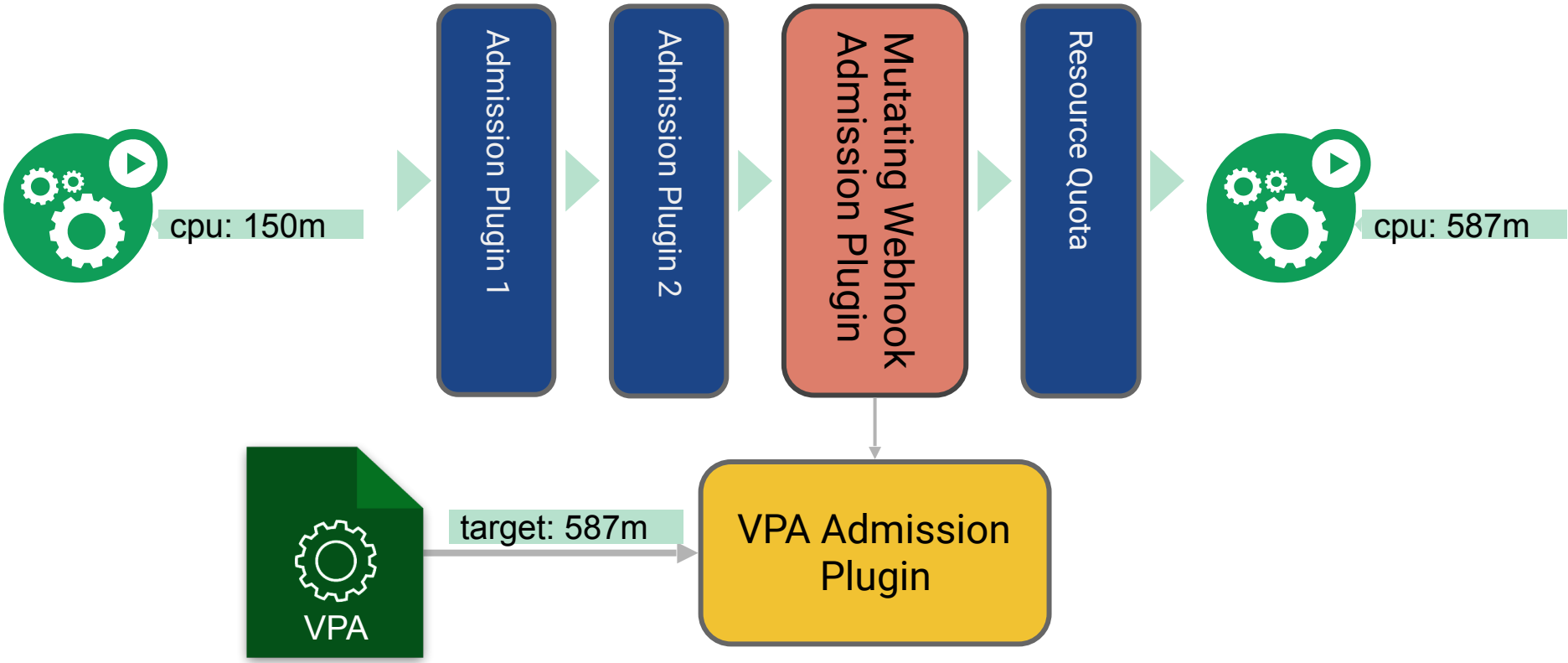
Bits and pieces



Watch and recommend

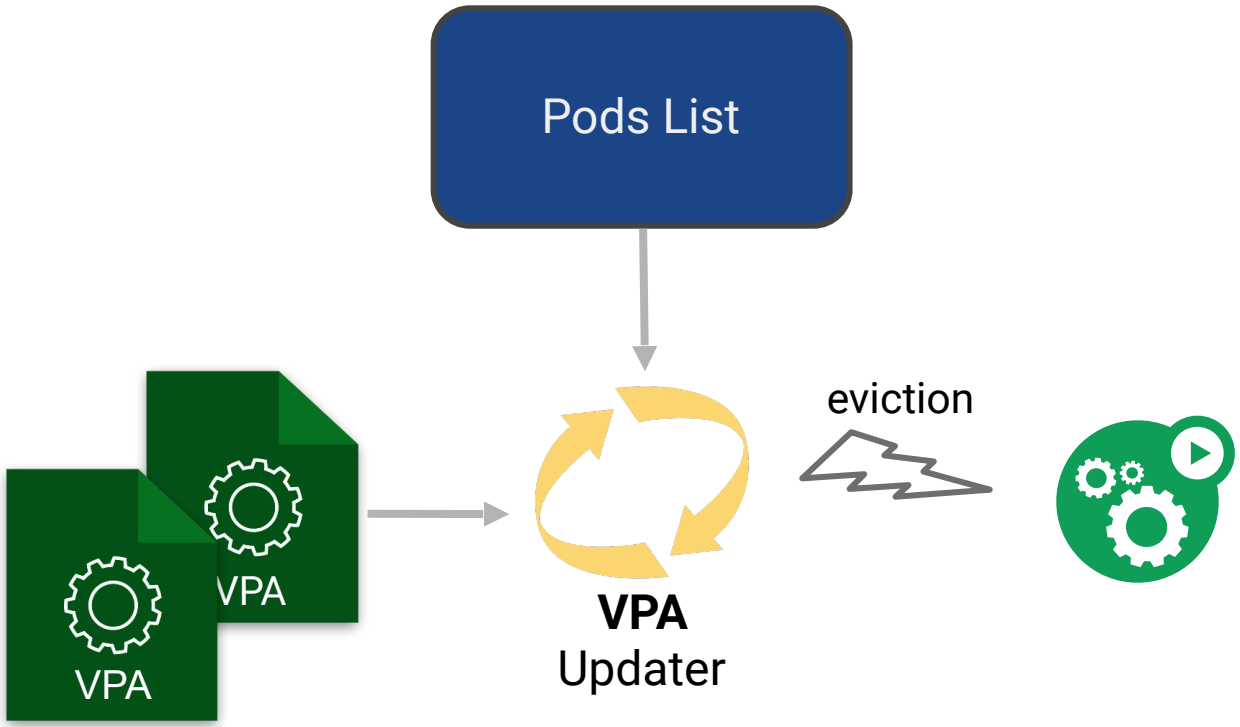


Apply recommendations



Restart running workloads

EXPERIMENTAL



Project status

Status & Plans

2018/04	VPA Alpha
2018/12	VPA Beta

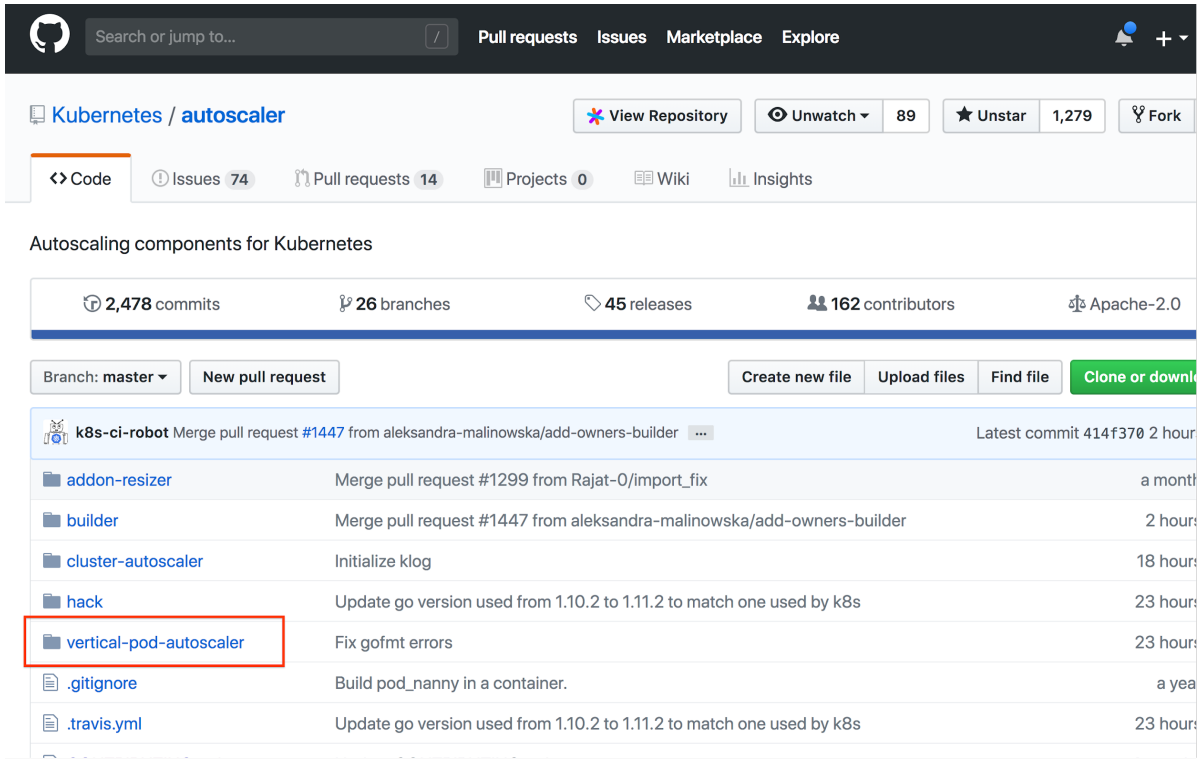
Waiting for your feedback!

Where is the limit?

Can you not restart my workloads?

Check it out!

https://github.com/kubernetes/autoscaler/



KubeCon
CloudNativeCon
North America 2018
Thank You