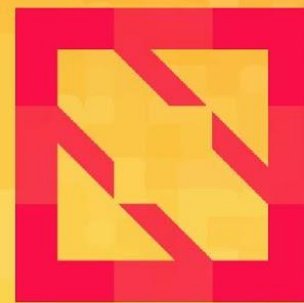




KubeCon



CloudNativeCon

North America 2019



KubeCon



CloudNativeCon

North America 2019

Staying in Tune: Optimize Kubernetes for Stability and Utilization

Randy Johnson, Field Engineer <jrandy@vmware.com>

Koushik Radhakrishnan, Field Engineer <radhakrishnk@vmware.com>

Why Kubernetes?

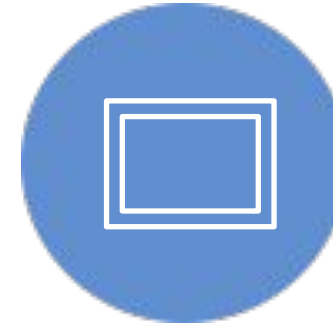
Kubernetes has potential



Declarative API
Software Interfacing with
software



Self Healing
Autonomously
converging on building
blocks



Bin Packing
Approximation with
performance guarantee

How it actually feels...



Goals

Stability

Utilization



Developer



IT Ops

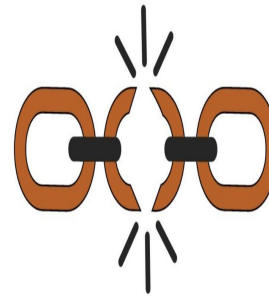
Challenges



Kubernetes isn't configured for our app



Configuration is especially important as utilization increases



Disruptions occur, how can we maintain stability?



The "right" solution is a moving target

Now what?

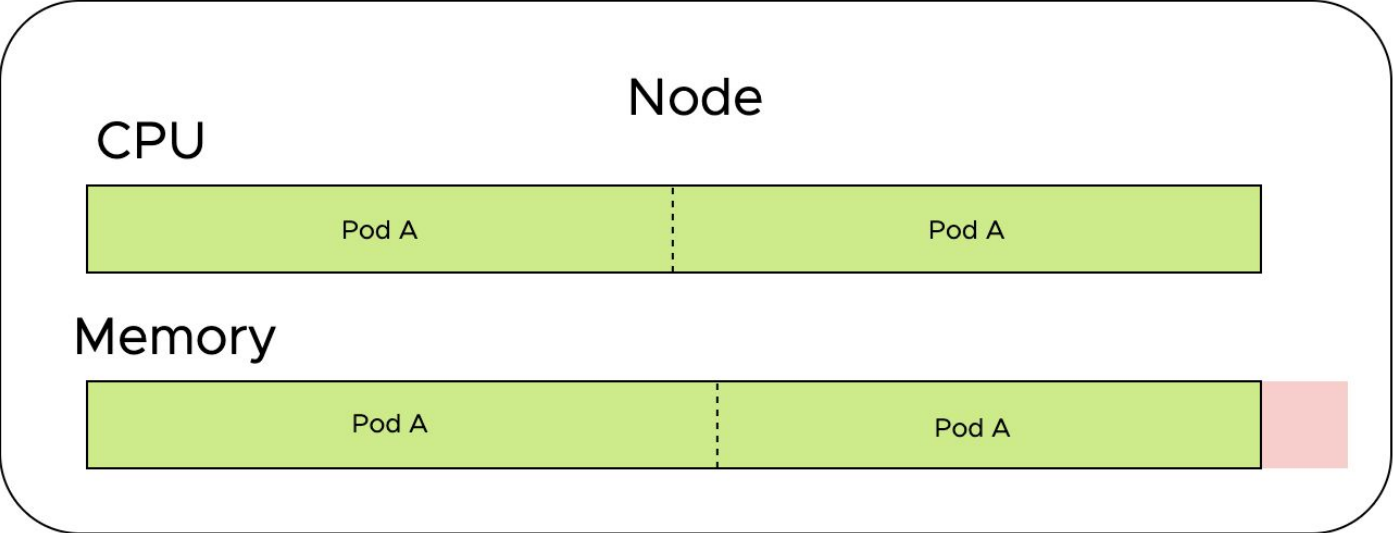
Limits and Requests

- Scheduling
- Overcommit
- Eviction

Eviction Policies

- Defaults
- Allocatable
- Fault Tolerance

Unbounded Resource Consumption



```
resources:
  limits:
    cpu:
    memory:
  requests:
    cpu:
    memory:
```

Pod A

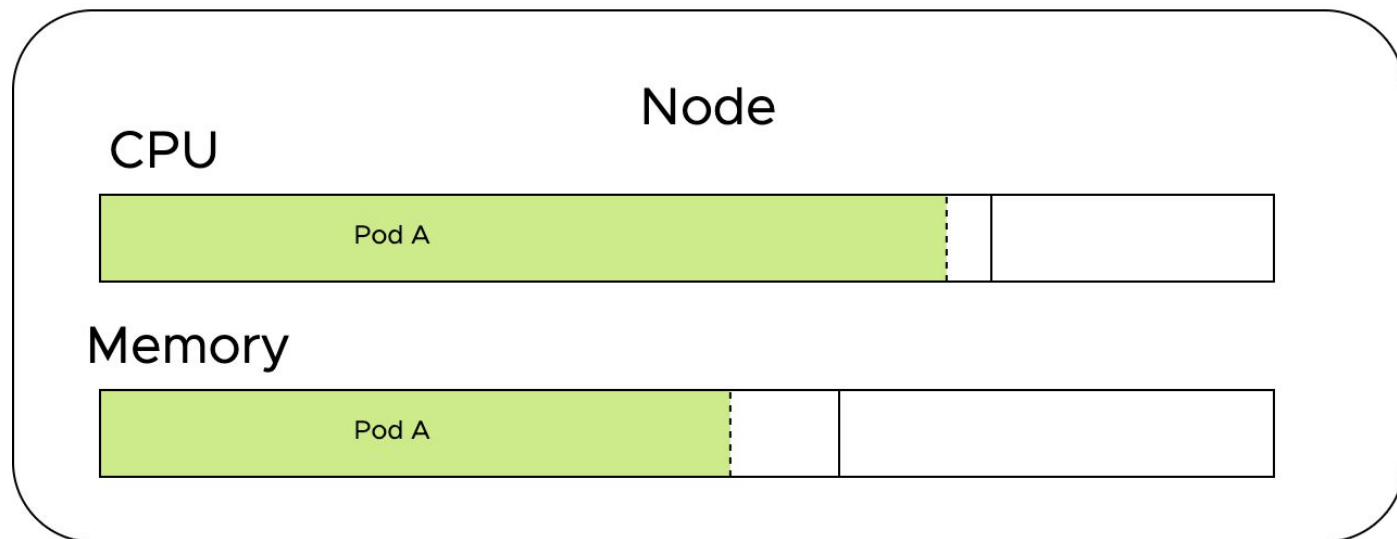
```
resources:
  limits:
    cpu:
    memory:
  requests:
    cpu:
    memory:
```

Pod A

Incompressible, like water



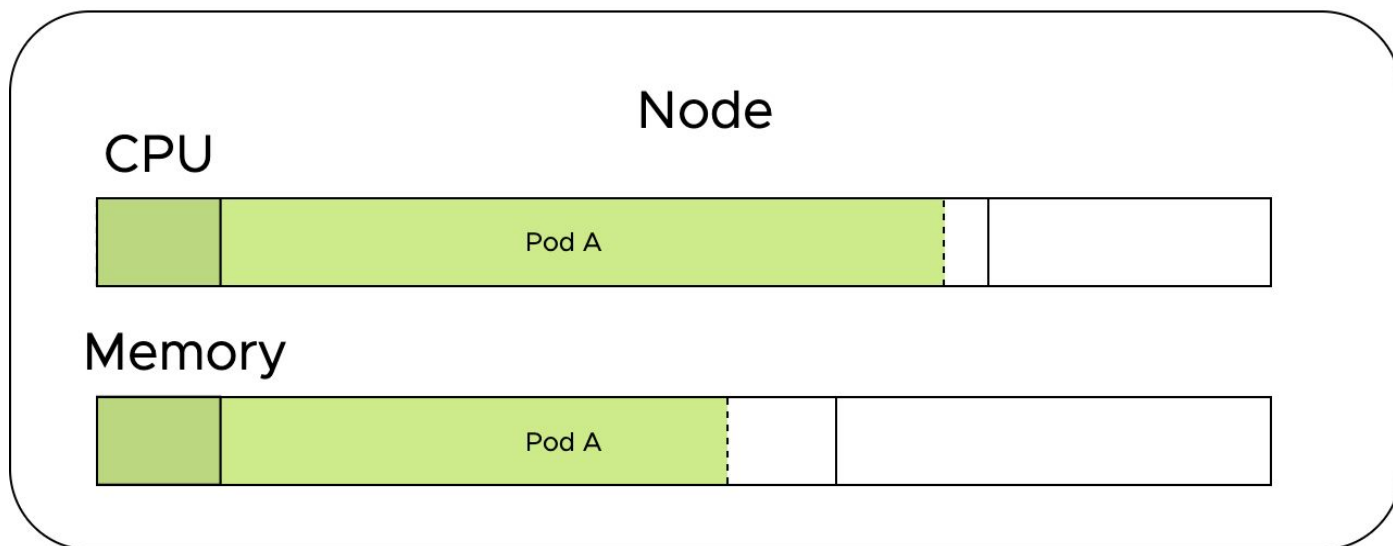
Limits



```
resources:  
  limits:  
    cpu: "700m"  
    memory: "600Mi"  
  requests:  
    cpu:  
    memory:
```

Pod A

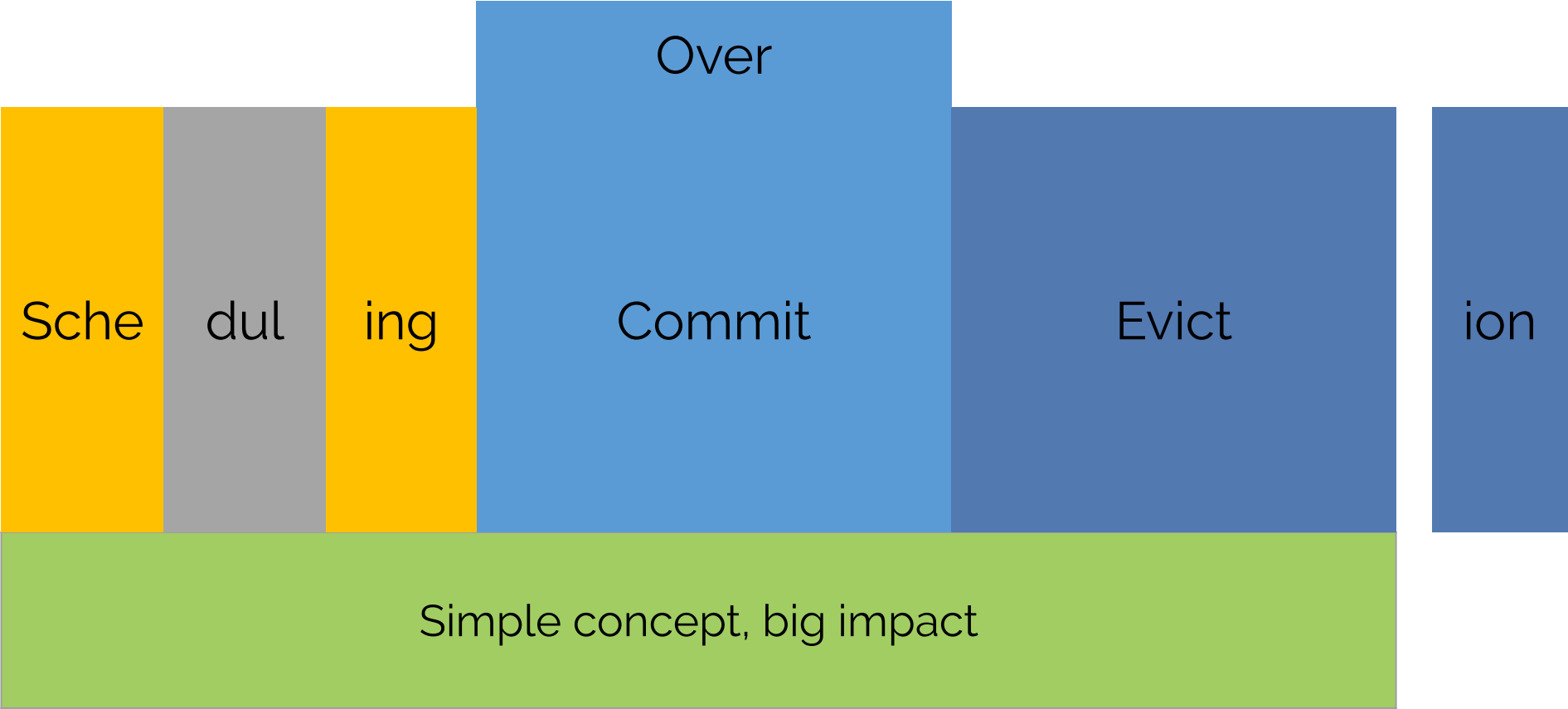
Requests



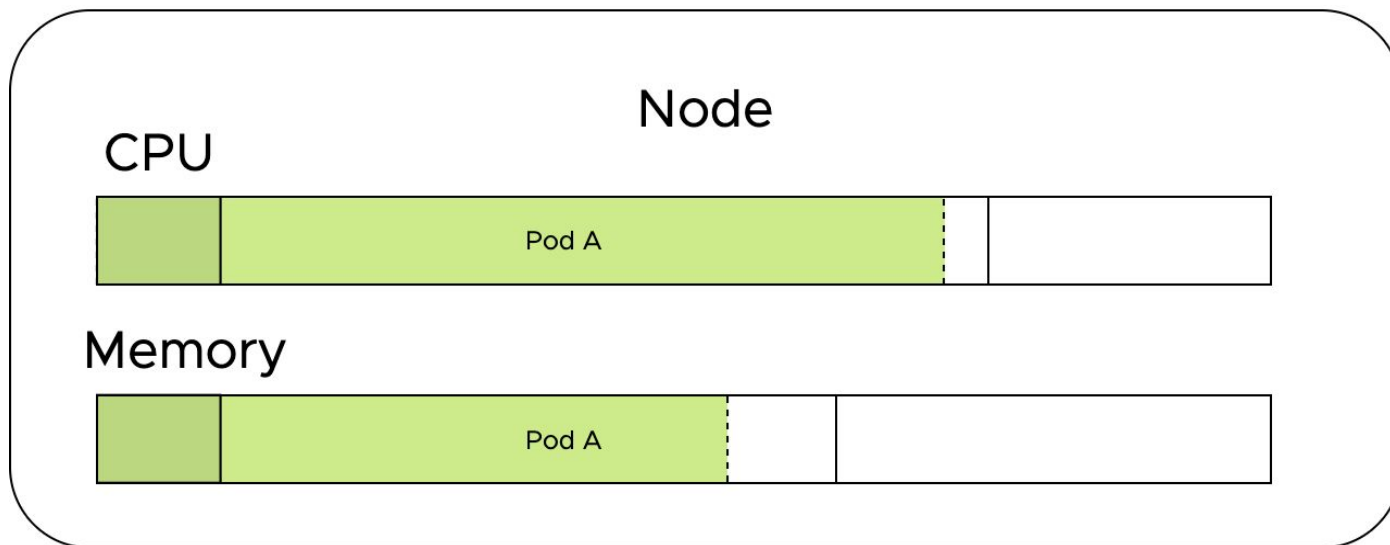
```
resources:  
  limits:  
    cpu: "700m"  
    memory: "600Mi"  
  requests:  
    cpu: "100m"  
    memory: "100Mi"
```

Pod A

Limits and Requests



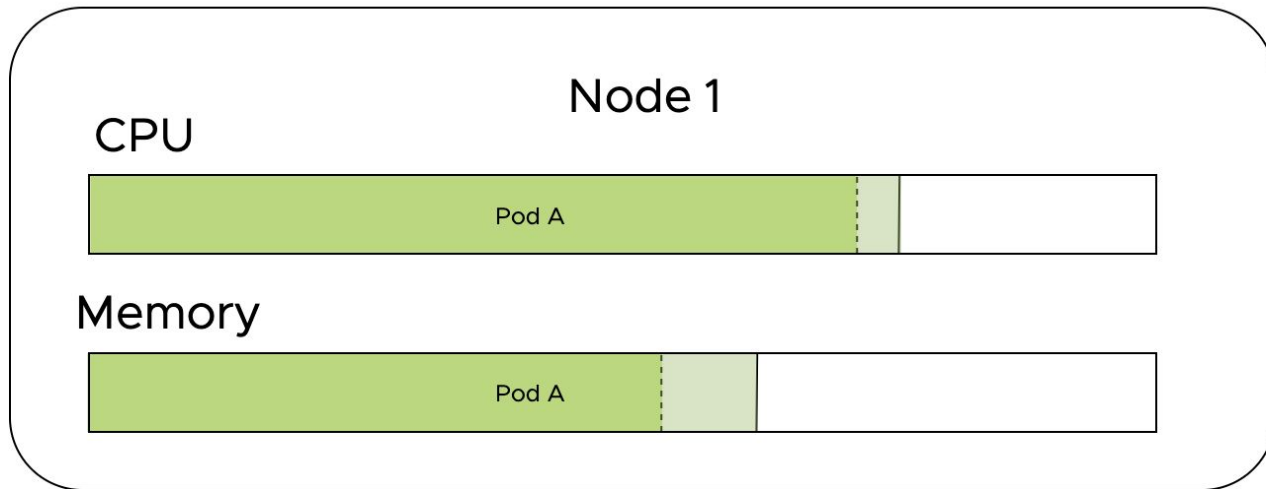
Scheduling



```
resources:  
  limits:  
    cpu: "700m"  
    memory: "600Mi"  
  requests:  
    cpu: "100m"  
    memory: "100Mi"
```

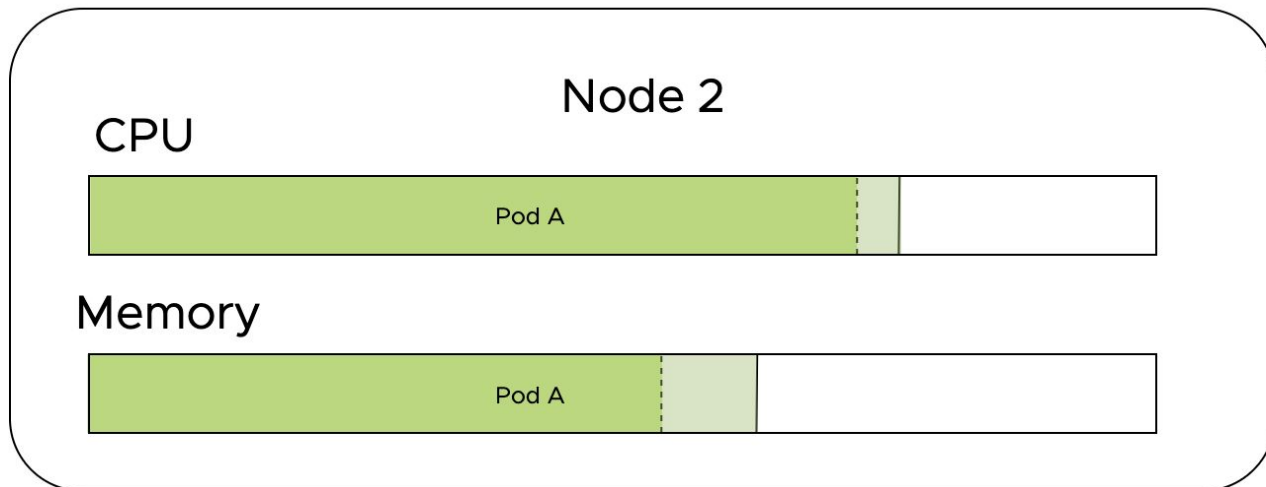
Pod A

Scheduling



```
resources:  
  limits:  
    cpu: "700m"  
    memory: "600Mi"  
  requests:  
    cpu: "700m"  
    memory: "600Mi"
```

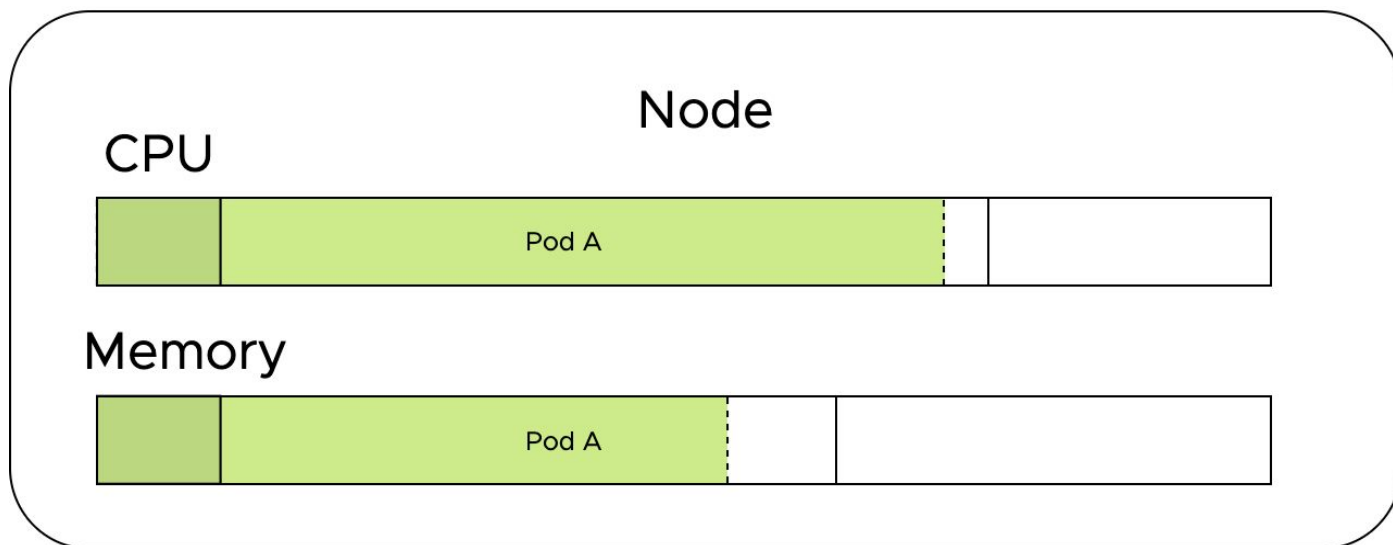
Pod A



```
resources:  
  limits:  
    cpu: "700m"  
    memory: "600Mi"  
  requests:  
    cpu: "700m"  
    memory: "600Mi"
```

Pod A

Overcommit



```
resources:  
  limits:  
    cpu: "700m"  
    memory: "600Mi"  
  requests:  
    cpu: "100m"  
    memory: "100Mi"
```

Pod A



KubeCon



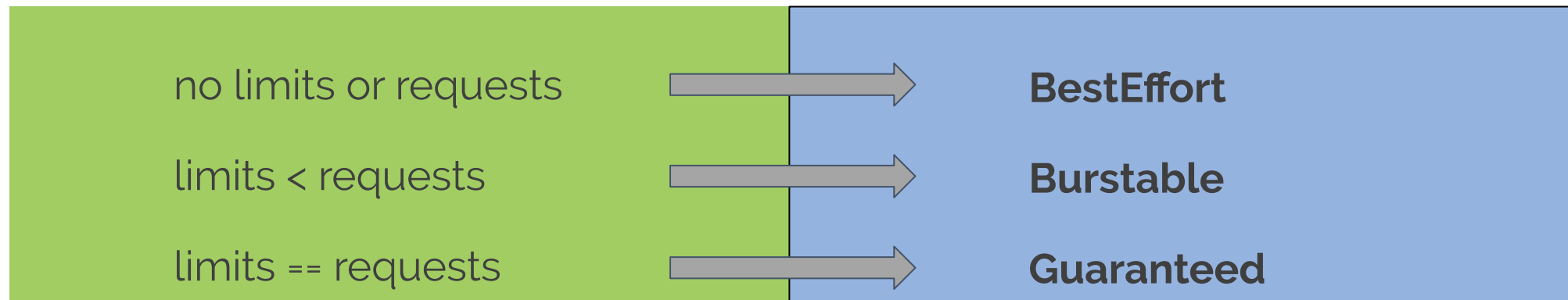
CloudNativeCon

North America 2019

QOS, Eviction Thresholds and Resource Quotas

Quality of Service

Examples so far



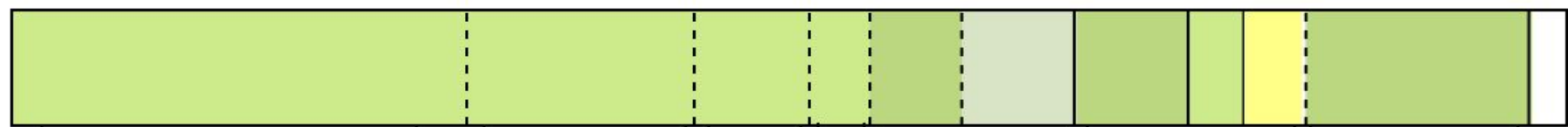
Eviction

Node - memoryPressure = true

CPU



Memory



A

B

C

D

E

F

G

Eviction

Pod	QoS	Priority	Utilization	Usage/ Request	Eviction Order
A	BestEffort	3	20%	N/A	3
B	BestEffort	3	10%	N/A	4
C	BestEffort	2	5%	N/A	2
D	BestEffort	1	2.5%	N/A	1
E	Burstable	2	N/A	0.5	6
F	Burstable	2	N/A	2	5
G	Guaranteed	1	N/A	1	7

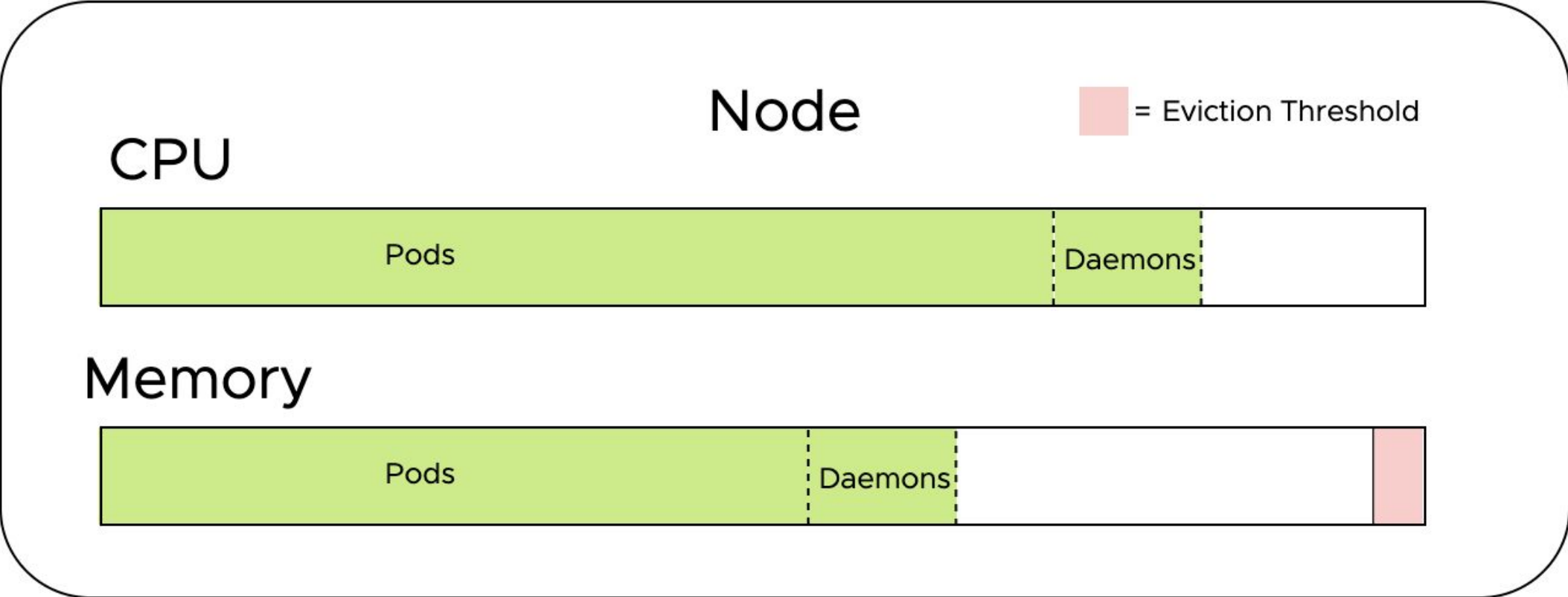
LimitRanges

- A *LimitRange* object enforces:
 - minimum
 - maximum
 - ratio
 - default

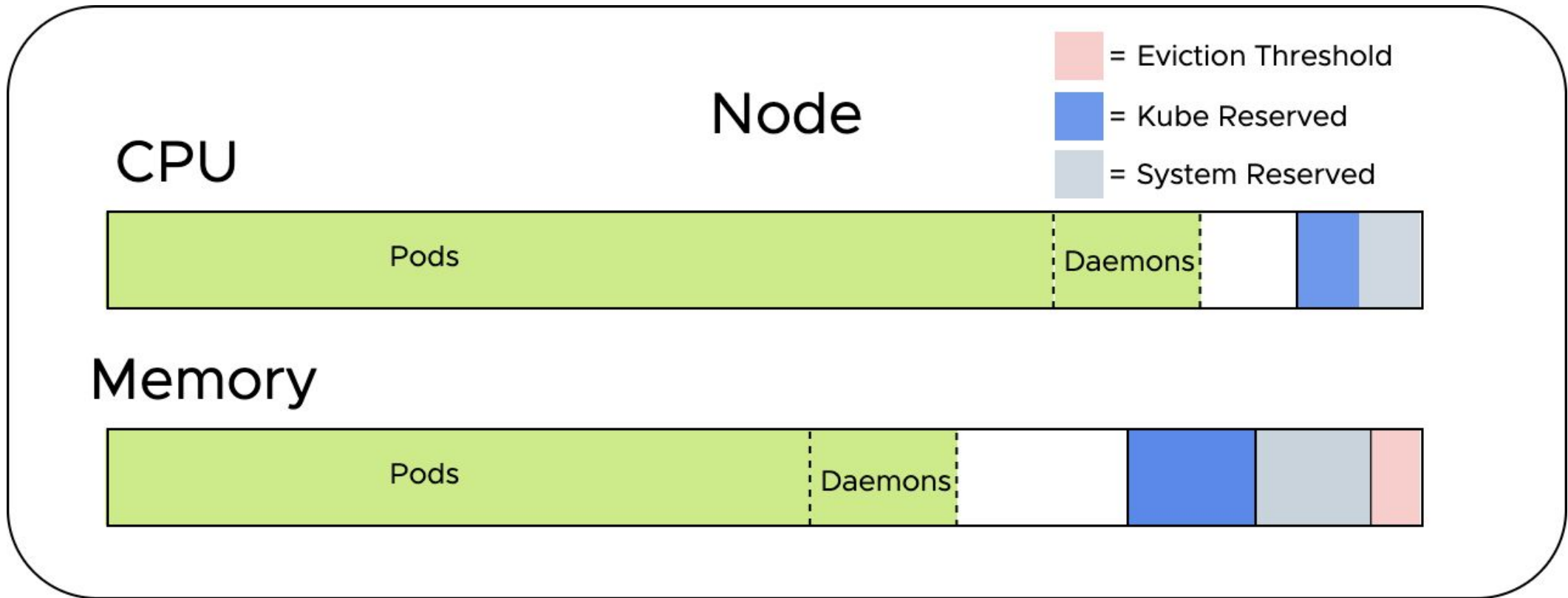
Resource Quotas

- A *ResourceQuota* enforces aggregate limits at the namespace level

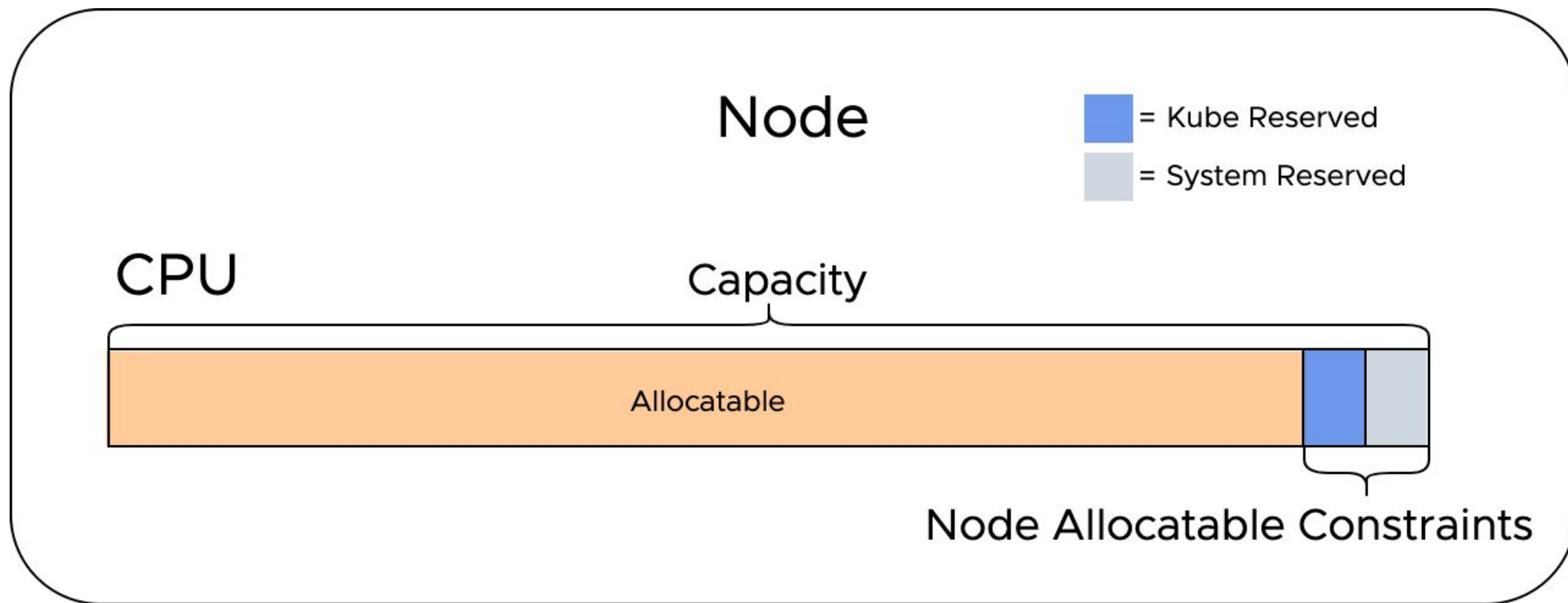
Eviction Thresholds



Kube & System Reserved



Allocatable CPU



Allocatable Memory & Ephemeral Storage

Node

\$resource

Allocatable

Allocatable Constraints



Kubelet Defaults

- Default flags to be aware of
 - `--eviction-hard=memory.available<100Mi`
 - `--eviction-pressure-transition-period=5m`
 - `--housekeeping-interval=10s`
 - `--enforce-node-allocatable=pods`
- Flags to consider
 - `--eviction-soft`
 - `--eviction-soft-grace-period`
 - `--system-reserved`
 - `--kube-reserved`

Node Allocatable

Node 1

\$resource



Node 2

\$resource



Node 3

\$resource



Key Metrics

- How well am I assigning requests?
 - utilization / request
- How well am I assigning memory limits?
 - container_restarts
- How close am I to triggering pod eviction?
 - allocatable - pod_utilization > 0
 - capacity - eviction_threshold - pod_utilization - daemon_utilization > 0
- What if I frequently encounter eviction caused by
 - utilizing allocatable?
 - Tune limits & requests to adjust overcommit
 - crossing an eviction threshold?
 - Tune kube-reserved & system-reserved

Max Node Utilization

$$\mathit{maxUtilization}(n, f, c) = \frac{n - f - c}{n}$$

for $n \geq 2, f \geq 0, c \geq 0$

Staying in Tune: Optimize Kubernetes for Stability and Utilization

Tell us your experience



KubeCon



CloudNativeCon

North America 2019

Randy Johnson, Field Engineer
<jrandy@vmware.com>

Koushik Radhakrishnan, Field Engineer
<radhakrishnk@vmware.com>

