zalando

KubeCon | CloudNativeCon

Europe 2020

Virtual

Autoscaling at Scale

How we manage capacity @ Zalando

Mikkel Larsen, August 2020

zalando

$ whoami

Mikkel Larsen

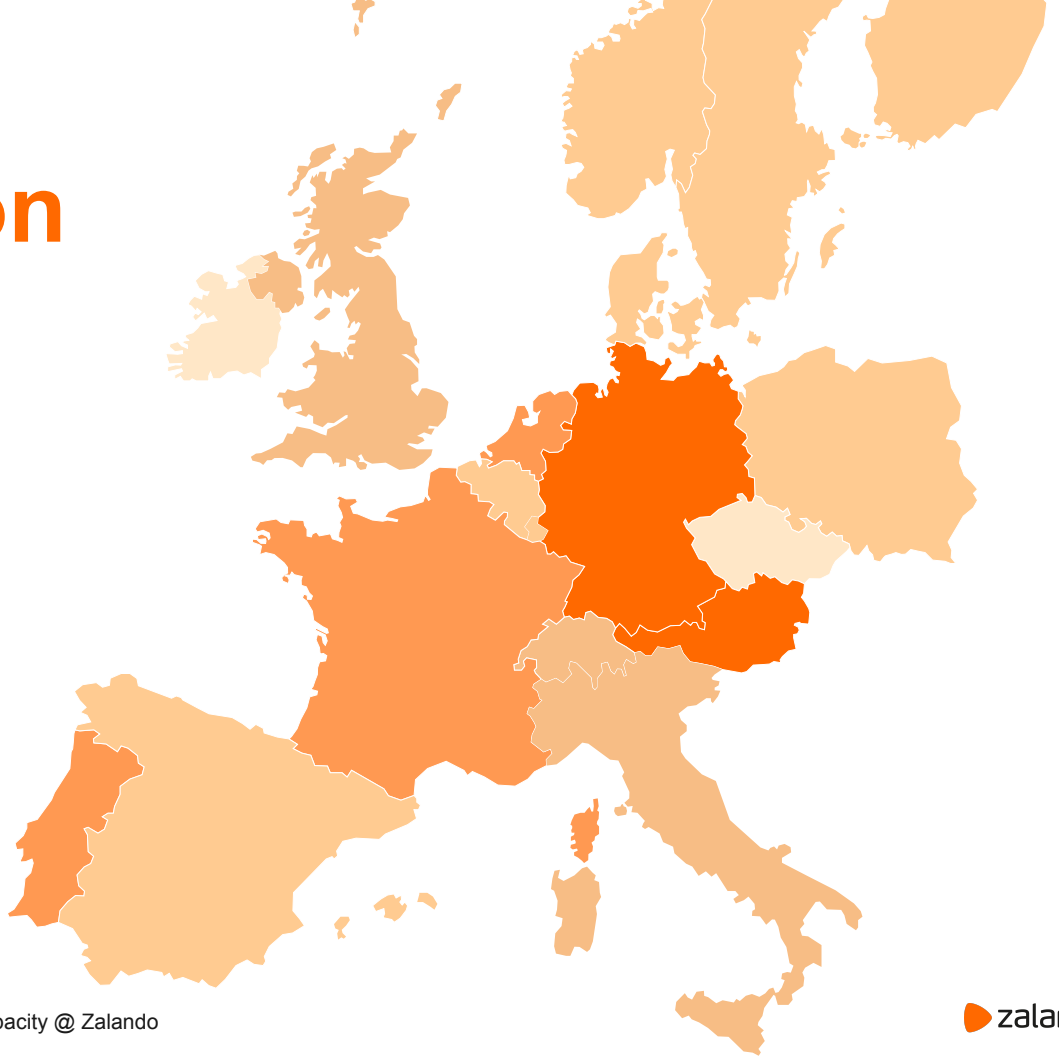
Software Engineer

Cloud Infrastructure @ Zalando SE

@mikkeloscar @mikkeloscar

Bringing Fashion to 17 Countries

- 11 Fulfillment centers
- 32 million active customers
- 380 million visits per month
- 8.2 € billion GMV



Infrastructure

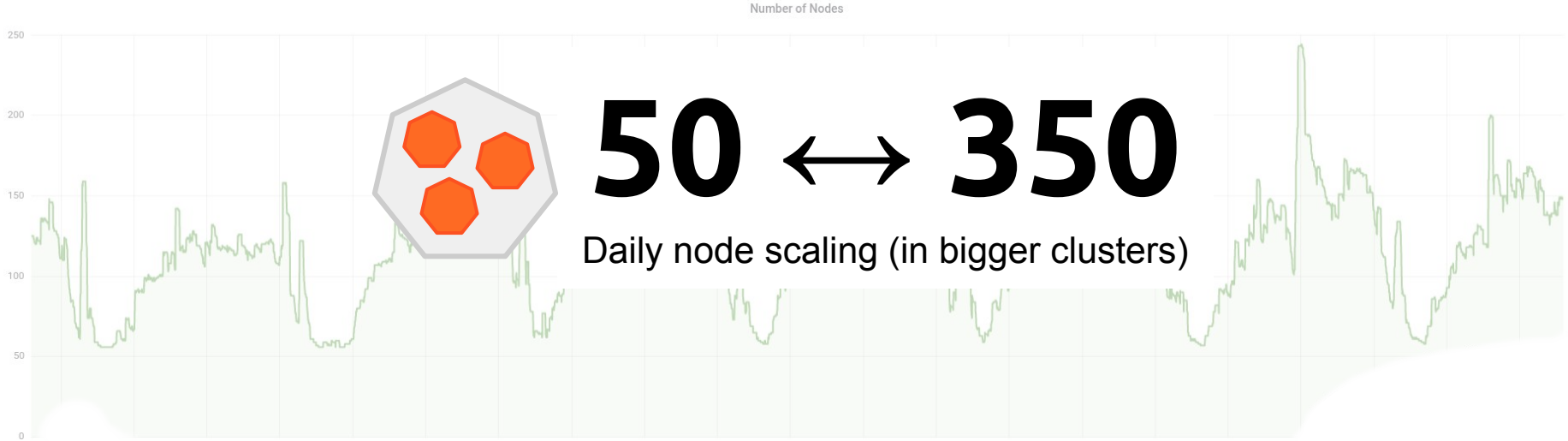
aws

150+

Kubernetes Clusters

4000+

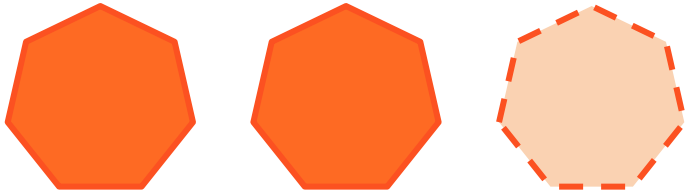
Services (85%+ in K8s)



50 ↔ 350

Daily node scaling (in bigger clusters)

zalando



Horizontal Pod Autoscaling

Horizontal Pod Autoscaler (HPA)

Official algorithm explanation

Configured on HPA

```
replicas = ceil[currentReplicas * (currentMetric / desiredMetric)]
```

Metric observed for running pods
(e.g. 50% average CPU)

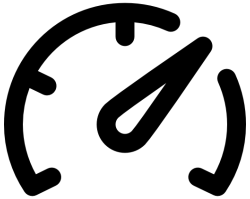
Supported Metric types

CPU | Memory | Custom/External

zalando

HPA - Kube Metrics Adapter

Queue Length



Ingress Req/s

Prometheus Query

JSON Metrics

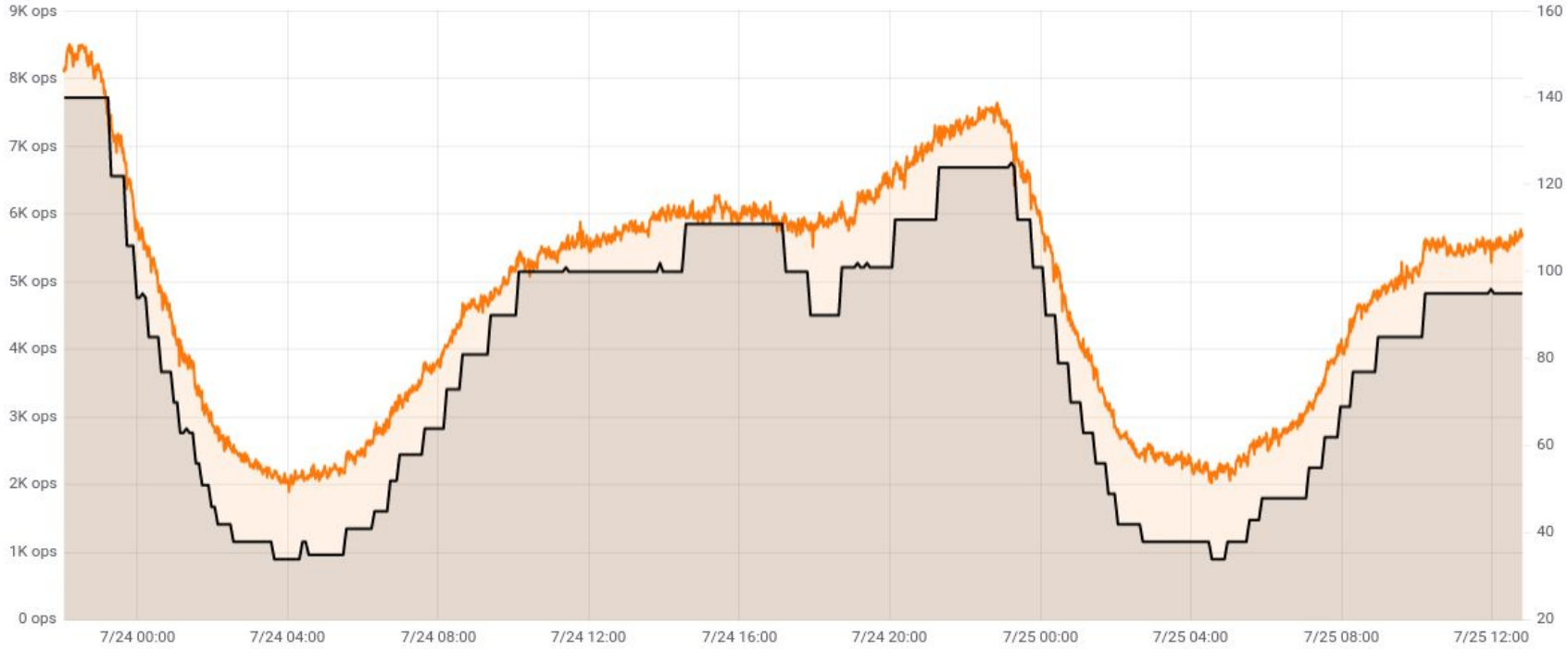
ZMON Check

InfluxDB Query

github.com/zalando-incubator/kube-metrics-adapter

zalando

HPA - Kube Metrics Adapter



● Req/s ● Pods

github.com/zalando-incubator/kube-metrics-adapter

zalando

HPA - Challenges

Scaling behavior was cluster wide until Kubernetes v1.18 (KEP)

Pods with multiple containers are not handled well (KEP)

Upstream contributions by Arjun Naik 🙏

zalando

HPA - Scaling Behavior

Until Kubernetes v1.17

```
initial-readiness-delay:     30s
autoscaler-tolerance:        0.1
cpu-initialization-period:   5m
downscale-stabilization:     5m
```

Clusterwide settings

From Kubernetes v1.18

Scaling behavior can be configured per HPA

```yaml
behavior:
  scaleDown:
    stabilizationWindowSeconds: 60
```

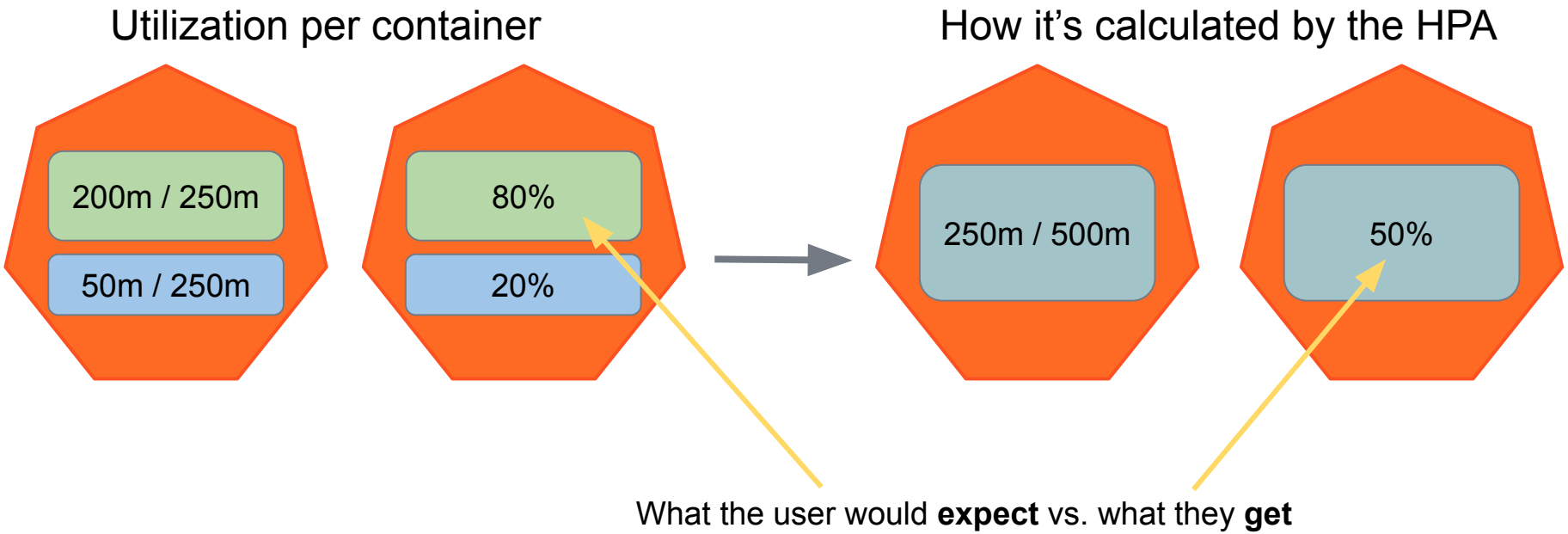
Reduce replicas by 10% every minutes

```yaml
behavior:
  policies:
  - type: Percent
    value: 10
    periodSeconds: 60
```

Configurable Scaling Docs

zalando

HPA - Multi Container Pods



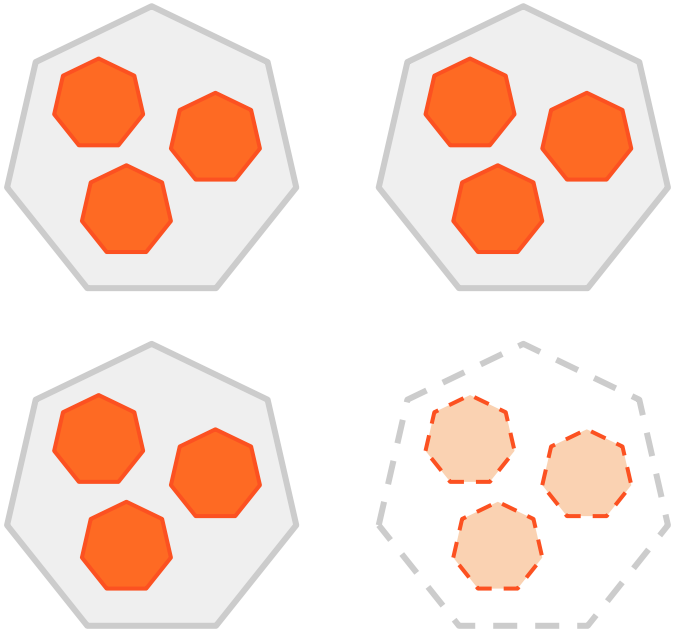
HPA - Multi Container Pods

Solution in Kubernetes >v1.19: **Container Resource metrics** (KEP, #90691)

```yaml
metrics:
- type: Resource
  resource:
    name: cpu
    target:
      type: Utilization
       averageUtilization: 30
```

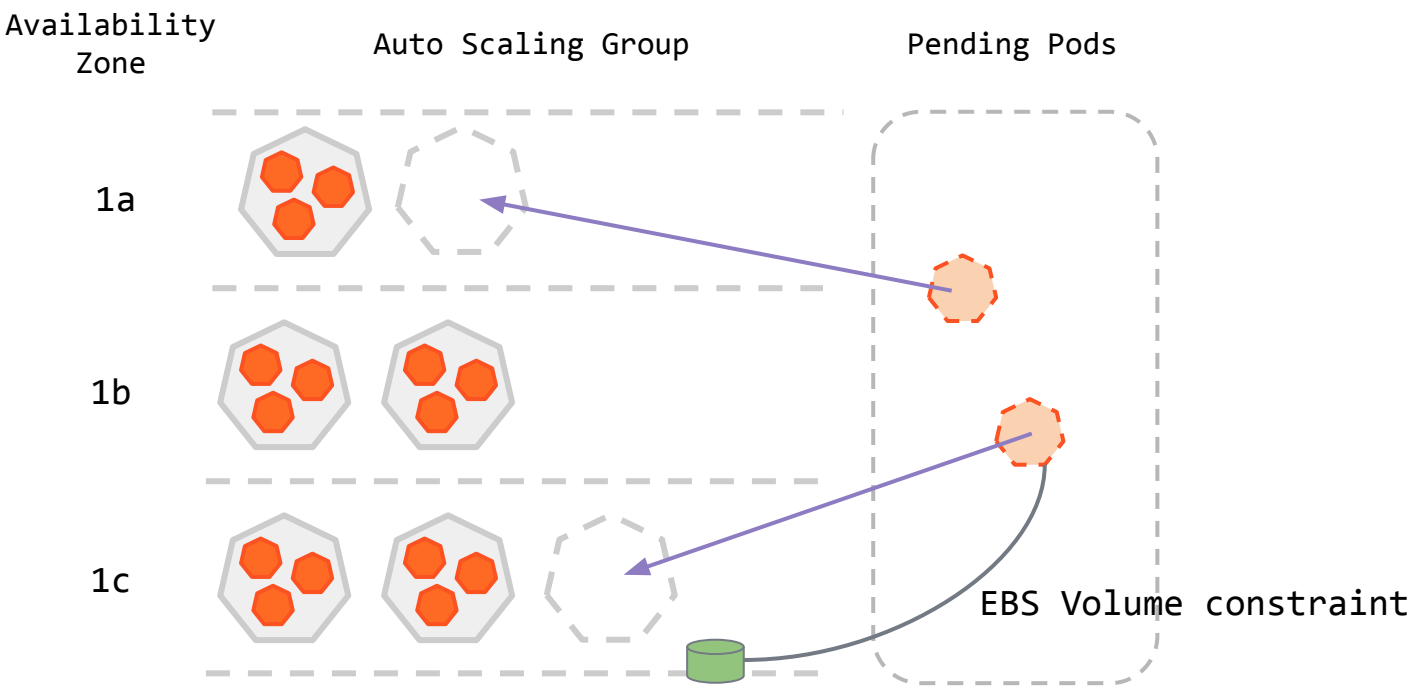
```yaml
metrics:
- type: ContainerResource
  resource:
    name: cpu
    container: application
    target:
      type: Utilization
       averageUtilization: 30
```


zalando

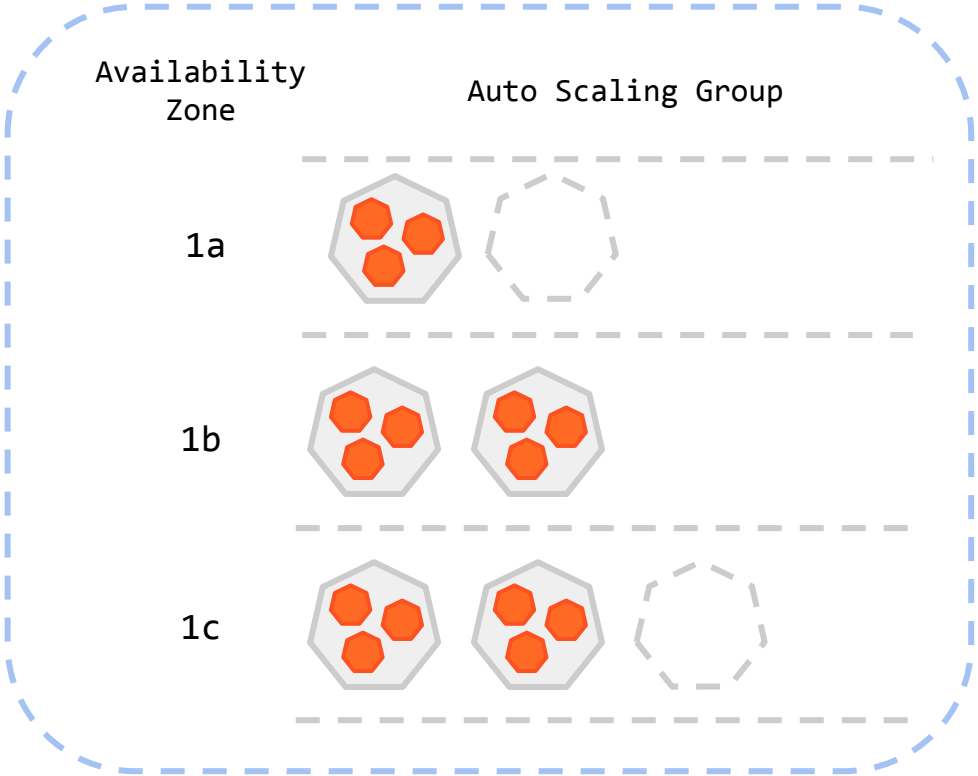


Cluster Autoscaling

Cluster Autoscaler (CA)



Cluster Autoscaler (CA) - Node Pools



Node Pool spanning 3 availability zones

Same instance type(s)

Same node labels and taints

Same min/max size

Different zone topology

zalando

Cluster Autoscaler (CA) - Node Pools

Name	Instance Types	Min Size	Max Size	Config Items
default	m5.large	0	300	
custom	c5d.xlarge	0	300	Label: dedicated=storage Taint: dedicated=storage

Pods must specify `nodeSelector` and `toleration` to land here!

zalando

Cluster Autoscaler (CA) - Fork

Custom Changes made to the Official Cluster Autoscaler

- **More robust template node generation**
  - Predict what a node would look like based on empty Auto Scaling Group
- **Support for AWS autoscaling groups with multiple instance types (spot)**
- **Customisable backoff settings**
  - Faster fallback to healthy Auto Scaling Groups
- **Priority based expander**
  - Custom defined priority for different node pools

Zalando Changes In Detail

zalando

Cluster Autoscaler (CA) - Node Pool Priority

Name	Instance Types	Min Size	Max Size	Config Items
default	m5.large	0	300	Priority: 0
fallback-m4-large	m4.large	0	300	Priority: -100
fallback-m5-xlarge	m5.xlarge	0	300	Priority: -200
fallback-m5-2xlarge	m5.2xlarge	0	300	Priority: -300
fallback-m5-4xlarge	m5.4xlarge	0	300	Priority: -400
spot	m5.large,m5.xlarge,...	0	300	Label: spot=true

zalando

Cluster Autoscaler (CA) - Limits

Easy to hit the default AWS limits!

Network layout limits the maximum number of nodes in a Cluster!

zalando

Cluster Autoscaler (CA) - Limits (AWS)

Limits

Calculate vCPU limit | Request

vcpu | All limits

Name	Limit type	Current limit	Description
Running On-Demand All G instances	Running instances	920 vCPUs	Running On-Demand G instances
Running On-Demand All Inf instances	Running instances	64 vCPUs	Running On-Demand Inf instances
Running On-Demand All P instances	Running instances	692 vCPUs	Running On-Demand P instances
Running On-Demand All Standard (A, C, D, H, I, ...	Running instances	4800 vCPUs	Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances
Running On-Demand All X instances	Running instances	560 vCPUs	Running On-Demand X instances
Running On-Demand All F instances	Running instances	176 vCPUs	Running On-Demand F instances

- **We use a Cronjob to bump limits.**
  - Creates automatic support requests to AWS based on node pool max sizes for all clusters/accounts.

zalando

Cluster Autoscaler (CA) - Limits (Network)

Network layout limits the maximum number of nodes in a Cluster!

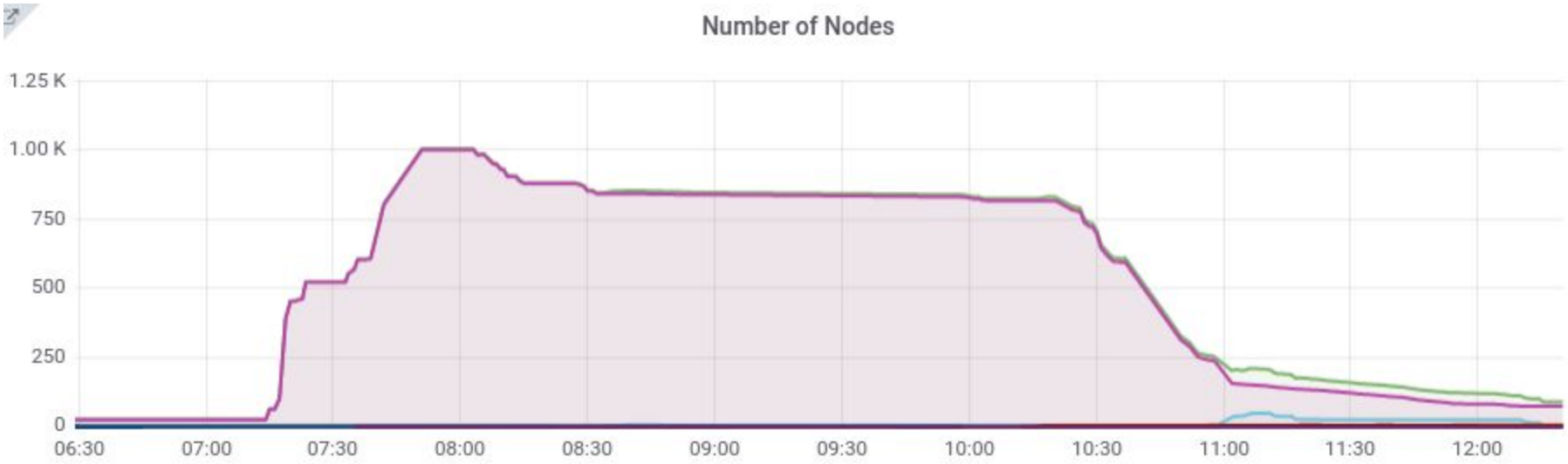
Pod CIDR: **10.2.0.0/16 = 65536** addresses

Node CIDR	Addresses per Node	Max Nodes (Total addresses / addresses per node)
/24 (default)	256	~ 256
/25	128	~ 512
/26	64	~ 1024
/27	32	~ 2048

Kubelet has a limit of **110** pods by default

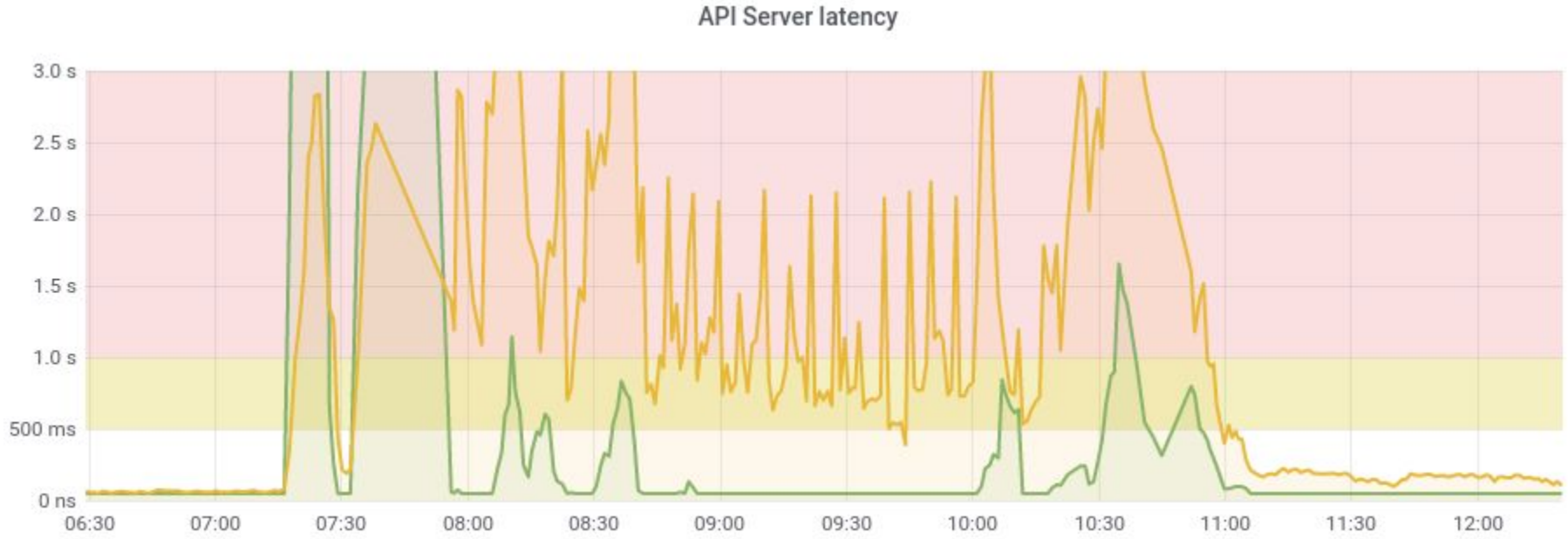
zalando

Cluster Autoscaler (CA) - No Limits!



Cluster Autoscaler (CA) - No Limits!

Control Plane Load Increases With Cluster Size



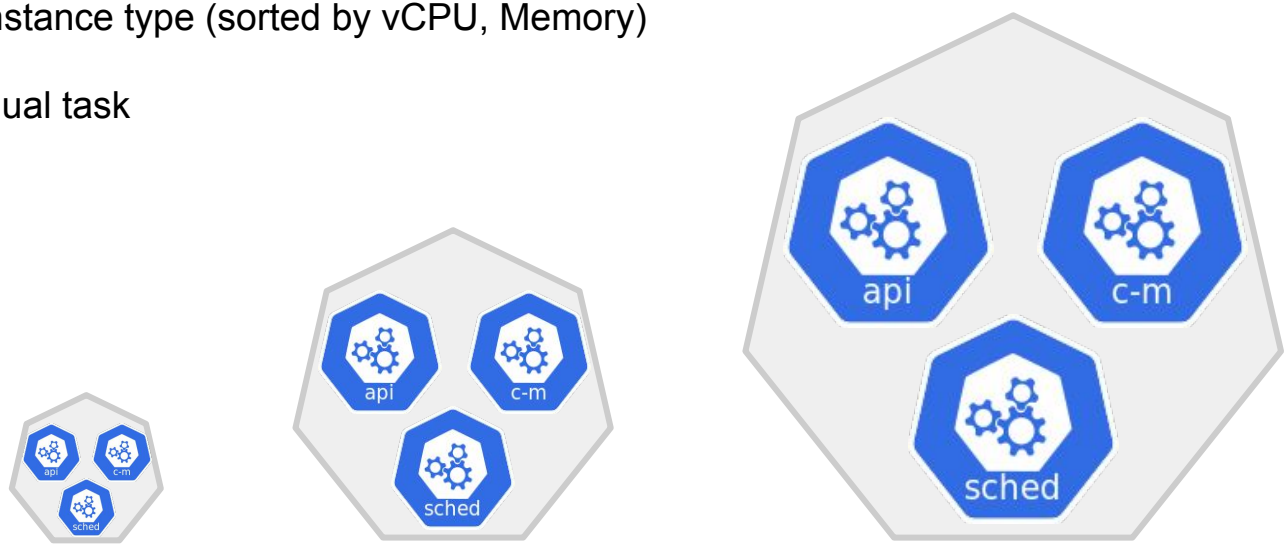
Control Plane Autoscaler

Scales vertically (needs to read 1000s of Pods at a time from etcd in big clusters)

Uses CPU load as indicator for scaling

Scales by changing EC2 instance type (sorted by vCPU, Memory)

Automates a previous manual task



Vertical Pod Autoscaling

Vertical Pod Autoscaler (VPA)

Scales Pods vertically by changing **requests/limits**

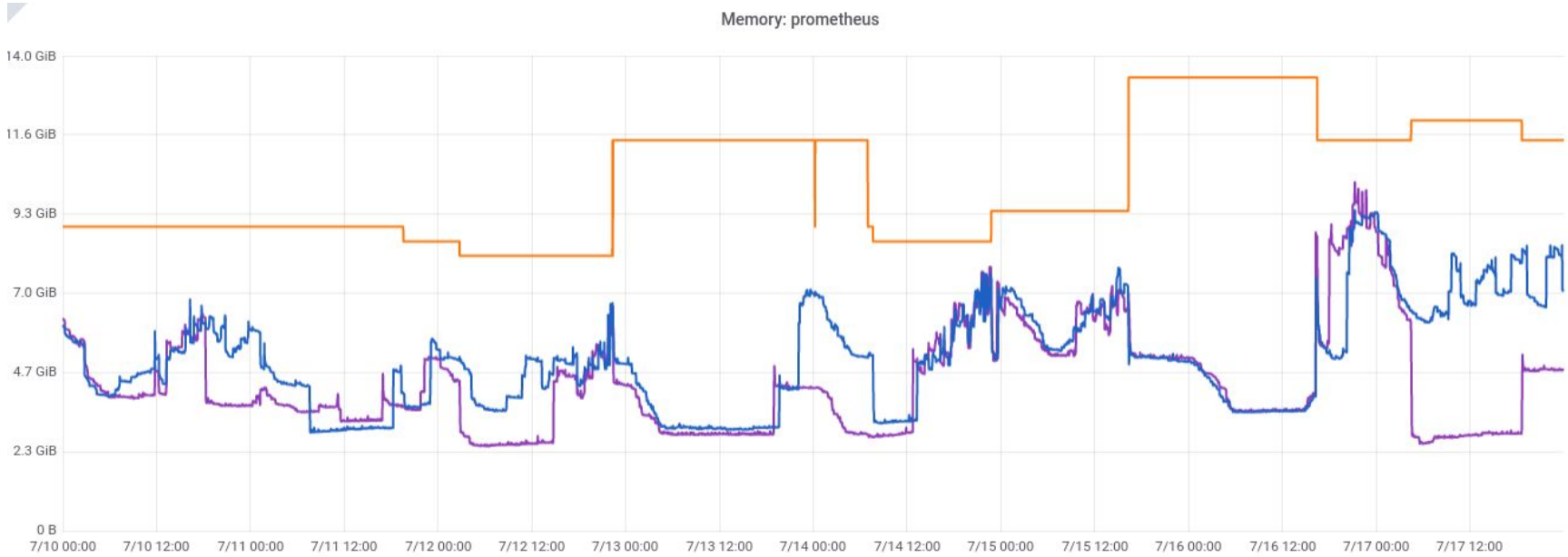
Scales based on **CPU** and **Memory** (no custom metrics)

Useful for components that scales vertically with the **size** of the cluster:

Prometheus, Ingress Controllers, External DNS, etc.

zalando

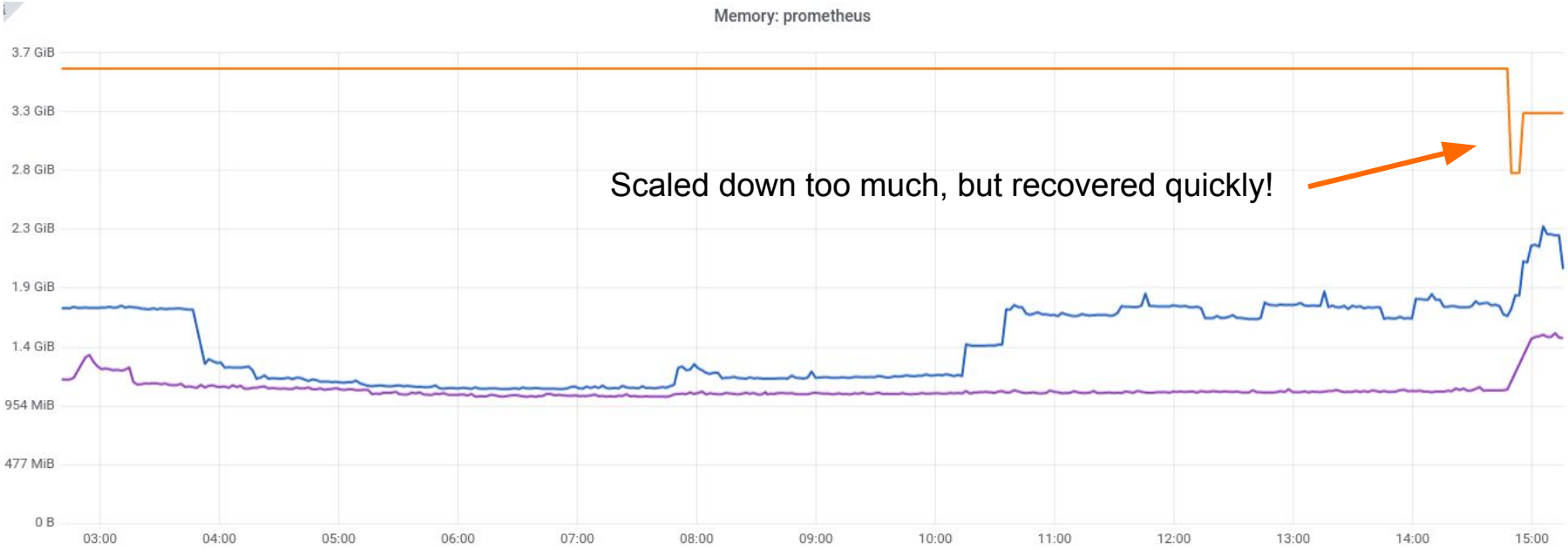
Vertical Pod Autoscaler (VPA)



- Requested/Limit
- Usage

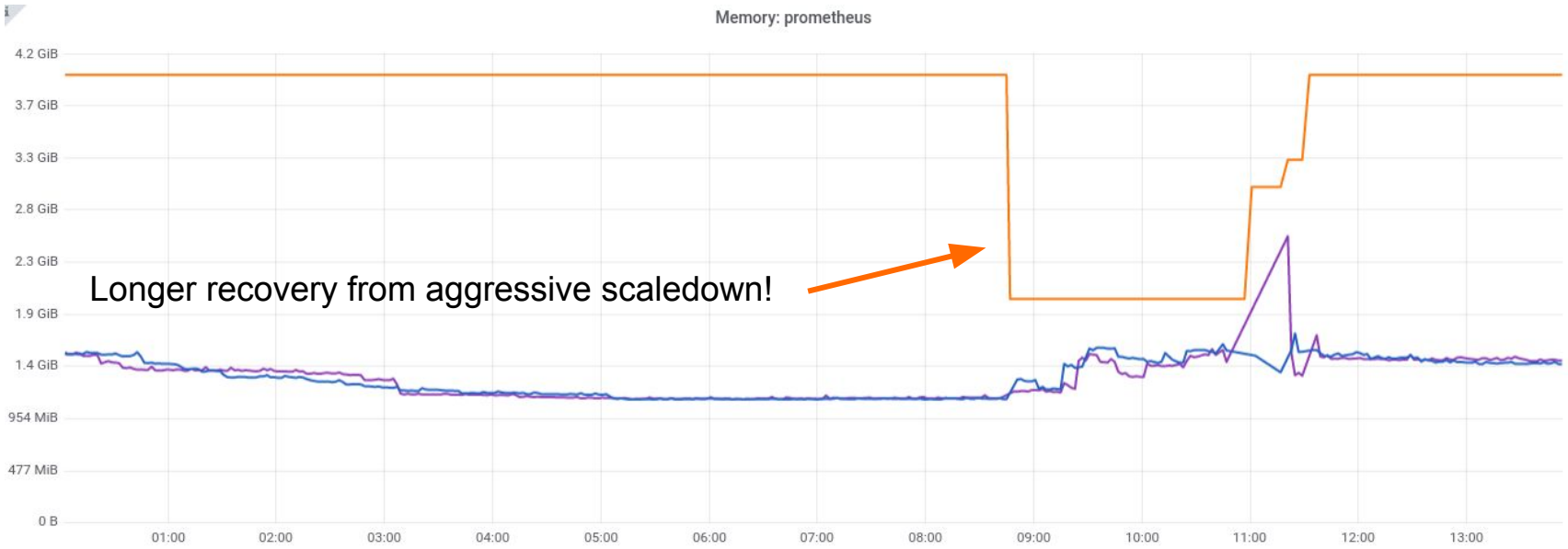
Failure Modes

Vertical Pod Autoscaler (VPA) - Failure modes



- Requested/Limit
- Usage

Vertical Pod Autoscaler (VPA) - Failure modes



- Requested/Limit
- Usage

VPA - Custom Changes

Improved OOMKilled handling

Quick OOM detection: Handle **all** containers

Always delete pods on quick OOMKills

Always record a sample for OOMKills

Various Small improvements

Reduced memory usage of VPA components

Timeouts for Admission Webhook

Zalando VPA Fork

Open Source

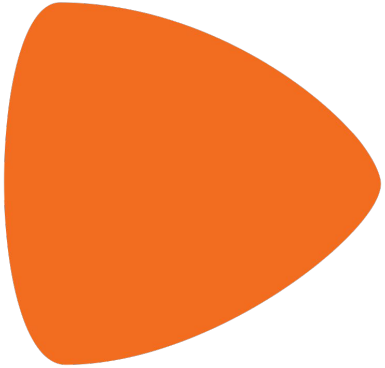
Kube Metrics Adapter

Cluster Autoscaler fork

Vertical Pod Autoscaler fork

KEP: Configurable scale up/down velocity for HPA

KEP: Container Resource Autoscaling



Thank you!

zalando