zalando

Kubernetes Failure Stories

MEETUP

HAMBURG

2019-02-11

HENNING JACOBS

@try_except_

ZALANDO AT A GLANCE

~ 4.5 billion EUR
revenue 2017

> 200 million
visits per month

> 24 million
active customers

> 300.000
product choices

~ 2.000
brands

17
countries

> 15.000
employees in Europe

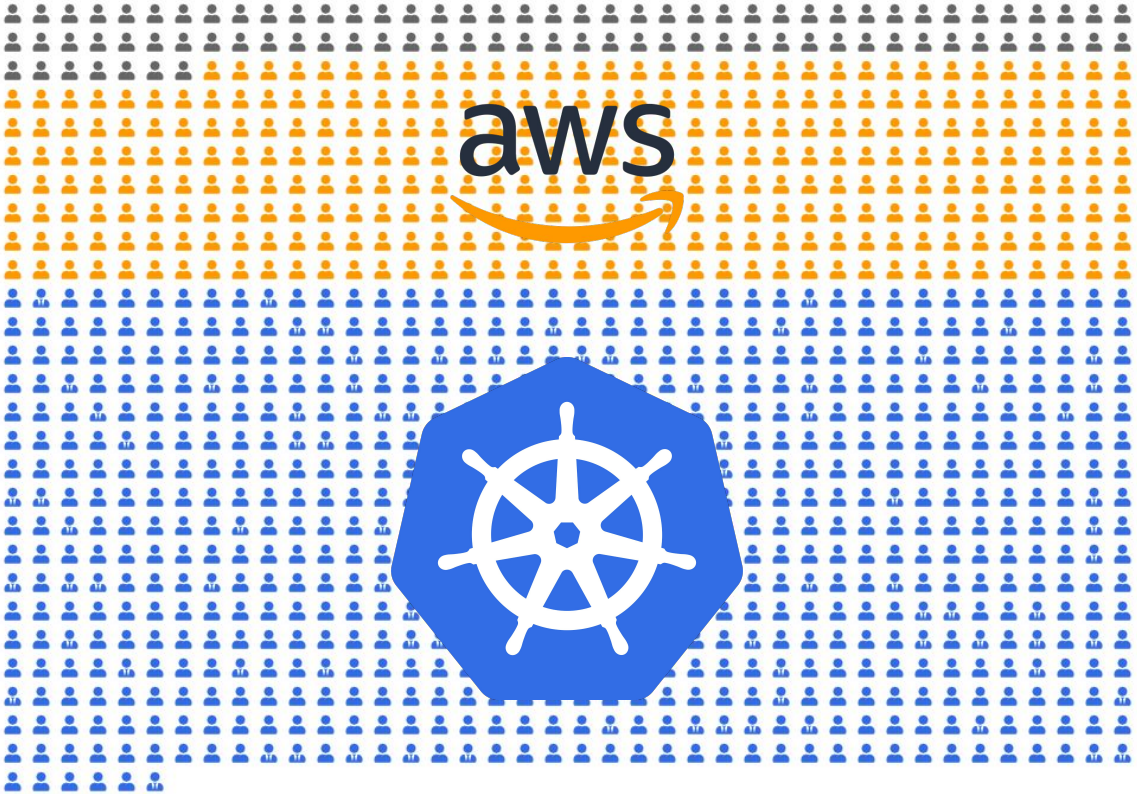
> 70%
of visits via mobile devices

SCALE

373 Accounts

100 Clusters

DEVELOPERS USING KUBERNETES

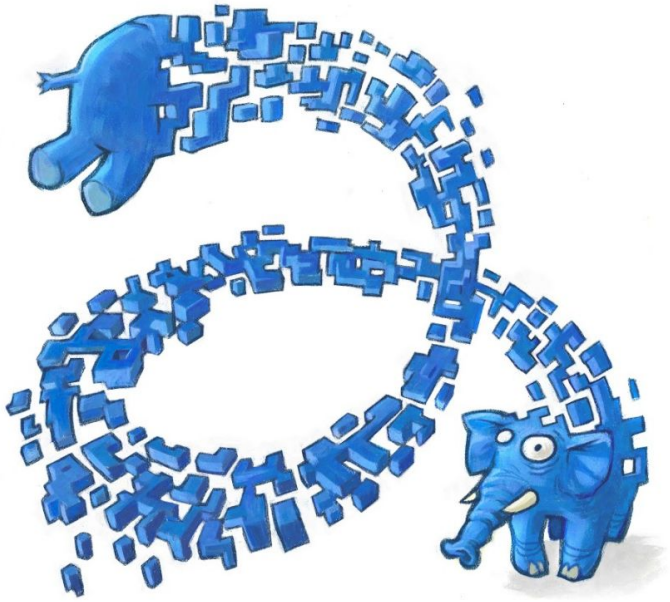


zalando

46+ cluster components

zalando

POSTGRES OPERATOR



Application to manage PostgreSQL clusters on Kubernetes

>500

clusters running on Kubernetes

https://github.com/zalando-incubator/postgres-operator


INCIDENTS ARE FINE

INCIDENT

#1

#1: LESS THAN 20% OF NODES AVAILABLE

NAME	STATUS	AGE	VERSION
ip-172-31-10-91...internal	NotReady	4d	v1.7.4+coreos.0
ip-172-31-11-16...internal	NotReady	4d	v1.7.4+coreos.0
ip-172-31-11-211...internal	Ready,SchedulingDisabled	5d	v1.7.4+coreos.0
ip-172-31-15-46...internal	Ready	4d	v1.7.4+coreos.0
ip-172-31-18-123...internal	NotReady	4d	v1.7.4+coreos.0
ip-172-31-19-46...internal	Ready	4d	v1.7.4+coreos.0
ip-172-31-19-75...internal	NotReady	4d	v1.7.4+coreos.0
ip-172-31-2-124...internal	NotReady	4d	v1.7.4+coreos.0
ip-172-31-3-58...internal	Ready	4d	v1.7.4+coreos.0
ip-172-31-5-211...internal	Ready	4d	v1.7.4+coreos.0
ip-172-31-7-147...internal	Ready,SchedulingDisabled	5d	v1.7.4+coreos.0

TRAIL OF CLUES

- Recovered automatically after 15 minutes
- Nodes unhealthy at same time, recover at same time
- API server is behind AWS ELB
- Seems to happen to others, too
- Some report it happening ~every month

UPSTREAM ISSUE

kubelet fails to heartbeat with API server with stuck TCP connections #48638

Closed **derekwaynecarr** opened this issue on Jul 7, 2017 · 32 comments

derekwaynecarr commented on Jul 7, 2017 • edited ▾ Member

Is this a BUG REPORT or FEATURE REQUEST?:
/kind bug

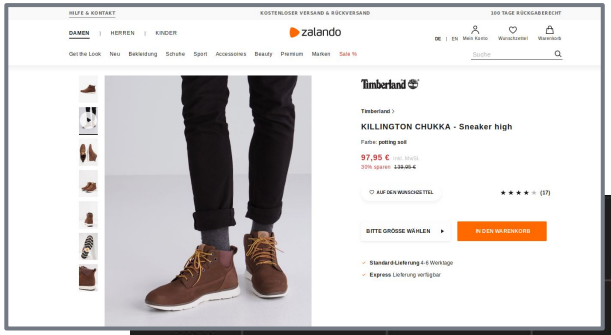
⇒ Fixed in 1.8 (backported to 1.7.8)

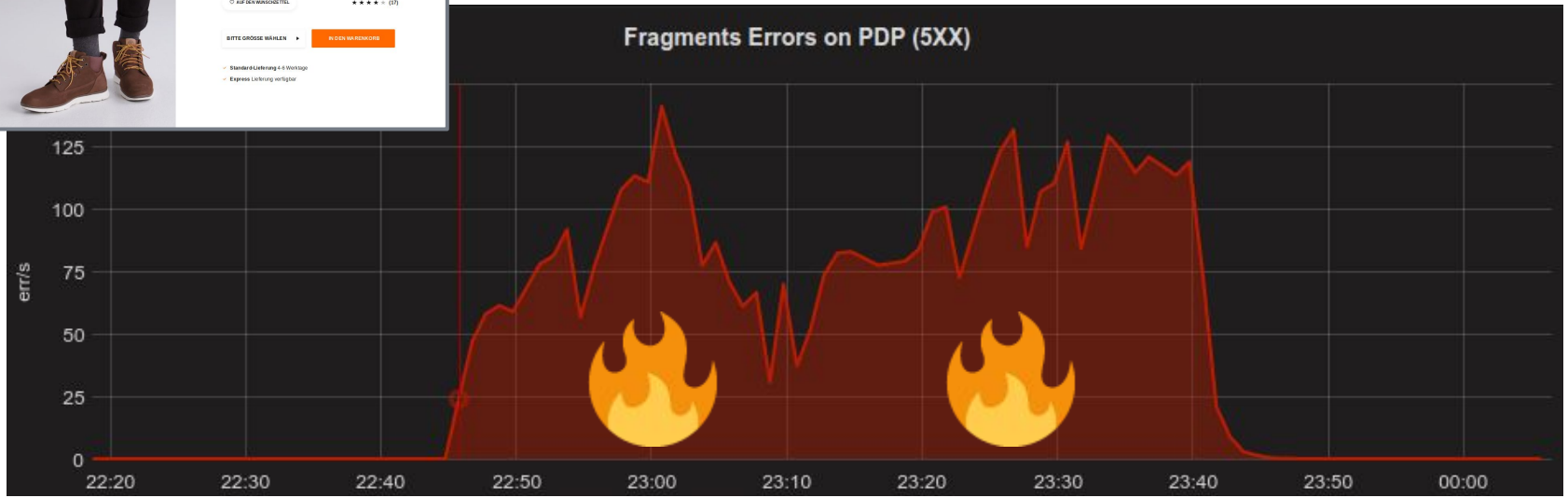
https://github.com/kubernetes/kubernetes/issues/48638

INCIDENT

#2

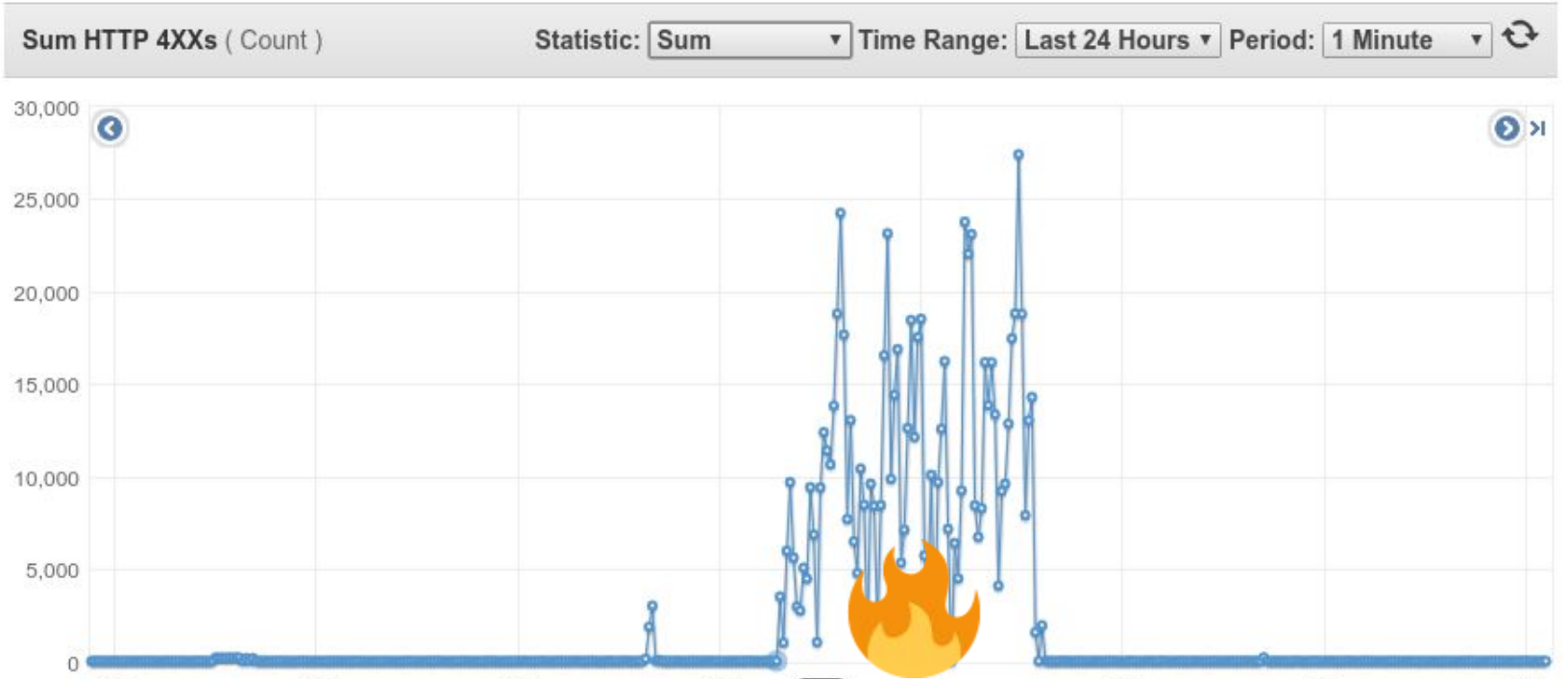
INCIDENT #2: CUSTOMER IMPACT





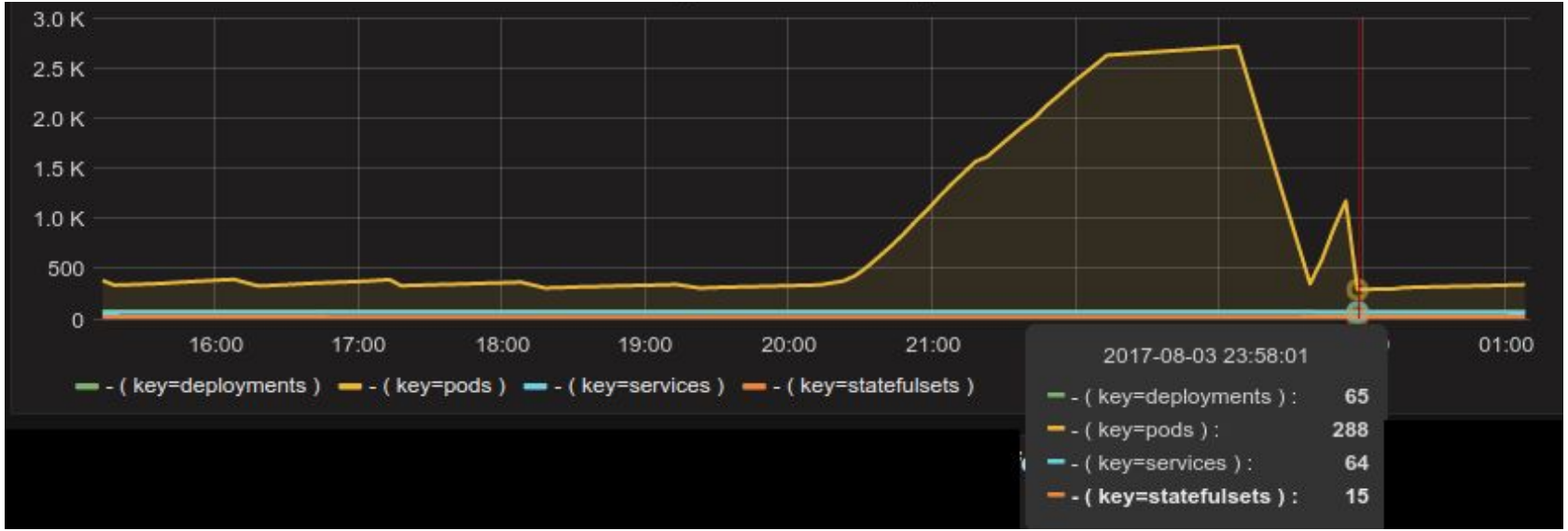
zalando

INCIDENT #2: IAM RETURNING 404



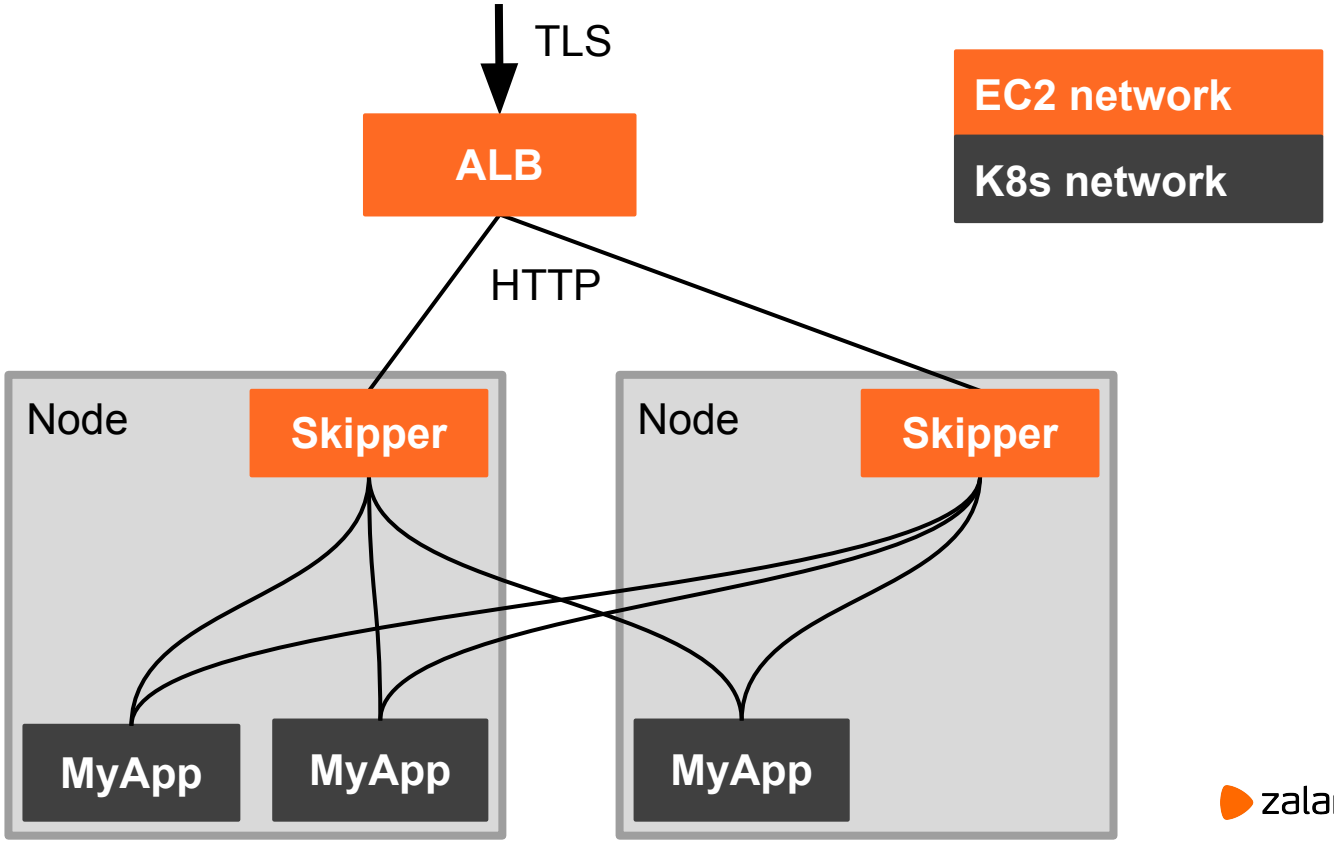
zalando

INCIDENT #2: NUMBER OF PODS



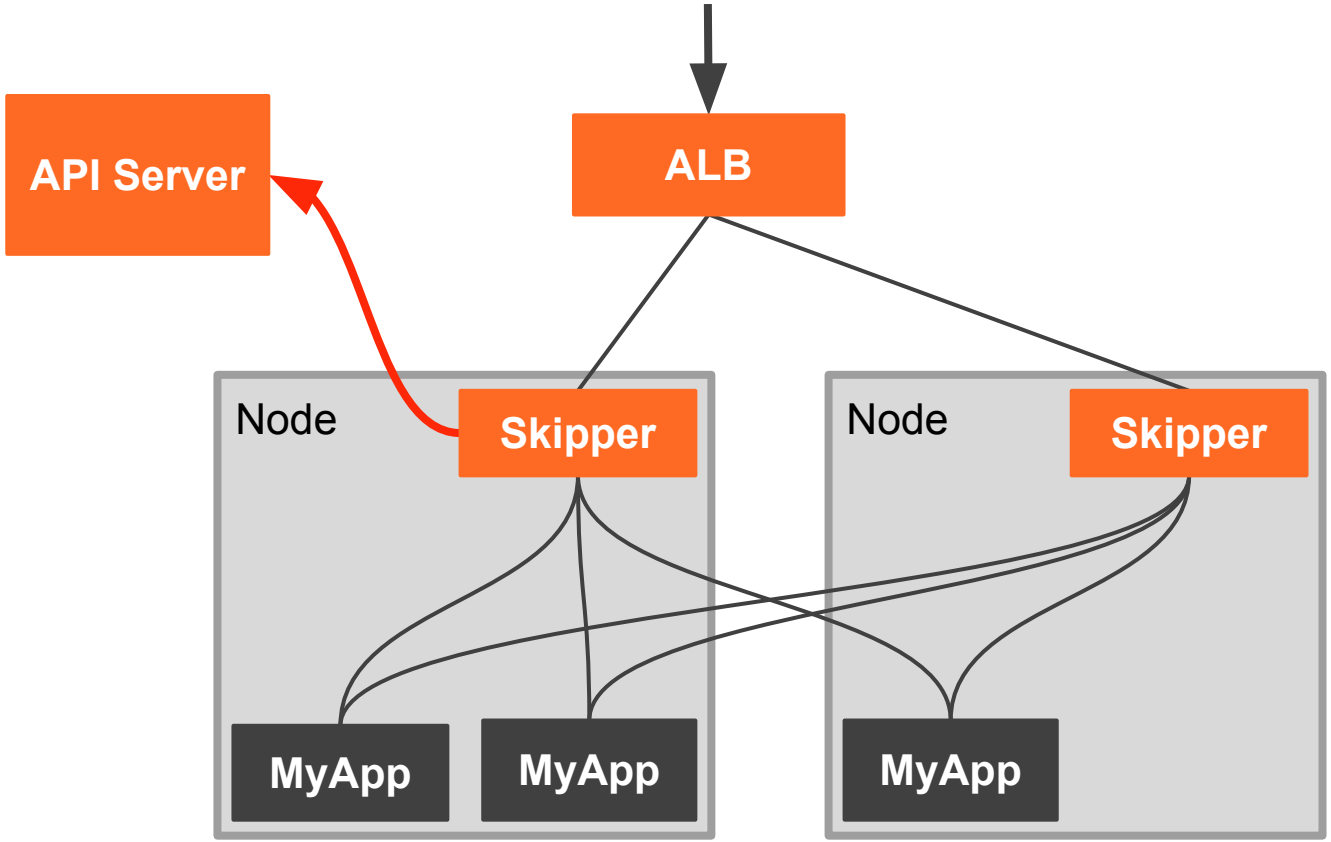
zalando

LIFE OF A REQUEST (INGRESS)



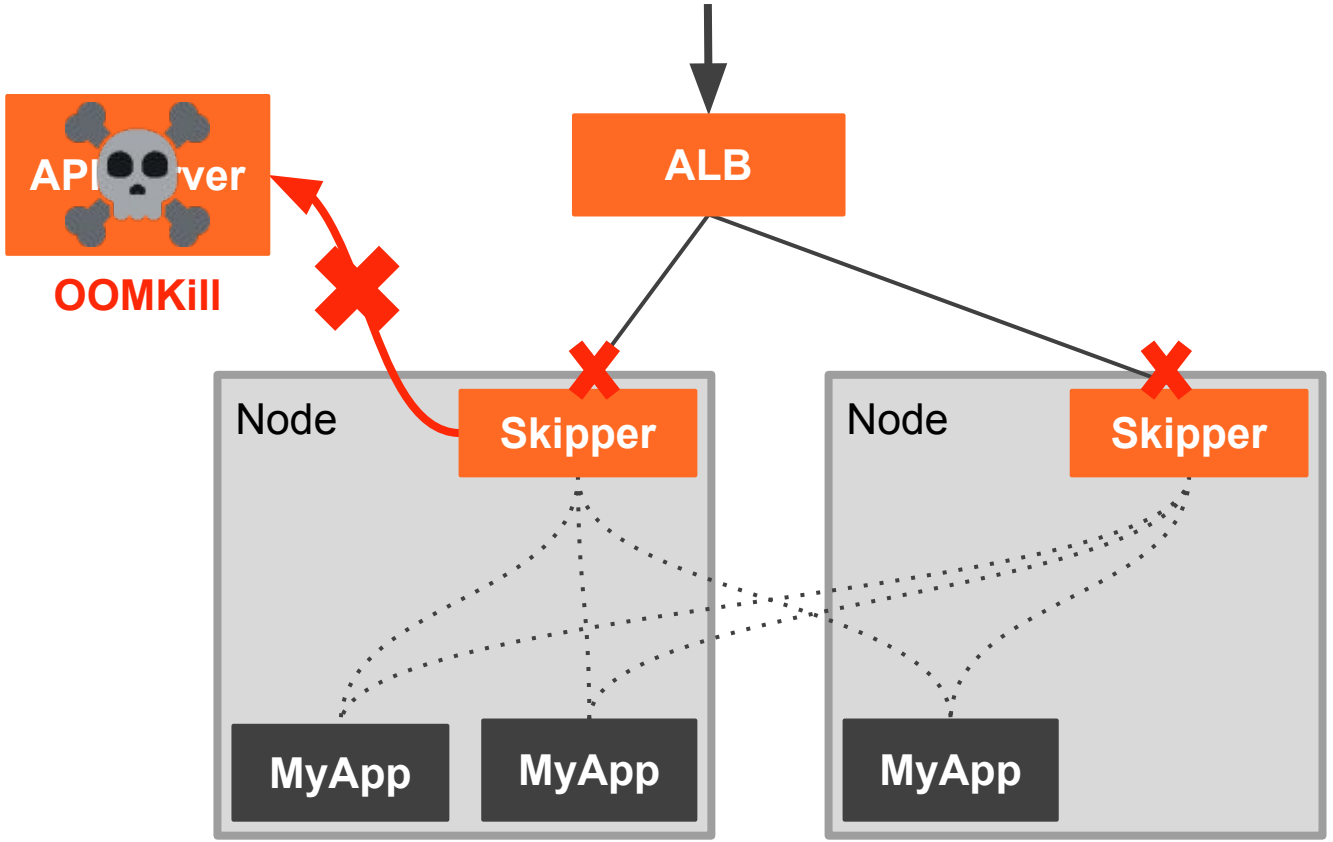
zalando

ROUTES FROM API SERVER



zalando

API SERVER DOWN



zalando

INCIDENT #2: INNOCENT MANIFEST

```
apiVersion: batch/v2alpha1
kind: CronJob
metadata:
  name: "foobar"
spec:
  schedule: "*/15 9-19 * * Mon-Fri"
  jobTemplate:
    spec:
      template:
        spec:
        restartPolicy: Never
        concurrencyPolicy: Forbid
        successfulJobsHistoryLimit: 1
        failedJobsHistoryLimit: 1
        containers:
        ...
```

INCIDENT #2: FIXED CRON JOB

```
apiVersion: batch/v2alpha1
kind: CronJob
metadata:
  name: "foobar"
spec:
  schedule: "7 8-18 * * Mon-Fri"
  concurrencyPolicy: Forbid
  successfulJobsHistoryLimit: 1
  failedJobsHistoryLimit: 1
  jobTemplate:
    spec:
      activeDeadlineSeconds: 120
      template:
        spec:
          restartPolicy: Never
          containers:
```

INCIDENT #2: CONTRIBUTING FACTORS

- Wrong CronJob manifest and no automatic job cleanup
- Reliance on Kubernetes API server availability
- Ingress routes not kept as-is in case of outage
- No quota for number of pods

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: compute-resources
spec:
  hard:
    pods: "1500"
```

INCIDENT

#3

INCIDENT #3: INGRESS ERRORS

INCIDENT #3: COREDNS OOMKILL

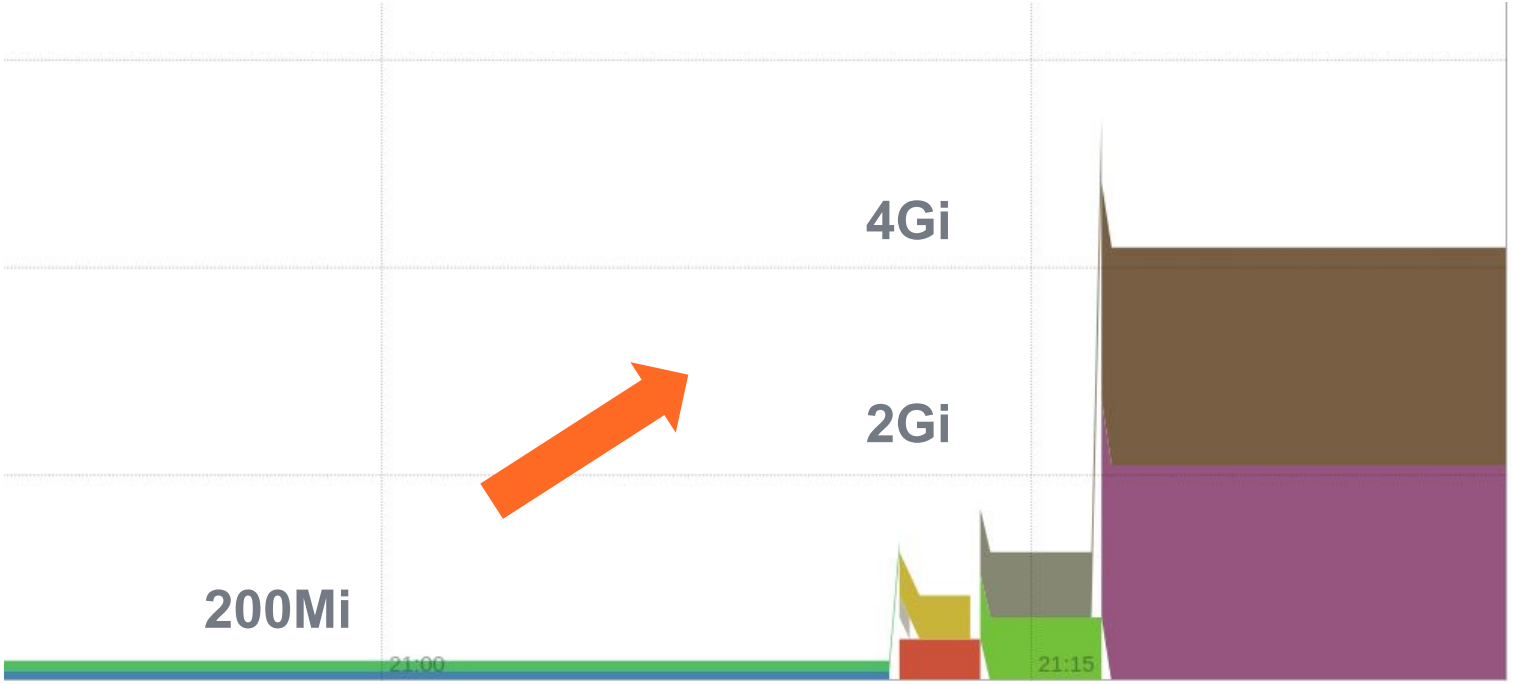
coredns invoked **oom-killer**:
gfp_mask=0x14000c0(GFP_KERNEL),
nodemask=(null), order=0, oom_score_adj=994

Memory cgroup out of memory: Kill process 6428 (coredns) score 2050 or sacrifice child

oom_reaper: reaped process 6428 (coredns), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB

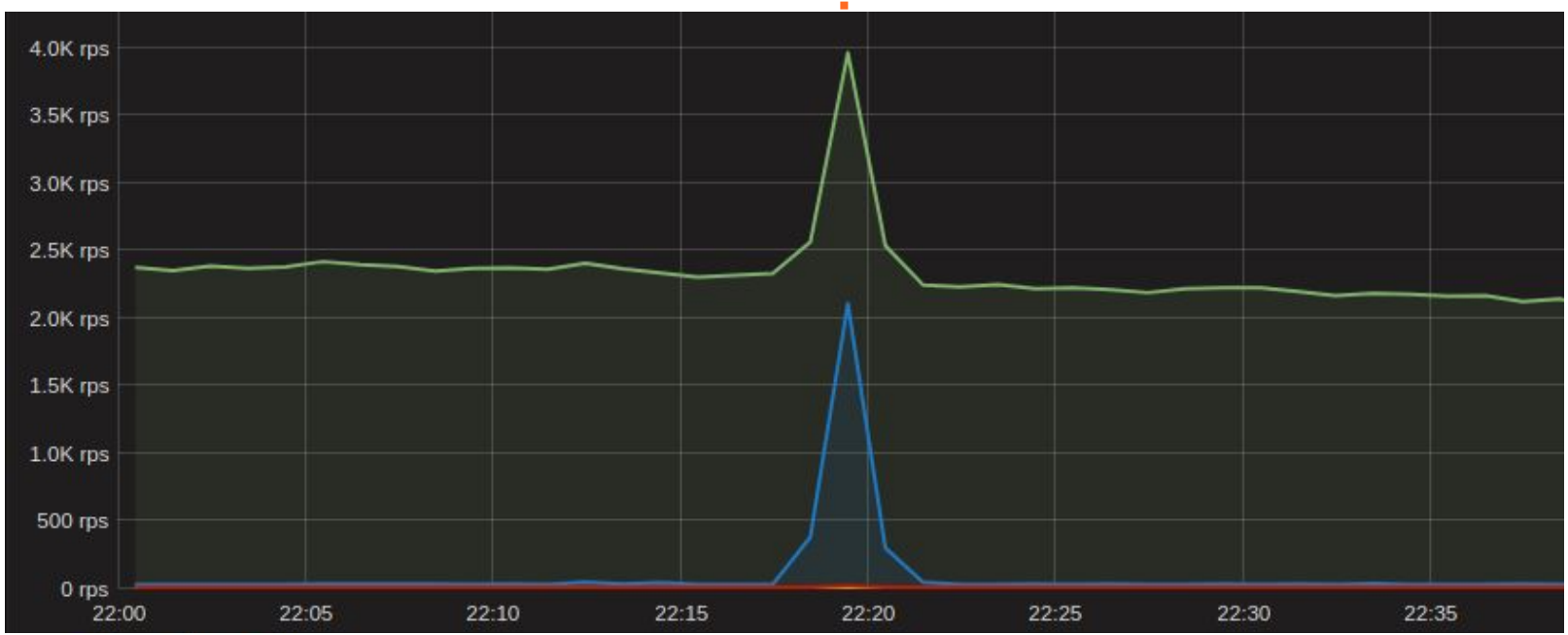
zalando

STOP THE BLEEDING: INCREASE MEMORY LIMIT



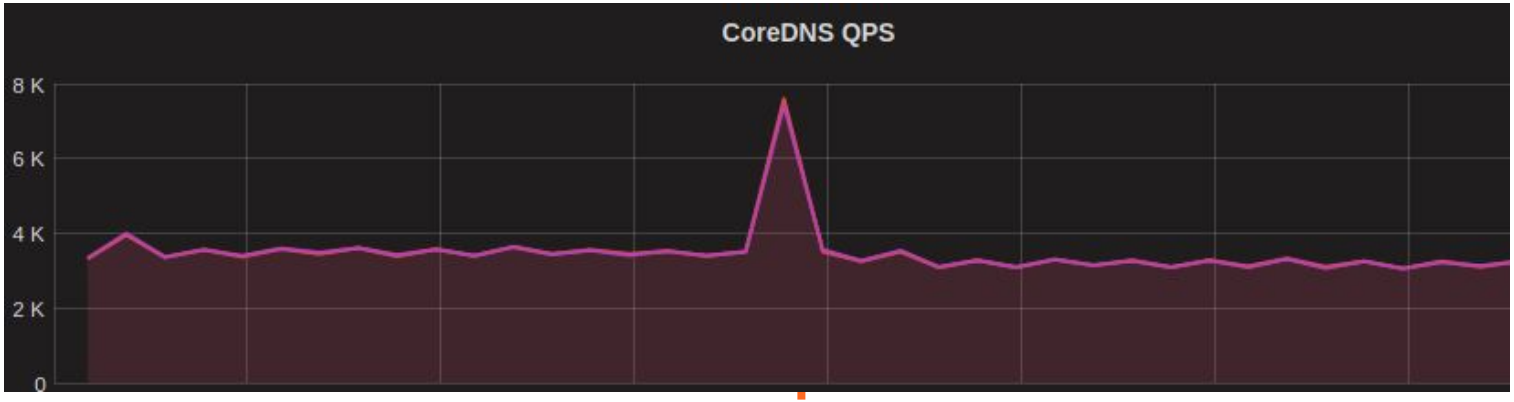
zalando

SPIKE IN HTTP REQUESTS



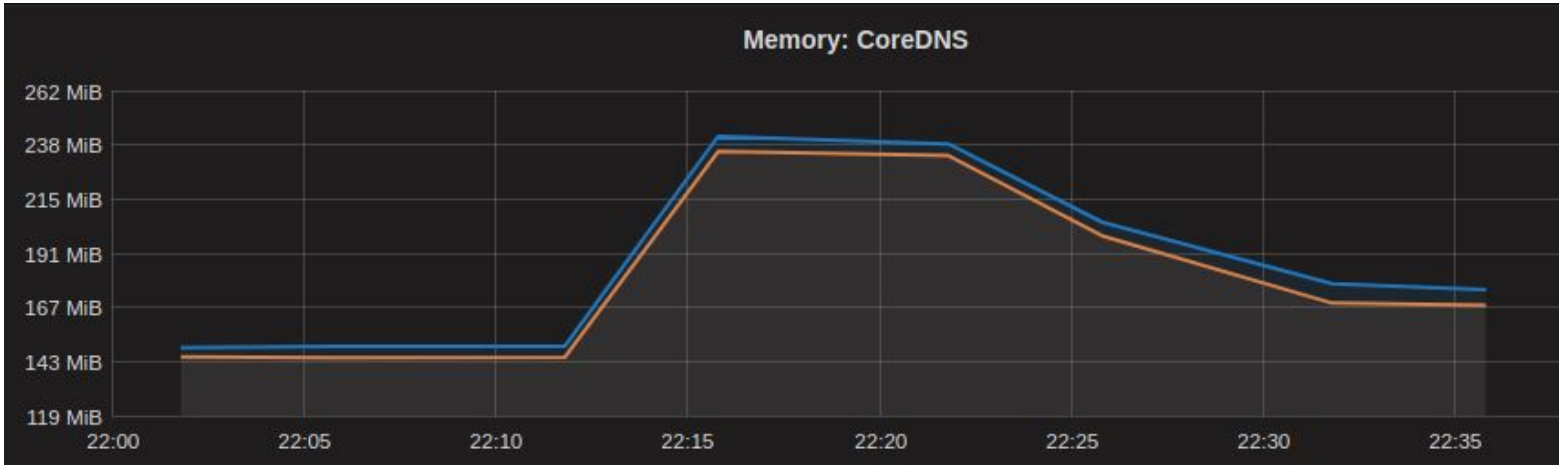
zalando

SPIKE IN DNS QUERIES



zalando

INCREASE IN MEMORY USAGE



zalando

INCIDENT #3: CONTRIBUTING FACTORS

- HTTP retries
- No DNS caching
- Kubernetes ndots:5 problem
- Short maximum lifetime of HTTP connections
- Fixed memory limit for CoreDNS
- Monitoring affected by DNS outage

github.com/zalando-incubator/kubernetes-on-aws/blob/dev/docs/postmortems/jan-2019-dns-outage.md

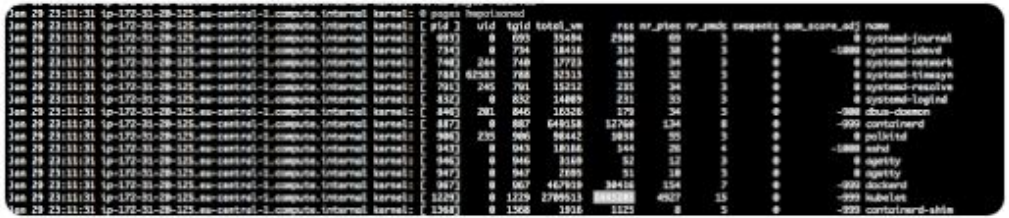
INCIDENT

#4

#4: KERNEL OOM KILLER

[redacted] Jan 30, 11:59 AM

so this is nice:



investigating a node in [redacted] kubelet apparently ate ~9gigs of ram and then the kernel oomkilled everything, including containerd

[redacted] Jan 30, 12:00 PM

Way to go KUBELET!!!!

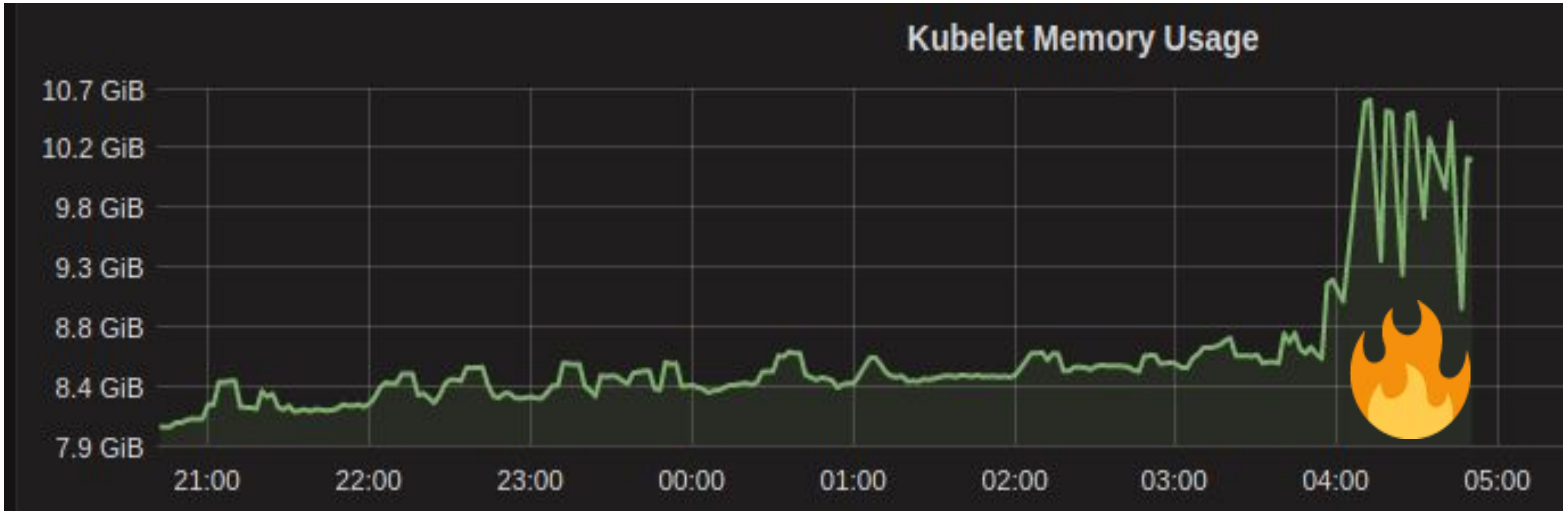
[redacted] Jan 30, 12:00 PM

well, it did solve the memory issues on the node!

⇒ all containers on this node down

zalando

INCIDENT #4: KUBELET MEMORY



zalando

UPSTREAM ISSUE REPORTED

memory leak in kubelet 1.12.5 #73587

Open **szuecs** opened this issue 10 days ago · 21 comments

szuecs commented 10 days ago • edited — Contributor

What happened:
After upgrading to kubernetes 1.12.5 we observe failing nodes, that are caused by kubelet eating all over the memory after some time.

https://github.com/kubernetes/kubernetes/issues/73587

INCIDENT #4: THE PATCH

szuecs commented 10 days ago — Contributor

For everyone that finds this issue and needs a patch to disable the reflector metrics:

```
diff --git c/pkg/util/reflector/prometheus/prometheus.go i/pkg/util/reflector/prometheus/prom
index 958a0007cd..63657e9c55 100644
--- c/pkg/util/reflector/prometheus/prometheus.go
+++ i/pkg/util/reflector/prometheus/prometheus.go
@@ -85,8 +85,6 @@ func init() {
        prometheus.MustRegister(watchDuration)
        prometheus.MustRegister(itemsPerWatch)
        prometheus.MustRegister(lastResourceVersion)
-
-       cache.SetReflectorMetricsProvider(prometheusMetricsProvider{})
 }

 type prometheusMetricsProvider struct{}
```

👍 4

https://github.com/kubernetes/kubernetes/issues/73587

INCIDENT

#5

INCIDENT #5: IMPACT

Error during Pod creation:

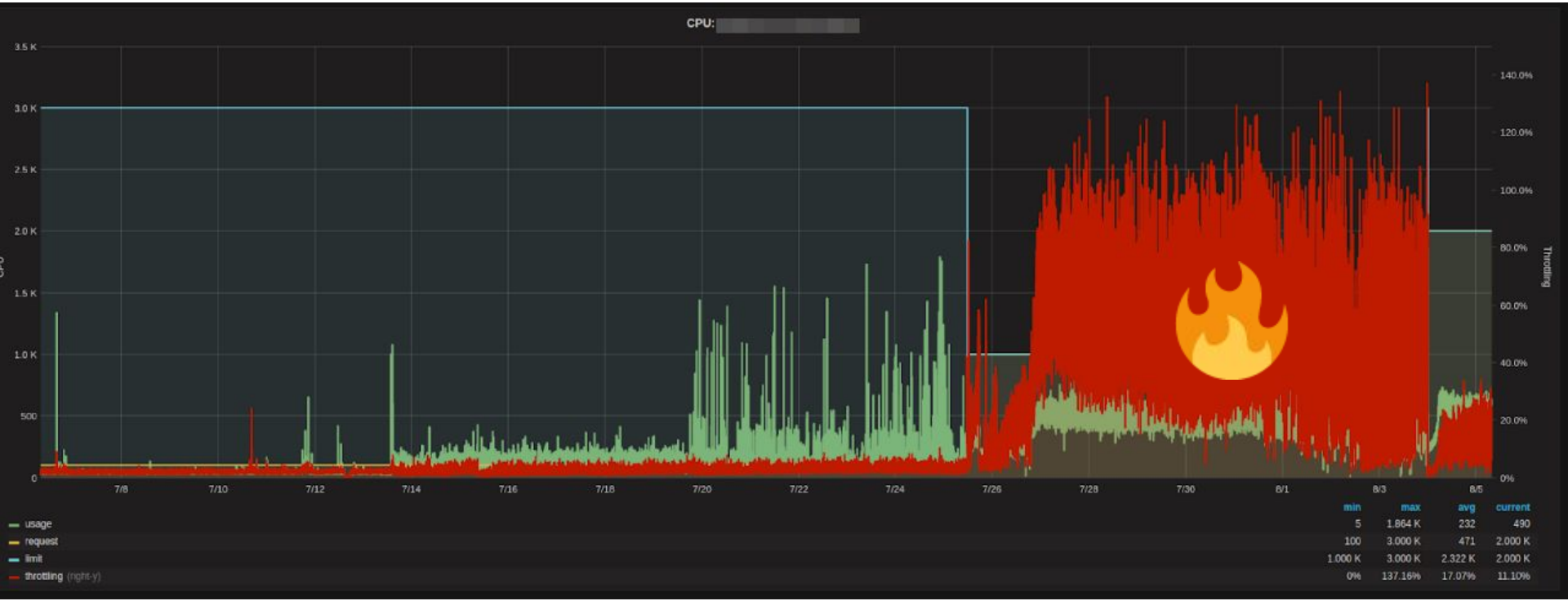
```
MountVolume.SetUp failed for volume
 "outfit-delivery-api-credentials" :
 secrets "outfit-delivery-api-credentials" not found
```

⇒ All new Kubernetes deployments fail 

INCIDENT #5: CREDENTIALS QUEUE

```
17:30:07 | [pool-6-thread-1    ] | Current queue size: 7115, current number of active workers: 20
17:31:07 | [pool-6-thread-1    ] | Current queue size: 7505, current number of active workers: 20
17:32:07 | [pool-6-thread-1    ] | Current queue size: 7886, current number of active workers: 20
..
17:37:07 | [pool-6-thread-1    ] | Current queue size: 9686, current number of active workers: 20
..
17:44:07 | [pool-6-thread-1    ] | Current queue size: 11976, current number of active workers: 20
..
19:16:07 | [pool-6-thread-1    ] | Current queue size: 58381, current number of active workers: 20
```

INCIDENT #5: CPU THROTTLING



zalando

INCIDENT #5: WHAT HAPPENED

Scaled down IAM provider
to reduce **Slack**

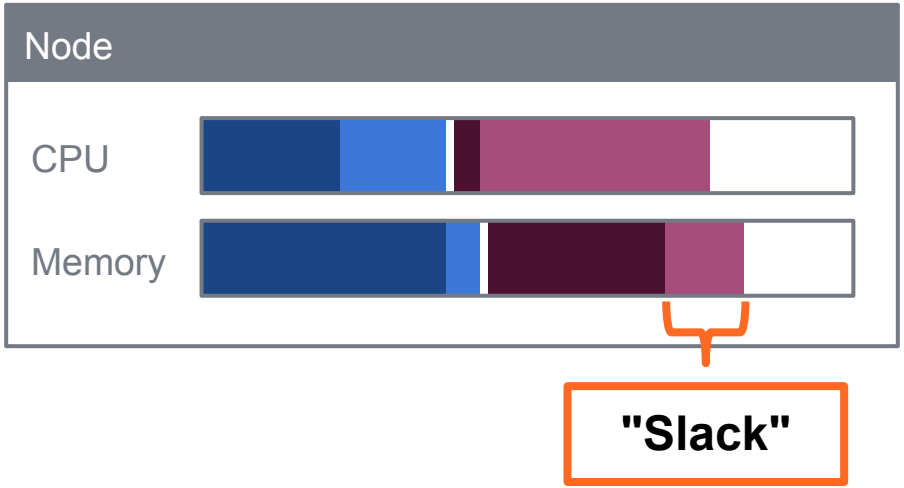
\+ Number of deployments increased

⇒ Process could not process credentials fast enough

SLACK

CPU/memory requests "block" resources on nodes.

Difference between actual usage and requests → **Slack**



zalando

DISABLING CPU THROTTLING

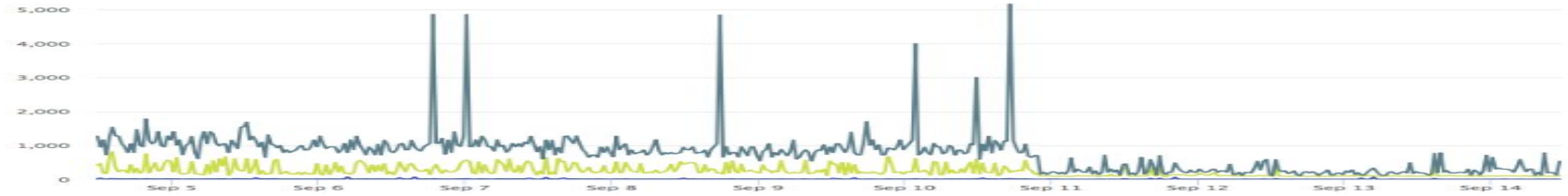
```
kubelet ... --cpu-cfs-quota=false
```

[Announcement] CPU limits will be disabled

TLDR: to **improve performance** and efficiency we will disable CPU limits in Kubernetes clusters. Please **revise your resource requests** if necessary.

We're going to disable CPU limits in the Kubernetes clusters. According to our experiments, this should improve the latencies for your applications and allow us to use the nodes more efficiently. To ensure that your applications get their fair share of CPU, please update your deployments' resource requests so they match the actual usage. You can use the Application Dashboard to find out how much CPU your applications use.

⇒ Ingress Latency Improvements



zalando

MANAGED KUBERNETES?

WILL MANAGED K8S SAVE US?

Amazon EKS Announces 99.9% Service Level Agreement

Posted On: Jan 16, 2019

AWS has published a service level agreement (SLA) for Amazon Elastic Container Service for Kubernetes (EKS), which provides availability guarantees for Amazon EKS.

GKE: monthly uptime percentage at 99.95% for regional clusters

WILL MANAGED K8S SAVE US?

NO

(not really)

e.g. AWS EKS uptime SLA is only for API server

PRODUCTION PROOFING AWS EKS

- Networking
- Networking—Limited pod capacity per subnet & VPC
- Networking—Limited pod capacity per worker node
- Networking—Kubernetes scheduler is unaware about actual IP availability
- Networking—Some pods cannot be accessed from peered networks by default
- Default worker AMI
- AMI—Based on Amazon Linux 2
- AMI—No docker log rotation
- AMI—Docker freezes
- AMI—Corrupted disk statistics
- Authentication and authorization
- Auth—RBAC enabled
- Auth—AWS IAM authentication
- Auth—API Server endpoint is public
- Limited availability
- Alpha Kubernetes features are disabled
- CronJobs are problematic
- CronJobs—Backoff limit does not work
- CronJobs don't work well with the Kubernetes network plugin
- Single kube-dns pod by default

List of things you might want to look at for EKS in production

https://medium.com/glia-tech/productionproofing-eks-ed52951ffd6c

zalando

AWS EKS IN PRODUCTION

DNS lookup scaling

Out of the box, AWS provides a `kube-dns` deployment containing a single pod of scale `1`. After a week or so in production, I was skimming our logs and came across this beauty. This reinforced something I had seen in our exception handling system.

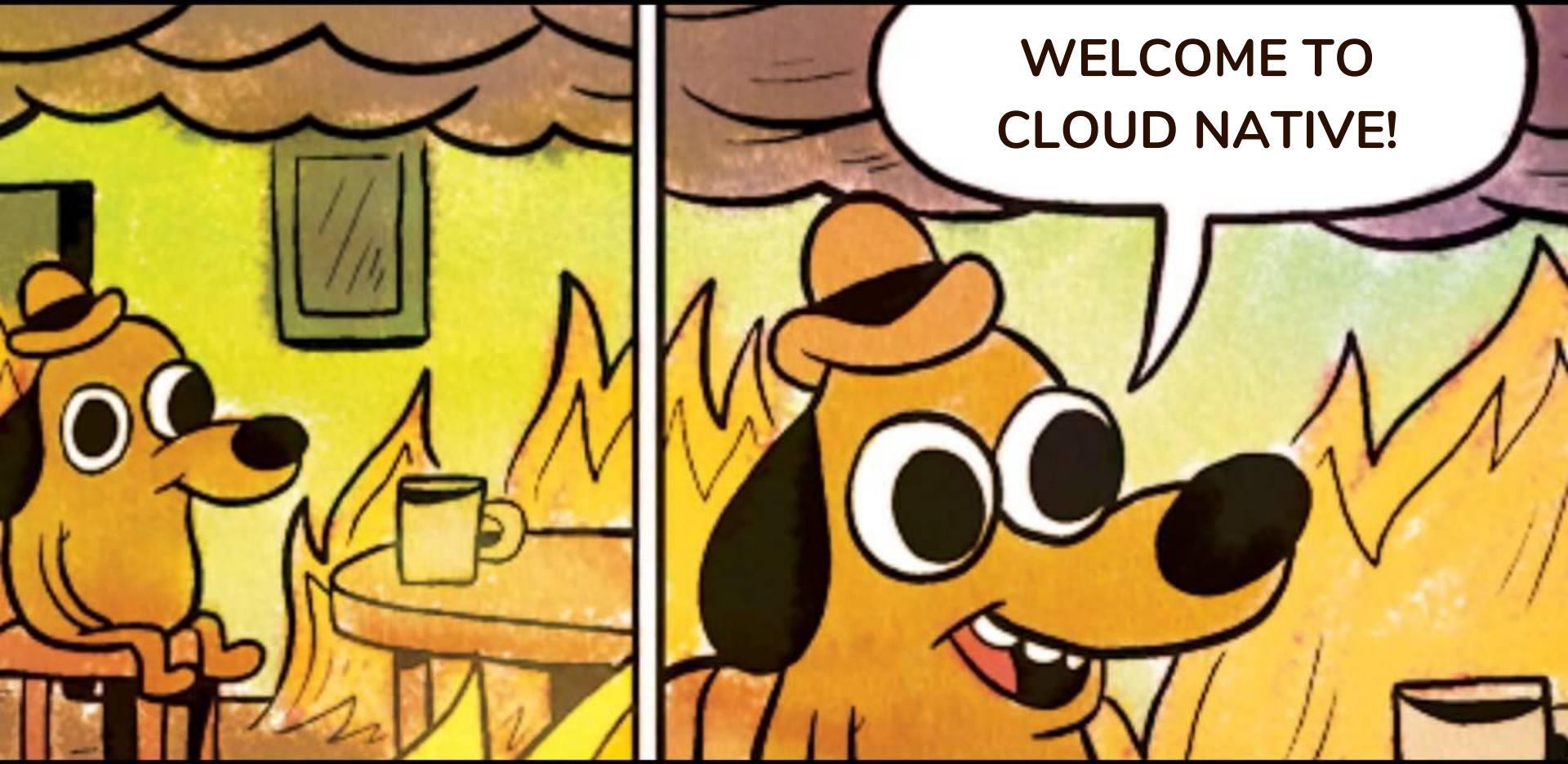
```
dnsmasq[14]: Maximum number of concurrent DNS queries reached (max: 150)
```

https://kubedex.com/90-days-of-aws-eks-in-production/

DOCKER.. (ON GKE)

```
25  # We simply kill the process when there is a failure. Another systemd service will
26  # automatically restart the process.
27  function docker_monitoring {
28    while [ 1 ]; do
29      if ! timeout 10 docker ps > /dev/null; then
30        echo "Docker daemon failed!"
31        pkill docker
32        # Wait for a while, as we don't want to kill it again before it is really up.
33        sleep 30
34      else
35        sleep "${SLEEP_SECONDS}"
36      fi
37    done
38  }
```

https://github.com/kubernetes/kubernetes/blob/8fd414537b5143ab039cb910590237cabf4af783/cluster/gce/gci/health-monitor.sh#L29


WELCOME TO
CLOUD NATIVE!


Henning Jacobs
@try_except_
Honest role description by @mikkeloscar
Tweet übersetzen
Folge ich
Mikkel Oscar Lyderik Larsen
@mikkeloscar Folgt dir
Writing #Kubernetes related post-mortems
@ZalandoTech
Berlin github.com/mikkeloscar
56 Folge ich 88 Follower
23:37 - 5. Juni 2018
1 Retweet 40 „Gefällt mir"-Angaben
1
40

zalando

Henning Jacobs
@try_except_

Follow

I started a compilation of #Kubernetes #Failure Stories --- the list is still short, please help! Thanks to @obeattie and @adman65 for their public postmortems

hjacobs/kubernetes-failure-stories
Compilation of public failure/horror stories related to Kubernetes - hjacobs/kubernetes-failure-stories
github.com

3:15 AM - 19 Jan 2019

86 Retweets **130** Likes

7 86 130

KUBERNETES FAILURE STORIES

- Total DNS outage in Kubernetes cluster - Zalando - postmortem 2019
  - involved: AWS, DNS, CoreDNS, `OOMKill`, `ndots:5`, HTTP retries
  - impact: production outage
- Maximize learnings from a Kubernetes cluster failure - NU.nl - blog post 2019
  - involved: AWS, `NotReady` nodes, `SystemOOM`, Helm, ElastAlert, no resource limits set
  - impact: user experience affected for internally used tools and dashboards
- Kubernetes Load Balancer Configuration - Beware when draining nodes - DevOps Hof - blog post 2019
  - involved: GCP Load Balancer, `externalTrafficPolicy`, ingress-nginx
  - impact: total ingress traffic outage
- On Infrastructure at Scale: A Cascading Failure of Distributed Systems - Target - Medium post January 2019
  - involved: on-premise, Kafka, large cluster, Consul, Docker daemon, high CPU usage
  - impact: development environment outage
- Running Kubernetes in Production: A Million Ways to Crash Your Cluster - Zalando - DevOpsCon Munich 2018
  - involved: AWS, Ingress, CronJob, etcd, flannel, Docker, CPU throttling
  - impact: production outages
- Outages? Downtime? - Veracode - blog post 2018
  - involved: AWS, AWS IAM, region migration, kubespray, Terraform, pod CIDR
  - impact: QA/dev cluster outage
- NRE Labs Outage Post-Mortem - NRE Labs - blog post 2018
  - involved: GCP, kubeadm, etcd, Terraform, `livenessProbe`
  - impact: production outage
- A Perfect DNS Storm - Toyota Connected - blog post 2018
  - involved: Azure, DNS, `ndots:5`, Alpine musl libc
  - impact: DNS resolution failures
- Kubernetes and the Menace ELB, the tale of an outage - Turnitin - blog post 2018
  - involved: AWS, kube-aws, ELB dynamic IPs, API server, kubelet, `NotReady` nodes
  - impact: 15 minutes cluster outage
- Moving the Entire Stack to K8s Within a Year – Lessons Learned - ThredUP - DevOpsStage 2018
  - involved: AWS, kops, HAProxy, `livenessProbe`, DNS, too many open files
  - impact: unknown outages, DNS errors
- AirMap Platform Service Outage - AirMap - incident report 2018
  - involved: Azure, `NotReady` nodes, kubelet PLEG, CNI
  - impact: production AirMap platform outage
- Anatomy of a Production Kubernetes Outage - Monzo - KubeCon Europe 2018
  - involved: AWS, etcd, Linkerd, `NullPointerException`, gRPC client, services without endpoints, incompatible Kubernetes API change
  - impact: production ledger/platform outage
- 101 Ways to "Break and Recover" Kubernetes Cluster - Oath/Yahoo - KubeCon Europe 2018
  - involved: on-premise, namespace deletion, domain name collision, `NotReady` nodes, etcd empty dir, TLS certificate refresh, DNS issues, OOM
  - impact: unknown cluster outages
- 101 Ways to Crash Your Cluster - Nordstrom - KubeCon North America 2017
  - involved: AWS, `NotReady` nodes, OOM, eviction thresholds, ELB dynamic IPs, kubelet, cluster autoscaler, etcd split
  - impact: full production cluster outage, other outages
- Major Outage: Current account payments may fail - Monzo - Monzo Community post 2017
  - involved: AWS, etcd, Linkerd, `NullPointerException`, services without endpoints
  - impact: major production outage, full platform outage, current account payments fail
- Fallacies of Distributed Computing with Kubernetes on AWS - Zalando - AWS User Group Hamburg October 2017
  - involved: AWS, unhealthy nodes, Ingress, CronJob
  - impact: production outage
- Search and Reporting Outage - Universe - incident report 2017
  - involved: Job, `RestartPolicy`, consume node resources
  - impact: production Universe search and reporting outage
- Our First Kubernetes Outage - Saltside - blog post 2017
  - involved: AWS, kops, Helm, `NotReady` nodes, resource exhaustion
  - impact: nonproduction cluster outage
- Our Failure Migrating to Kubernetes - Saltside - blog post 2017
  - involved: AWS, kops, ELB, `BackendConnectionErrors`, `LoadBalancer` service
  - impact: aborted application migration
- SaleMove US System Issue - SaleMove - incident report 2017
  - involved: AWS, ELB dynamic IPs, DNS `A` record for master, API server
  - impact: production issues with SaleMove US System

20 failure stories so far

What about yours?

github.com/hjacobs/kubernetes-failure-stories

zalando

zalando

QUESTIONS?

HENNING JACOBS

HEAD OF

DEVELOPER PRODUCTIVITY

henning@zalando.de

@try_except_

Illustrations by @01k