



DATADOG

100% Numérique

Telecom
Valley

SophiaConf

Le cycle azurée de conférences et workshops Open Source

Autoscaling in Kubernetes: HPA, WPA, ClusterAutoscaler

Par David BENQUE

■ 30 Juin | 18h20 ■

Gratuit sur inscription www.sophiaconf.fr





Why do we want to scale?

Up: to cope with demand

Down: to save money

Why do we want to autoscale?

so you can sleep sometimes

because we have fat fingers and we break things



Why do we want to scale?

Up:

to cope with demand

Down:

to save money

The difficult part

Why do we want to autoscale?

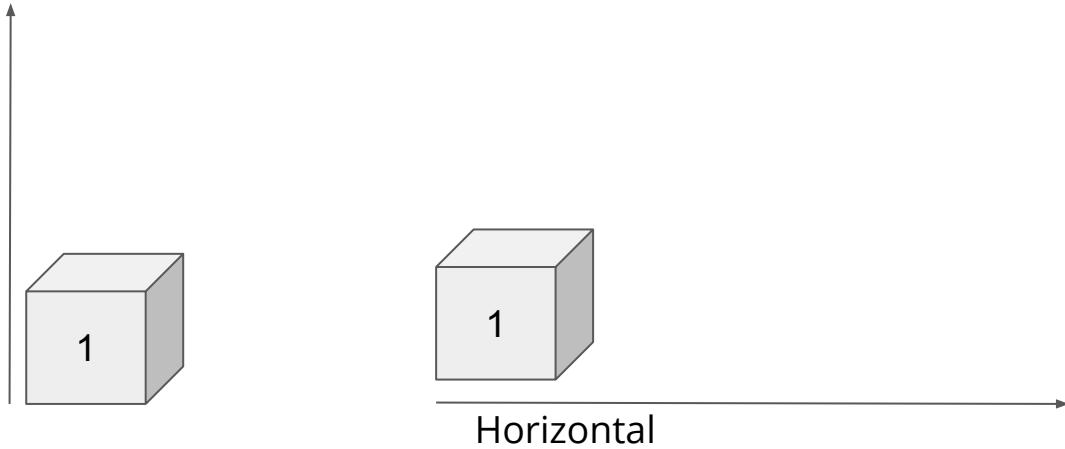
so you can sleep sometimes

The easy part

because we have fat fingers and we break things

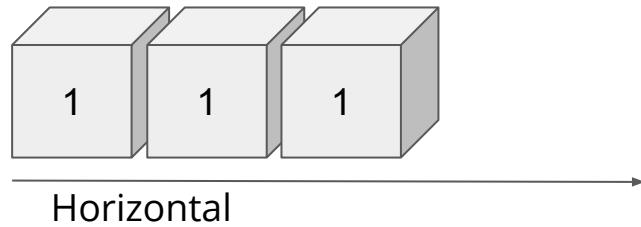
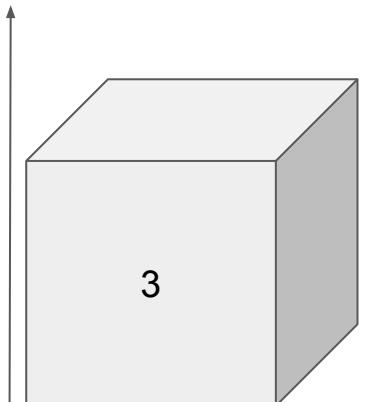
Scaling types

Vertical



Scaling types

Vertical





Scaling resources



kubernetes



pod

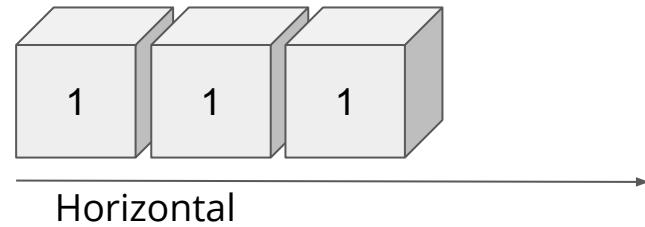
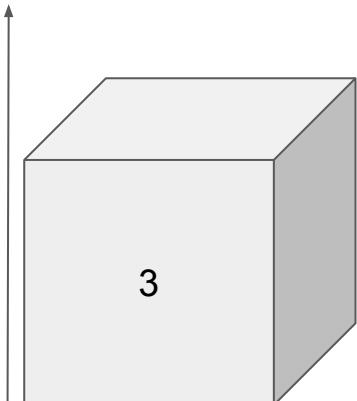


node



Cluster

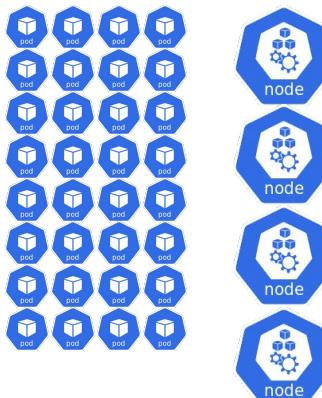
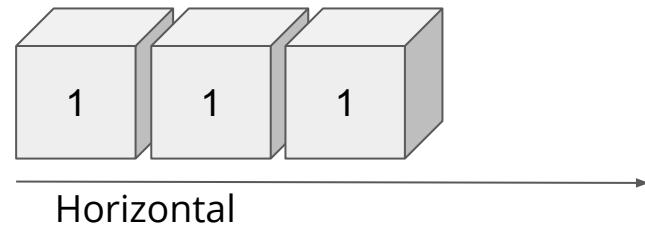
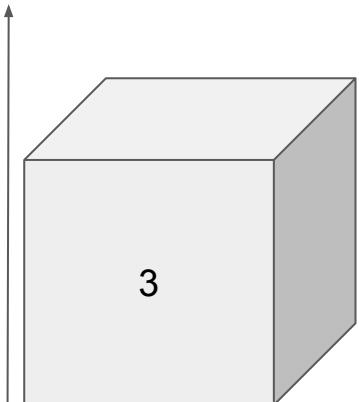
Vertical





Cluster vertical scaling

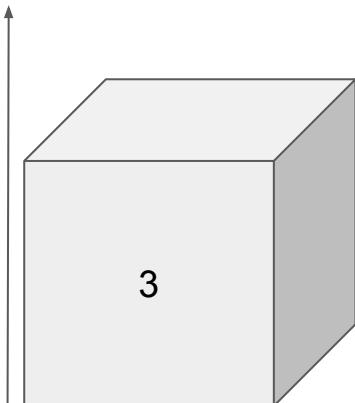
Vertical



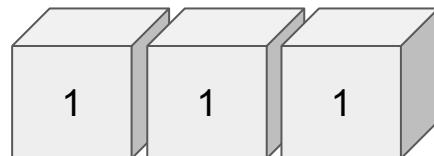


Cluster vertical scaling

Vertical



Horizontal



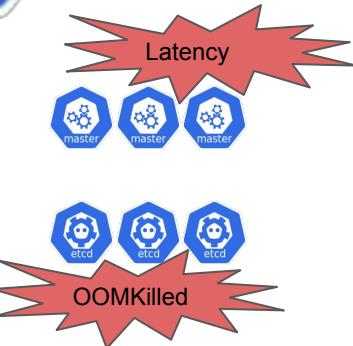
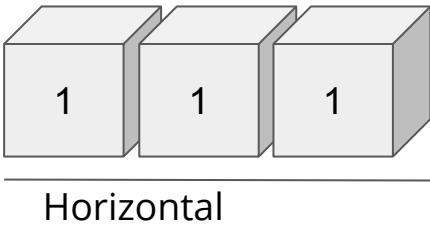
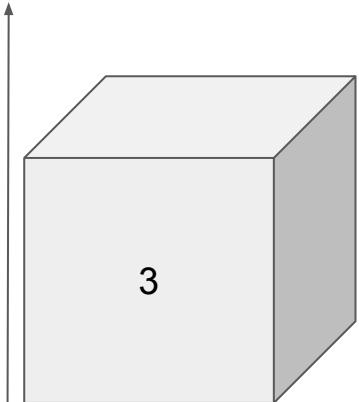
kubernetes





Cluster vertical scaling

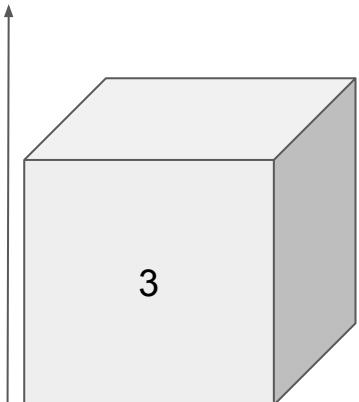
Vertical



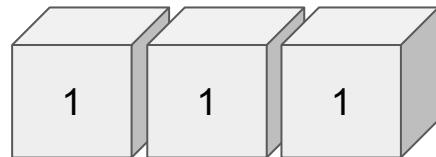


Cluster vertical scaling

Vertical



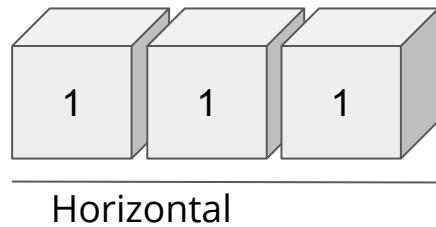
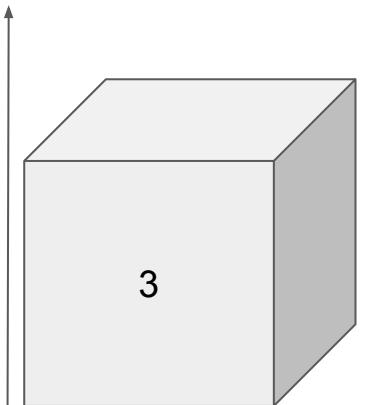
Horizontal





Cluster vertical scaling

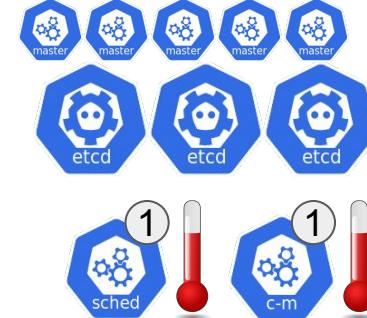
Vertical



Horizontal

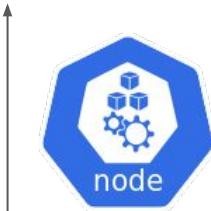


kubernetes



Vertical scaling

Vertical



change instance type:

c4.xlarge → c4.2xlarge



change resources request (and limits) in the pod definition:

resources:
limits:
cpu: 500m
memory: 500Mi
requests:
cpu: 500m
memory: 500Mi



resources:
limits:
cpu: 2
memory: 2Gi
requests:
cpu: 2
memory: Gi

Vertical scaling

Vertical



change instance type:

c4.xlarge → c4.2xlarge

**apply only on new
nodes/pods**



change resources request (and limits) in the pod definition:

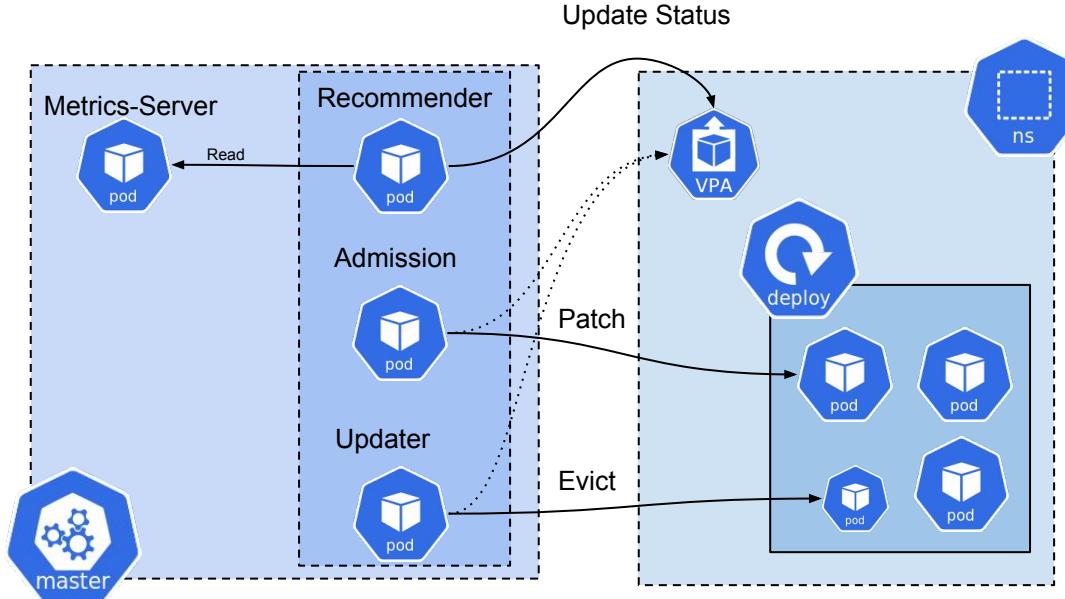
resources:
limits:
cpu: 500m
memory: 500Mi
requests:
cpu: 500m
memory: 500Mi



resources:
limits:
cpu: 2
memory: 2Gi
requests:
cpu: 2
memory: 2Gi

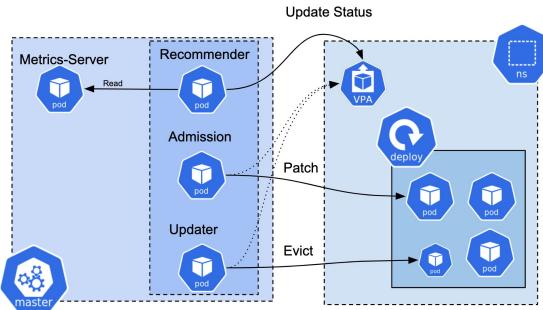


Vertical Pod Autoscaler (VPA)





Vertical Pod Autoscaler



```
Recommendation:  
Container Recommendations:  
Container Name: hamster  
Lower Bound:  
  Cpu:      550m  
  Memory:   262144k  
Target:  
  Cpu:      587m  
  Memory:   262144k  
Uncapped Target:  
  Cpu:      587m  
  Memory:   262144k  
Upper Bound:  
  Cpu:      21147m  
  Memory:  387863636
```

Comments:

- kubernetes native support
- pod admission !!!
- per container recommendation
- historical data depth
- update frequency → disruption/eviction
- eviction
 - pod disruption budget (PDB)
 - not suitable for stateful application
 - pending pod
- maintain ratio requests/limits

Configuration:

- dedicated VPA resource
- target deployment
- min/max range
- dry-run mode:
 - updatePolicy:
 - updateMode: "Off"



Vertical scaling on nodes, how ?

Vertical



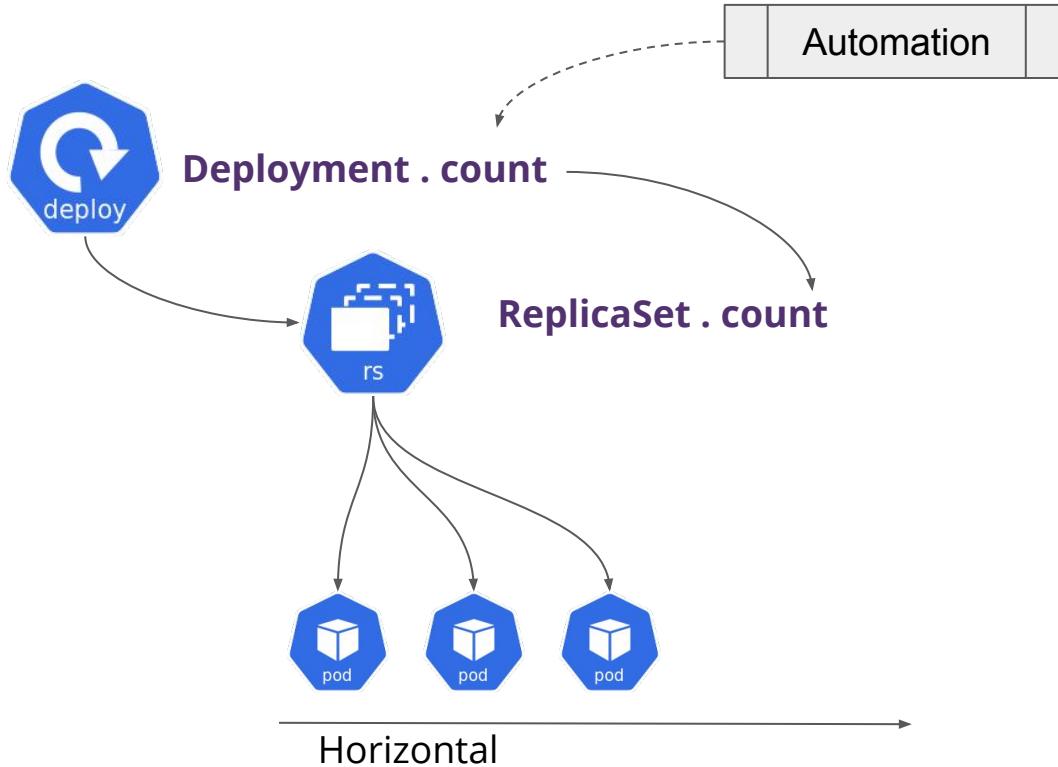
change instance type:

c4.xlarge → c4.2xlarge

How do you provision your nodes ?



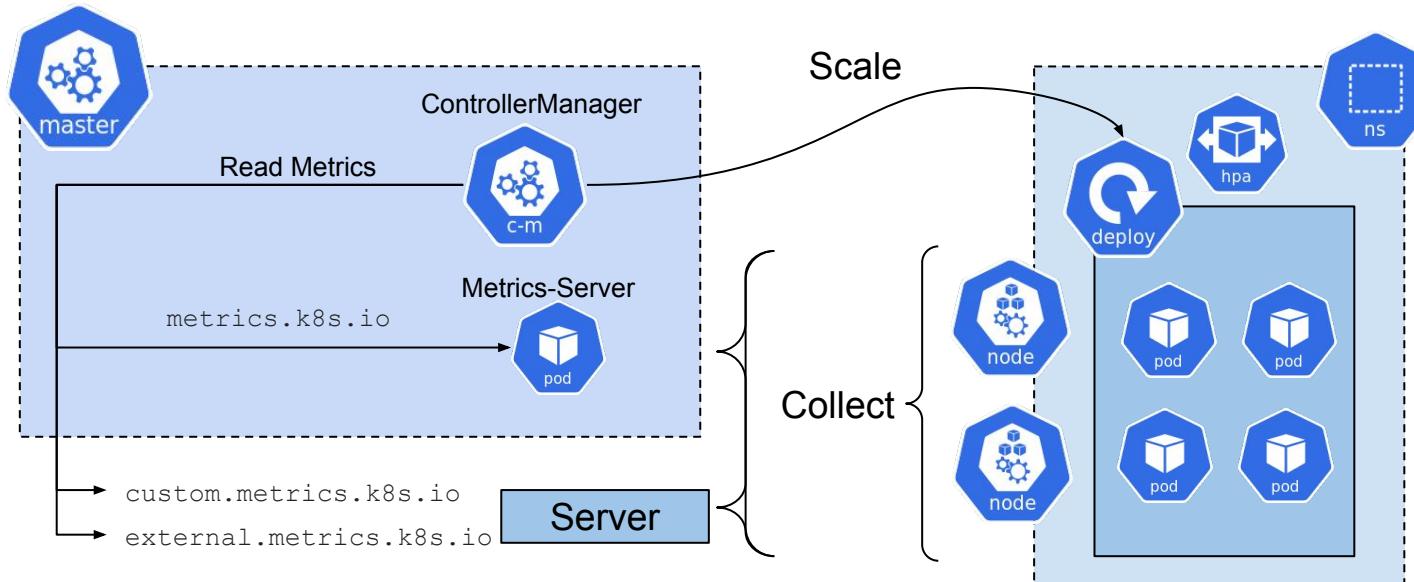
Horizontal scaling for pods





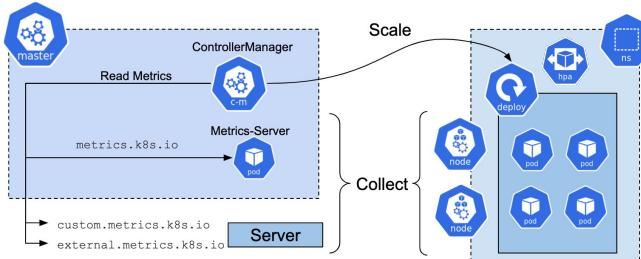
Horizontal Pod Autoscaler (HPA)

HPA runs inside ControllerManager





Horizontal Pod Autoscaler (HPA)



Comments:

- **kubernetes native support**

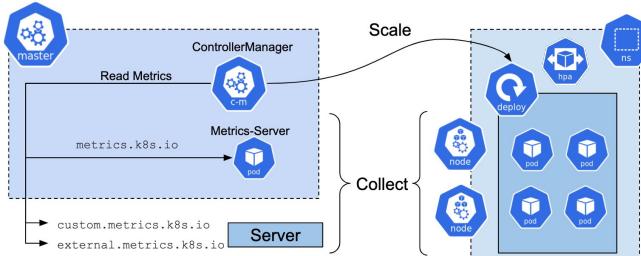
```
desiredReplicas = ceil[currentReplicas * (currentMetricValue / desiredMetricValue)]
```

Configuration:

- **dedicated HPA resource**
- **min/max range**
- **different source of metrics**
- **forbiddenWindow defined at controller level**
 - Improved in 1.18 (1)
- **tolerance defined at controller level (10%)**

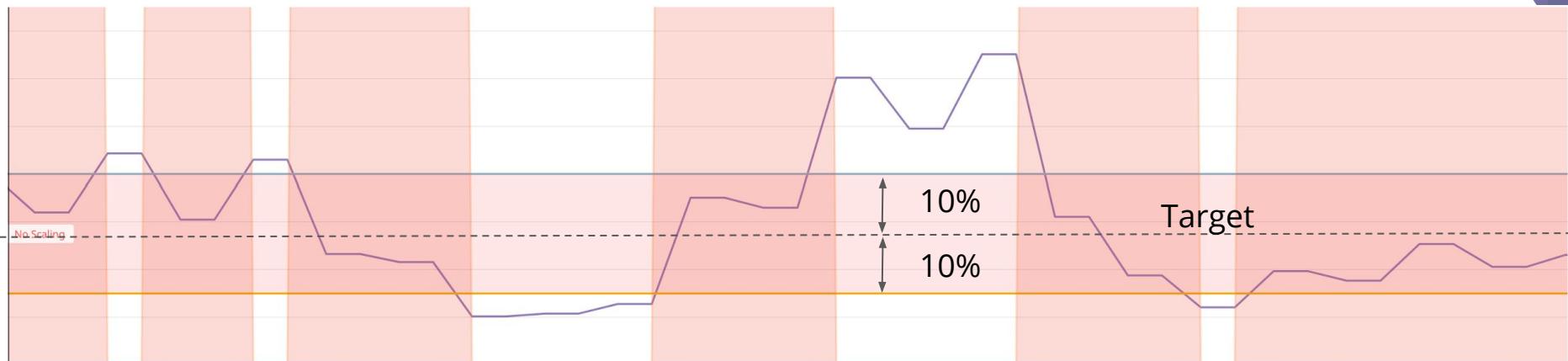


Horizontal Pod Autoscaler (HPA)

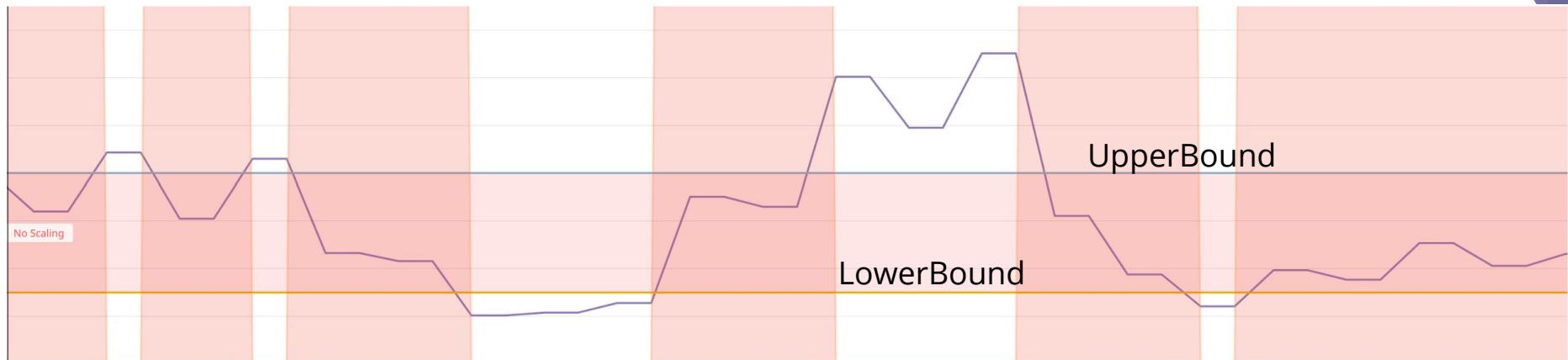


tolerance is 10% for all HPAs in the cluster

target is defined for each HPA

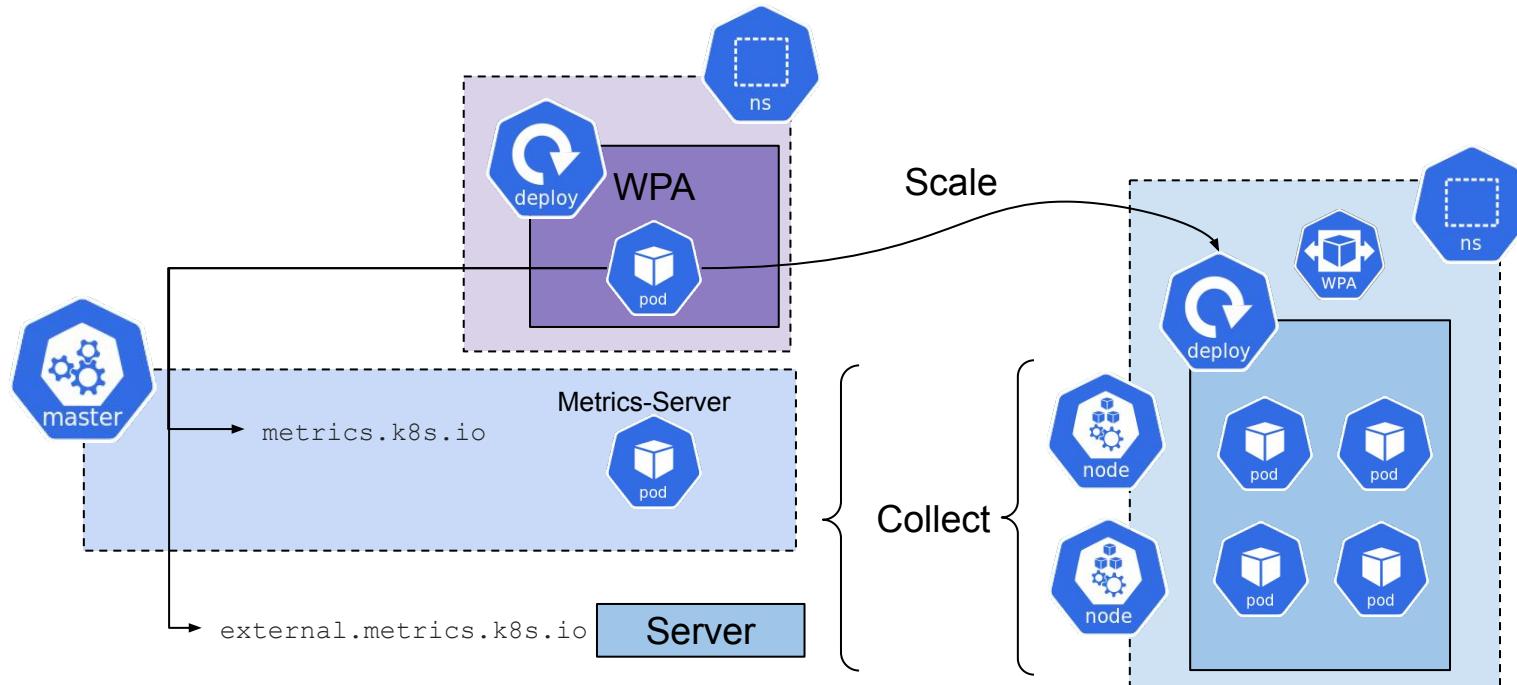


bounds are defined for each WPA



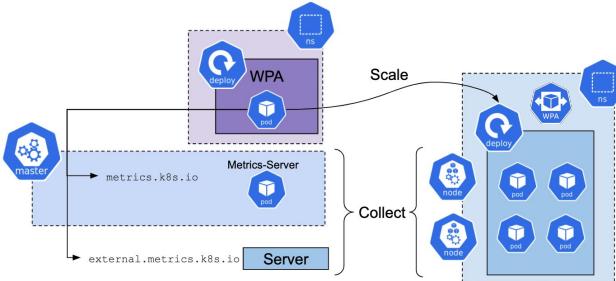


Watermark Pod Autoscaler (WPA) by





Watermark Pod Autoscaler (WPA) by



Comments:

- opensource project by Datadog
- Can be used even without Datadog product

Configuration:

- dedicated WPA resource
- same as HPA with more:
 - tolerance replaced by upper/lower bound
 - velocity

c: current number of replicas
x: sum of external metrics provider
h,l: HighWaterMark, LowWaterMark
y: computed number of replicas

$$\forall x \in \mathbb{R}^+, c \in \mathbb{N}^*, \{l, h\} \in \mathbb{R}^2 [l < h], \exists ! y \in \mathbb{N}^* /$$
$$\begin{cases} x < l < h \implies y = \lfloor \frac{x*c}{l} \rfloor \\ l < x < h \implies y = c \\ l < h < x \implies y = \lceil \frac{x*c}{h} \rceil \end{cases}$$



What about ?

serveless...

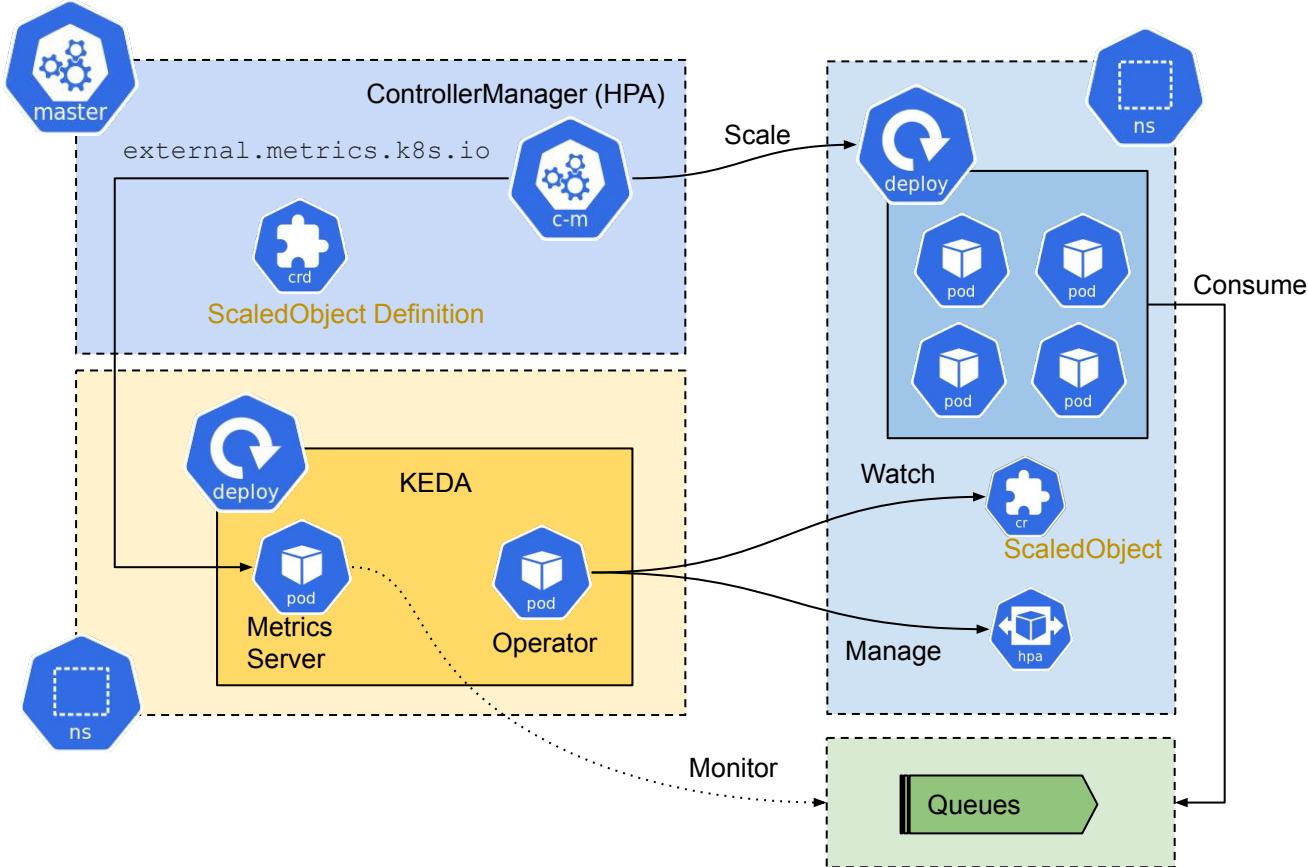
scale-down to 0...

CPU when 0 instance ...

event driven architectures ...

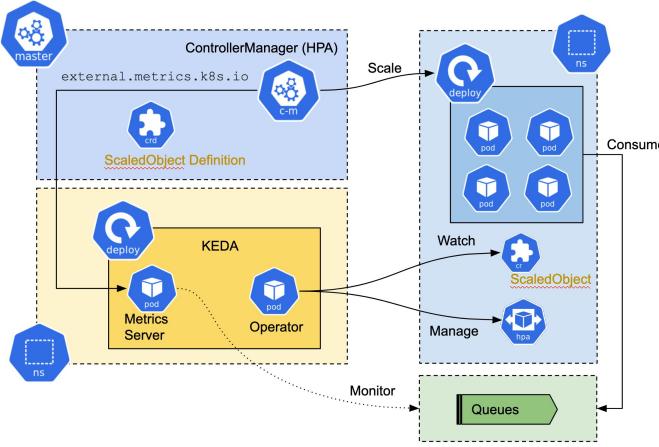


Kubernetes-based Event Driven Autoscaling





Kubernetes-based Event Driven Autoscaling



Comments:

- opensource project: www.keda.sh
- can create jobs (long running scenario)

Configuration:

- dedicated **ScaledObject** resource
- **19 built-in scalers**
 - kafka
 - redis
 - rabbitMQ
 - NATS
 - AWS (Kinesis Stream, SQS Queue, CloudWatch)
 - Azure (Event Hubs, Monitor, Storage Queue, Blob Storage ...)



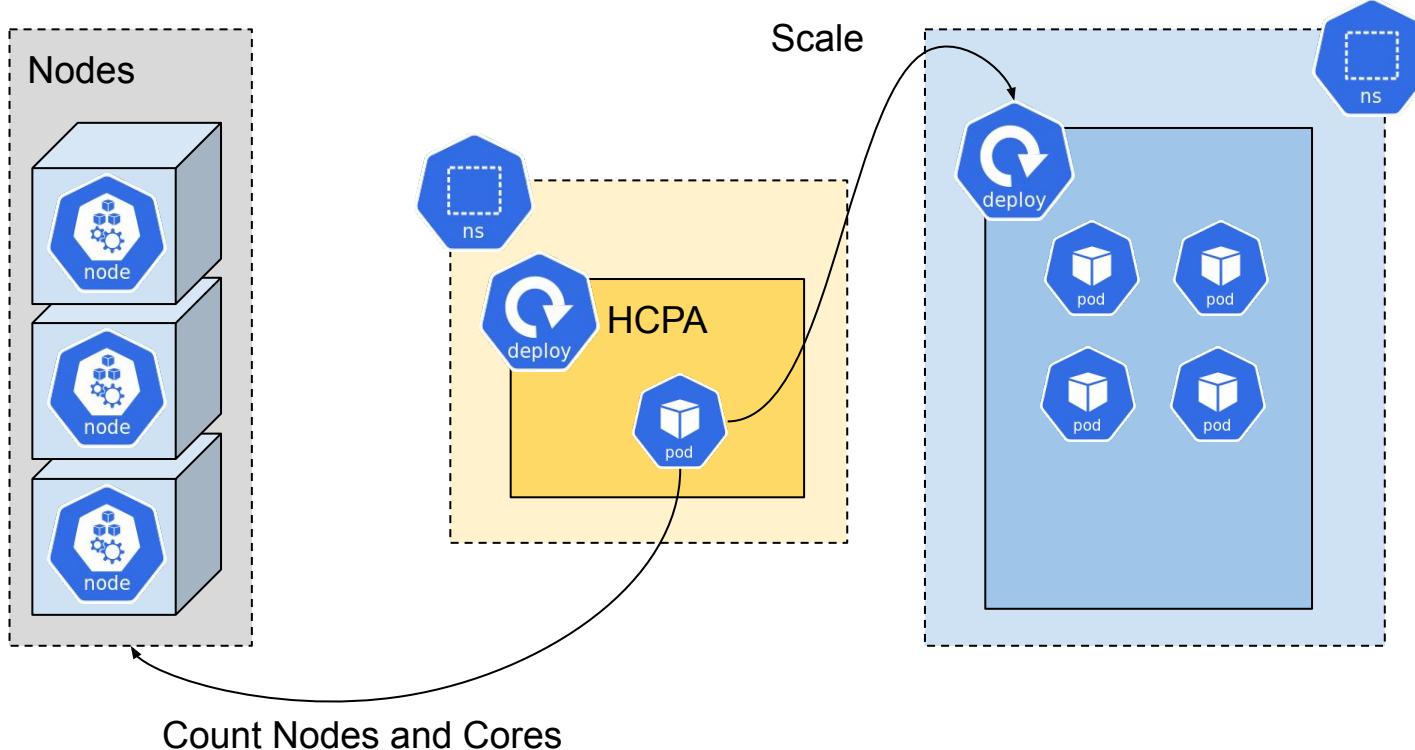
What about ?

cluster size...

growing with number of nodes...

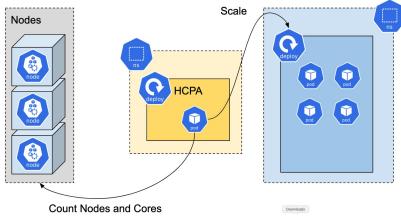


Horizontal cluster-proportional-autoscaler





Horizontal cluster-proportional-autoscaler



```
replicas = max(  
    ceil( cores * 1/coresPerReplica ) ,  
    ceil( nodes * 1/nodesPerReplica ) )
```

```
replicas = min(replicas, max)  
replicas = max(replicas, min)
```

Comments:

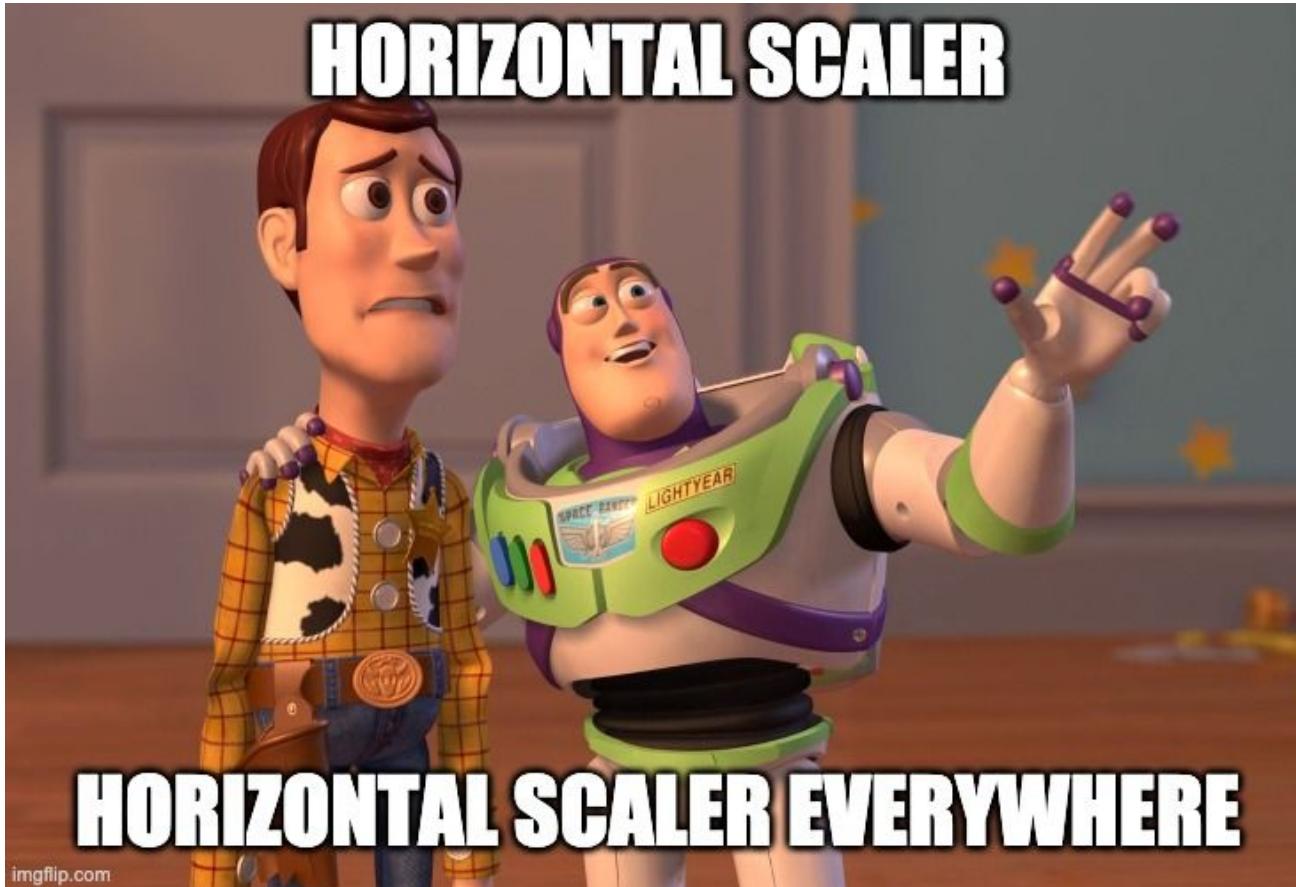
- one HCPA per target deployment
- nodes scoped by labels
- not suitable for general applications
- suitable for cluster application (coredns)

Configuration:

- ConfigMap
- linear (node or core)
- ladder (node or core)
- no timing definition
- includeUnschedulableNodes **recommended**



Any other horizontal scaler for Pods?





Don't, don't *

Vertical Pod Autoscaler

+

Horizontal Pod Autoscaler

=

FUN

*at least not on the same metrics

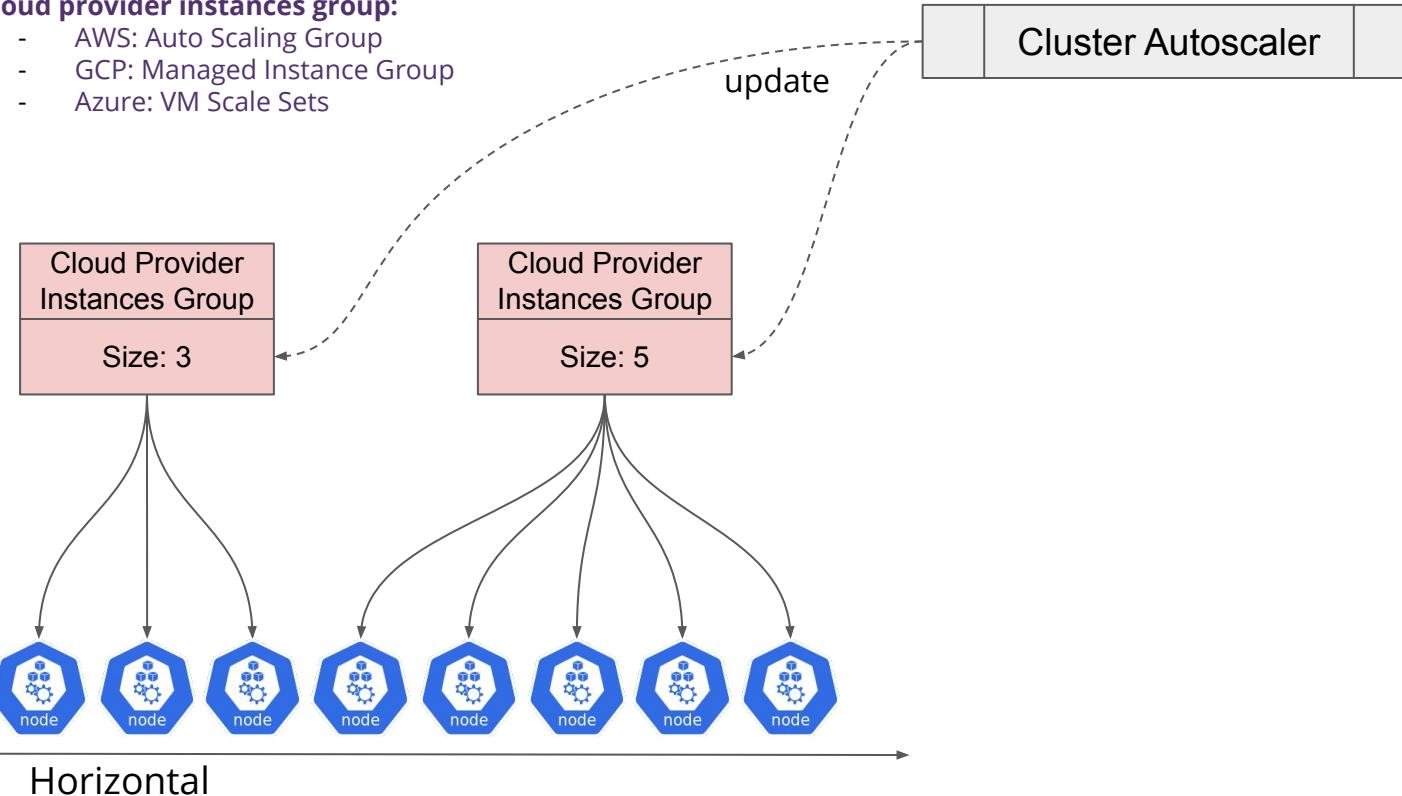




Horizontal scaling for nodes

cloud provider instances group:

- AWS: Auto Scaling Group
- GCP: Managed Instance Group
- Azure: VM Scale Sets

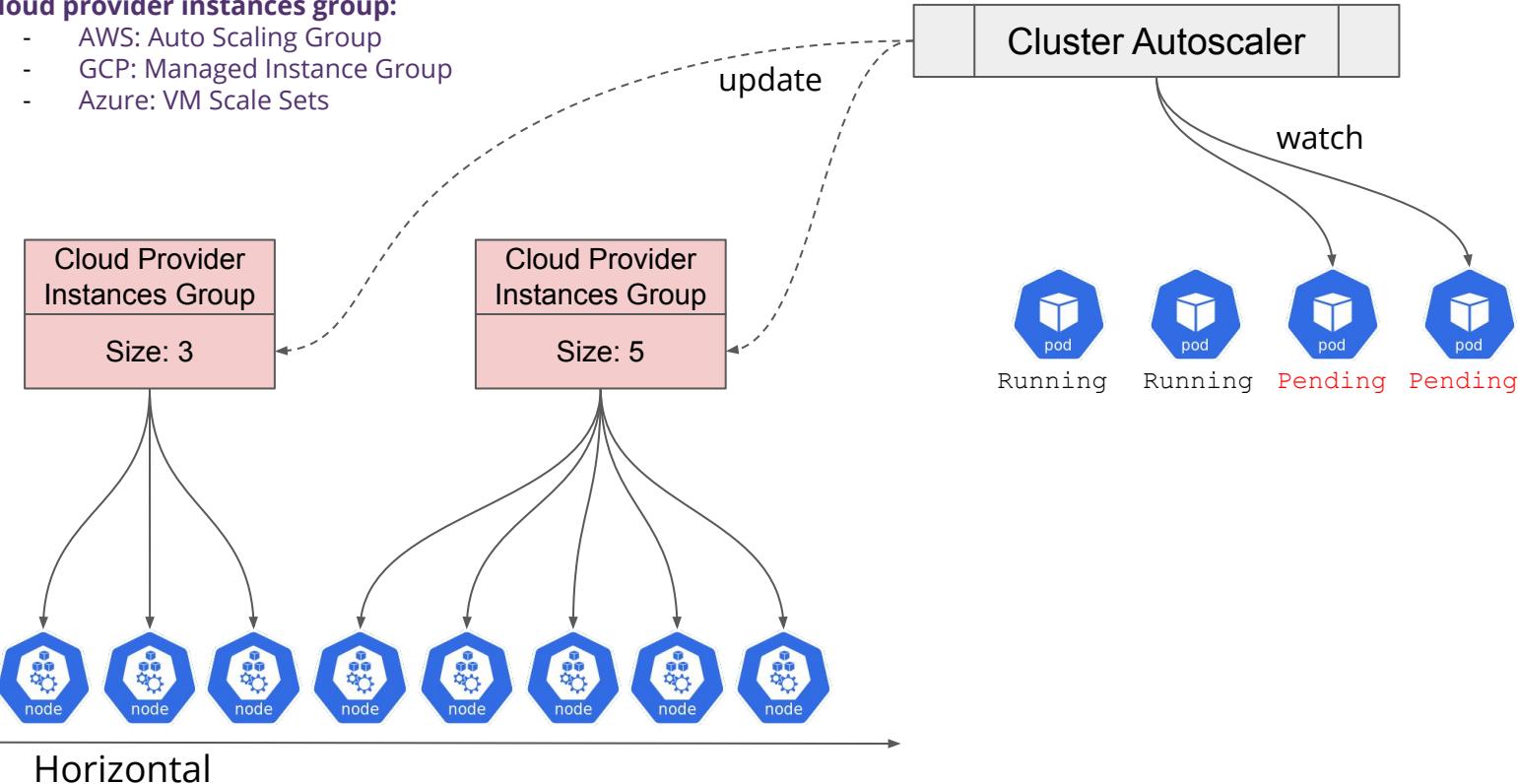




Horizontal scaling for nodes

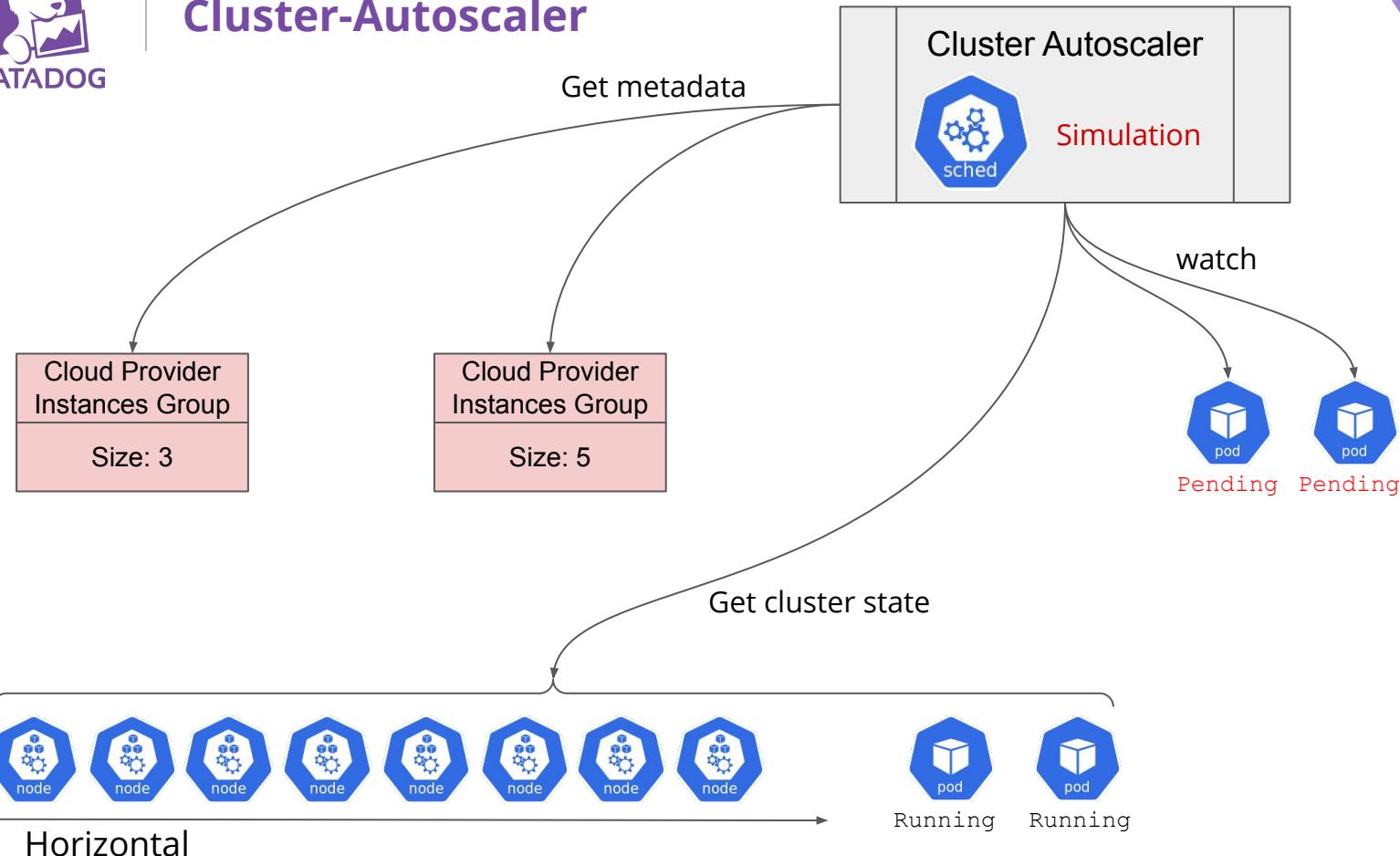
cloud provider instances group:

- AWS: Auto Scaling Group
- GCP: Managed Instance Group
- Azure: VM Scale Sets



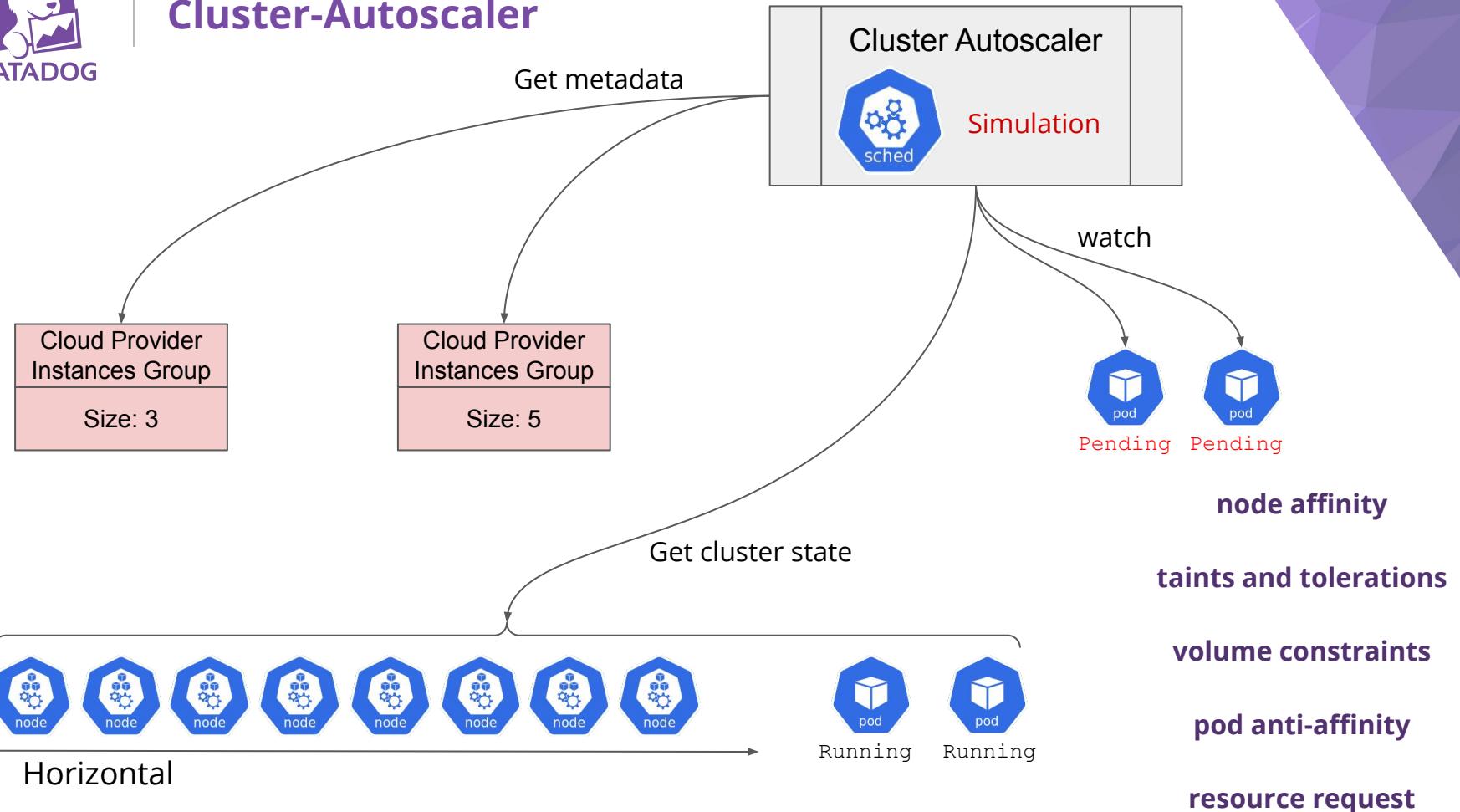


Cluster-Autoscaler





Cluster-Autoscaler





Cluster-Autoscaler events

Index:k8s-audit × Kubernetes Cluster: 1 × User Username:system:serviceaccount:kube-system:cluster-autoscaler × - Resource:configmaps × Verb:create ×

Facets Saved Views Hide Controls 3,829 results found

Search facets

Showing 869 of 1119 Add +

Core Index Source Host Service kube-apiserver-audit 3.83k Status Error 0 Warn 0 Info 3.83k

DATE	RESP...	REQUESTOBJECT	REASON	ESTOBJECT.MESSAGE
Jun 23 14:05:31.190	201	NotTriggerScaleUp	pod didn't trigger scale-up (it wouldn't fit if a new node is added): 502 node(s) had taints that the pod didn't tolerate, 17 max node group size reached	
Jun 23 14:05:09.636	201	TriggeredScaleUp	pod triggered scale-up: [{us1-prod-1-asg-6f7614da5bf 1->2 (max: 256)}]	
Jun 23 14:04:58.821	201	TriggeredScaleUp	pod triggered scale-up: [{us1-prod-1-asg-944fc044f38 0->1 (max: 256)}]	
Jun 23 14:01:13.985	201	NotTriggerScaleUp	pod didn't trigger scale-up (it wouldn't fit if a new node is added): 502 node(s) had taints that the pod didn't tolerate, 17 max node group size reached	
Jun 23 14:01:03.526	201	TriggeredScaleUp	pod triggered scale-up: [{us1-prod-1-asg-9ab63eab069 0->1 (max: 256)}]	
Jun 23 14:00:28.742	201	TriggeredScaleUp	pod triggered scale-up: [{us1-prod-1-asg-ea18569f0450 0->1 (max: 1)}]	
Jun 23 14:00:28.573	201	ScaledUpGroup	Scale-up: setting group us1-prod-1-asg-ea18569f0450 size to 1	
Jun 23 13:52:15.826	201	NotTriggerScaleUp	pod didn't trigger scale-up (it wouldn't fit if a new node is added): 16 max node group size reached, 503 node(s) had taints that the pod didn't tolerate	
Jun 23 13:48:19.594	201	ScaleDown	node removed by cluster autoscaler	
Jun 23 13:48:07.875	201	ScaleDownEmpty	Scale-down: removing empty node ip-	
Jun 23 13:42:03.648	201	NotTriggerScaleUp	pod didn't trigger scale-up (it wouldn't fit if a new node is added): 16 max node group size reached, 503 node(s) had taints that the pod didn't tolerate	
Jun 23 13:40:38.243	201	NotTriggerScaleUp	pod didn't trigger scale-up (it wouldn't fit if a new node is added): 503 node(s) had taints that the pod didn't tolerate, 16 max node group size reached	
Jun 23 13:37:37.251	201	TriggeredScaleUp	pod triggered scale-up: [{us1-prod-1-asg-eeca4666f82c 4->5 (max: 18)}]	
Jun 23 13:37:37.003	201	ScaledUpGroup	Scale-up: setting group us1-prod-1-asg-eeca4666f82c size to 5	
Jun 23 13:35:17.805	201	NotTriggerScaleUp	pod didn't trigger scale-up (it wouldn't fit if a new node is added): 503 node(s) had taints that the pod didn't tolerate, 16 max node group size reached	
Jun 23 13:34:24.921	201	ScaleDown	node removed by cluster autoscaler	
Jun 23 13:30:52.965	201	NotTriggerScaleUp	pod didn't trigger scale-up (it wouldn't fit if a new node is added): 503 node(s) had taints that the pod didn't tolerate, 16 max node group size reached	
Jun 23 13:30:21.109	201	ScaleDown	node removed by cluster autoscaler	
Jun 23 13:30:10.113	201	ScaleDownEmpty	(combined from similar events): Scale-down: removing empty node ip-internal	
Jun 23 13:27:30.179	201	ScaleDown	node removed by cluster autoscaler	

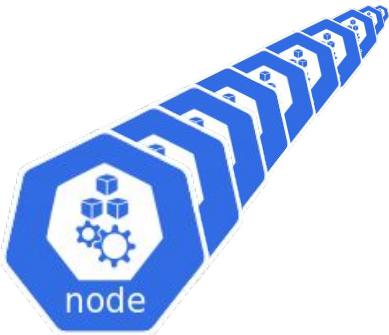
Cluster-Autoscaler NotTriggerScaleUp

The usual suspects:

- max group size reached
- PVC bound to existing node
- Resources does not match ( *I am looking at you*)
- user definition issue:
 - tolerations
 - node selectors



Cluster-Autoscaler Infinite ScaleUp



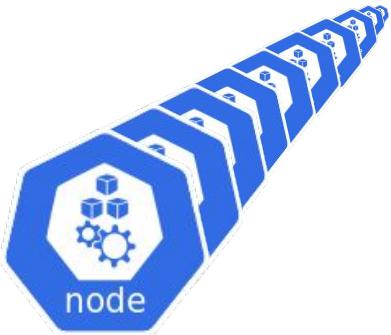
N nodes scaled-up just with 1 pod



Pending



Cluster-Autoscaler Infinite ScaleUp



N nodes scaled-up just with 1 pod



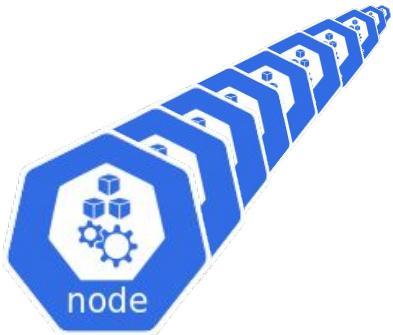
Simulation != Real



Pending



Cluster-Autoscaler Infinite ScaleUp



N nodes scaled-up just with 1 pod



Simulation != Real



Pending

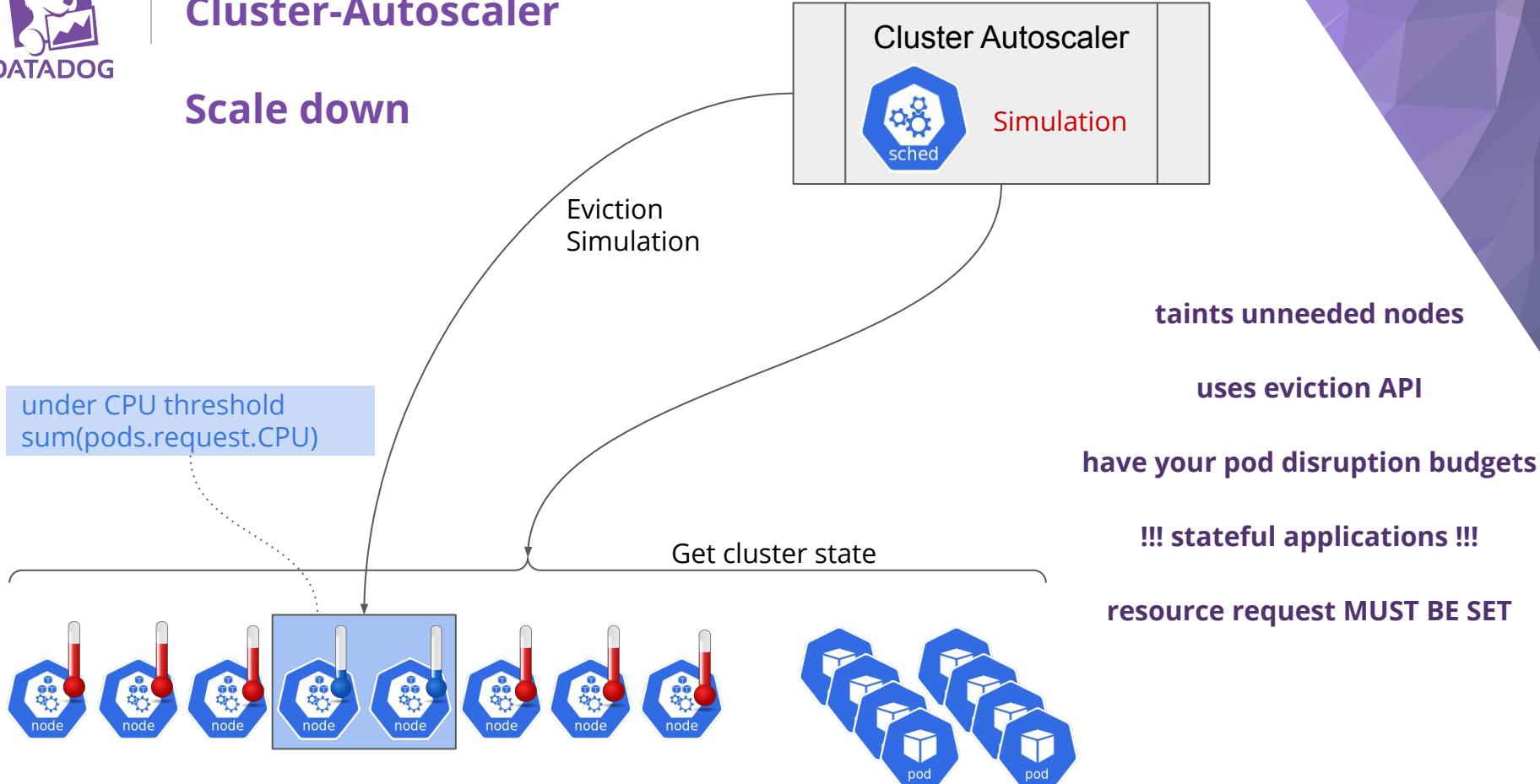
node capacity != node allocatable

metadata cache != live metadata (aws at least)



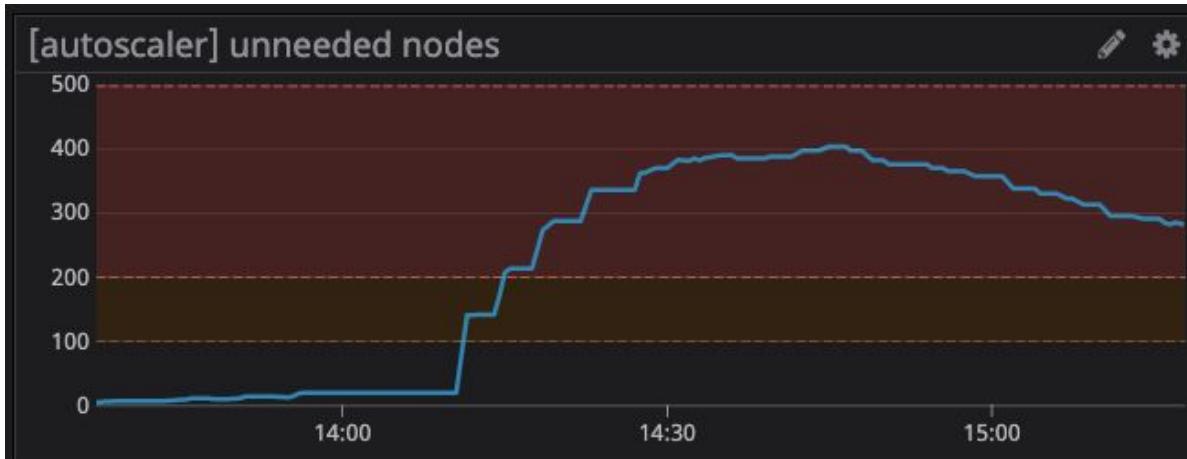
Cluster-Autoscaler

Scale down





CoolDown





Cluster Autoscaler - velocity

[autoscaler] scale up (+) / down (-)

Quick Functions



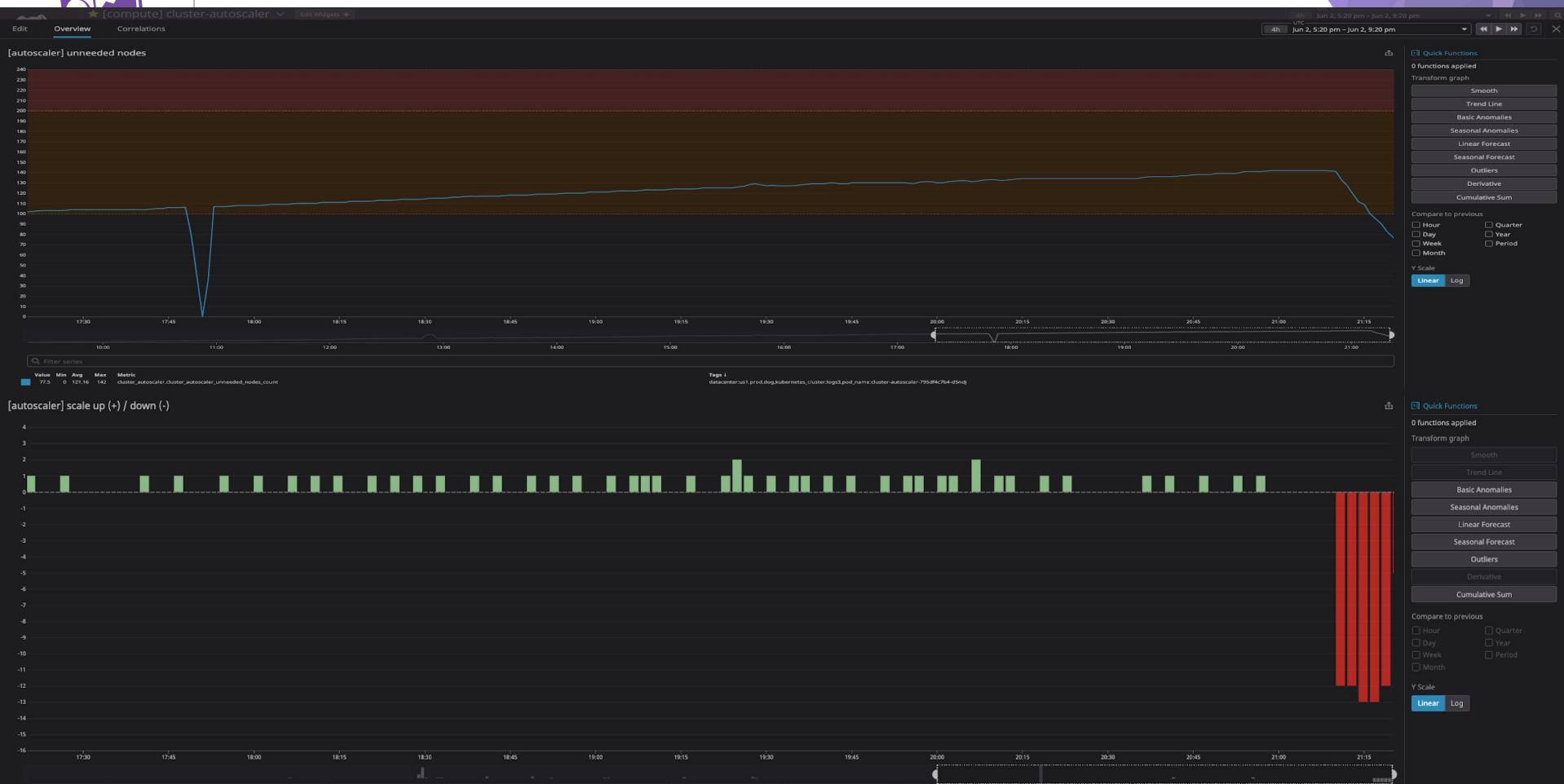
Filter series

Value	Sum	Metric
0	-175	(0 - cluster_autoscaler.cluster_autoscaler_scaled_down_nodes_total)
0	428	cluster_autoscaler.cluster_autoscaler_scaled_up_nodes_total

Tags ↓
datacenter:eu1.prod.dog.kubernetes_cluster:app3
datacenter:eu1.prod.dog.kubernetes_cluster:app3



Infinite scale-up



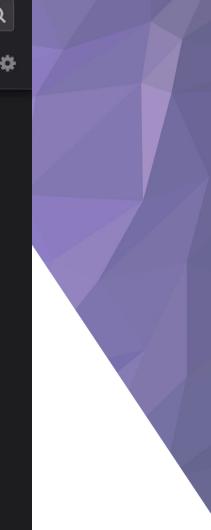


★ [compute] cluster-autoscaler

Edit Widgets +

UTC

12h May 22, 6:02 am - May 22, 5:57 pm



Q Search...

Save or select views

\$datacenter *

\$kubernetes_cluster app3 *

\$env * \$k8s_cluster *

edit



Watchdog

Events

Dashboards

Infrastructure

Monitors

Metrics

Integrations

APM

Notebooks

Logs

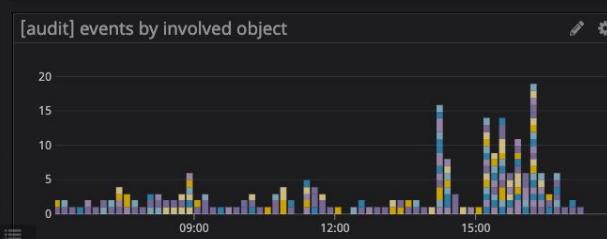
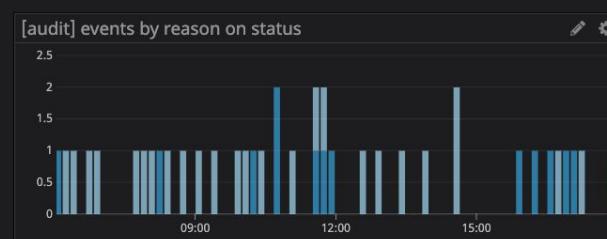
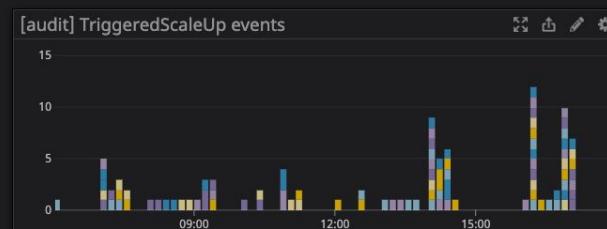
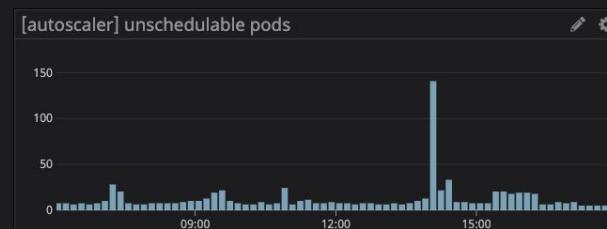
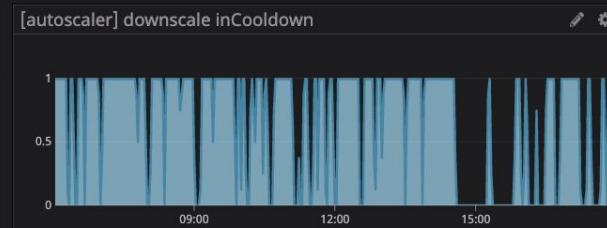
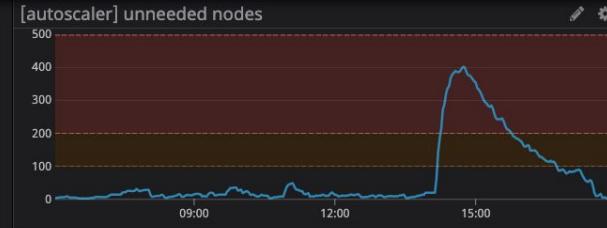
Security

UX Monitoring

Help

Team

david.benque...
Datadog HQ





Cluster-Autoscaler Scaleup to ?

You can set your limit!

Official test ~1000 nodes, ~30 pods per node

Loop iteration set to 10s

... it remains under 30s

The complexity is $O(\text{nodes} \times \text{pods})$

so with less pods on each nodes, >1000 nodes is ok
some feature (podAntiAffinity increase complexity)

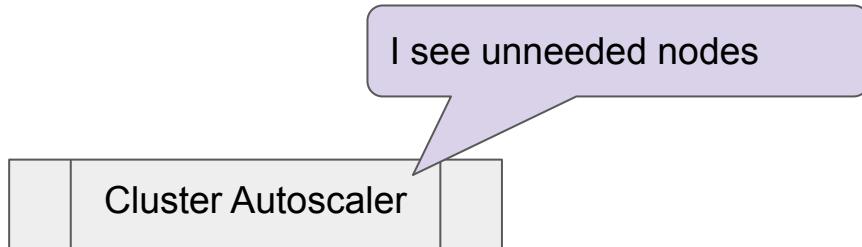


Cluster-Autoscaler, the case of jobs ...

Cluster autoscaler is doing a loop every 10 seconds (parameter)

The job may last only few seconds

If you have groups of nodes dedicated to jobs:



solution: create a node holder deployment



Combining solutions

cluster-autoscaler

+

any horizontal pod autoscaler

=

fine

use VPA alone or to get
recommendations

apply the recommendations
manually

High
Frequency

Low
Frequency



Combining solutions

cluster-autoscaler

+

any horizontal pod autoscaler

=

fine

use VPA alone or to get
recommendations

apply the recommendations
manually

high frequency

low frequency

so you can sleep and
save money... in theory



Questions

<https://github.com/kubernetes/autoscaler/tree/master/vertical-pod-autoscaler>

<https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler>

<https://github.com/kubernetes-sigs/cluster-proportional-autoscaler>

<https://github.com/kedacore/keda>

<https://github.com/DataDog/watermarkpodautoscaler>

<https://github.com/kubernetes/enhancements/tree/master/keps/sig-autoscaling>

The graphic features a dark purple background with white and light blue text. At the top right, it says "100% Numérique". Below that, "Telecom Valley" is on the left and "SophiaConf" is on the right, with the subtitle "Le cycle azuréen de conférences et workshops Open Source". In the center, the title "Autoscaling in Kubernetes: HPA, WPA, ClusterAutoscaler" is followed by "Par David BENQUE". At the bottom, the date "■ 30 Juin | 18h20 ■" is shown. A small "Gratuit sur inscription" and the website "www.sophiaconf.fr" are at the very bottom. On the right side, there is a black and white portrait of a smiling man with dark hair and a beard, identified as David Benque.