zalando

Kubernetes Failure Stories

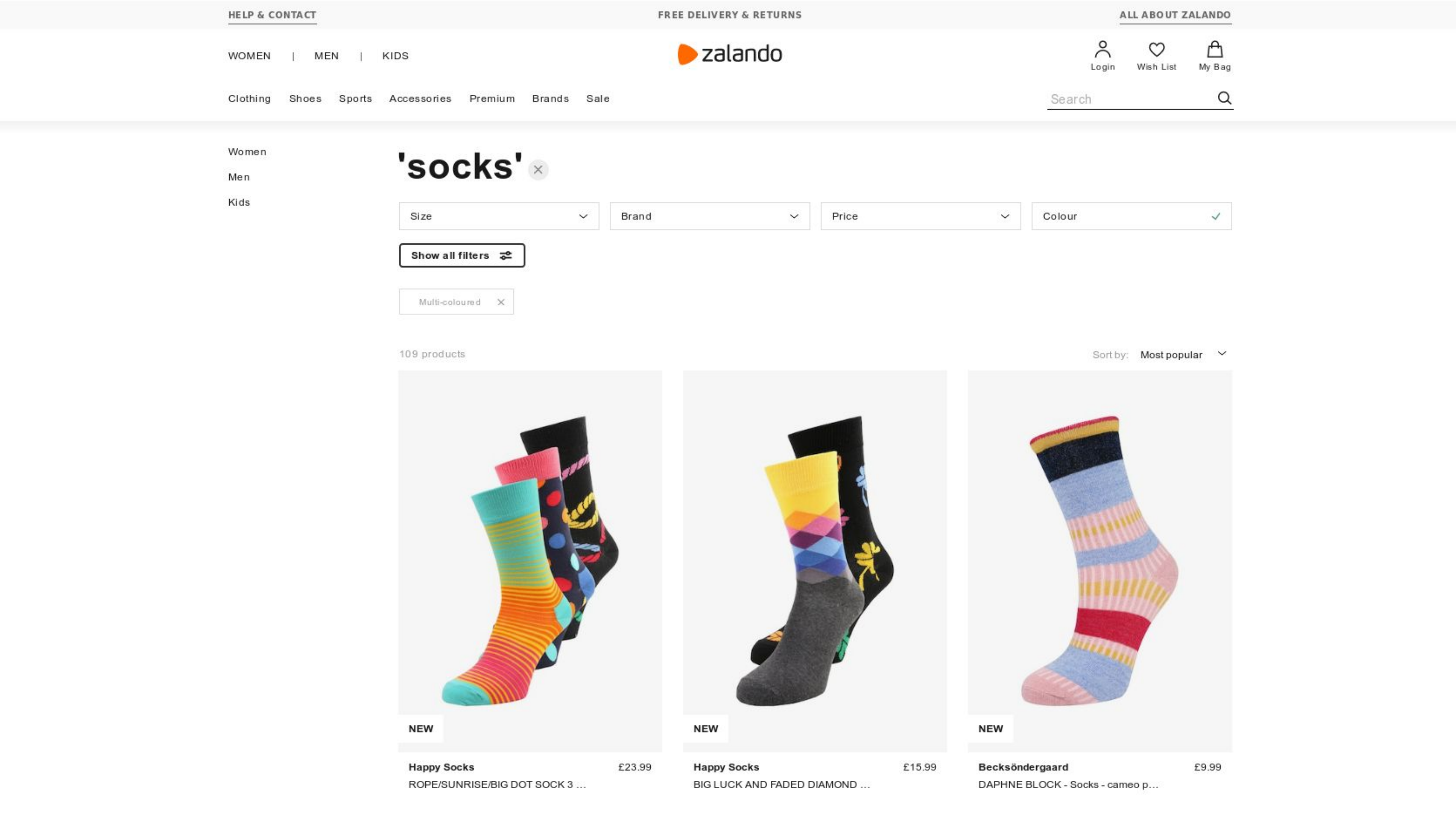
HENNING JACOBS

@try_except_

KubeCon | CloudNativeCon

Europe 2019

You sell socks, ergo: build your own Kubernetes!



HELP & CONTACT FREE DELIVERY & RETURNS ALL ABOUT ZALANDO

WOMEN | MEN | KIDS

zalando

Login Wish List My Bag

Clothing Shoes Sports Accessories Premium Brands Sale

Search

Women

Men

Kids

'socks' ✕

Size Brand Price Colour ✓

Show all filters

Multi-coloured ✕

109 products

Sort by: Most popular

NEW

Happy Socks £23.99

ROPE/SUNRISE/BIG DOT SOCK 3 ...

NEW

Happy Socks £15.99

BIG LUCK AND FADED DIAMOND ...

NEW

Becksöndergaard £9.99

DAPHNE BLOCK - Socks - cameo p...

ZALANDO AT A GLANCE

~ 5.4 billion EUR

revenue 2018

> 250 million visits per month

> 26 million active customers

> 300.000 product choices

> 15.000 employees in Europe

> 79% of visits via mobile devices

~ 2.000 brands

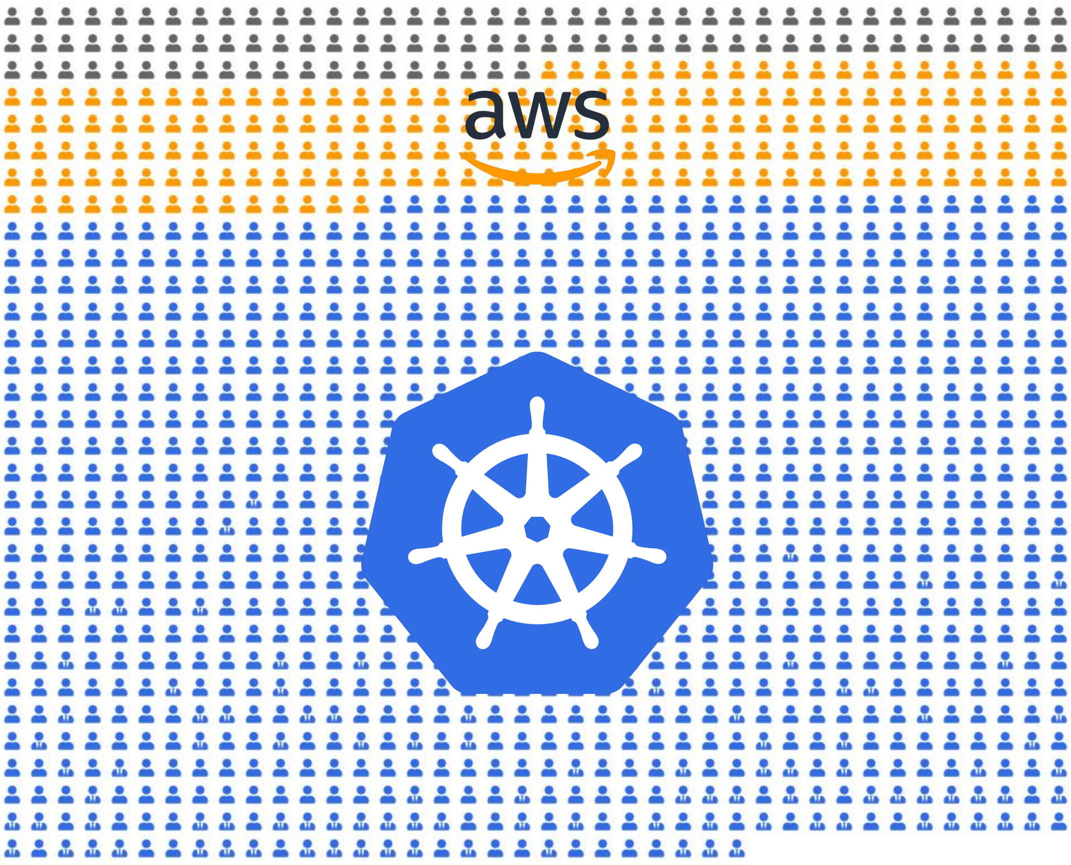
17 countries

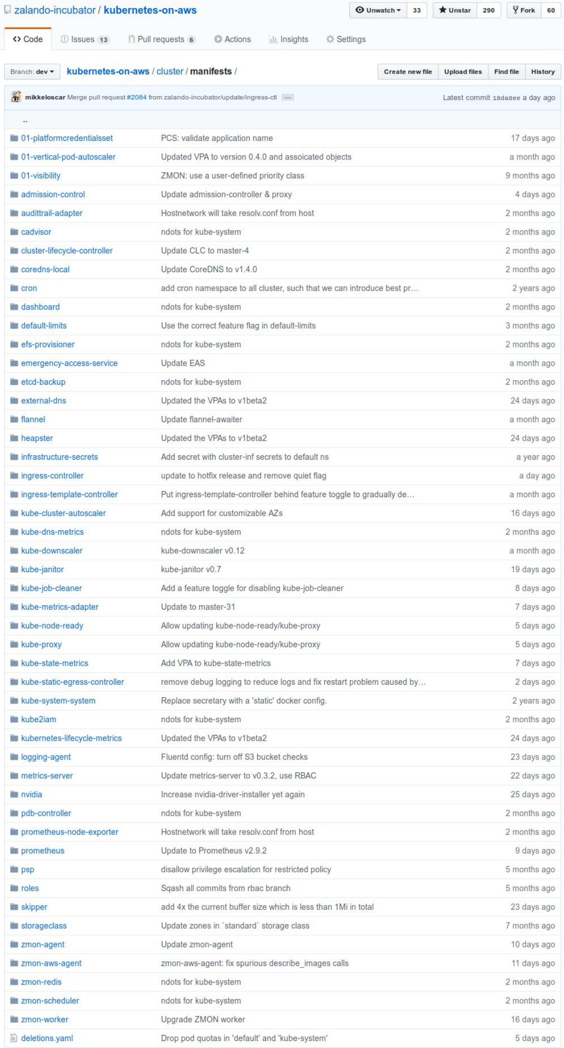
SCALE

380 Accounts

118 Clusters

DEVELOPERS USING KUBERNETES





kubernetes-on-aws / cluster / manifests /

mikkeloscar Merge pull request #2084 from zalando-incubator/update/ingress-ctl … Latest commit 18da8ee a day ago

..		
01-platformcredentialsset	PCS: validate application name	17 days ago
01-vertical-pod-autoscaler	Updated VPA to version 0.4.0 and assoicated objects	a month ago
01-visibility	ZMON: use a user-defined priority class	9 months ago
admission-control	Update admission-controller & proxy	4 days ago
audittrail-adapter	Hostnetwork will take resolv.conf from host	2 months ago
cadvisor	ndots for kube-system	2 months ago
cluster-lifecycle-controller	Update CLC to master-4	2 months ago
coredns-local	Update CoreDNS to v1.4.0	2 months ago
cron	add cron namespace to all cluster, such that we can introduce best pr…	2 years ago
dashboard	ndots for kube-system	2 months ago
default-limits	Use the correct feature flag in default-limits	3 months ago
efs-provisioner	ndots for kube-system	2 months ago
emergency-access-service	Update EAS	a month ago
etcd-backup	ndots for kube-system	2 months ago
external-dns	Updated the VPAs to v1beta2	24 days ago
flannel	Update flannel-awaiter	a month ago
heapster	Updated the VPAs to v1beta2	24 days ago
infrastructure-secrets	Add secret with cluster-inf secrets to default ns	a year ago
ingress-controller	update to hotfix release and remove quiet flag	a day ago
ingress-template-controller	Put ingress-template-controller behind feature toggle to gradually de…	a month ago
kube-cluster-autoscaler	Add support for customizable AZs	16 days ago
kube-dns-metrics	ndots for kube-system	2 months ago
kube-downscaler	kube-downscaler v0.12	a month ago
kube-janitor	kube-janitor v0.7	19 days ago
kube-job-cleaner	Add a feature toggle for disabling kube-job-cleaner	8 days ago
kube-metrics-adapter	Update to master-31	7 days ago
kube-node-ready	Allow updating kube-node-ready/kube-proxy	5 days ago
kube-proxy	Allow updating kube-node-ready/kube-proxy	5 days ago
kube-state-metrics	Add VPA to kube-state-metrics	7 days ago
kube-static-egress-controller	remove debug logging to reduce logs and fix restart problem caused by…	2 days ago
kube-system-system	Replace secretary with a 'static' docker config.	2 years ago
kube2iam	ndots for kube-system	2 months ago
kubernetes-lifecycle-metrics	Updated the VPAs to v1beta2	24 days ago
logging-agent	Fluentd config: turn off S3 bucket checks	23 days ago
metrics-server	Update metrics-server to v0.3.2, use RBAC	22 days ago
nvidia	Increase nvidia-driver-installer yet again	25 days ago
pdb-controller	ndots for kube-system	2 months ago
prometheus-node-exporter	Hostnetwork will take resolv.conf from host	2 months ago
prometheus	Update to Prometheus v2.9.2	9 days ago
psp	disallow privilege escalation for restricted policy	5 months ago
roles	Sqash all commits from rbac branch	5 months ago
skipper	add 4x the current buffer size which is less than 1Mi in total	23 days ago
storageclass	Update zones in `standard` storage class	7 months ago
zmon-agent	Update zmon-agent	10 days ago
zmon-aws-agent	zmon-aws-agent: fix spurious describe_images calls	11 days ago
zmon-redis	ndots for kube-system	2 months ago
zmon-scheduler	ndots for kube-system	2 months ago
zmon-worker	Upgrade ZMON worker	16 days ago
deletions.yaml	Drop pod quotas in 'default' and 'kube-system'	5 days ago

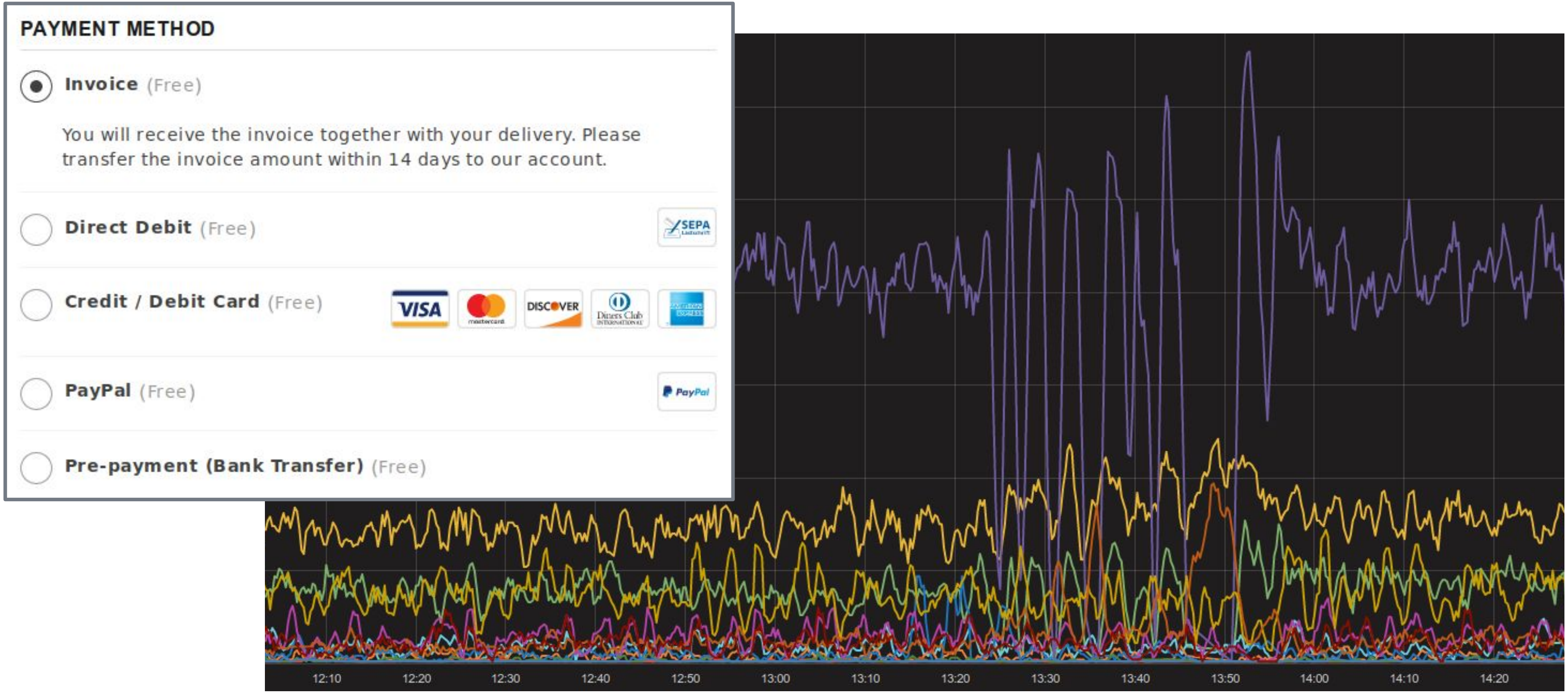
47+ cluster components

INCIDENTS ARE FINE

INCIDENT

#1

INCIDENT #1: CUSTOMER IMPACT



PAYMENT METHOD

● **Invoice** (Free)

You will receive the invoice together with your delivery. Please transfer the invoice amount within 14 days to our account.

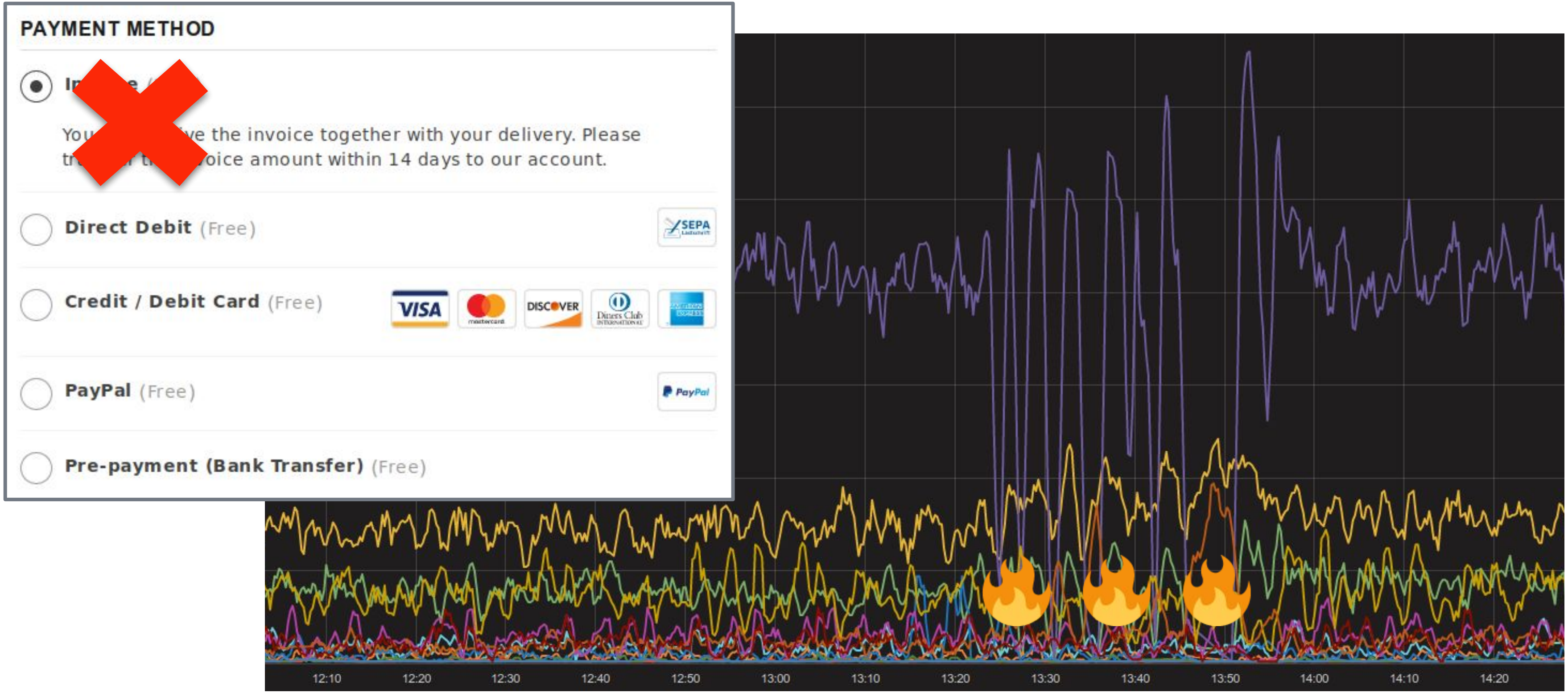
○ **Direct Debit** (Free)

○ **Credit / Debit Card** (Free)

○ **PayPal** (Free)

○ **Pre-payment (Bank Transfer)** (Free)

INCIDENT #1: CUSTOMER IMPACT



PAYMENT METHOD

● In e (

You ve the invoice together with your delivery. Please tr t oice amount within 14 days to our account.

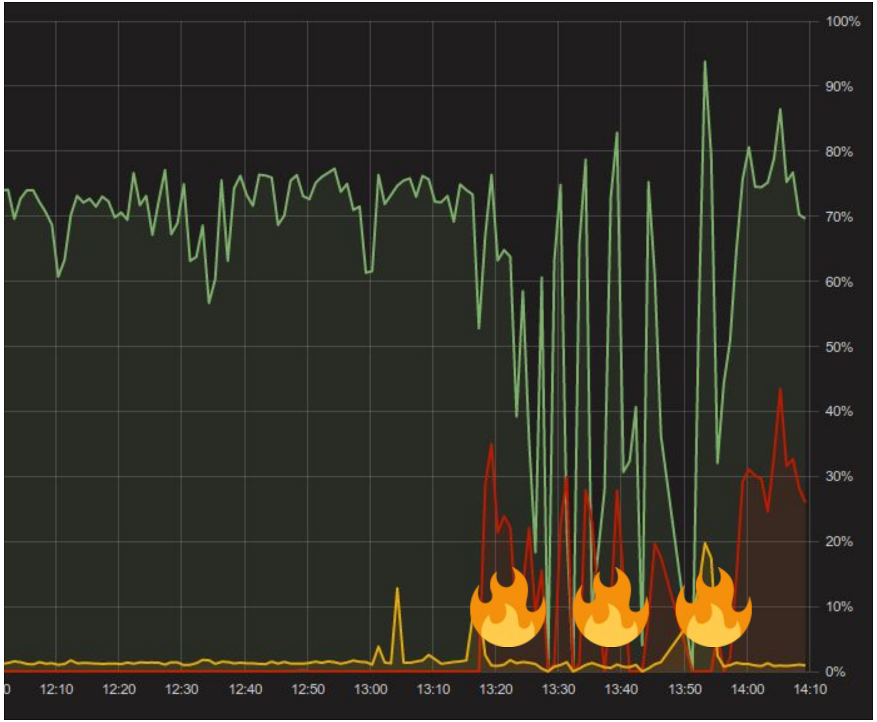
○ Direct Debit (Free)

○ Credit / Debit Card (Free)

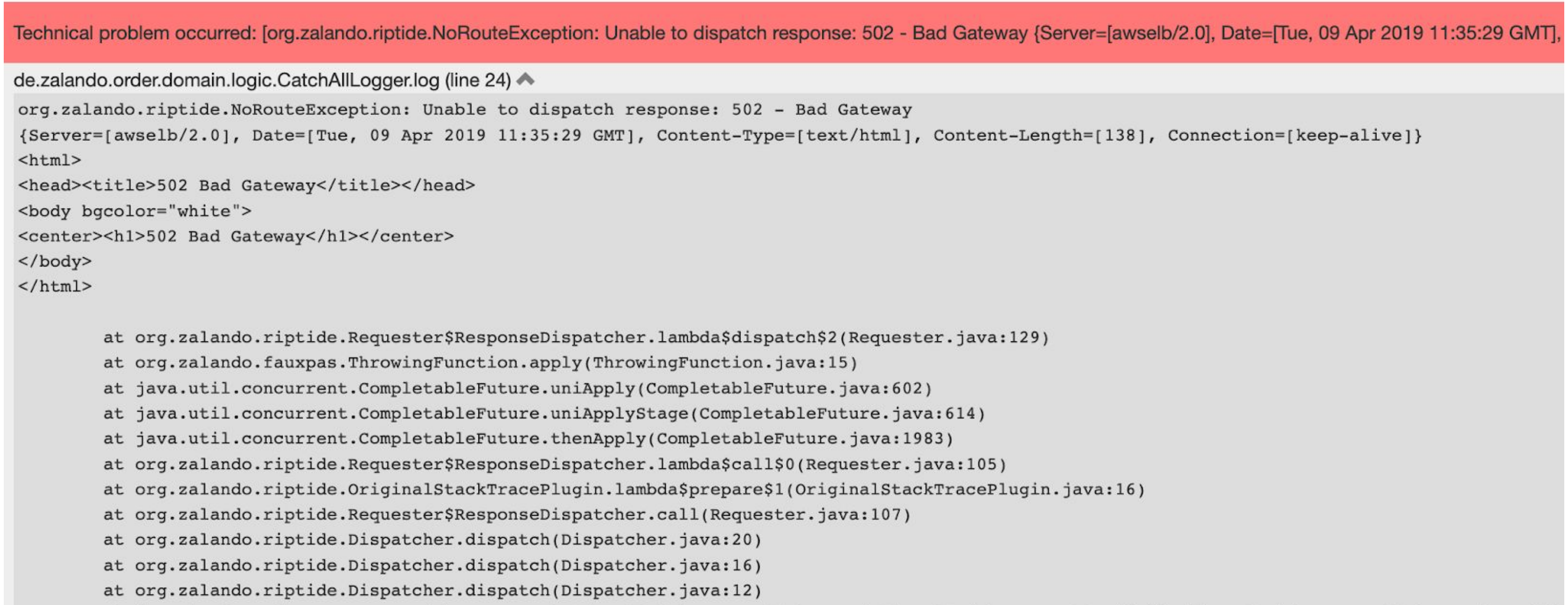
○ PayPal (Free)

○ Pre-payment (Bank Transfer) (Free)

INCIDENT #1: INGRESS ERRORS



INCIDENT #1: AWS ALB 502



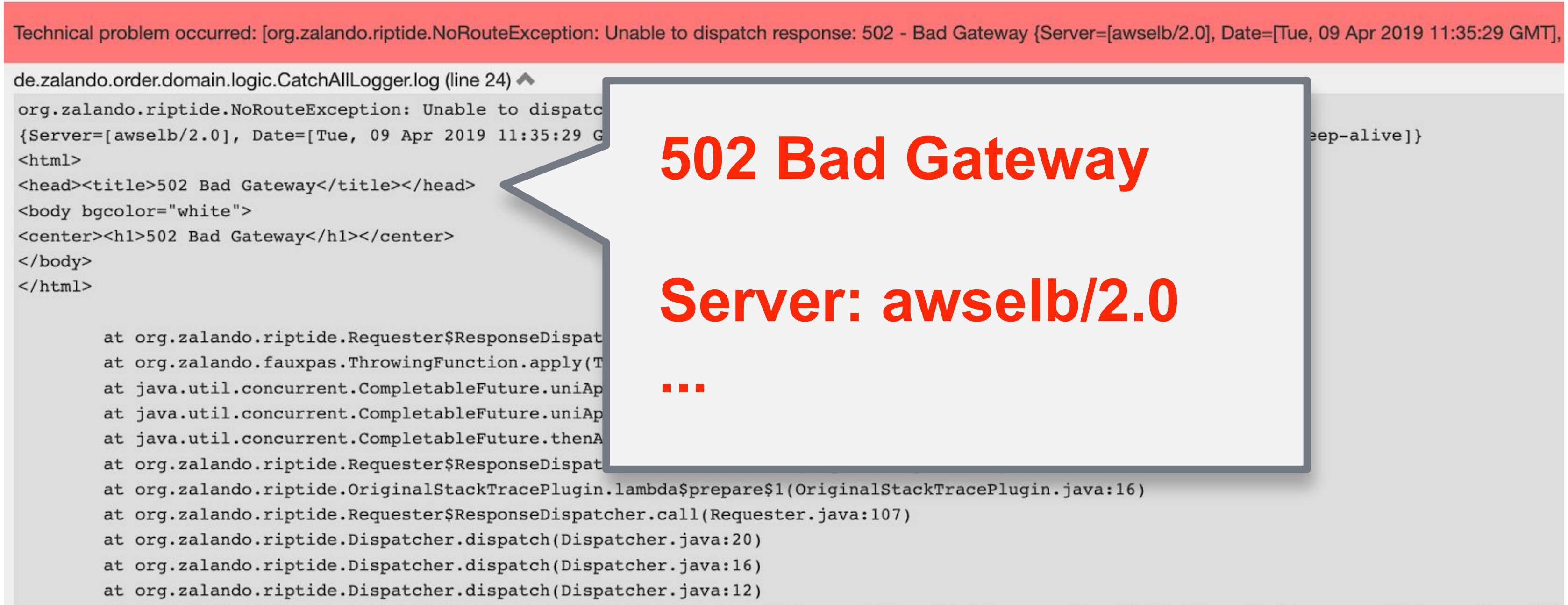
Technical problem occurred: [org.zalando.riptide.NoRouteException: Unable to dispatch response: 502 - Bad Gateway {Server=[awselb/2.0], Date=[Tue, 09 Apr 2019 11:35:29 GMT],

de.zalando.order.domain.logic.CatchAllLogger.log (line 24)

```
org.zalando.riptide.NoRouteException: Unable to dispatch response: 502 - Bad Gateway
{Server=[awselb/2.0], Date=[Tue, 09 Apr 2019 11:35:29 GMT], Content-Type=[text/html], Content-Length=[138], Connection=[keep-alive]}
<html>
<head><title>502 Bad Gateway</title></head>
<body bgcolor="white">
<center><h1>502 Bad Gateway</h1></center>
</body>
</html>

        at org.zalando.riptide.Requester$ResponseDispatcher.lambda$dispatch$2(Requester.java:129)
        at org.zalando.fauxpas.ThrowingFunction.apply(ThrowingFunction.java:15)
        at java.util.concurrent.CompletableFuture.uniApply(CompletableFuture.java:602)
        at java.util.concurrent.CompletableFuture.uniApplyStage(CompletableFuture.java:614)
        at java.util.concurrent.CompletableFuture.thenApply(CompletableFuture.java:1983)
        at org.zalando.riptide.Requester$ResponseDispatcher.lambda$call$0(Requester.java:105)
        at org.zalando.riptide.OriginalStackTracePlugin.lambda$prepare$1(OriginalStackTracePlugin.java:16)
        at org.zalando.riptide.Requester$ResponseDispatcher.call(Requester.java:107)
        at org.zalando.riptide.Dispatcher.dispatch(Dispatcher.java:20)
        at org.zalando.riptide.Dispatcher.dispatch(Dispatcher.java:16)
        at org.zalando.riptide.Dispatcher.dispatch(Dispatcher.java:12)
```

github.com/zalando/riptide

INCIDENT #1: AWS ALB 502



Technical problem occurred: [org.zalando.riptide.NoRouteException: Unable to dispatch response: 502 - Bad Gateway {Server=[awselb/2.0], Date=[Tue, 09 Apr 2019 11:35:29 GMT],

de.zalando.order.domain.logic.CatchAllLogger.log (line 24)

```
org.zalando.riptide.NoRouteException: Unable to dispatc
{Server=[awselb/2.0], Date=[Tue, 09 Apr 2019 11:35:29 G                                          eep-alive]}
<html>
<head><title>502 Bad Gateway</title></head>
<body bgcolor="white">
<center><h1>502 Bad Gateway</h1></center>
</body>
</html>

        at org.zalando.riptide.Requester$ResponseDispat
        at org.zalando.fauxpas.ThrowingFunction.apply(T
        at java.util.concurrent.CompletableFuture.uniAp
        at java.util.concurrent.CompletableFuture.uniAp
        at java.util.concurrent.CompletableFuture.thenA
        at org.zalando.riptide.Requester$ResponseDispat
        at org.zalando.riptide.OriginalStackTracePlugin.lambda$prepare$1(OriginalStackTracePlugin.java:16)
        at org.zalando.riptide.Requester$ResponseDispatcher.call(Requester.java:107)
        at org.zalando.riptide.Dispatcher.dispatch(Dispatcher.java:20)
        at org.zalando.riptide.Dispatcher.dispatch(Dispatcher.java:16)
        at org.zalando.riptide.Dispatcher.dispatch(Dispatcher.java:12)
```

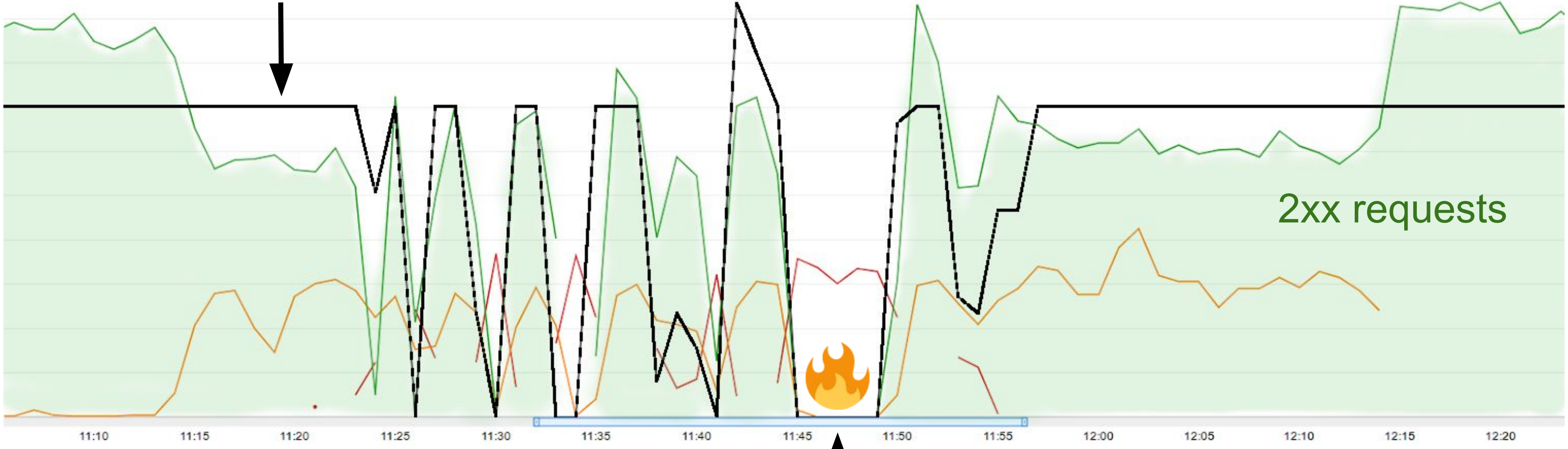
502 Bad Gateway

Server: awselb/2.0

...

github.com/zalando/riptide

INCIDENT #1: ALB HEALTHY HOST COUNT

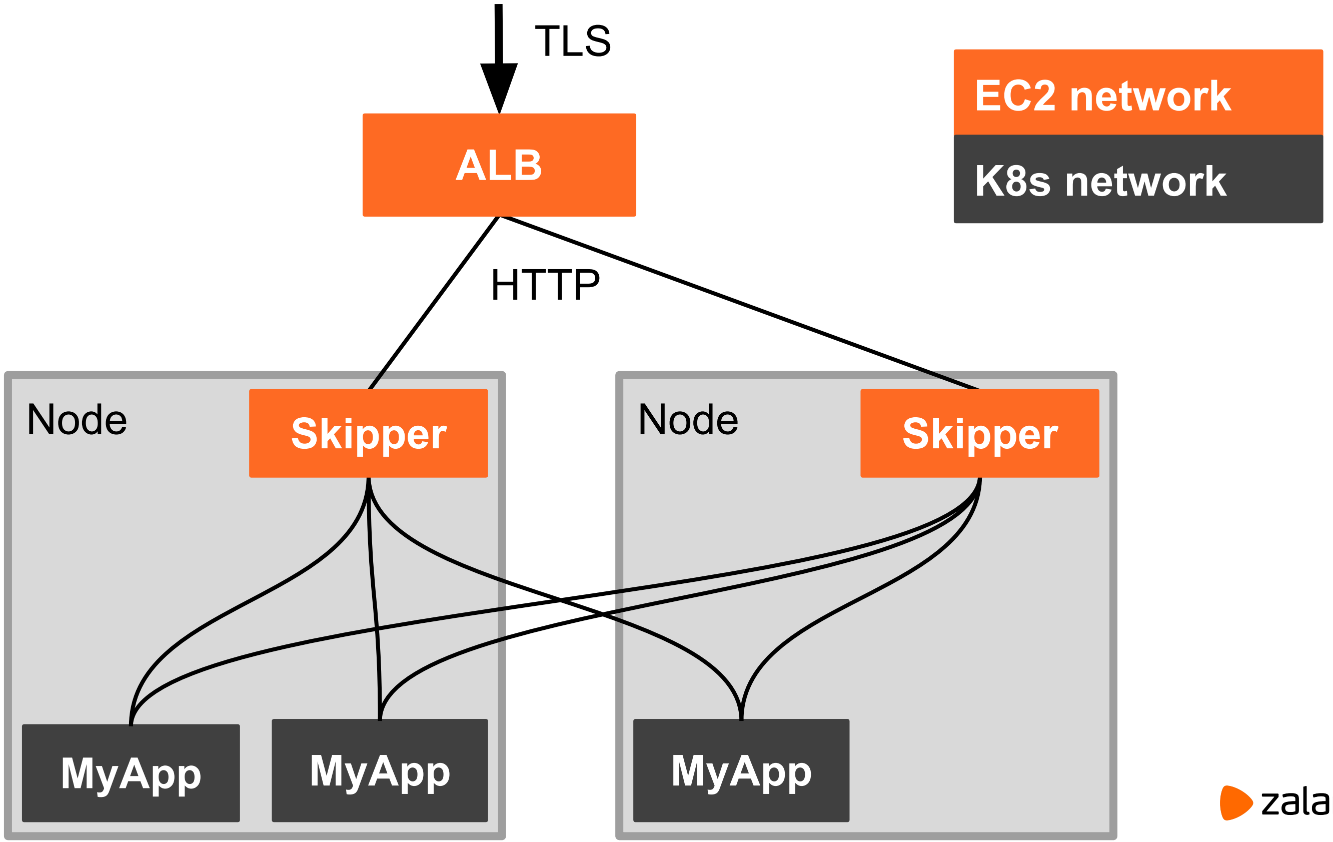


3 healthy hosts

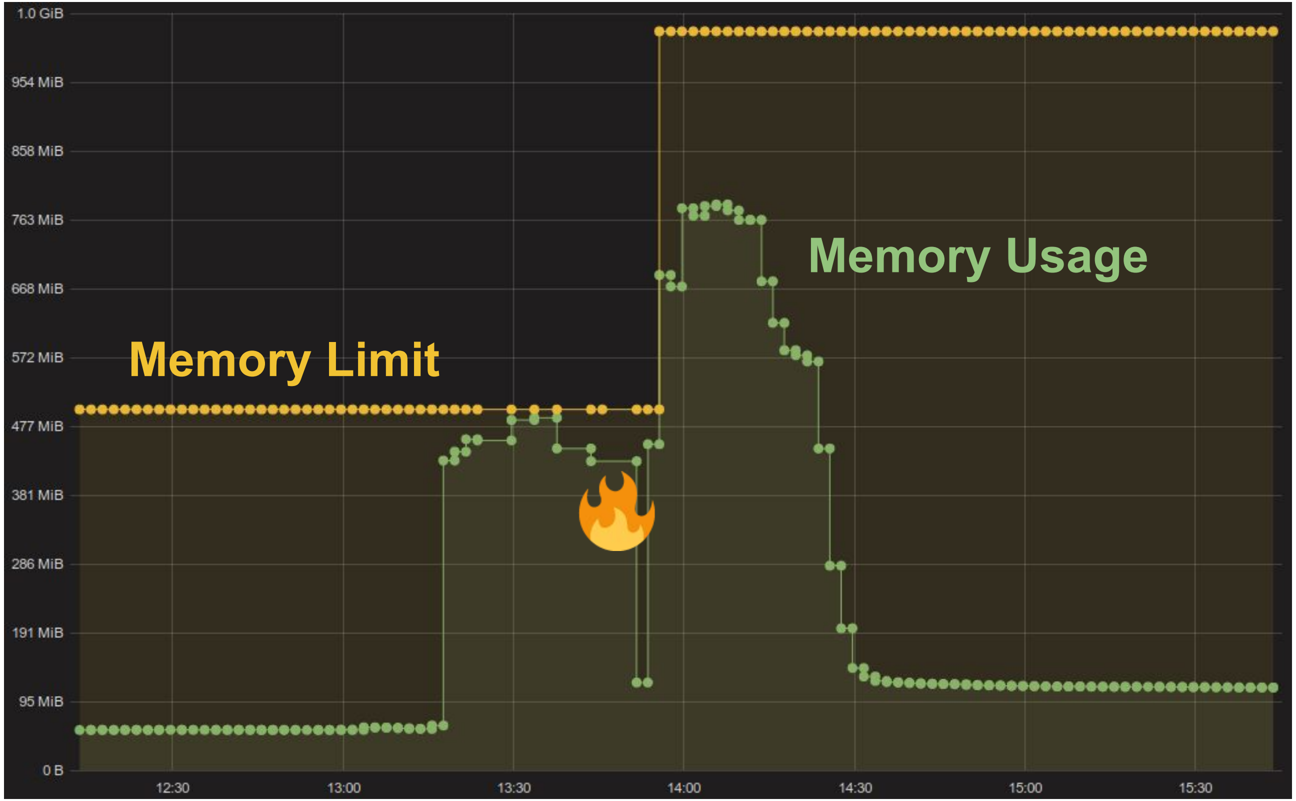
2xx requests

zero healthy hosts

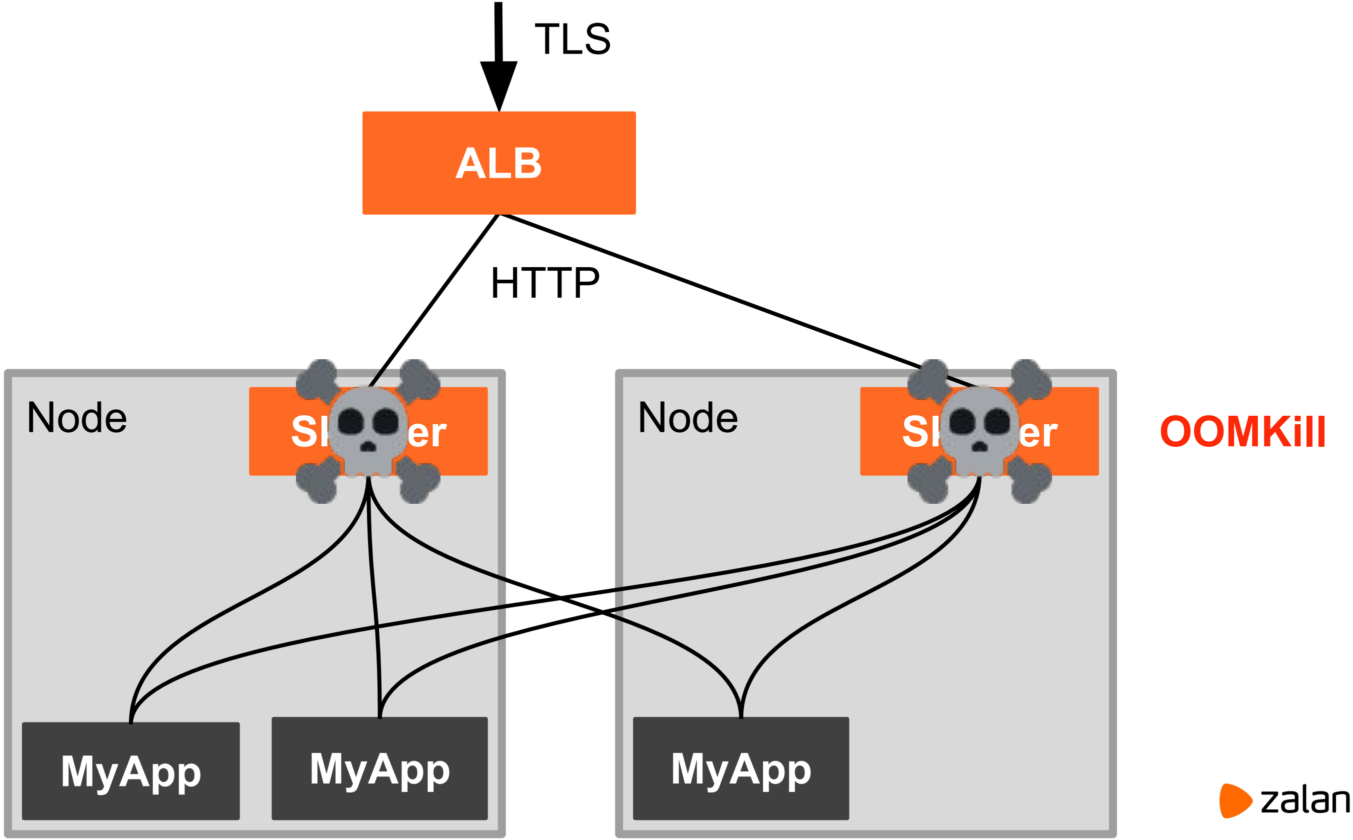
LIFE OF A REQUEST (INGRESS)



INCIDENT #1: SKIPPER MEMORY USAGE



INCIDENT #1: SKIPPER OOM



INCIDENT #1: CONTRIBUTING FACTORS

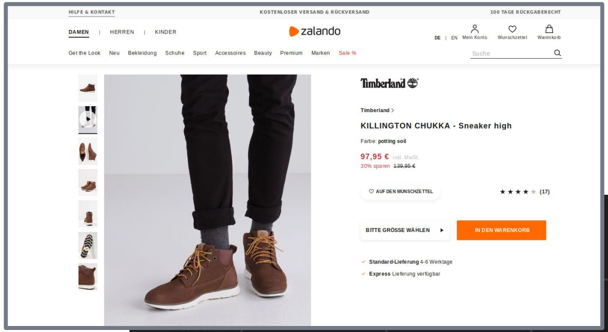
- Shared Ingress (per cluster)
- High latency of unrelated app (Solr) caused high number of in-flight requests
- Skipper creates goroutine per HTTP request. Goroutine costs 2kB memory + http.Request
- Memory limit was fixed at 500Mi (4x regular usage)

Fix for the memory issue in Skipper:
https://opensource.zalando.com/skipper/operation/operation/#scheduler

INCIDENT

#2

INCIDENT #2: CUSTOMER IMPACT



HILFE & KONTAKT

KOSTENLOSER VERSAND & RÜCKVERSAND

100 TAGE RÜCKGABERECHT

Timberland >

KILLINGTON CHUKKA - Sneaker high

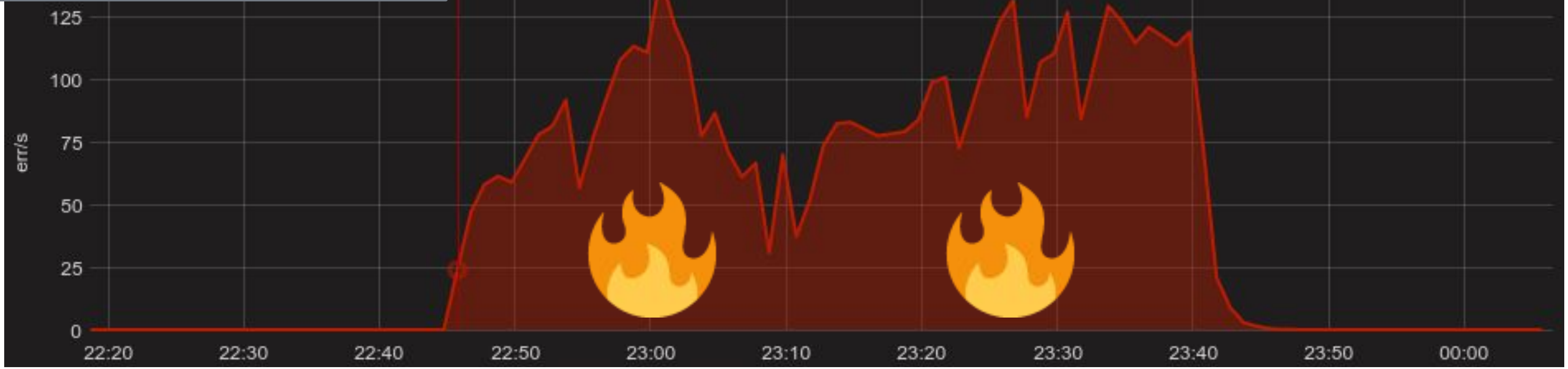
Farbe: **potting soil**

97,95 € inkl. MwSt.

30% sparen ~~139,95 €~~

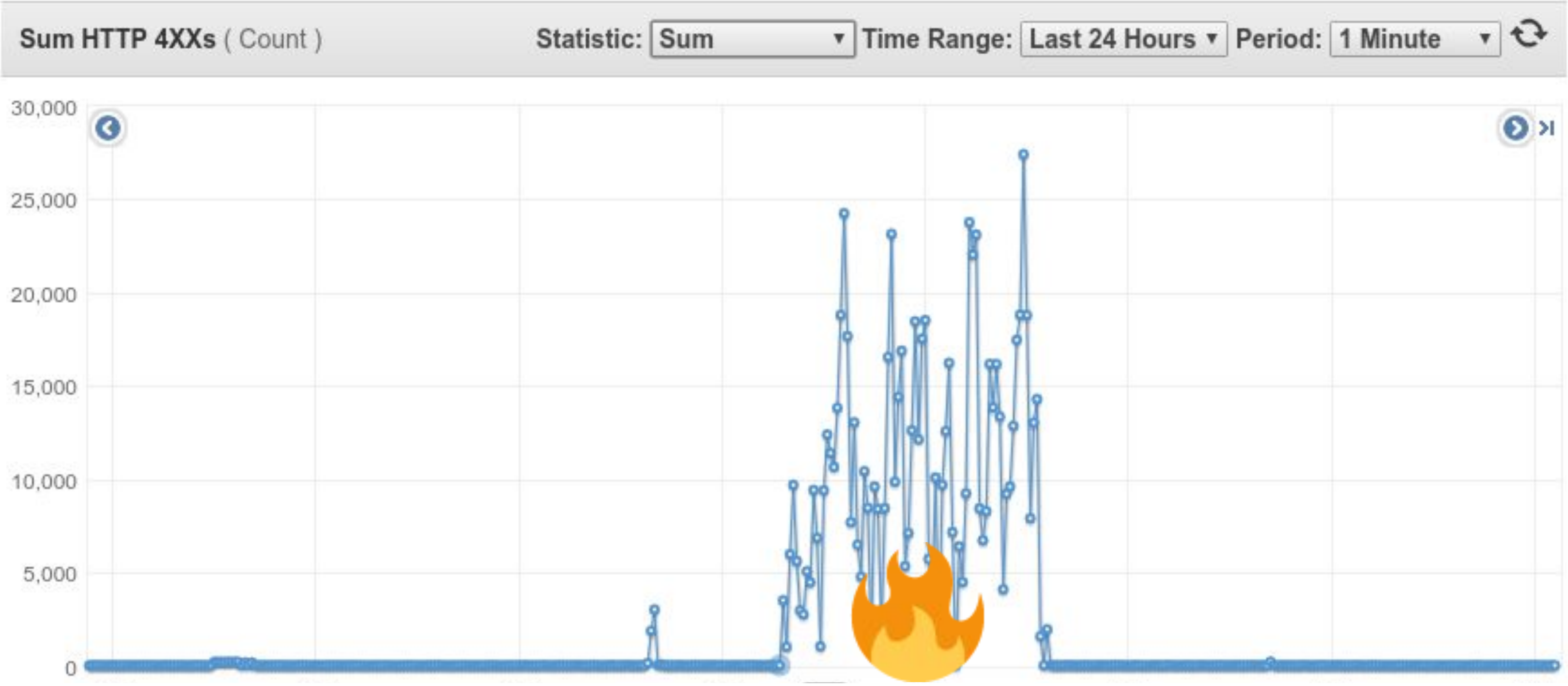
✓ **Standard-Lieferung** 4-6 Werktage

✓ **Express** Lieferung verfügbar



Fragments Errors on PDP (5XX)

INCIDENT #1: IAM RETURNING 404



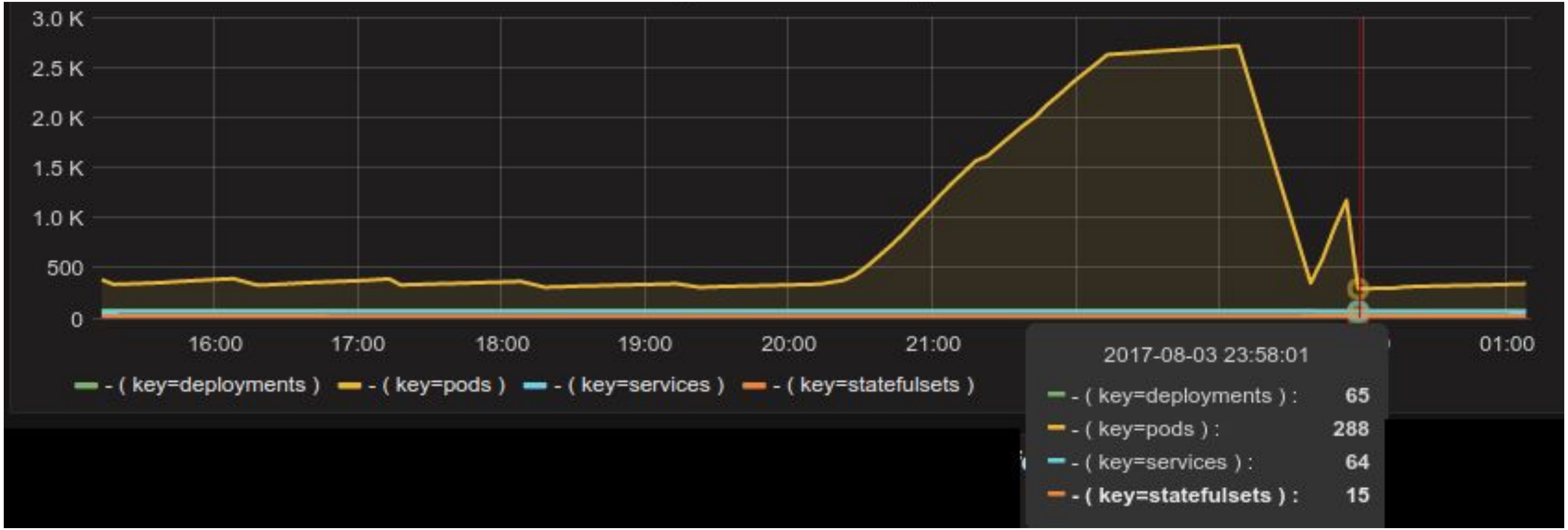
Sum HTTP 4XXs (Count)

Statistic: Sum

Time Range: Last 24 Hours

Period: 1 Minute

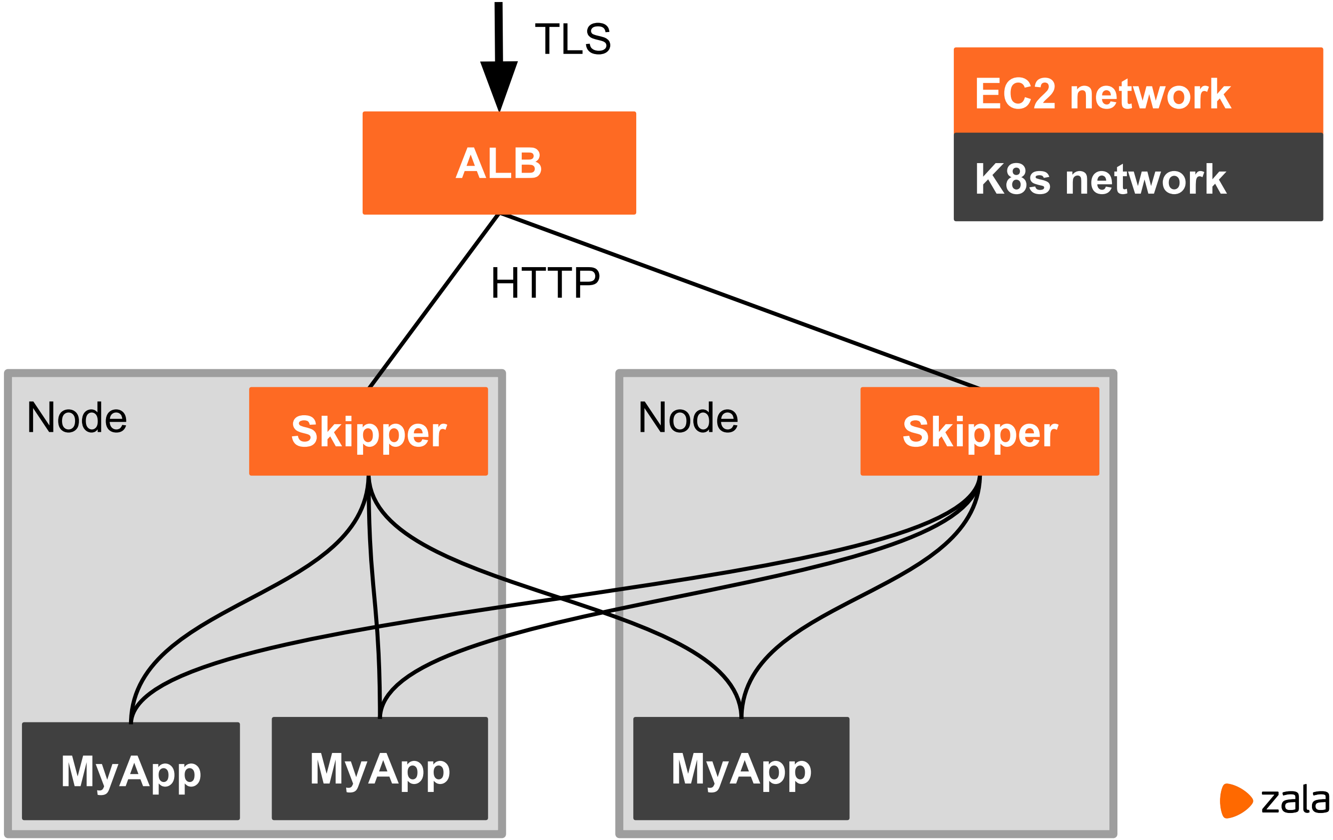
INCIDENT #1: NUMBER OF PODS



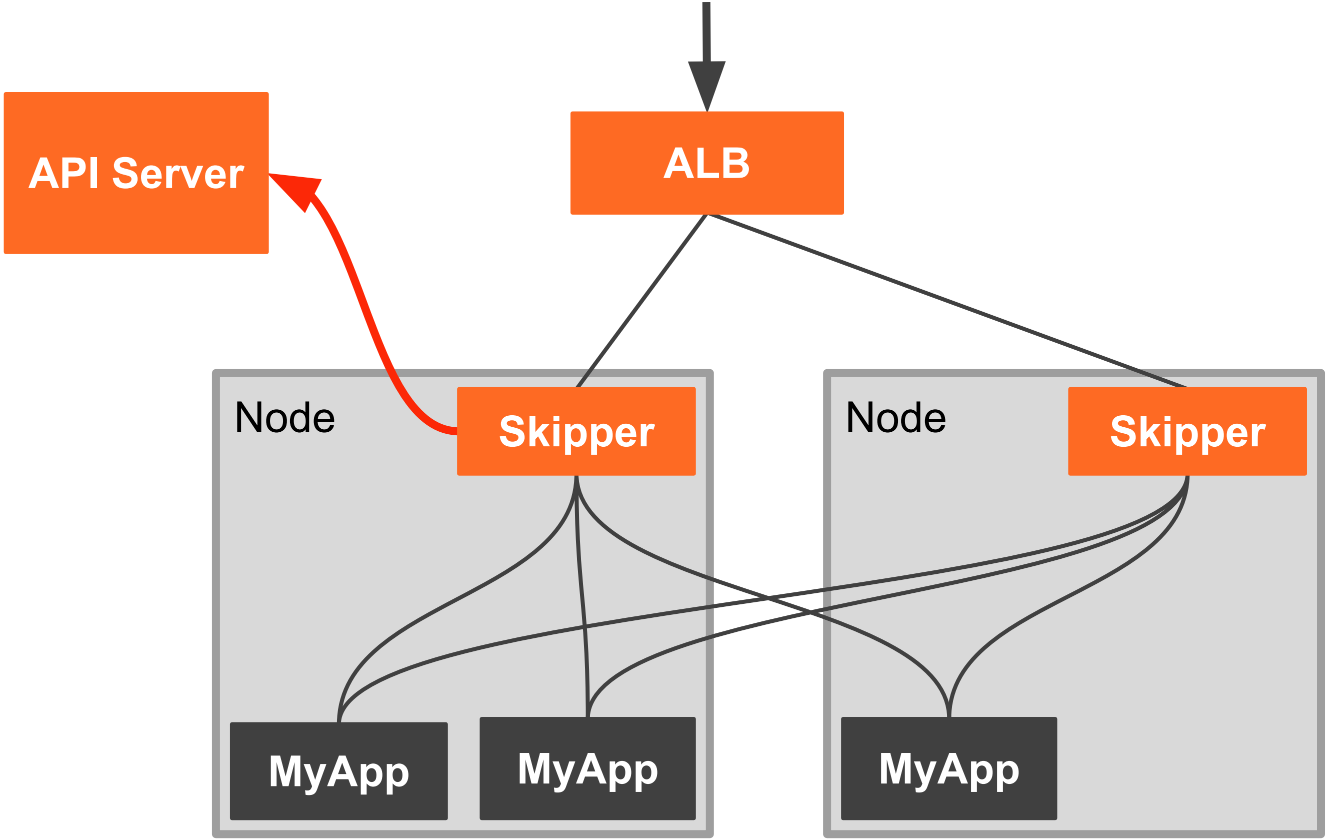
2017-08-03 23:58:01

- (key=deployments) : 65
- (key=pods) : 288
- (key=services) : 64
- (key=statefulsets) : 15

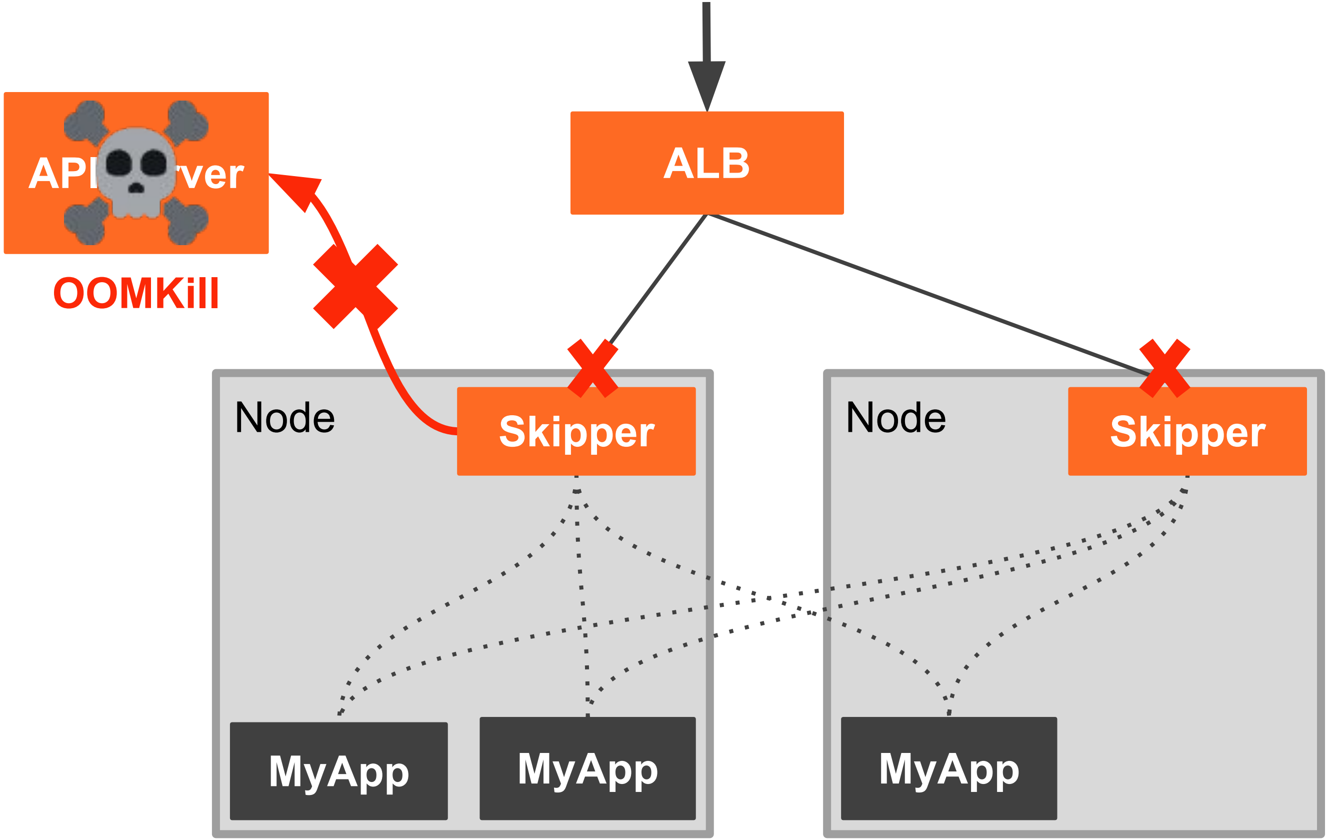
LIFE OF A REQUEST (INGRESS)



ROUTES FROM API SERVER



API SERVER DOWN



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: LESSONS LEARNED

- Fix Ingress to stay “healthy” during API server problems
- Fix Ingress to retain last known set of routes
- Use quota for number of pods

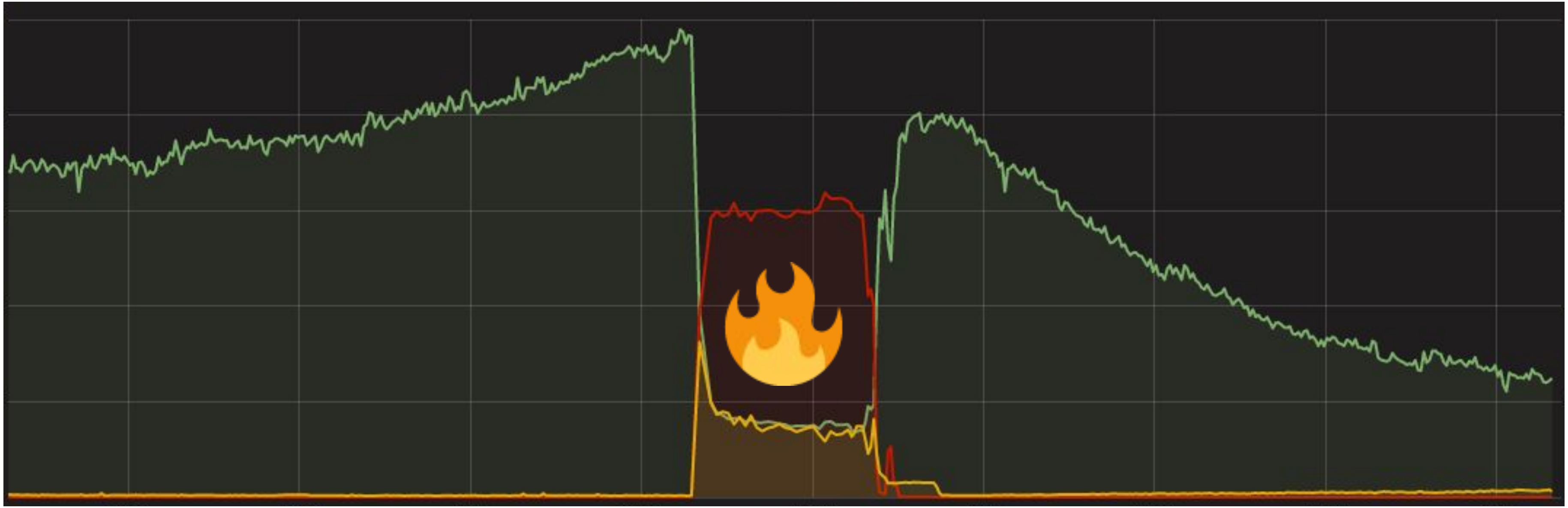
```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: compute-resources
spec:
  hard:
    pods: "1500"
```

NOTE: we dropped quotas recently github.com/zalando-incubator/kubernetes-on-aws/pull/2059

INCIDENT

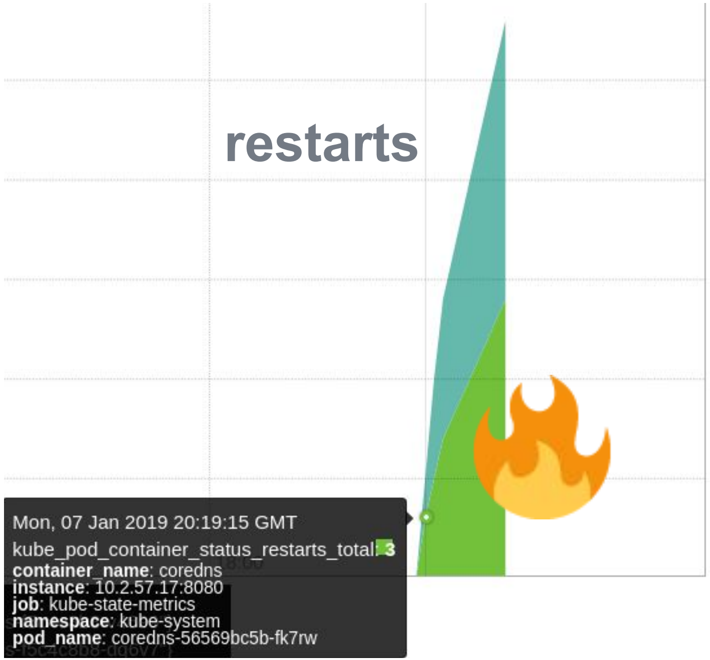
#3

INCIDENT #3: INGRESS ERRORS



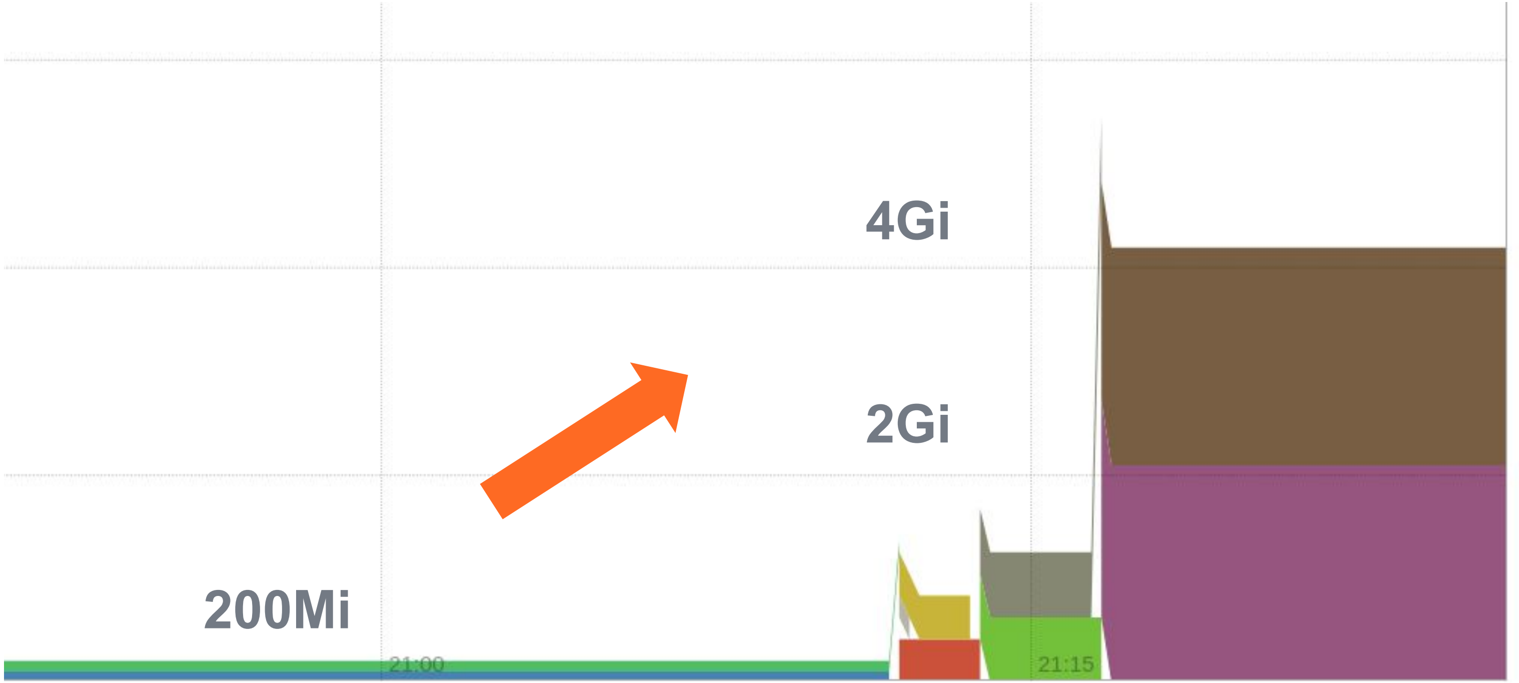
INCIDENT #3: COREDNS OOMKILL

```
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
```

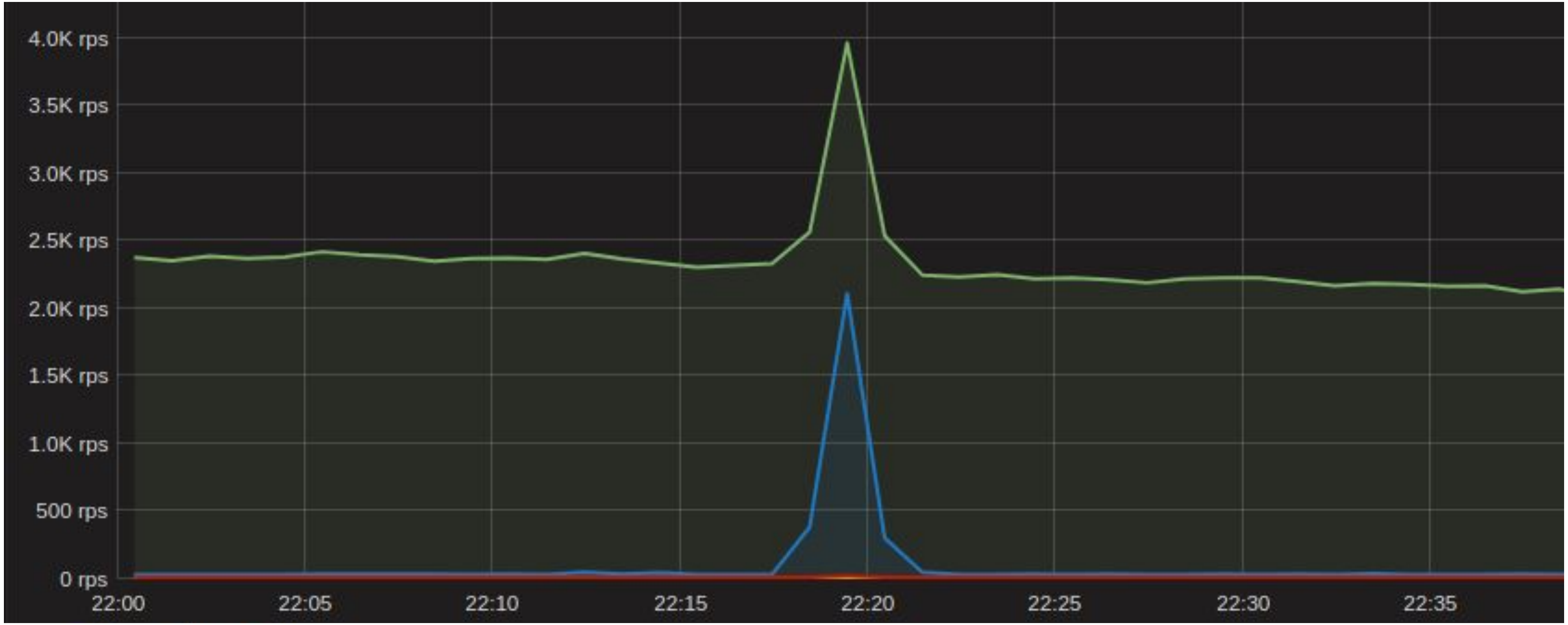


```
Mon, 07 Jan 2019 20:19:15 GMT
kube_pod_container_status_restarts_total: 3
container_name: coredns
instance: 10.2.57.17:8080
job: kube-state-metrics
namespace: kube-system
pod_name: coredns-56569bc5b-fk7rw
```

STOP THE BLEEDING: INCREASE MEMORY LIMIT

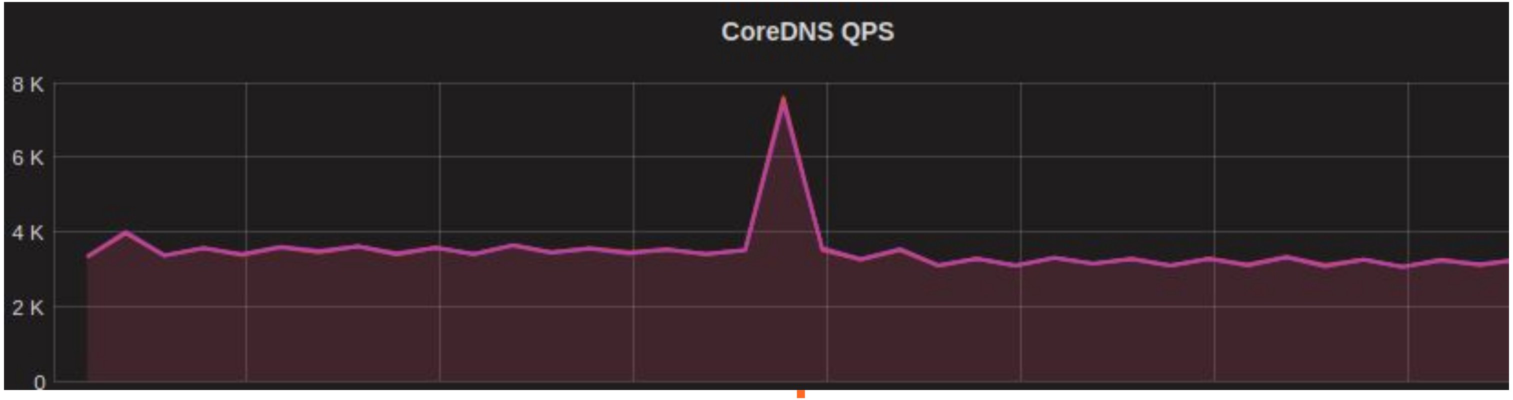


SPIKE IN HTTP REQUESTS

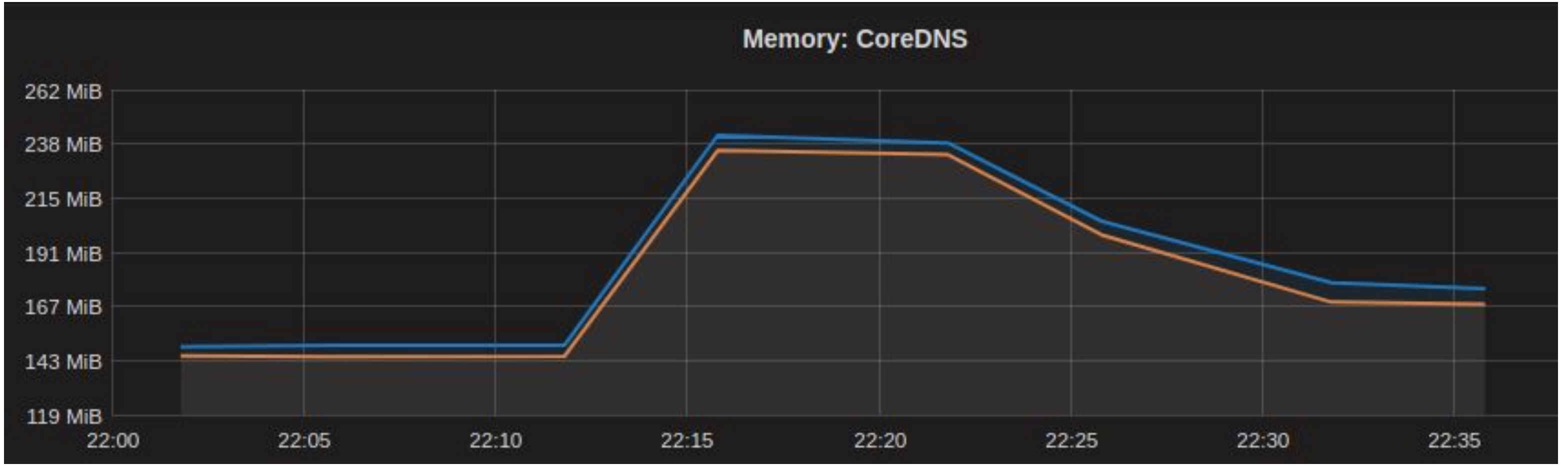


zalando

SPIKE IN DNS QUERIES



INCREASE IN MEMORY USAGE



INCIDENT #3: CONTRIBUTING FACTORS

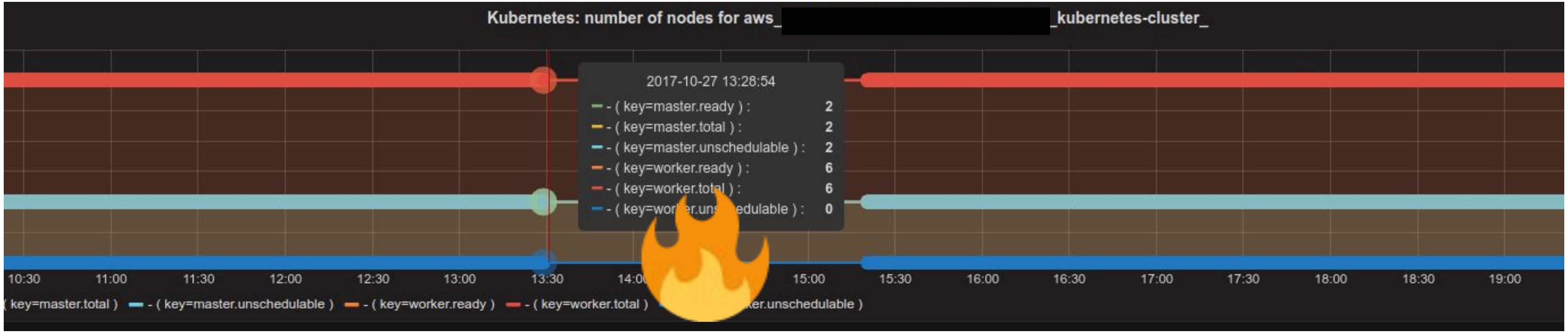
- HTTP retries
- No DNS caching
- Kubernetes ndots:5 problem
- Short maximum lifetime of HTTP connections
- Fixed memory limit for CoreDNS
- Monitoring affected by DNS outage

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

INCIDENT

#4

INCIDENT #4: CLUSTER DOWN



Kubernetes: number of nodes for aws_[redacted]_kubernetes-cluster_

2017-10-27 13:28:54

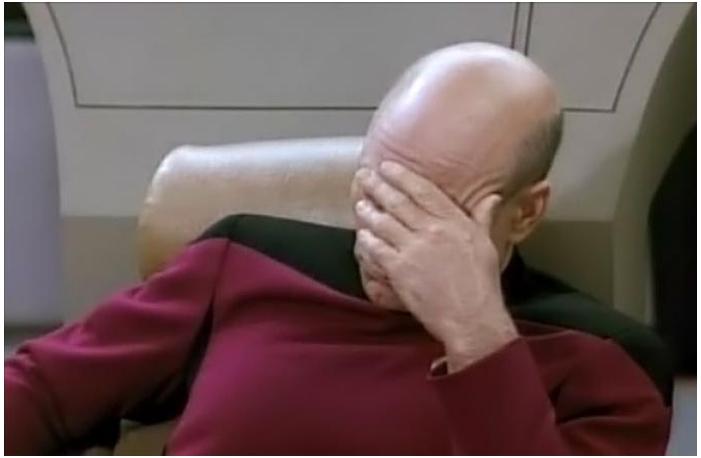
```
- ( key=master.ready ) :           2
- ( key=master.total ) :           2
- ( key=master.unschedulable ) :   2
- ( key=worker.ready ) :           6
- ( key=worker.total ) :           6
- ( key=worker.unschedulable ) :   0
```

INCIDENT #4: MANUAL OPERATION

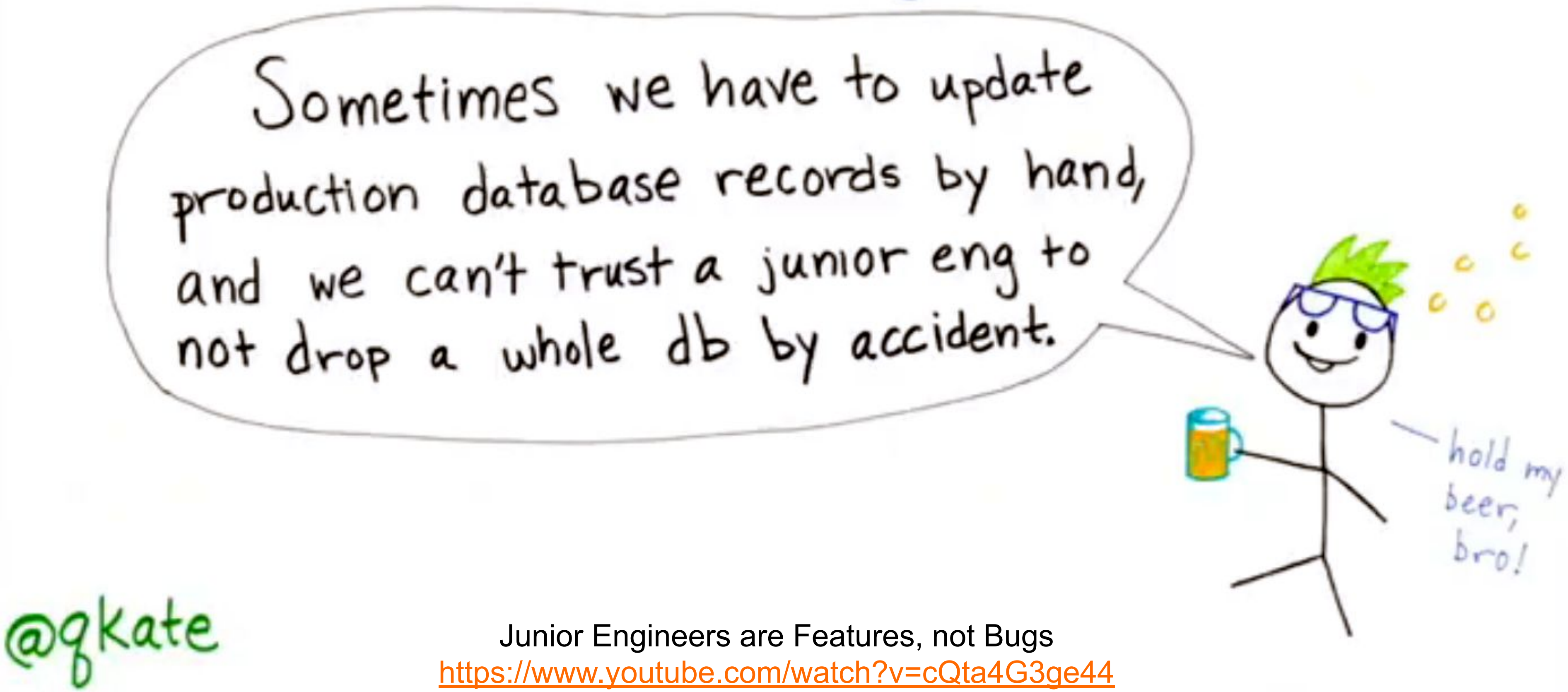
```
% etcdctl del -r /registry-kube-1/certificatesigningrequest prefix
```

INCIDENT #4: RTFM

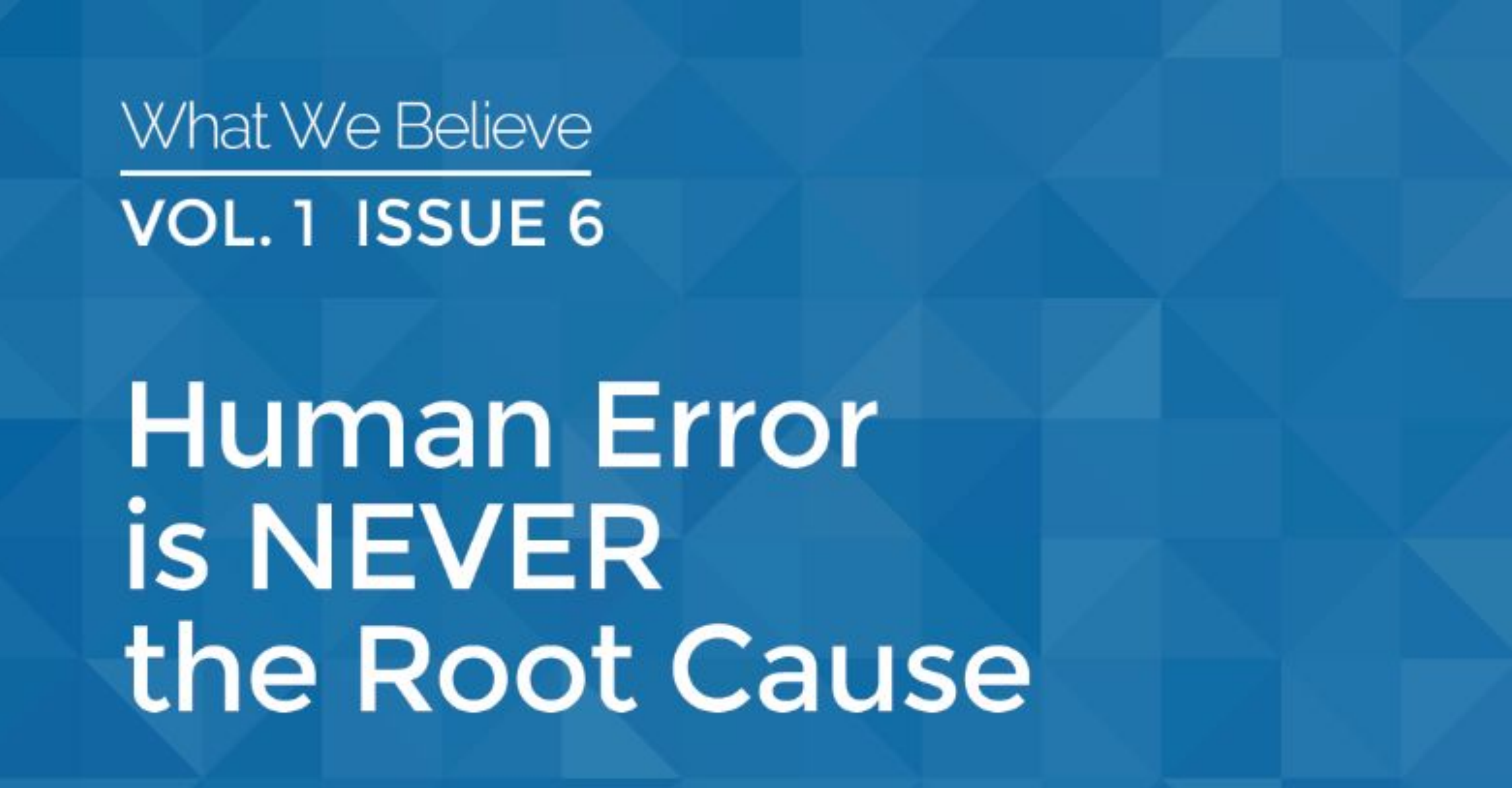
```
% etcdctl del -r /registry-kube-1/certificatesigningrequest prefix

help: etcdctl del [options] <key> [range_end]
```



MYTH #3: Tooling


Sometimes we have to update production database records by hand, and we can't trust a junior eng to not drop a whole db by accident.
hold my beer, bro!
@qkate

Junior Engineers are Features, not Bugs
https://www.youtube.com/watch?v=cQta4G3ge44



What We Believe

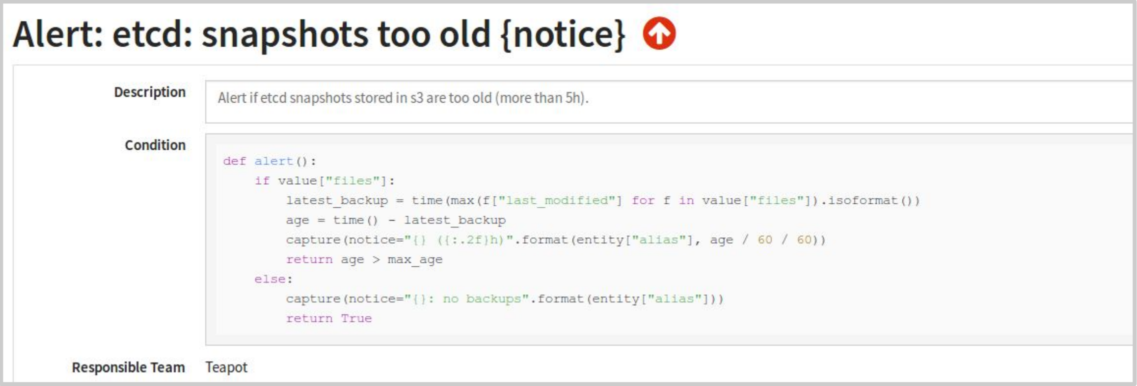
VOL. 1 ISSUE 6

Human Error is NEVER the Root Cause

https://www.outcome-eng.com/human-error-never-root-cause/

INCIDENT #4: LESSONS LEARNED

- Disaster Recovery Plan?
- Backup etcd to S3
- Monitor the snapshots



Alert: etcd: snapshots too old {notice}

Description Alert if etcd snapshots stored in s3 are too old (more than 5h).

Condition

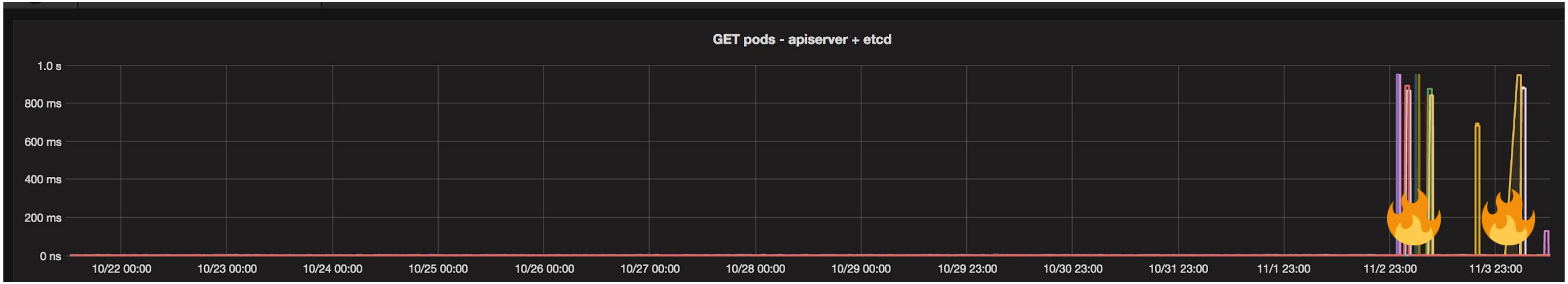
```
def alert():
    if value["files"]:
        latest_backup = time(max(f["last_modified"] for f in value["files"]).isoformat())
        age = time() - latest_backup
        capture(notice="{} ({:.2f}h)".format(entity["alias"], age / 60 / 60))
        return age > max_age
    else:
        capture(notice="{}: no backups".format(entity["alias"]))
        return True
```

Responsible Team Teapot

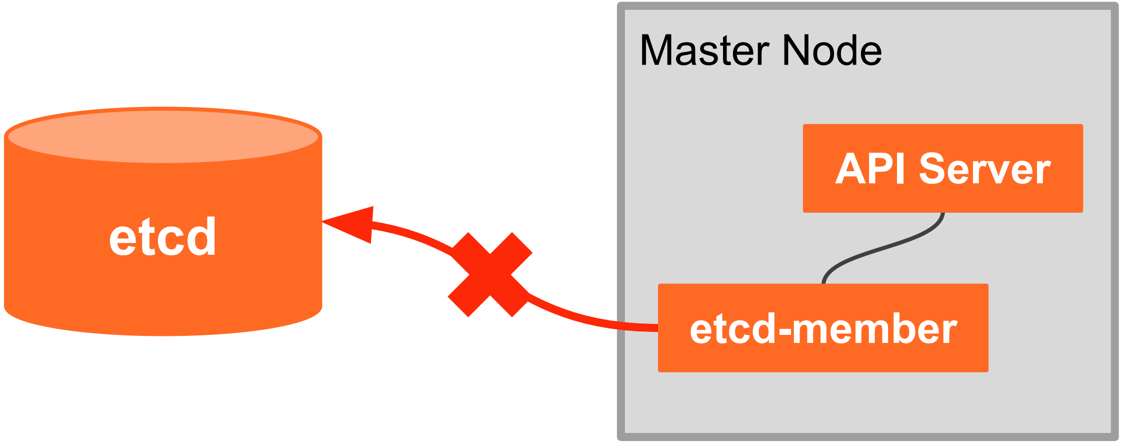
INCIDENT

#5

INCIDENT #5: API LATENCY SPIKES



INCIDENT #5: CONNECTION ISSUES



...
Kubernetes worker and master nodes sporadically fail to connect to etcd causing timeouts in the APIserver and disconnects in the pod network.
...

INCIDENT #5: STOP THE BLEEDING

```bash
#!/bin/bash

while true; do
  echo "sleep for 60 seconds"
  sleep 60
  timeout 5 curl http://localhost:8080/api/v1/nodes > /dev/null
  if [ $? -eq 0 ]; then
    echo "all fine, no need to restart etcd member"
    continue
  else
    echo "restarting etcd-member"
    systemctl restart etcd-member
  fi
done
```

INCIDENT #5: CONFIRMATION FROM AWS

[...]

We can’t go into the details [...] that resulted the networking problems during the “non-intrusive maintenance”, as it relates to internal workings of EC2. We can confirm this only affected the T2 instance types, ...

[...]

We don’t explicitly recommend against running production services on T2

[...]

🤔

INCIDENT #5: LESSONS LEARNED

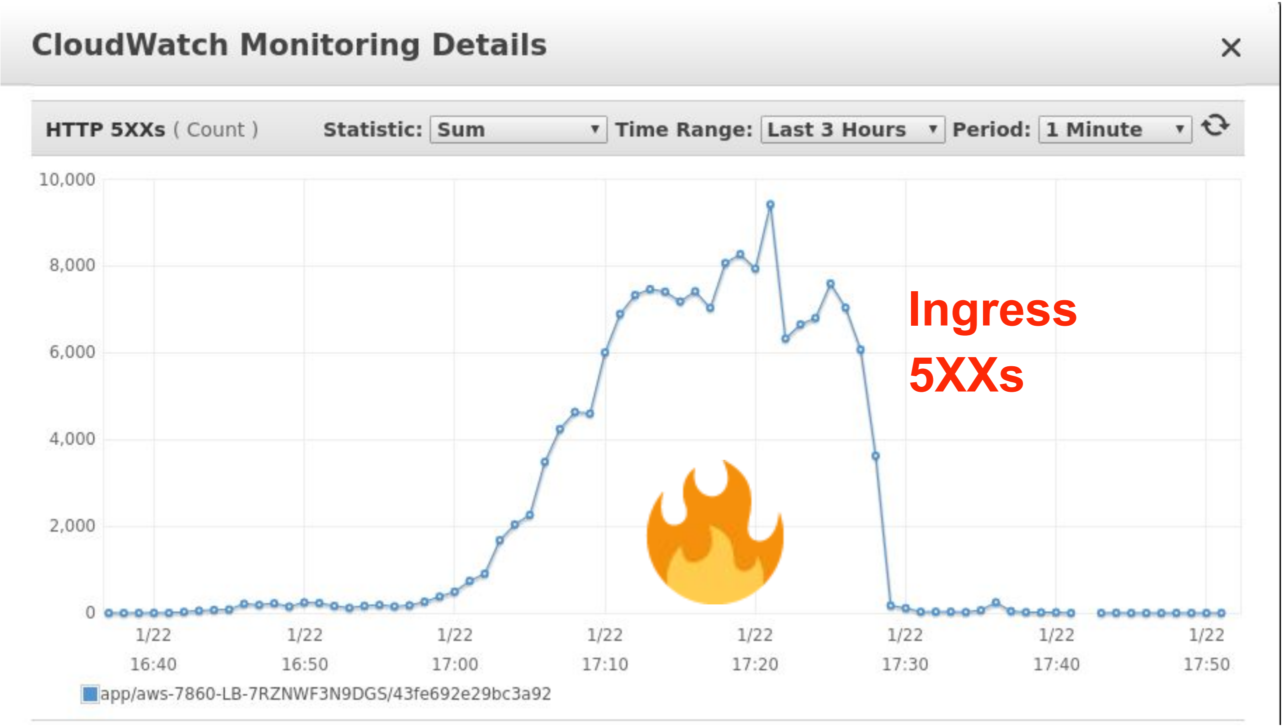
- It's never the AWS infrastructure until it is
- Treat t2 instances with care
- Kubernetes components are not necessarily "cloud native"

Cloud Native? Declarative, dynamic, **resilient**, and scalable

INCIDENT

#6

INCIDENT #6: IMPACT

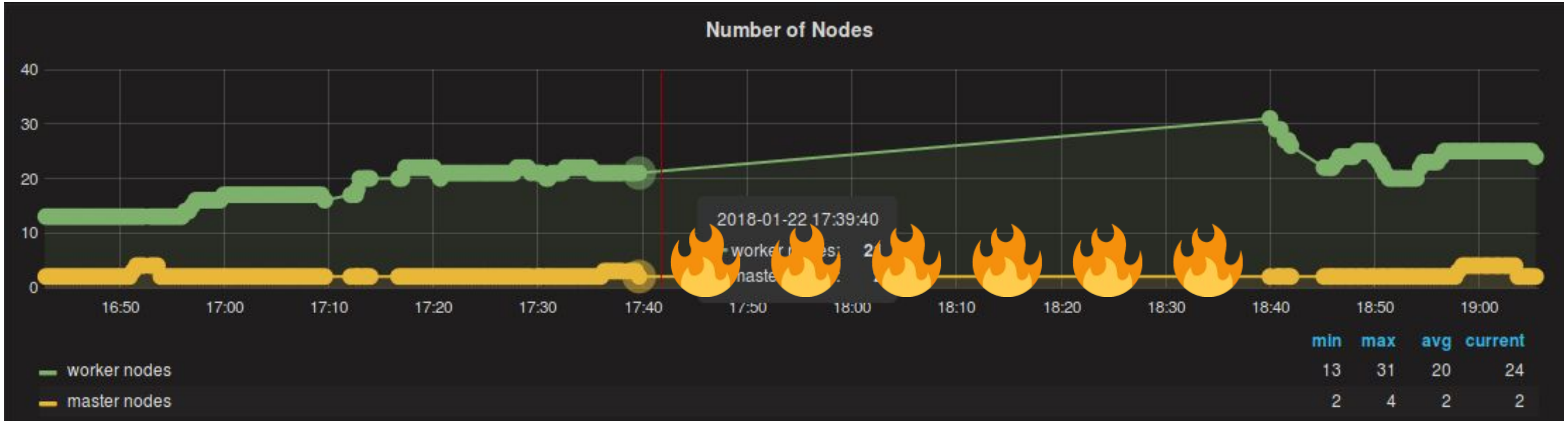


CloudWatch Monitoring Details

HTTP 5XXs (Count) Statistic: Sum Time Range: Last 3 Hours Period: 1 Minute

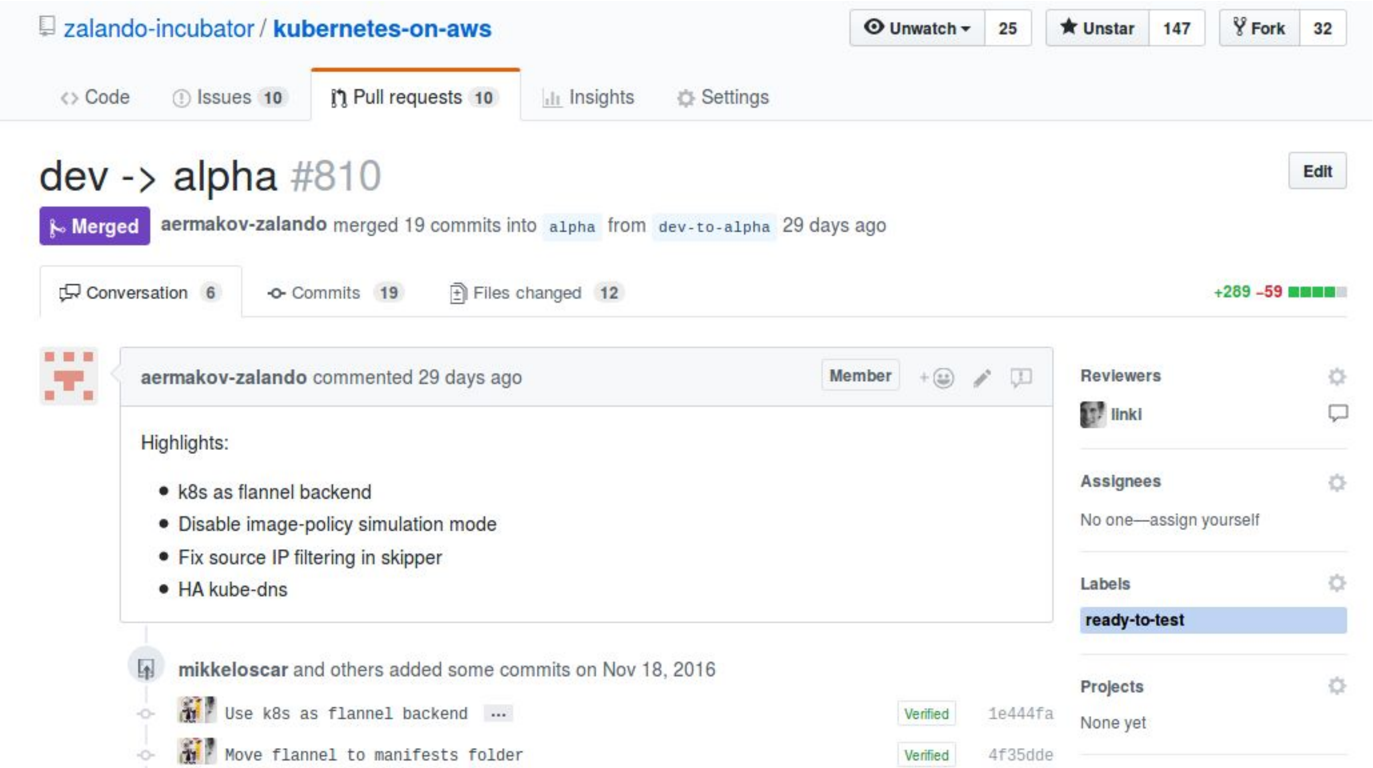
Ingress 5XXs

INCIDENT #6: CLUSTER DOWN?



	min	max	avg	current
▬ worker nodes	13	31	20	24
▬ master nodes	2	4	2	2

INCIDENT #6: THE TRIGGER



zalando-incubator / **kubernetes-on-aws**

dev -> alpha #810

Merged **aermakov-zalando** merged 19 commits into `alpha` from `dev-to-alpha` 29 days ago

aermakov-zalando commented 29 days ago

Highlights:

- k8s as flannel backend
- Disable image-policy simulation mode
- Fix source IP filtering in skipper
- HA kube-dns

mikkeloscar and others added some commits on Nov 18, 2016

```
Use k8s as flannel backend …        Verified  1e444fa
Move flannel to manifests folder    Verified  4f35dde
```

Reviewers: linki

Assignees: No one—assign yourself

Labels: **ready-to-test**

Projects: None yet

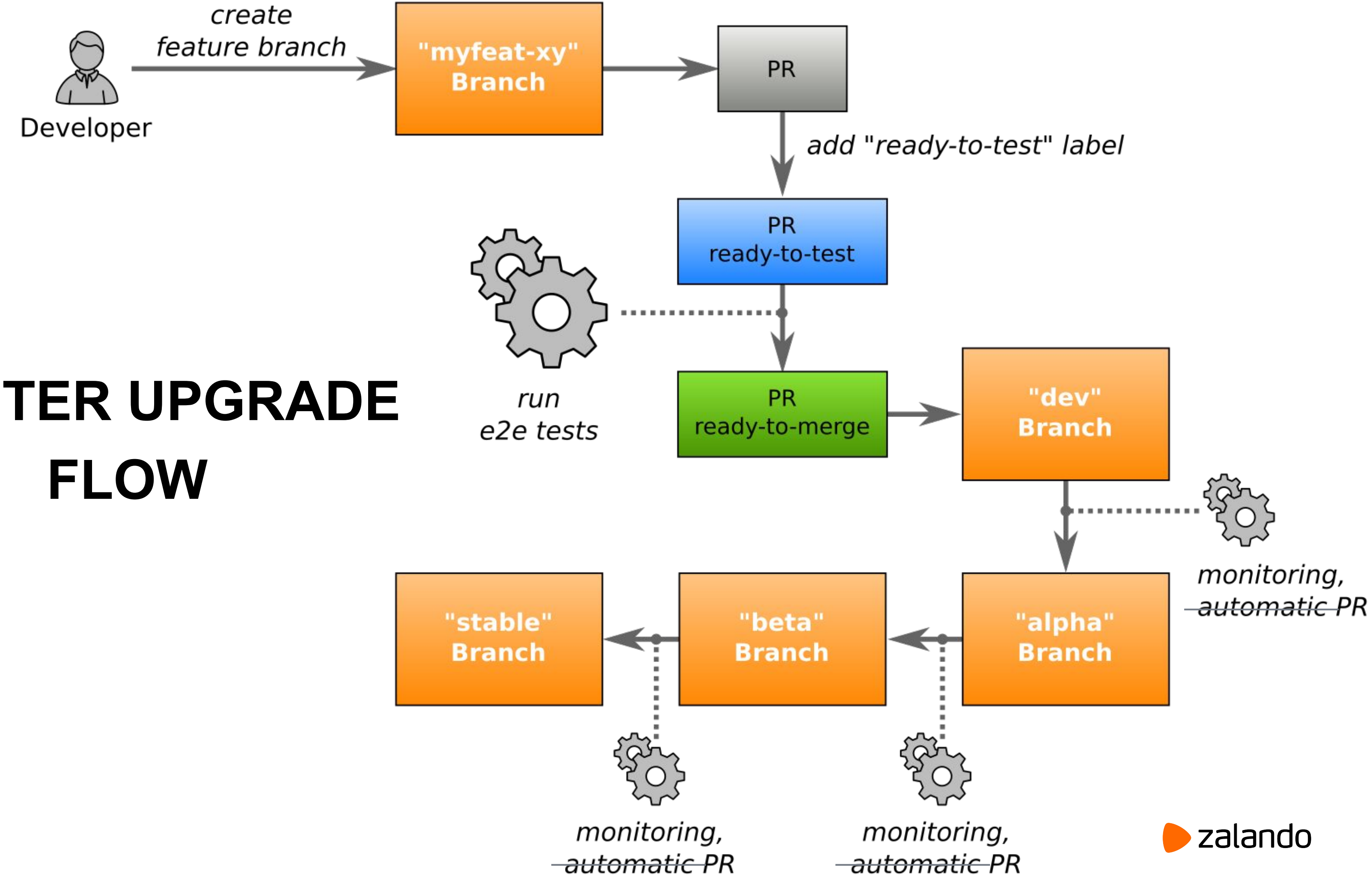
What We Believe

VOL. 1 ISSUE 6

Human Error is NEVER the Root Cause

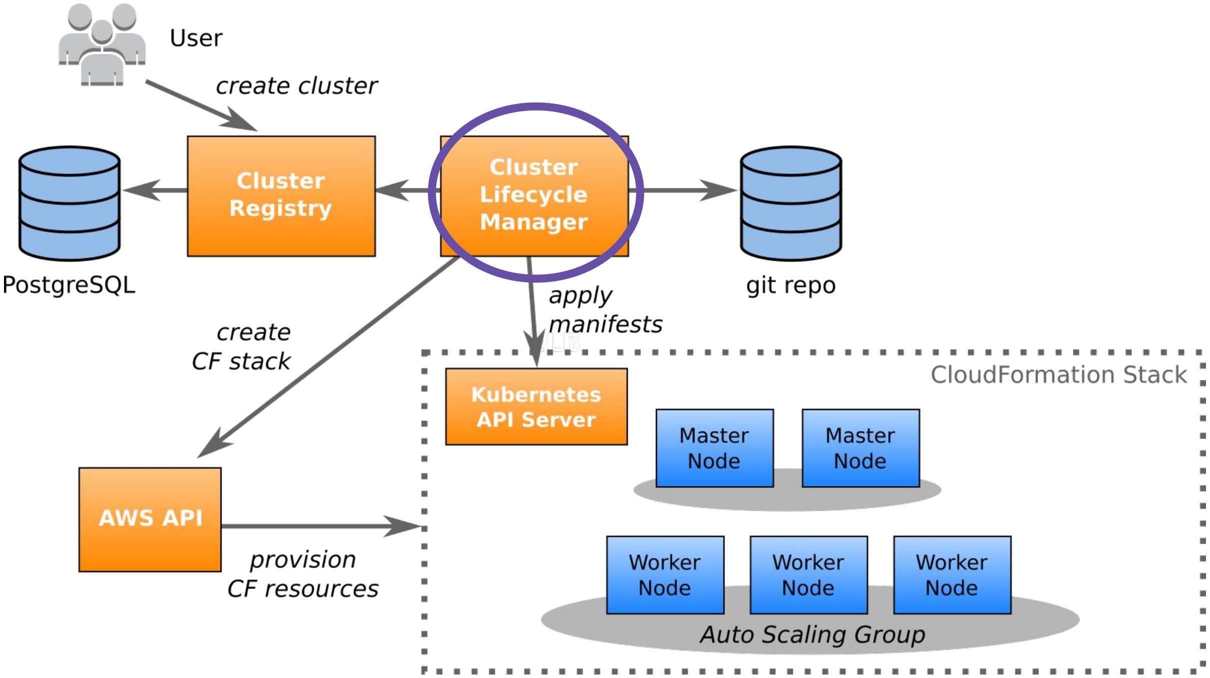
https://www.outcome-eng.com/human-error-never-root-cause/

CLUSTER UPGRADE FLOW



zalando

CLUSTER LIFECYCLE MANAGER (CLM)



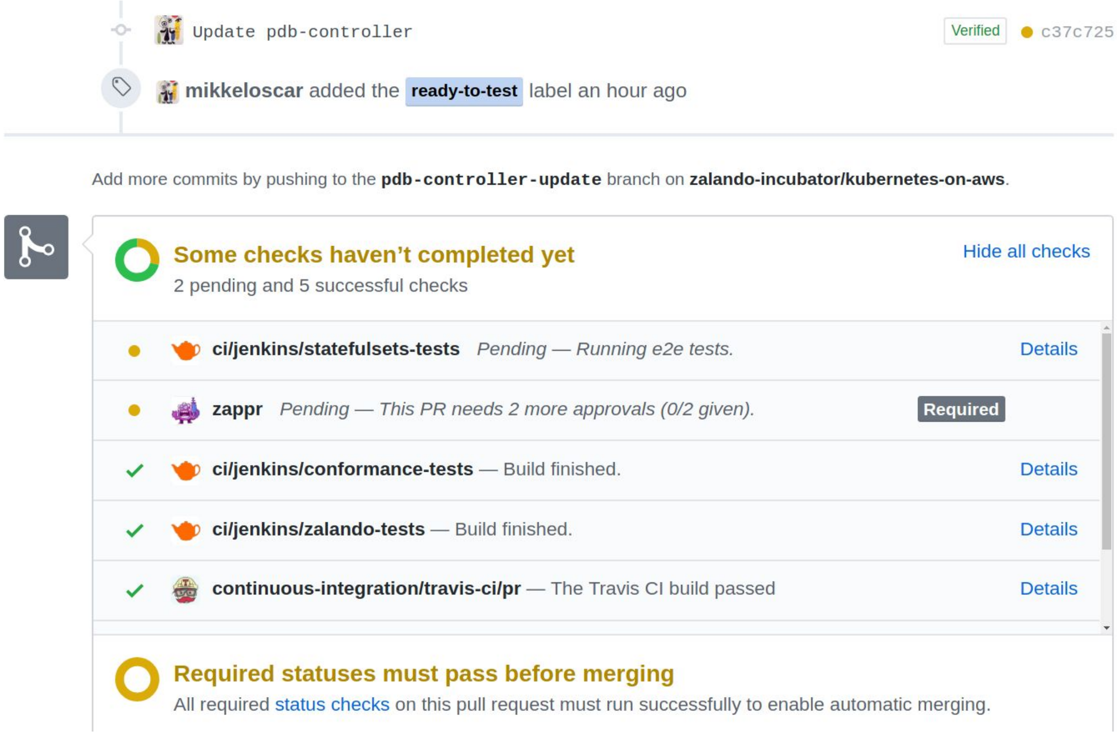
github.com/zalando-incubator/cluster-lifecycle-manager

CLUSTER CHANNELS

Channel	Description	Clusters
`dev`	Development and playground clusters.	**3**
`alpha`	Main infrastructure clusters (**important to us**).	**2**
`beta`	Product clusters for the rest of the organization (non-prod).	**57+**
`stable`	Product clusters for the rest of the organization (prod).	**57+**

github.com/zalando-incubator/kubernetes-on-aws

E2E TESTS ON EVERY PR



Update pdb-controller

Verified ● c37c725

mikkeloscar added the **ready-to-test** label an hour ago

Add more commits by pushing to the **pdb-controller-update** branch on **zalando-incubator/kubernetes-on-aws**.

Some checks haven't completed yet

2 pending and 5 successful checks

Hide all checks

● **ci/jenkins/statefulsets-tests** *Pending — Running e2e tests.* Details

● **zappr** *Pending — This PR needs 2 more approvals (0/2 given).* **Required**

✓ **ci/jenkins/conformance-tests** — Build finished. Details

✓ **ci/jenkins/zalando-tests** — Build finished. Details

✓ **continuous-integration/travis-ci/pr** — The Travis CI build passed Details

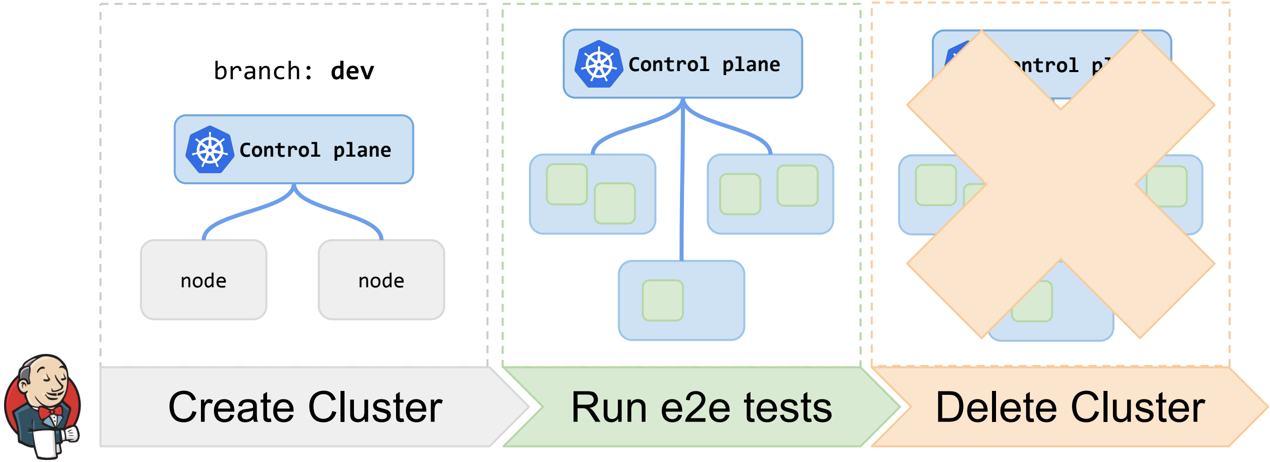
Required statuses must pass before merging

All required status checks on this pull request must run successfully to enable automatic merging.

github.com/zalando-incubator/kubernetes-on-aws

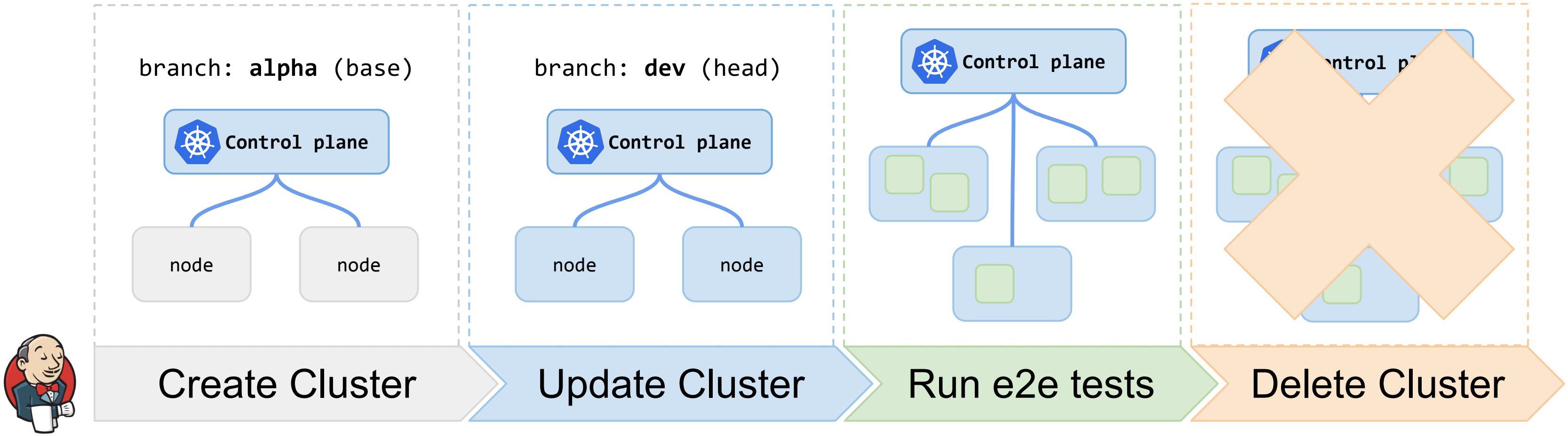
RUNNING E2E TESTS (BEFORE)

Testing **dev** to **alpha** upgrade



RUNNING E2E TESTS (NOW)

Testing **dev** to **alpha** upgrade



INCIDENT #6: LESSONS LEARNED

- Automated e2e tests are pretty good, but not enough
- Test the diff/migration automatically
  - Bootstrap new cluster with previous configuration
  - Apply new configuration
  - Run end-to-end & conformance tests

github.com/zalando-incubator/kubernetes-on-aws/tree/dev/test/e2e

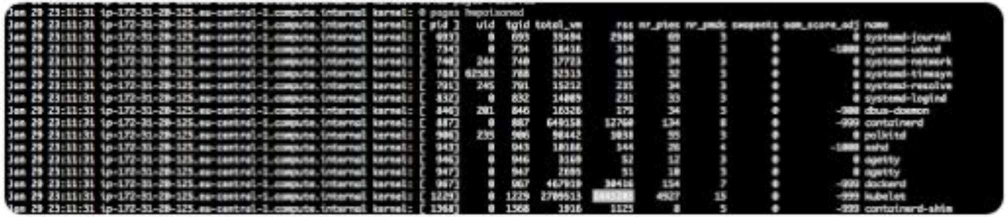
INCIDENT

#7

#7: KERNEL OOM KILLER

[redacted] Jan 30, 11:59 AM

so this is nice:



```
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: 0 pages hwpoisoned
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [ pid ]   uid  tgid total_vm      rss nr_ptes nr_pmds swapents oom_score_adj name
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  693]     0   693    35494     2900      69       3        0             0 systemd-journal
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  734]     0   734    10416      314      38       3        0         -1000 systemd-udevd
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  740]   244   740    17723      485      34       3        0             0 systemd-network
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  788] 62583   788    32513      133      32       3        0             0 systemd-timesyn
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  791]   245   791    15212      235      34       3        0             0 systemd-resolve
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  832]     0   832    14009      231      33       3        0             0 systemd-logind
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  846]   201   846    16326      179      34       3        0          -900 dbus-daemon
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  887]     0   887   640158    12760     134       8        0          -999 containerd
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  906]   235   906    90442     1038      55       3        0             0 polkitd
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  943]     0   943    18186      144      26       4        0         -1000 sshd
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  946]     0   946     3160       52      12       3        0             0 agetty
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  947]     0   947     2889       51      10       3        0             0 agetty
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [  967]     0   967   467939    30416     154       7        0          -999 dockerd
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [ 1229]     0  1229  2709613 [illegible]    4927      15        0          -999 kubelet
Jan 29 23:11:31 ip-172-31-28-125.eu-central-1.compute.internal kernel: [ 1368]     0  1368     1916     1125       8       5        0          -999 containerd-shim
```

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!!!!

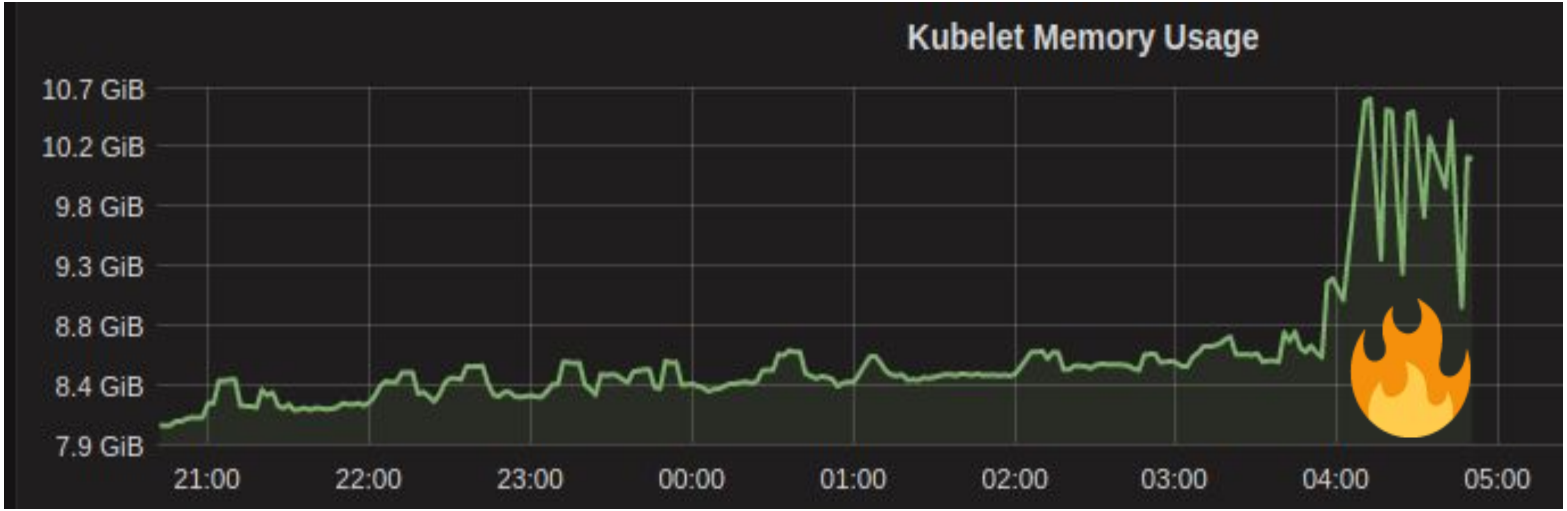
[redacted] Jan 30, 12:00 PM

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

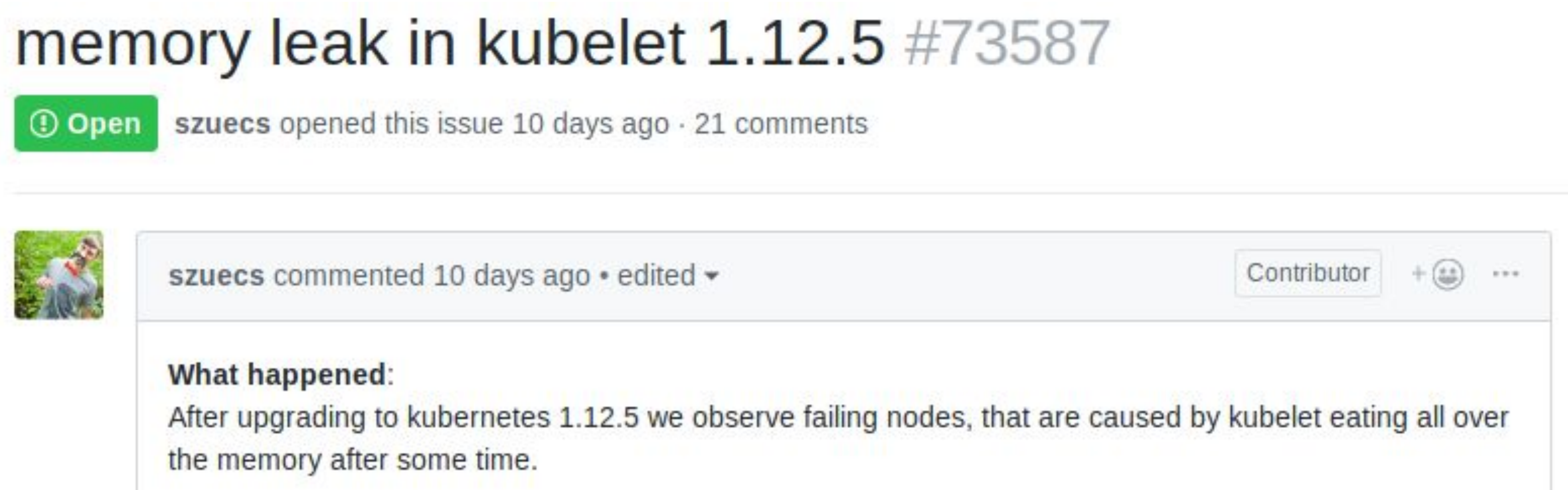
🔥

⇒ all containers on this node down

INCIDENT #7: KUBELET MEMORY



UPSTREAM ISSUE REPORTED



memory leak in kubelet 1.12.5 #73587

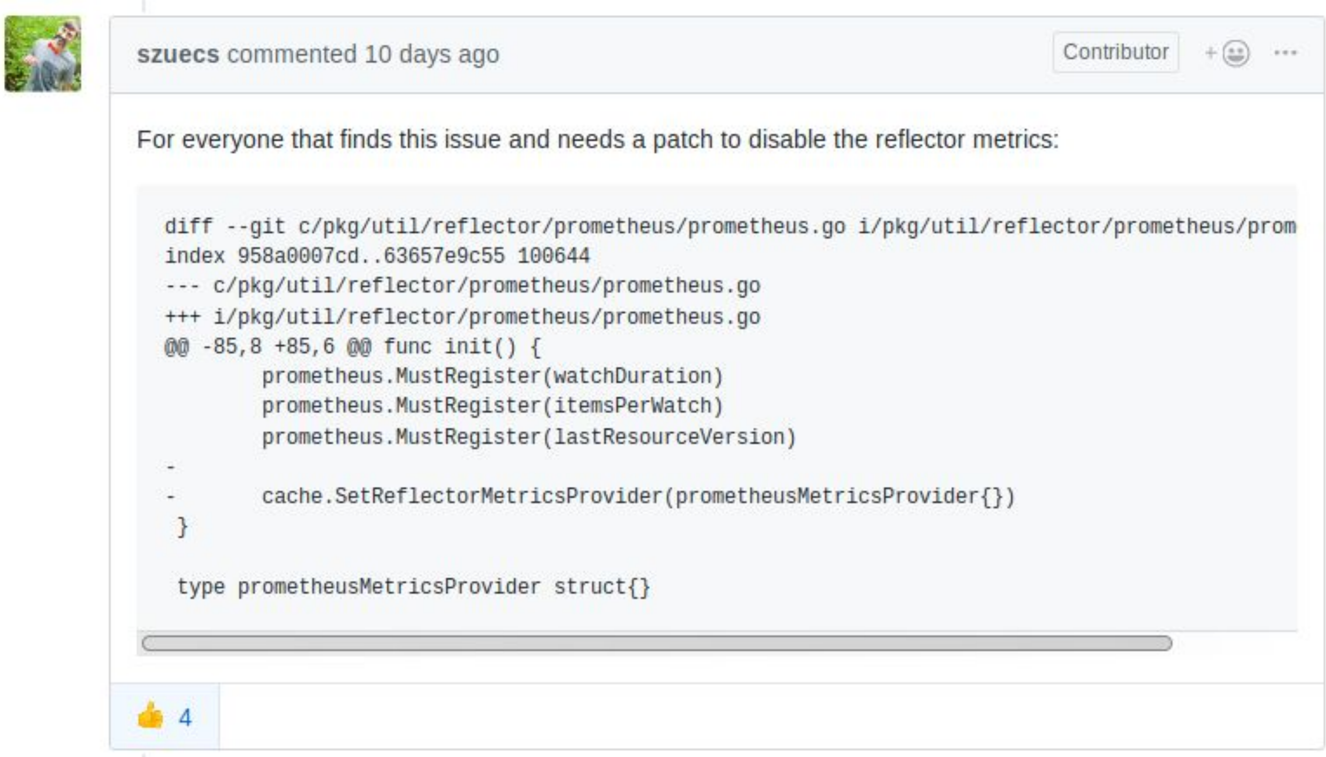
szuecs opened this issue 10 days ago · 21 comments

szuecs commented 10 days ago • edited

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 #7: THE PATCH



szuecs commented 10 days ago

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{}
```

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

INCIDENT

#8

INCIDENT #8: IMPACT

Error during Pod creation:

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

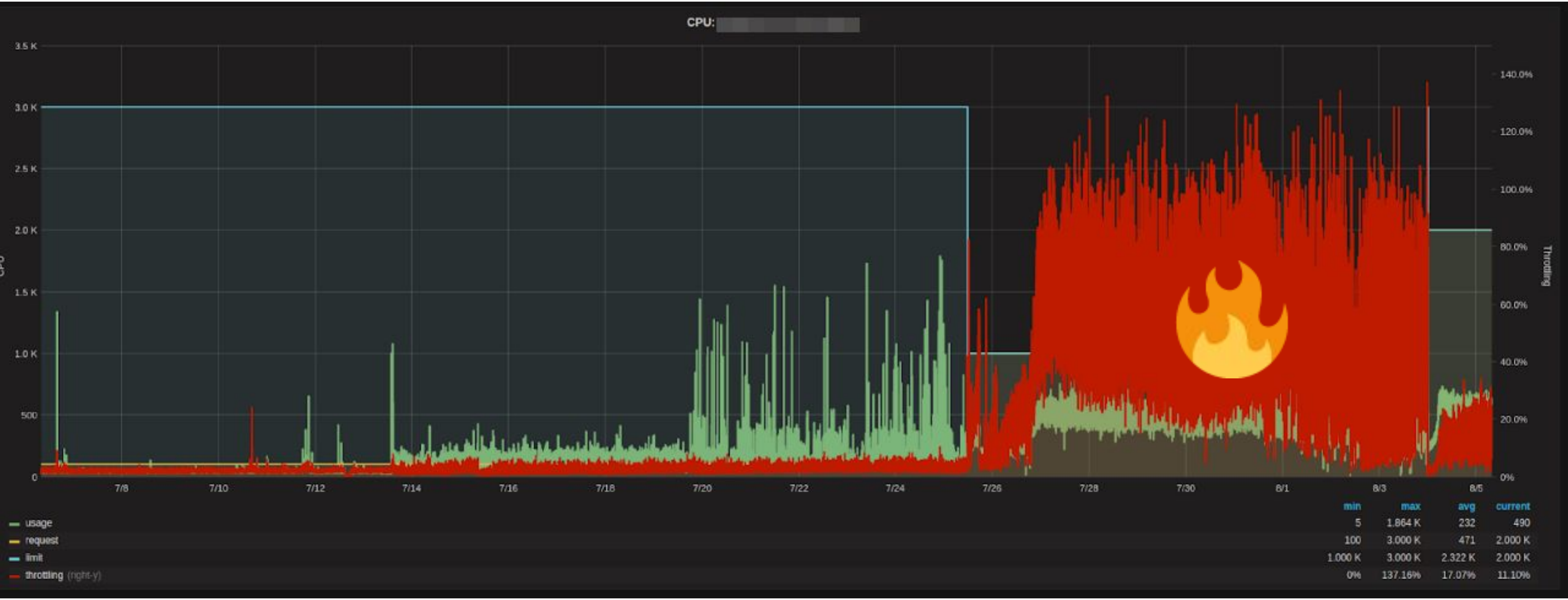
⇒ All new Kubernetes deployments fail 🔥

INCIDENT #8: 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 #8: CPU THROTTLING



CPU: [redacted]

	min	max	avg	current
🟩 usage	5	1.864 K	232	490
🟨 request	100	3.000 K	471	2.000 K
🟦 limit	1.000 K	3.000 K	2.322 K	2.000 K
🟥 throttling (right-y)	0%	137.16%	17.07%	11.10%

zalando

INCIDENT #8: WHAT HAPPENED

Scaled down IAM provider
to reduce **Slack**

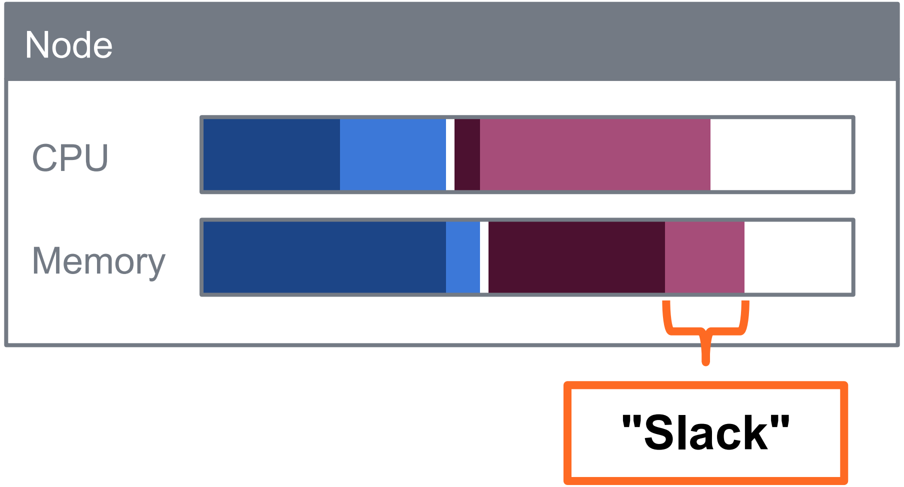
+ Number of deployments increased

⇒ Process could not process credentials fast enough

SLACK

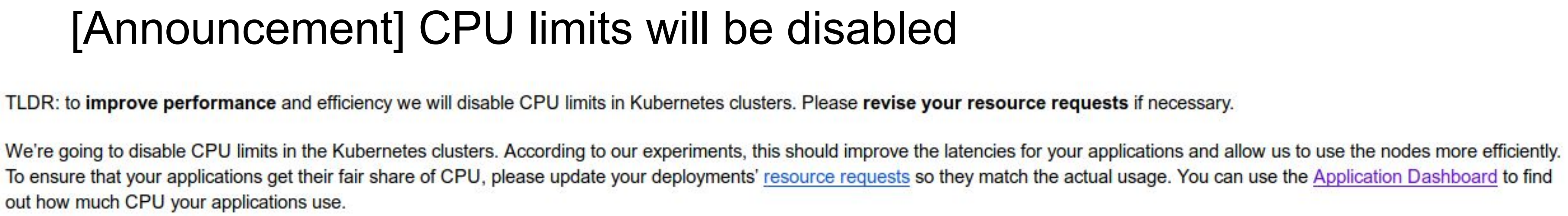
CPU/memory requests "block" resources on nodes.

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



DISABLING CPU THROTTLING

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

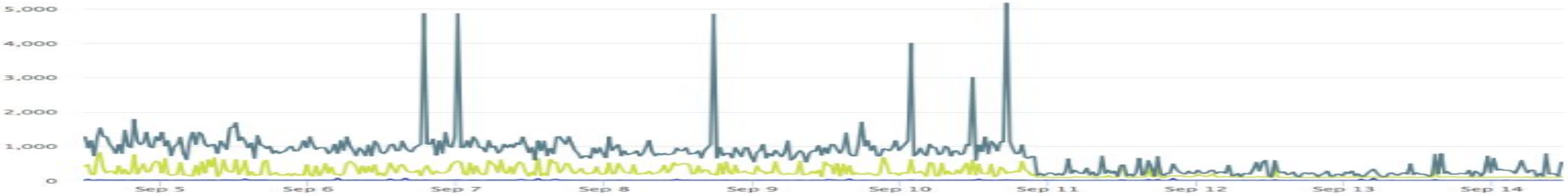


[Announcement] CPU limits will be disabled

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

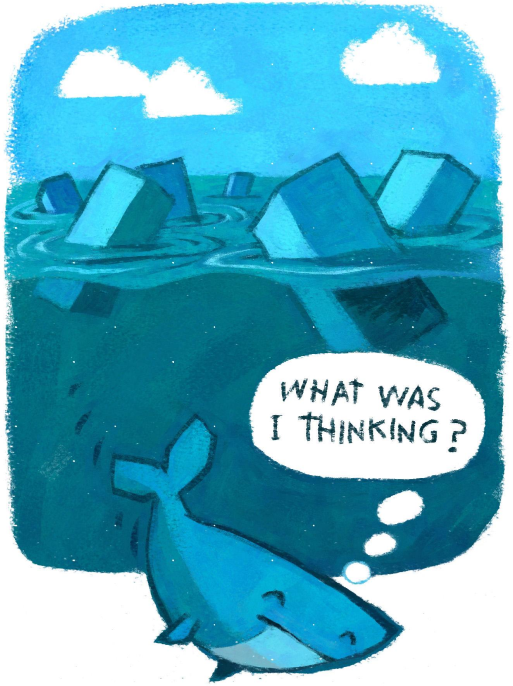


A MILLION WAYS TO CRASH YOUR CLUSTER?

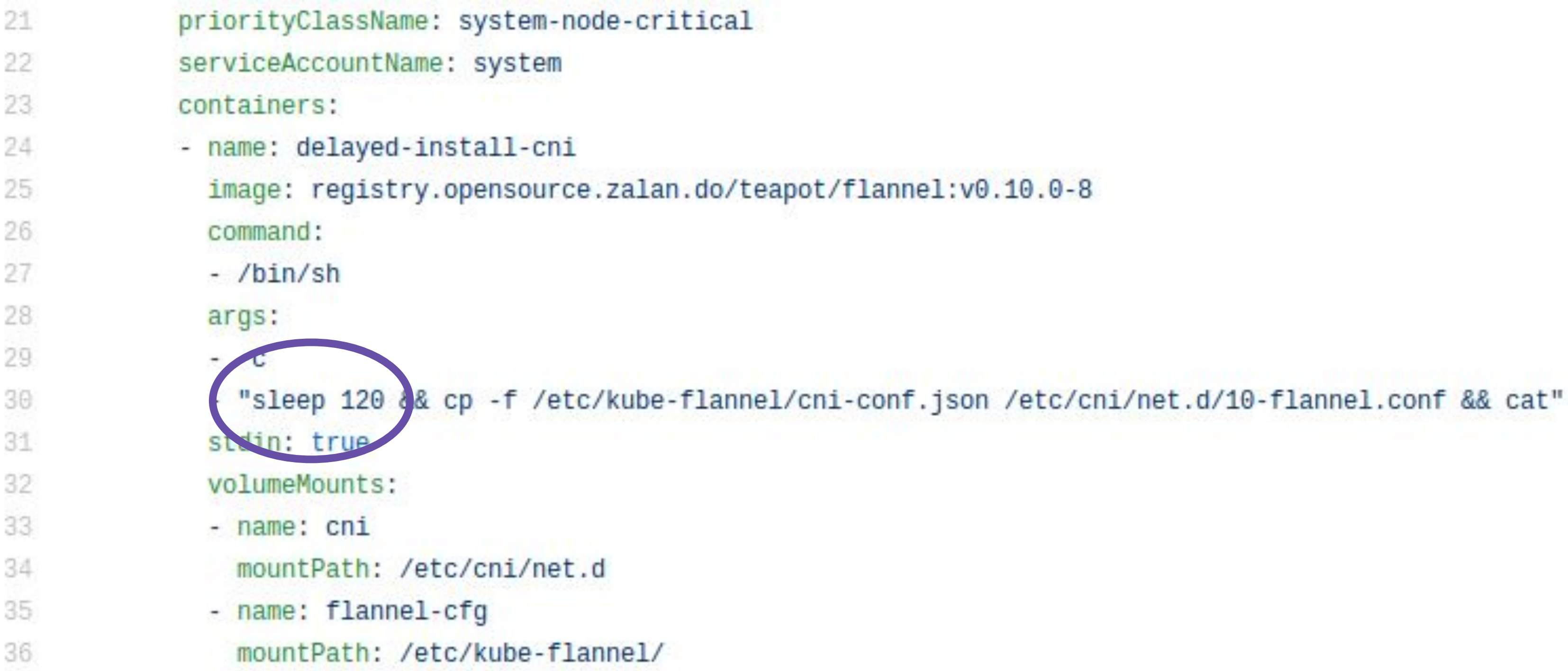
- 🔥 Switch to latest Docker to fix issues with **Docker daemon freezing**
- 🔥 Redesign of DNS setup due to **high DNS latencies** (5s), switch from kube-dns to node-local dnsmasq+CoreDNS
- 🔥 Disabling CPU throttling (CFS quota) to avoid **latency issues**
- 🔥 Quick fix for timeouts using etcd-proxy: client-go still seems to have **issues with timeouts**
- 🔥 **502's** during cluster updates: race condition during network setup

MORE TOPICS

- 🔥 **Graceful Pod shutdown** and race conditions (endpoints, Ingress)
- 🔥 **Incompatible Kubernetes changes**
- 🔥 CoreOS **ContainerLinux** "stable" won't boot
- 🔥 Kubernetes **EBS volume handling**
- 🔥 **Docker**



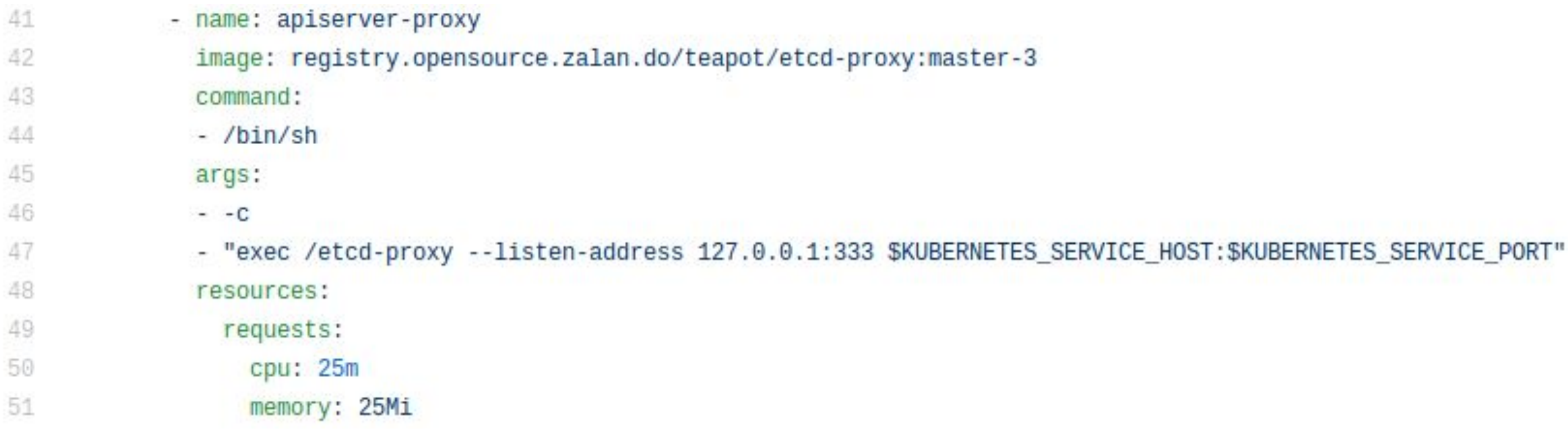
RACE CONDITIONS..



```
priorityClassName: system-node-critical
serviceAccountName: system
containers:
- name: delayed-install-cni
  image: registry.opensource.zalan.do/teapot/flannel:v0.10.0-8
  command:
  - /bin/sh
  args:
  - -c
  - "sleep 120 && cp -f /etc/kube-flannel/cni-conf.json /etc/cni/net.d/10-flannel.conf && cat"
  stdin: true
  volumeMounts:
  - name: cni
    mountPath: /etc/cni/net.d
  - name: flannel-cfg
    mountPath: /etc/kube-flannel/
```

github.com/zalando-incubator/kubernetes-on-aws

TIMEOUTS TO API SERVER..

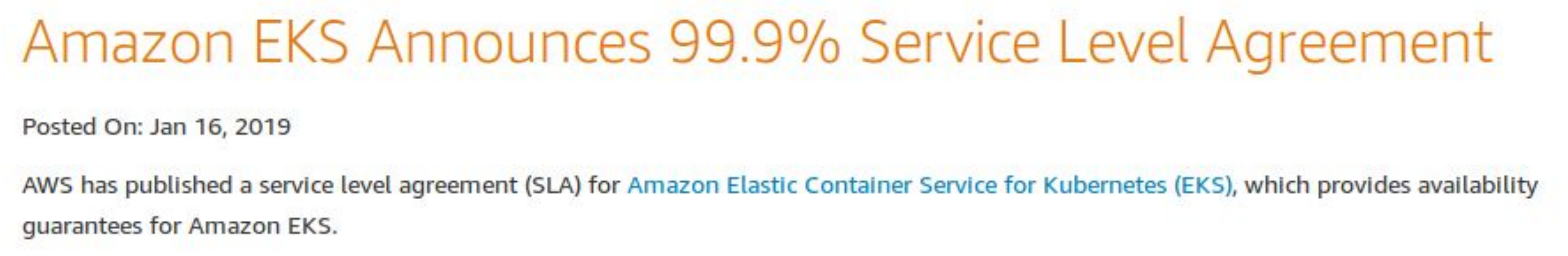


```yaml
- name: apiserver-proxy
  image: registry.opensource.zalan.do/teapot/etcd-proxy:master-3
  command:
  - /bin/sh
  args:
  - -c
  - "exec /etcd-proxy --listen-address 127.0.0.1:333 $KUBERNETES_SERVICE_HOST:$KUBERNETES_SERVICE_PORT"
  resources:
    requests:
      cpu: 25m
      memory: 25Mi
```

github.com/zalando-incubator/kubernetes-on-aws

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

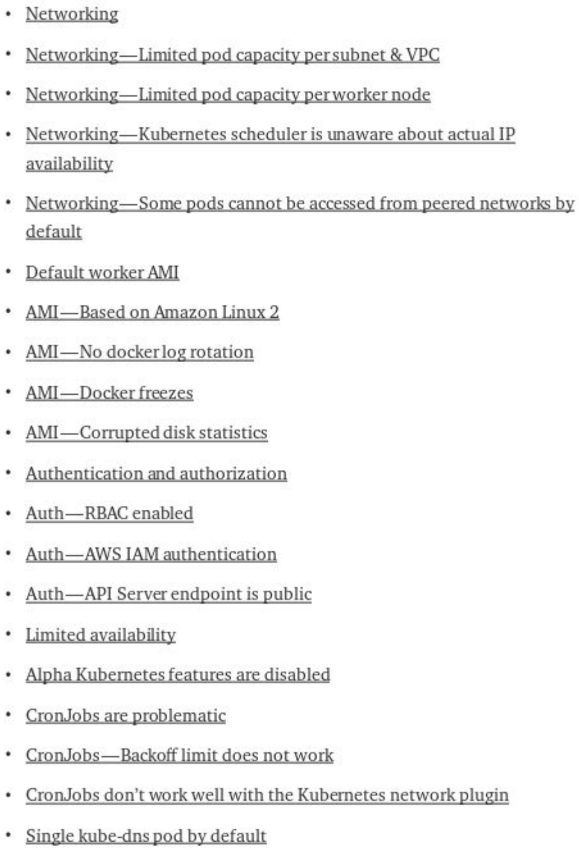
WILL MANAGED K8S SAVE US?

NO

(not really)

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

PRODUCTION PROOFING AWS EKS

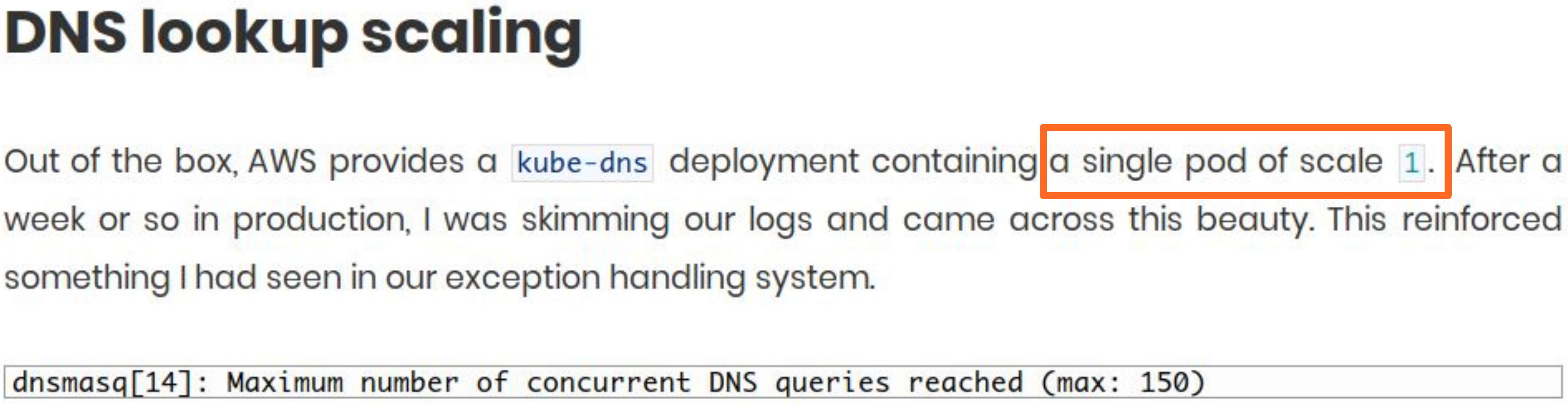


- Networking
- Networking—Limited pod capacity per subnet & VPC
- Networking—Limited pod capacity per worker node
- Networking—Kubernetes scheduler is unaware about actual IP availability
- Networking—Some pods cannot be accessed from peered networks by default
- Default worker AMI
- AMI—Based on Amazon 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

AWS EKS IN PRODUCTION



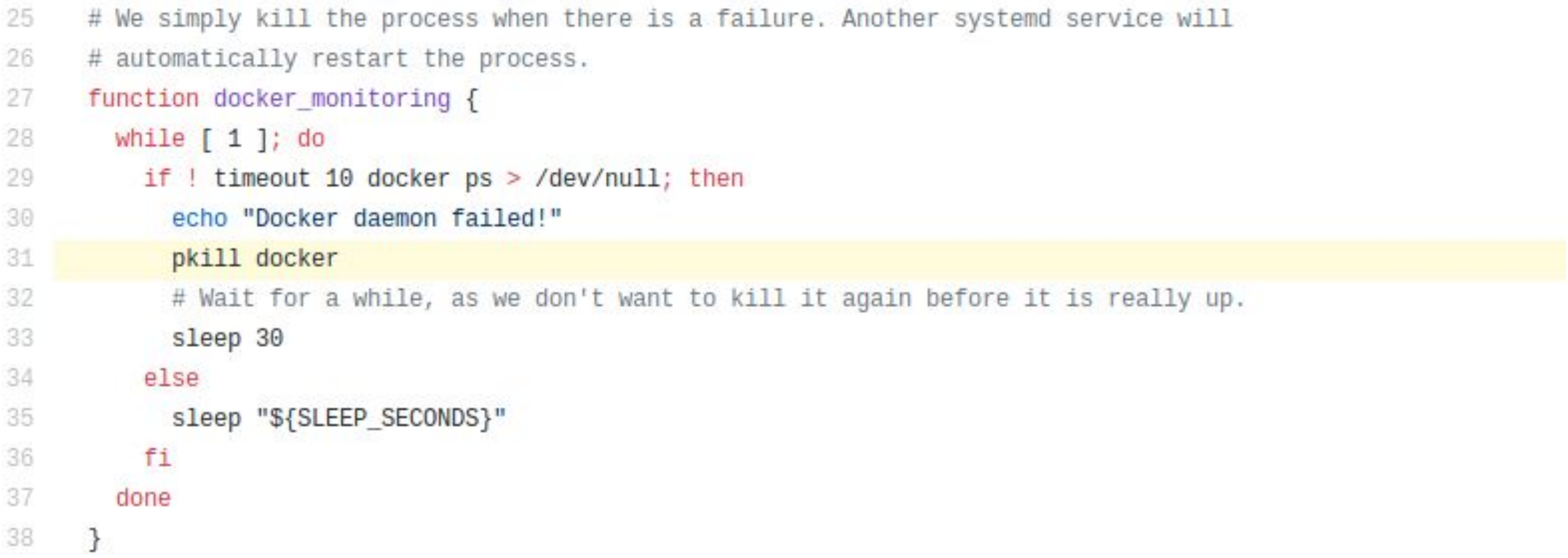
DNS lookup scaling

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)
```

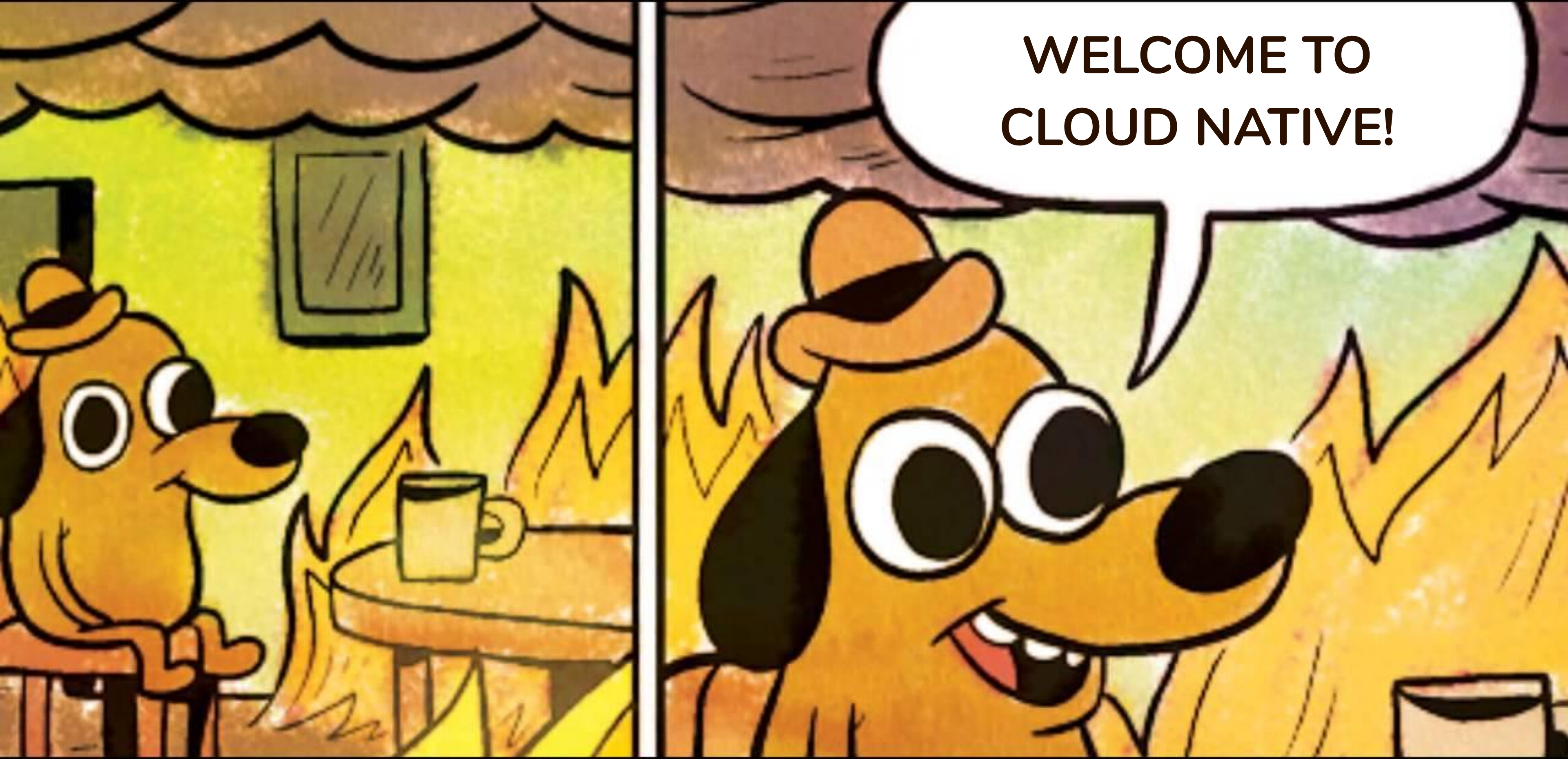
https://kubedex.com/90-days-of-aws-eks-in-production/

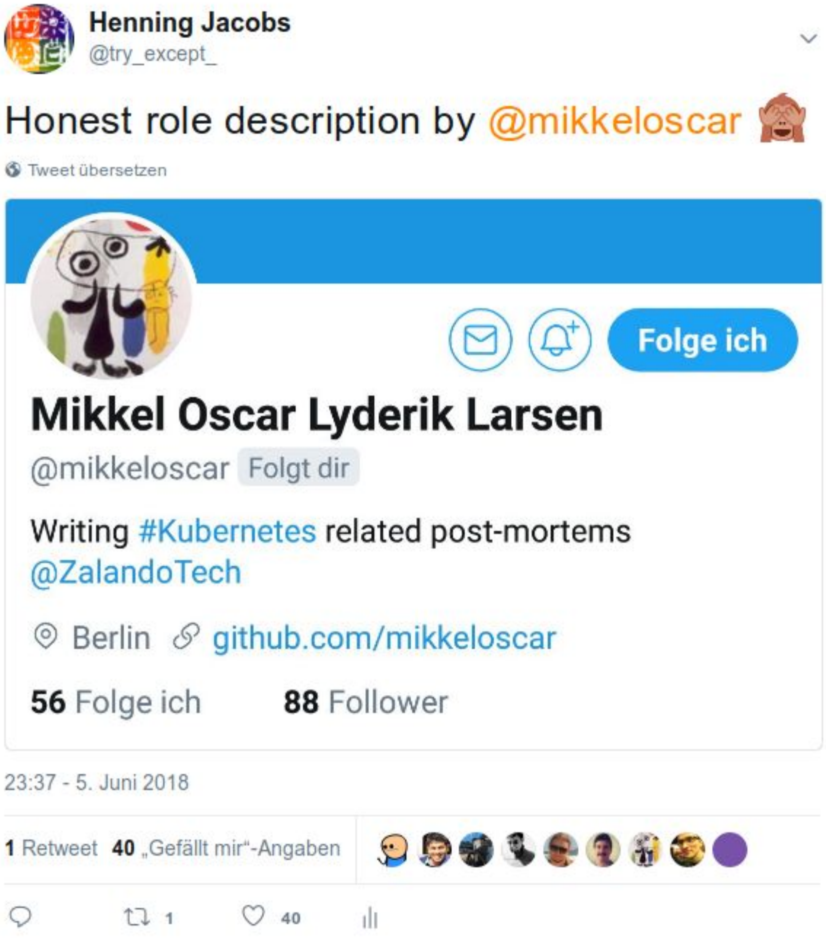
DOCKER.. (ON GKE)



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

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

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

KUBERNETES FAILURE STORIES

A compiled list of links to public failure stories related to Kubernetes.

k8s.af

We need more failure talks!

Istio? Anyone?

OPEN SOURCE

Kubernetes on AWS
github.com/zalando-incubator/kubernetes-on-aws

AWS ALB Ingress controller
github.com/zalando-incubator/kube-ingress-aws-controller

Skipper HTTP Router & Ingress controller
github.com/zalando/skipper

External DNS
github.com/kubernetes-incubator/external-dns

Postgres Operator
github.com/zalando-incubator/postgres-operator

Kubernetes Resource Report
github.com/hjacobs/kube-resource-report

Kubernetes Downscaler
github.com/hjacobs/kube-downscaler

zalando

QUESTIONS?

HENNING JACOBS

HEAD OF

DEVELOPER PRODUCTIVITY

henning@zalando.de

@try_except_

Illustrations by @01k