



Why Kubernetes?

OWL TECH &
INNOVATION DAY

2019-09-26

HENNING JACOBS
@try_except_



ROLLING OUT KUBERNETES?

Y **Hacker News** [new](#) | [threads](#) | [past](#) | [comments](#) | [ask](#) | [show](#) | [jobs](#) | [submit](#)

▲ Ask HN: Do's/don'ts of working with Kubernetes you learned through experience?

33 points by fiddlerINT 1 day ago | [flag](#) | [hide](#) | [past](#) | [web](#) | [favorite](#) | 26 comments

"We are rolling out Kubernetes to production next month and I'm interested to hear from people who made that step already."

DON'T USE IT !!!!!

Y **Hacker News** [new](#) | [threads](#) | [past](#) | [comments](#) | [ask](#) | [show](#) | [jobs](#) | [submit](#)

▲ Ask HN: Do's/don'ts of working with Kubernetes you learned through experience?

33 points by fiddlerINT 1 day ago | [flag](#) | [hide](#) | [past](#) | [web](#) | [favorite](#) | 26 comments

We are rolling out Kubernetes to production next month and I'm interested to hear from people who made that step already.

▲ iamnothere123 5 hours ago [-]

DON'T USE IT !!!!!

[reply](#)

▲ anon284271 19 hours ago [-]

Don't use Kubernetes.

[reply](#)



MATTHIAS ENDLER

Backend Engineer at trivago.
Likes just-in-time compilers
and hot chocolate. [About me.](#)



Search

Published on 21st of March, 2019

Maybe You Don't Need Kubernetes





Corey Quinn

@QuinnyPig

Follow



Nuclear hot take: nobody will care about Kubernetes in five years.

CZnative @ home @pczarkowski

Replying to @tmclaughbos @iteration1 @behemphi

As I keep telling people, if you have a kubernetes strategy you've already failed. Kubernetes should be an implementation detail at the tactical level to deal with the strategic imperative of solving the problems that are halting the flow of money.

6:32 PM - 6 Feb 2019

97 Retweets 439 Likes



41



97



439



KUBERNETES FAILURE STORIES



François Zaninotto

@francoisz



Kubernetes Failure Stories. The fact that this list has a dedicated website is a serious symptom of the complexity problem. [#Docker](#) [#DevOps](#) [k8s.af](#)



[Tweet übersetzen](#)

Kubernetes Failure Stories

A compiled list of links to public failure stories related to Kubernetes. Most recent publications on top.

- 10 Ways to Shoot Yourself in the Foot with Kubernetes, #9 Will Surprise You - Datadog - KubeCon Barcelona 2019
 - involved: CoreDNS, `ndots:5`, IPVS conntrack, `imagePullPolicy: Always`, DaemonSet, NAT instances, `latest` tag, API server `OOMKill`, kube2iam, cluster-autoscaler, PodPriority, audit logs, `spec.replicas`, AWS ASG rebalance, CronJob, Pod toleration, zombies, `readinessProbe.exec`, cgroup freeze, kubect!
 - impact: unknown, API server outage, pending pods, slow deployments
- How Spotify Accidentally Deleted All its Kube Clusters with No User Impact - Spotify - KubeCon Barcelona 2019
 - involved: GKE, cluster deletion, browser tabs, Terraform, global state file, git PRs, GCP permissions
 - impact: no impact on end users
- Kubernetes Failure Stories - Zalando - KubeCon Barcelona 2019
 - involved: Skipper-Ingress, AWS, `OOMKill`, high latency, CronJob, CoreDNS, `ndots:5`, etcd, CPU throttling
 - impact: multiple production outages
- Oh Sh*t! The Config Changed! - Pusher - KubeCon Barcelona 2019
 - involved: AWS, nginx, ConfigMap change
 - impact: production outage
- Misunderstanding the behaviour of one templating line - Skyscanner - blog post 2019
 - involved: HAProxy-Ingress, Service VIPs, Golang templating

ZALANDO AT A GLANCE

~ 5.4 billion EUR
revenue 2018

> 15.000
employees in
Europe

> 79%
of visits via
mobile devices

**> 250
million** visits
per
month

**> 26
million** product choices
active customers

> 300.000
~ 2.000
brands

17
countries



A BRIEF HISTORY OF ZALANDO TECH

2010

"Sysop-Test"

"QA-Test"

Zalando (Magento-)Shop-Deploy Checkliste

Tag: _____ SysOP: _____ Datum: _____

- () Releasefreigabe erfolgt
- (X) SVN-Tag angelegt
- () **create tag zal_1X_XX-XX-XX\ln<Freigabe Mail>**
- () Datenbankänderungen auf Reihenfolge prüfen
- () Falls Solr-Update (<https://devwiki.zalando.de/Sysop/Deployment>)
 - o Booster-Config im LiveOverlay auf **solr02** ändern (solr01 auskommentieren)
 - o 'set_update live' mit dem alten Release (neu config deployen)
 - o Replikation von Solr02 beenden
 - o booster-cron 'cron.d/zalando-booster' deaktivieren
 - o Booster-Config in LiveOverlay auf **solr01** ändern (solr02 auskommentieren)
- () Konfigurationsänderungen aus Datenbankänderungen in LiveOverlay übernehmen
- () deploy-wizard mit neuem Tag starten
- (X) **Vor deaktivieren** von http01, im LB, **admin01 mit rsync-deploy-live_all updaten**
 - o Konfigurationsänderungen aus Wiki nach zalando-live übernehmen
admin wird nicht von LiveOverlay bedient!
 - o solr01 updaten
 - o booster-full-update starten und prüfen
- () Deaktivierung/Swtich von http01
- (X) Sysop-Test http01
- () Fehler, fehlende/falsche Bilder sofort an QA melden und weiteren verlauf klären
- (X) Aktivierung von http01 im LB
- (X) QA-Test auf http01
- (X) Deaktivierung/Switch von http01-http03
- (X) Sysop-Test http01-http03
- (X) Aktivierung von http01-http03 im LB
- (X) QA-Test auf http01-http03
- (X) Nach Okay: Switch auf allen https und squids
- () Falls kein Solr-Update, jetzt **admin01 mit rsync-deploy-live_all updaten**
 - o Konfigurationsänderungen aus Wiki nach zalando-live übernehmen
admin wird nicht von LiveOverlay bedient!
- () Crons testen und aktivieren

Bei Probleme die **nicht innerhalb von 5 Minuten** behoben werden können folgende Personen Anrufen und informieren!



2013: SELF SERVICE

```
$ grep description ~/Projects/zalando/deployctl/project.desc  
description: Deployment tool with ncurses frontend and  
project/instance database with web frontend
```

```
DEPLOYCTL  
  
Main menu: Select what to do  
Deploy  
Deploy (force rebuild of all projects)  
Deploy DB API schemas only  
Build only (does not switch instances)  
Deploy only (use precompiled WAR)  
Expert Mode  
View Log  
Exit  
  
DEBUG root: FESN segments: {u'customer': [(u'fesn01', u'9620'), (u'fesn04', u'9620'), (u'fesn07',
```

Overview Projects Instances Databases Jobs Teams Web Services
Configuration Issues

Deployctl

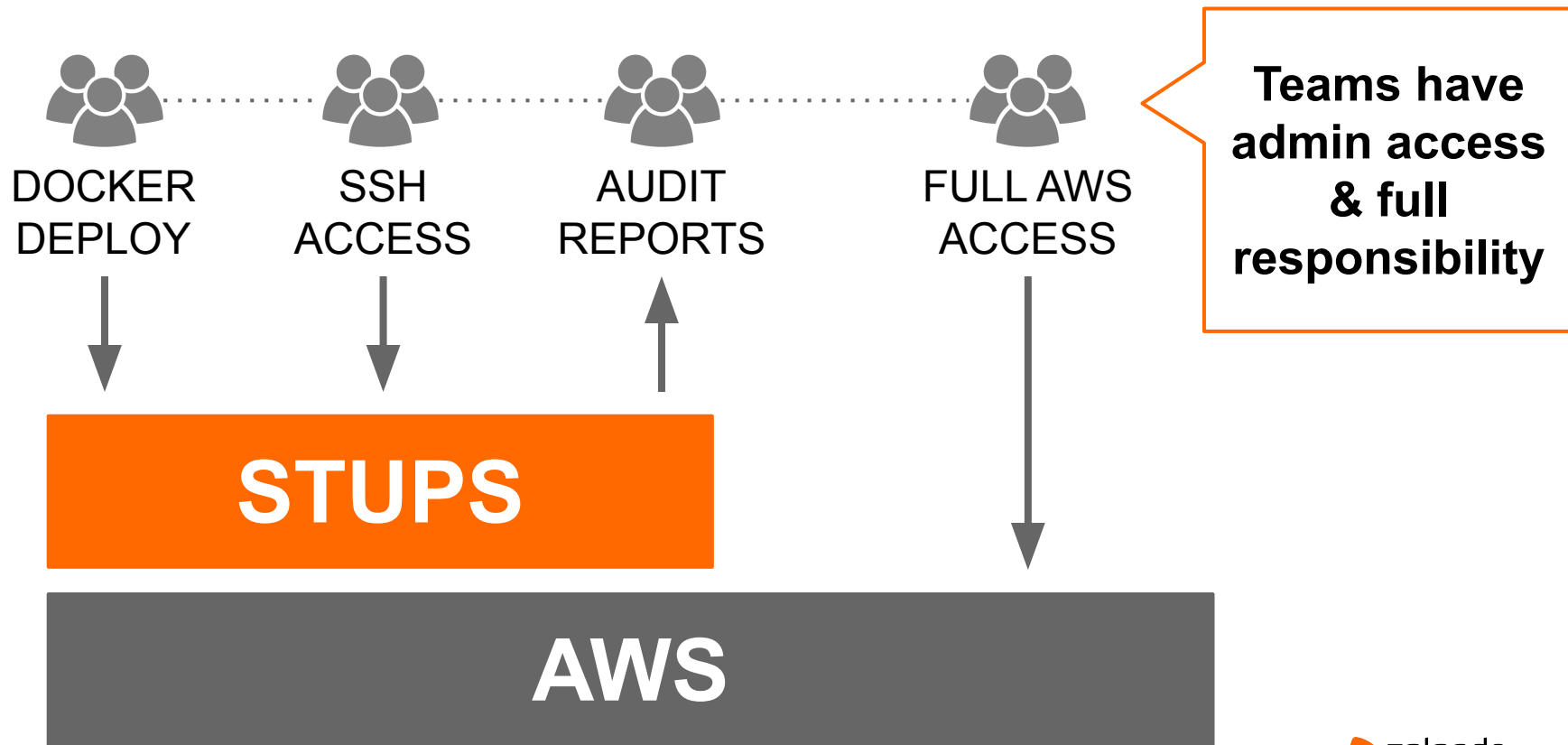
Current Deploy Locks (1 / 7 slots used)

Created	Environment	Tag	Projects	User	Stage	Comment
2013-10-31 11:41:29	12 m ago PATCH-STAGING	R13_00_44	shop cms export recommendations-servlet recommendations-solr shopadmin shop-config-service	mrosemann	switch	Instances

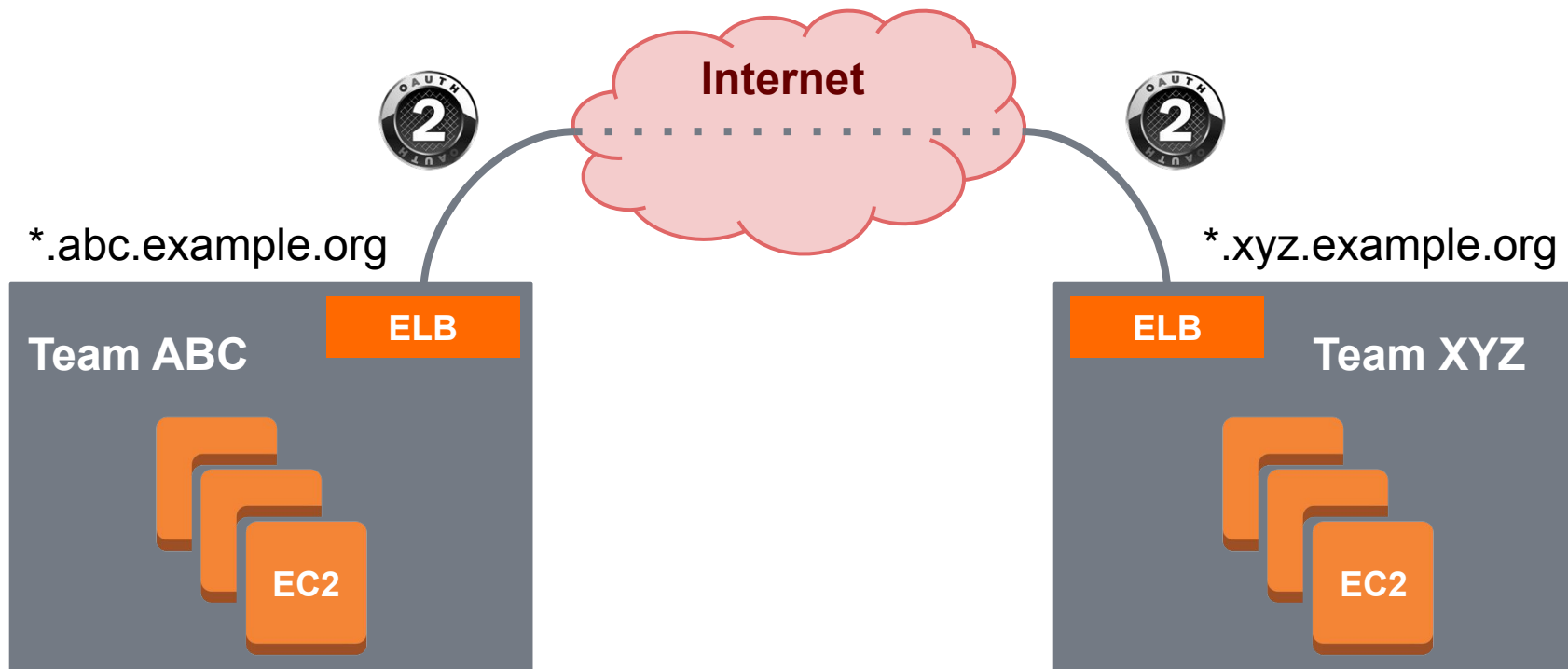
Recent Deployments

Time	Environment	Tag	Projects	User	Stage	Comment
2013-10-31 11:51:38	2 m ago INTEGRATION	trunk	production-tools-jet-ing	froske	done	Instances
2013-10-31 11:51:13	2 m ago INTEGRATION	trunk	shop	sharder	done	Instances
2013-10-31 11:50:12	3 m ago INTEGRATION	trunk	production-tools-jet-ing	froske	switch	Instances
2013-10-31 11:50:12	3 m ago INTEGRATION	trunk	production-tools-jet-ing	froske	api-schema-check	Instances

2015: RADICAL AGILITY



2015: ISOLATED AWS ACCOUNTS



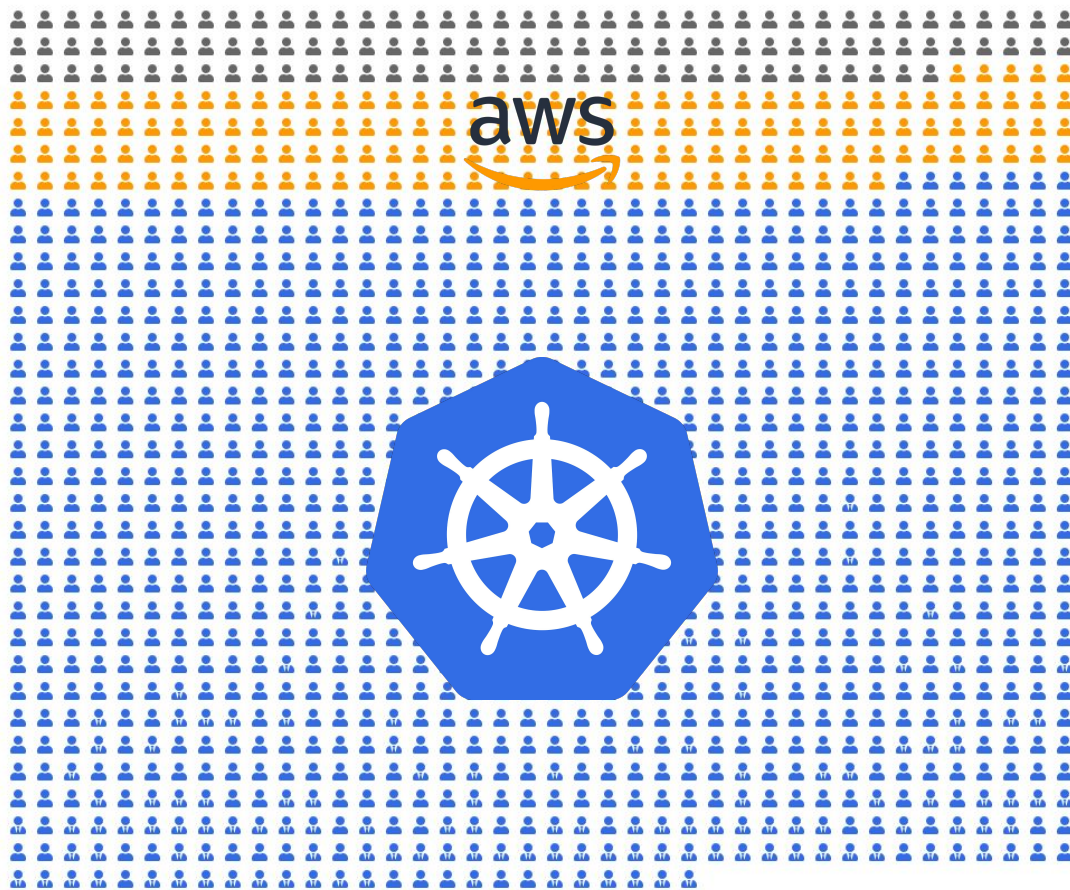
2019: SCALE

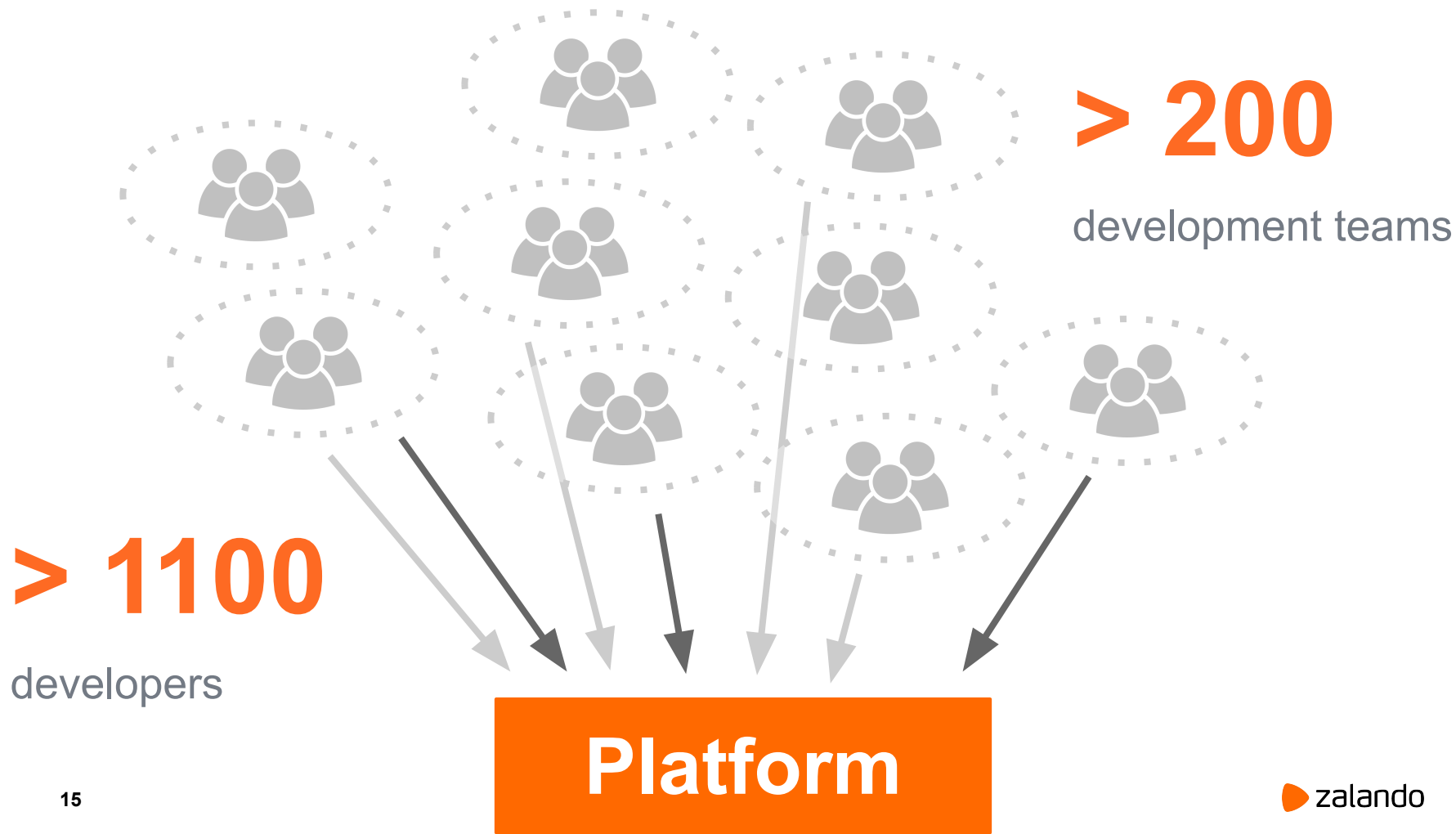
396 Accounts



140 Clusters

2019: DEVELOPERS USING KUBERNETES





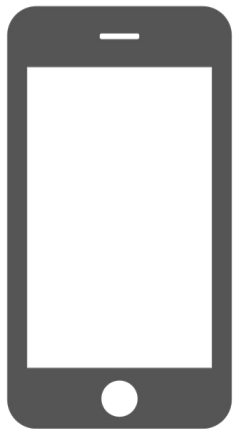
YOU BUILD IT, YOU RUN IT

The traditional model is that you take your software to the wall that separates development and operations, and throw it over and then forget about it. Not at Amazon.

***You build it, you run it.** This brings developers into contact with the day-to-day operation of their software. It also brings them into day-to-day contact with the customer.*

- [A Conversation with Werner Vogels, ACM Queue, 2006](#)

ON-CALL: YOU OWN IT, YOU RUN IT



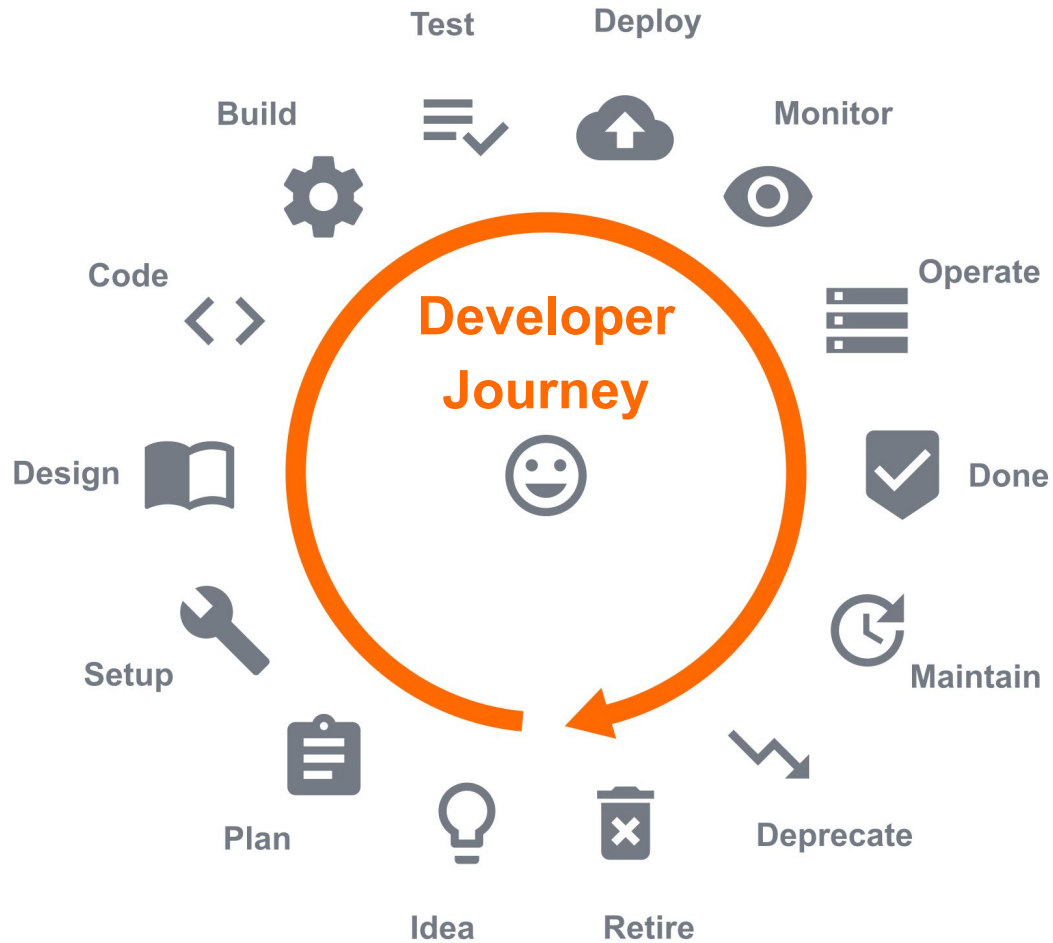
*When things are broken,
we want people with the best
context trying to fix things.*

- [Blake Scrivener, Netflix SRE Manager](#)



DEVELOPER JOURNEY

**Consistent story
that models
all aspects of SW dev**



Correctness
Compliance
GDPR
Security
Cost Efficiency
24x7 On Call
Governance
Resilience
Capacity
...



DEVELOPER PRODUCTIVITY



Cloud Native Application Runtime

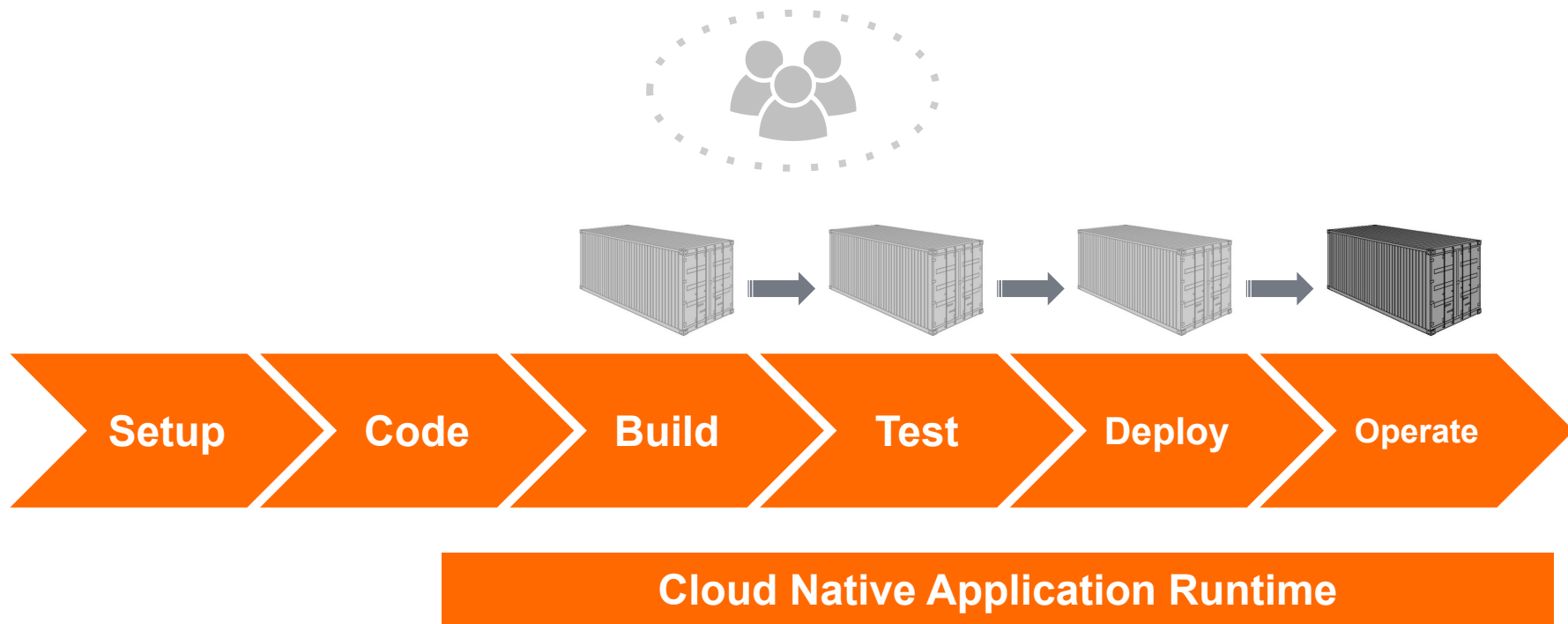
CLOUD NATIVE

*.. uses an **open source** software stack to deploy applications as **microservices**, packaging each part into its own **container**, and **dynamically orchestrating** those containers to optimize resource utilization.*

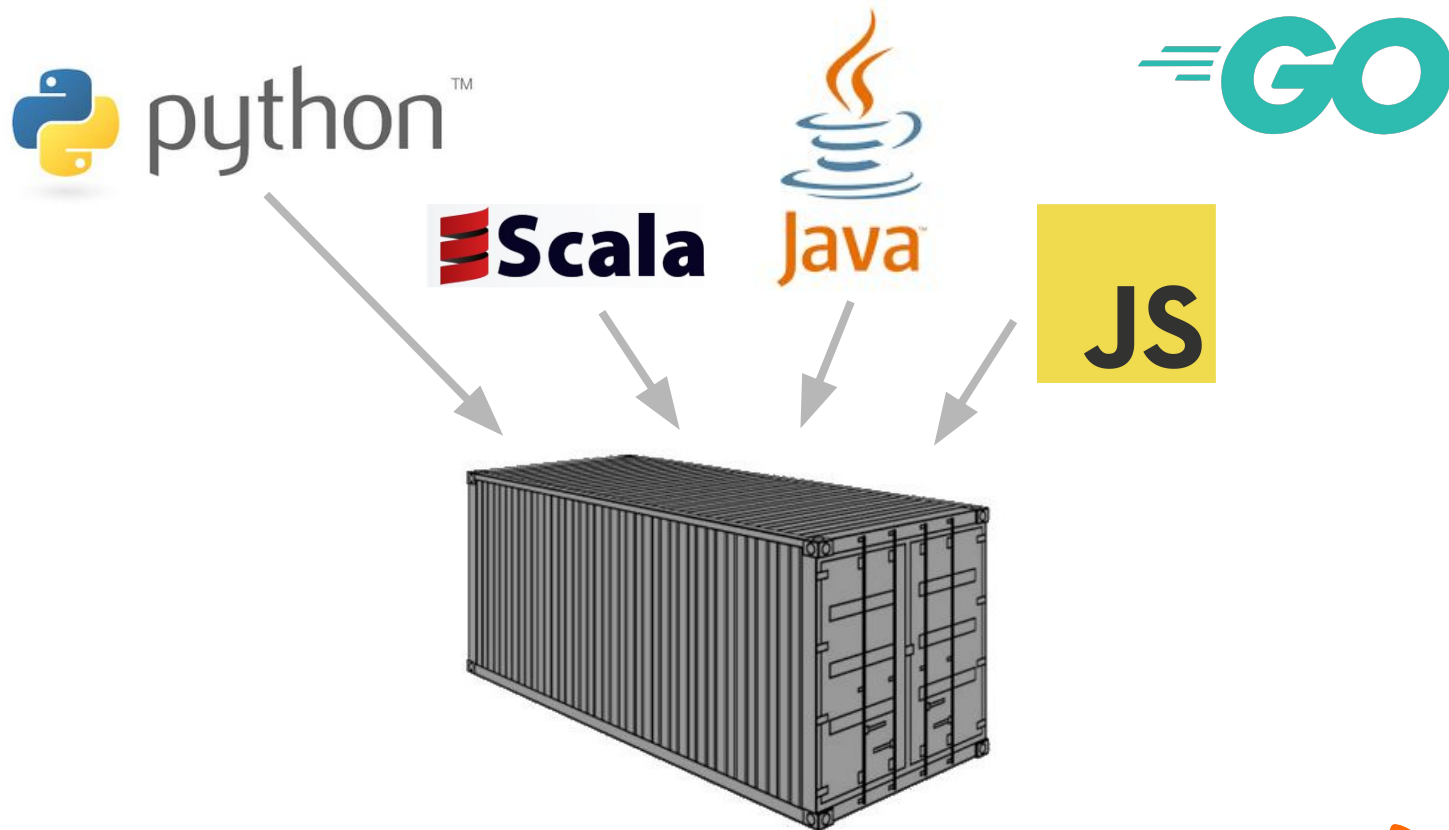
*Cloud native technologies enable software developers to **build great products faster**.*

- <https://www.cncf.io/>

CONTAINERS END-TO-END



CONTAINERS



CONTAINERS



kubernetes



IBM Cloud



Welcome to the Developer Journey

The Developer Console is your guide for all the things you need to take an application from conception to production within 30 minutes. Every application starts as an idea, which subsequently undergoes planning, design, coding and testing phases until, at the end of its lifetime, it is retired.

You will find links to documentation, tools and information for your journey in this Developer Console.

We here in Developer Productivity want to support you as best we can and if something is missing, you need more information or features, do not hesitate to reach out to us by clicking on the feedback icon in the top right. Any feedback is highly appreciated.

Welcome to the Developer Console

You can now find all the documentation you need for the whole Developer Journey in one place:

The Developer's Journey

Test

Unit tests, contract tests, integration tests, end-to-end tests and smoke tests can be automated via our build-tooling.

[Learn more...](#)



Build



Deploy



Monitor



Operate



Done



Maintain



Deprecate



Retire



Idea



Plan



Setup



Design



Code



Design



RESTful API Guidelines



API Portal



Zally API Linter



Fashion Content Platform (AI & Data APIs)



Test



Setup



ZACK Role Management



GitHub Enterprise



Create Git Repository



YOUR TURN



Zappr



Create database cluster



AWS Account Chooser



Build



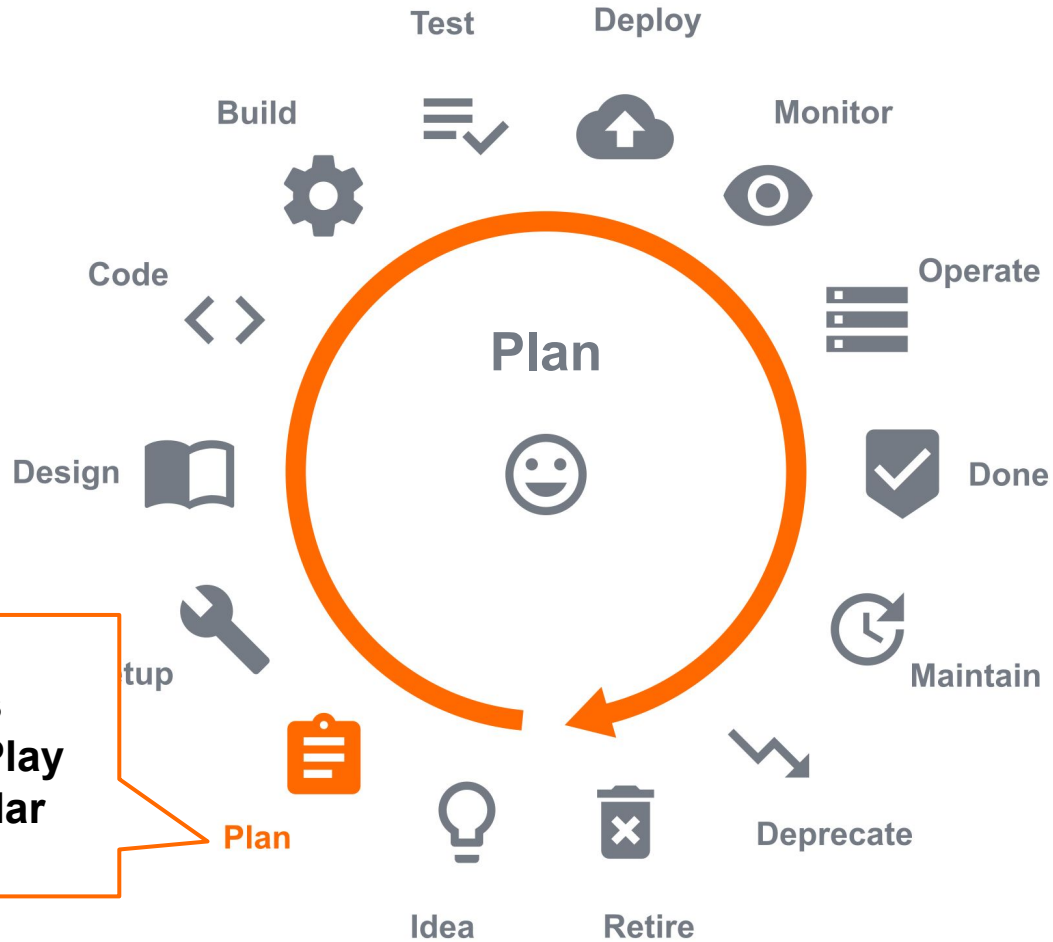
Continuous Delivery Platform (CDP)

☐ don't show me again

close



PLAN & SETUP



Zalando Tech Radar — 2019.09

Frameworks

ADOPT

1. Akka (Scala)
2. Node.js
3. Play (Scala)
4. ReactJS
5. RxJava (Android)
6. scikit-learn
7. Spring

TRIAL

8. Akka-Http
9. Angular
10. AspectJ
11. Camel
12. Camunda
13. OpenNLP
14. TensorFlow
15. Thymeleaf

ASSESS

16. Aurelia
17. Ember.js
18. gRPC
19. Http4s
20. JOOQ
21. Redux
22. Vert.x
23. Vue.js

HOLD

24. Activiti
25. AngularJS 1.x
26. BackboneJS
27. Drools
28. Spray

Infrastructure

ADOPT

68. Docker
69. Hystrix
70. Kubernetes
71. Nginx
72. OpenTracing
73. Tomcat
74. ZMON

ASSESS

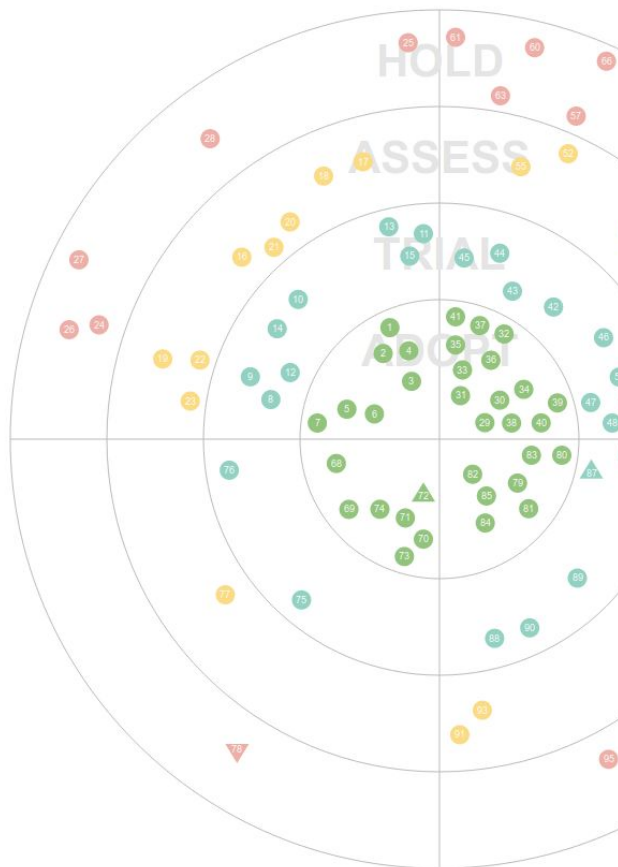
77. AWS Lambda

HOLD

78. STUPS

TRIAL

75. Failsafe
76. Undertow



Kotlin ASSESS

edit this page

Kotlin is a pragmatic, multi-paradigm, multi-platform, modern, safe, interoperable and toolable programming language by JetBrains. Its primary focus is and has been the JVM ecosystem.

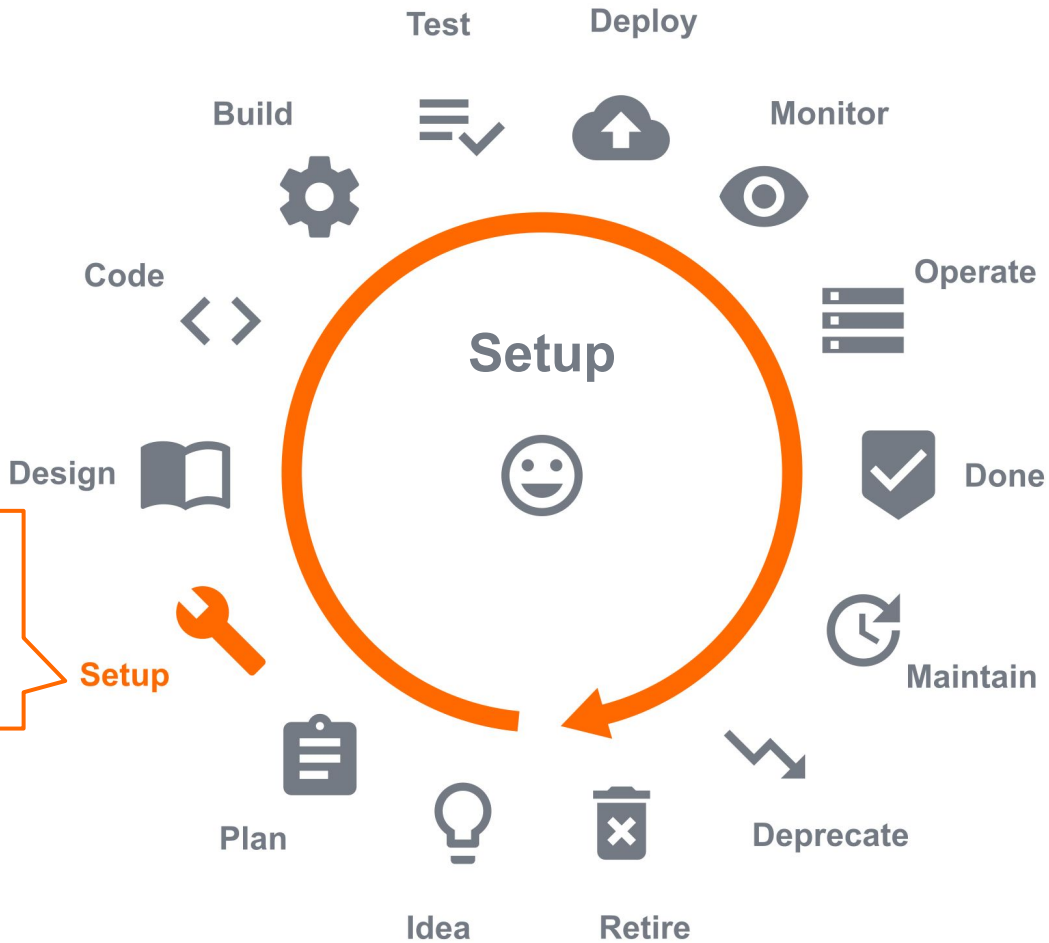
Nonetheless, it supports compilation to JavaScript and native code via LLVM. More than just compilation, Kotlin makes sure to integrate very well with the respective ecosystem, be it package managers or build tools. Plus, for each additional feature the designers deeply think about its implications with respect to toolability, compilation performance, learning costs and multi-platform suitability. All of this is a reflection of Kotlin's philosophy: Pragmatism and interoperability over fancy but costly language features.

Kotlin was developed mostly in the open with constant feedback from the community. A handful of companies were using the language in production even before the 1.0 release. Nowadays, many companies use Kotlin in critical applications. Notable examples include: Android, Gradle, Netflix, Pinterest, Atlassian, Coursera, Evernote, Trello and Uber.

Notable language features:

1. Nullable types — with compiler checking on dereferencing
2. Smart casts — type narrowing based on control flow
3. Extension methods — define methods on any type
 1. extend APIs with discoverable methods
4. Data classes — create POJOs easily with auto-generated equals, hashCode and toString methods
5. Properties — auto-generated properties for class fields
6. Functional programming — supports closures, (explicit) tail recursion and higher order functions

**Application
Bootstrapping**



Create a new repository

A repository contains all the files for your project, including the revision history

Organisation * hackweek	Repository Name * myapp
----------------------------	----------------------------

Description

Team ID *

Repository type *
Code


















Bootstrap with application template
web-java-spring-boot-rest

Add .gitignore template

* Mandatory fields

Create Repository

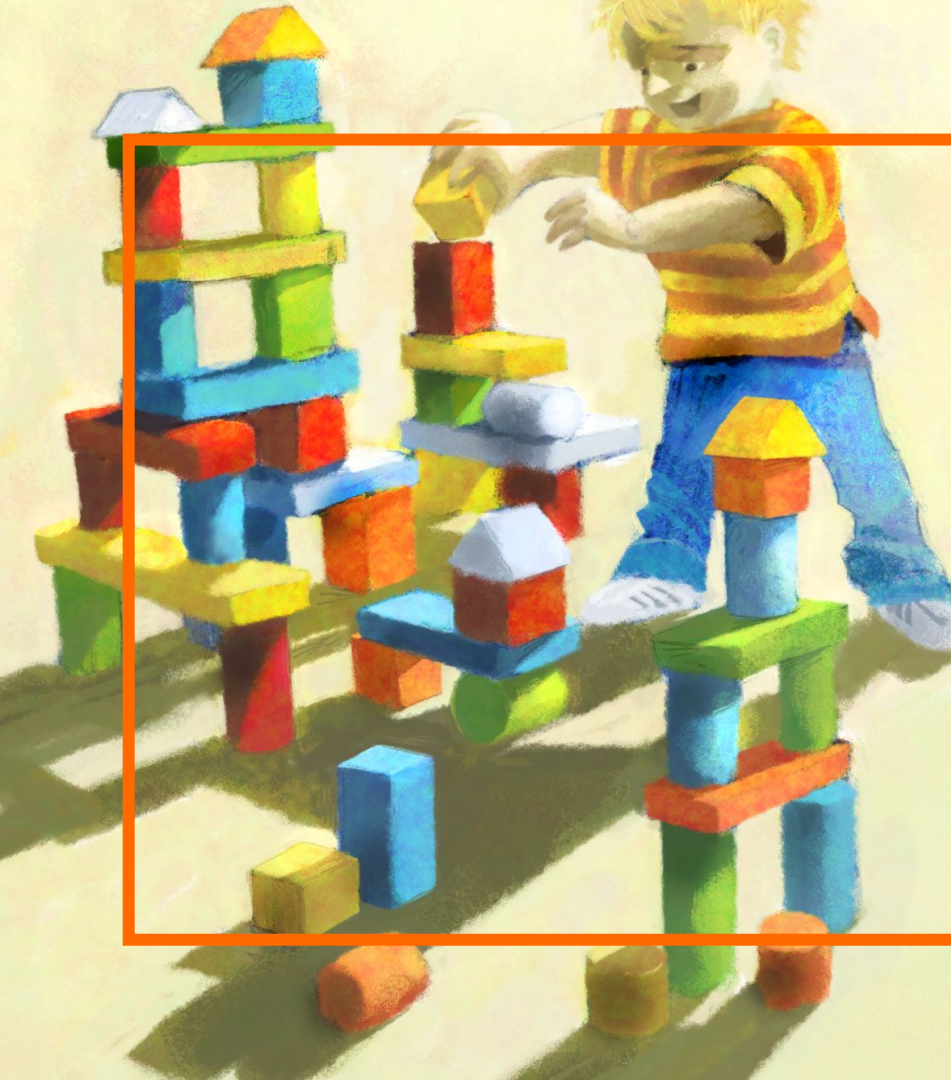
Branch: **master** ▾[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download ▾](#)

 committed with  Updated readme due to switch to riptide (#34)		Latest commit 965218e 7 hours ago
 api	Migrated out of twintip (#29)	20 days ago
 deploy/apply	Remove not existing uid privilege (#7)	7 months ago
 gradle/wrapper	Upgrade dependencies to newest versions (#16)	5 months ago
 src	Run Docker build from project root (#33)	18 days ago
 .gitignore	Move over to riptide (#31)	18 days ago
 .zappr.yaml	Add match label selector (#4)	7 months ago
 Dockerfile	Run Docker build from project root (#33)	18 days ago
 README.md	Updated readme due to switch to riptide (#34)	7 hours ago
 build.gradle	Move over to riptide (#31)	18 days ago
 configure.sh	Configure script adapted to osx (#30)	18 days ago
 delivery.yaml	Run Docker build from project root (#33)	18 days ago
 gradlew	Upgrade dependencies to newest versions (#16)	5 months ago
 gradlew.bat	Transition to single repo per template	7 months ago
 settings.gradle	Use  as team to prepare for deployment (#1)	7 months ago

README.md

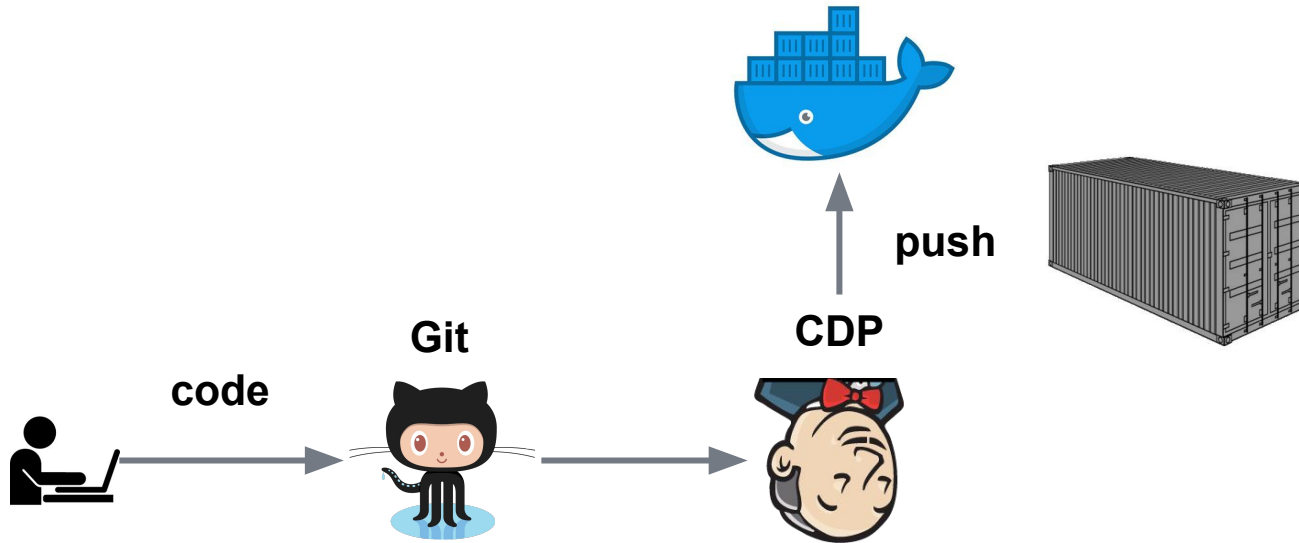
Java - Spring Boot - REST

This project is a highly opinionated template for a Java Spring Boot based REST service. It includes various Zalando specific libraries like e.g. [ZMON Actuator](#), [Tracer](#) and [Failsafe Actuator](#). See the full list below.



BUILD & TEST

CONTINUOUS DELIVERY PLATFORM: BUILD



Add more commits by pushing to the  branch on 



Some checks haven't completed yet

[Hide all checks](#)

1 pending and 5 successful checks



zappr — This PR needs 2 more approvals (0/2 given).

Required



CDP/pr/build — Build succeeded.

[Details](#)



CDP/pr/deployment-deploy-staging — Deployment succeeded.

[Details](#)



CDP/pr/deployment-deploy-prod — Deployment succeeded.

[Details](#)



CDP/pr/lifecycle — Lifecycle succeeded.

[Details](#)



Required statuses must pass before merging

All required [status checks](#) on this pull request must run successfully to enable automatic merging.

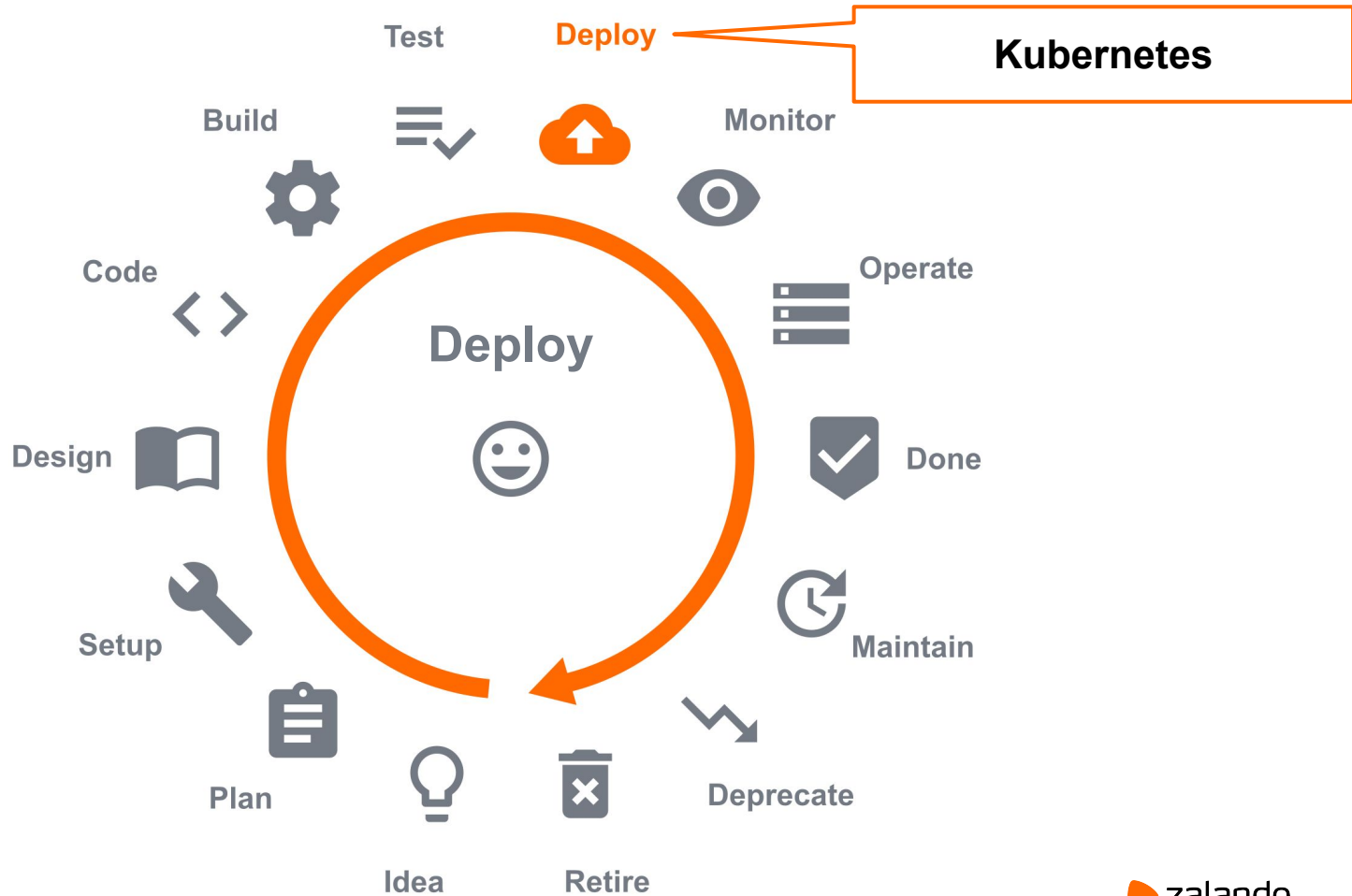
Squash and merge



or view [command line instructions](#).



DEPLOY



DEPLOYMENT CONFIGURATION



```
├── deploy/apply
│   ├── deployment.yaml
│   ├── credentials.yaml # Zalando IAM
│   ├── ingress.yaml
│   └── service.yaml
└── delivery.yaml # Zalando CI/CD
```

INGRESS.YAML



```
kind: Ingress
metadata:
  name: "..."/>
spec:
  rules:
    # DNS name your application should be exposed on
    - host: "myapp.foo.example.org"
      http:
        paths:
          - backend:
              serviceName: "myapp"
              servicePort: 80
```


TEMPLATING: MUSTACHE

```
kind: Ingress
metadata:
  name: "..."/>
spec:
  rules:
    # DNS name your application should be exposed on
    - host: "{{{APPLICATION}}}.example.org"
      http:
        paths:
          - backend:
              serviceName: "{{{APPLICATION}}}"
              servicePort: 80
```






























































































CONTINUOUS DELIVERY PLATFORM

DEPLOYMENT UNITS

RENDERING-ENGINE

Exclude PRs: ☐



Pipeline	Started	Action	Pipeline Runs				
 pr-1785-2   a4be6c  PR	3h ago 		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
 pr-1785-1   e65617  PR	3h ago 		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
Remove `any` usage from our code pr-1784-1   1bbf84  PR	6h ago  		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
master-1188 master-1188   1e6657	1d ago 		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
Do not bundle node_modules in re package pr-1783-1   d5aa17  PR	1d ago  		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
master-1187 master-1187   1e09f4	1d ago 		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
Document the renderer contribution workflow pr-1775-2   e1a92a  PR	2d ago  		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
master-1186 master-1186   06d929	2d ago 		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
 pr-1778-1   2099b1  PR	2d ago 		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
Document the renderer contribution workflow pr-1775-1   b6d0c7  PR	6d ago 		 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS

CDP: DEPLOY

DEPLOYMENT UNITS

TRACKING-DEPLOY

MASTER-91

TRACKING-DEPLOY

TEST

STAGING

PROD

GENERAL

DEPLOYMENT STATUS

LOGS

Ran for 1 m, 16 s

Deployment **merchant-parcels**

ReplicaSet **merchant-parcels-bd76cbc9b**

Pod **merchant-parcels-bd76cbc9b-s84kp** scalyr logs

Pod **merchant-parcels-bd76cbc9b-bwzff** scalyr logs

Pod **merchant-parcels-bd76cbc9b-hgj5p** scalyr logs

Ingress **merchant-parcels**

Hostname open

PlatformCredentialsSet

postgresql-db

Secret-credentials

Service **merchant-parcels**

Please give us your feedback

Do you like the new deployment status ?

☐ Yes ☐ No

What could be improved upon?

Submit

"glorified kubectl apply"

CDP: OPTIONAL APPROVAL

Developer Console

Search

Feedback

Support

DEPLOYMENT UNITS

FEED-SERVICE

MASTER-16

FEED-SERVICE

BUILD

DEPLOY-TEST

DEPLOY-PROD

GENERAL

LOGS

ACTIONS

APPROVE

REJECT

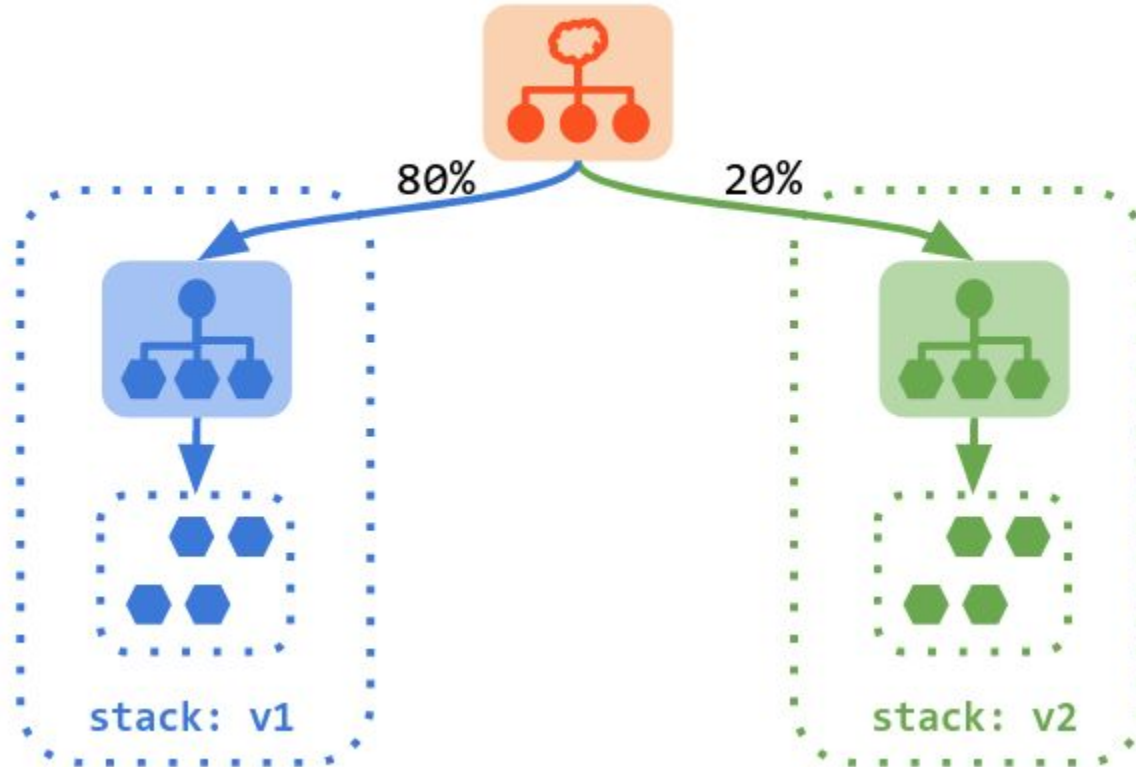
ERRORS & WARNINGS

Error	No Error
Warning	No Warnings

DETAILS

Id	n/a
Type	process
Status	PENDING_APPROVAL
Deployment	deploy-prod
Cluster	
Started at	n/a
Finished at	n/a

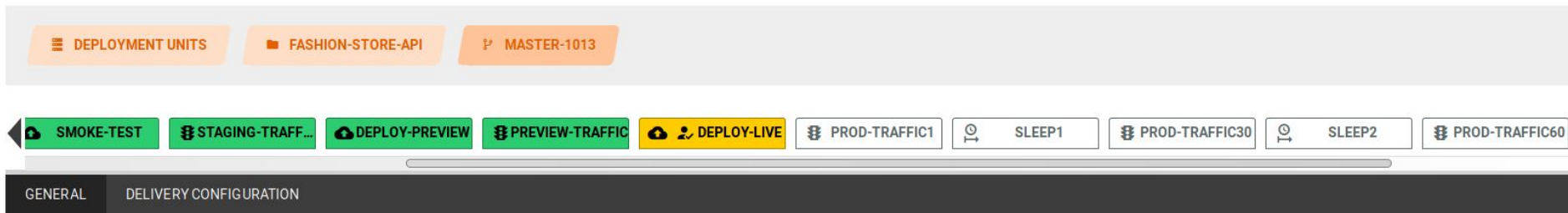
STACKSET: TRAFFIC SWITCHING

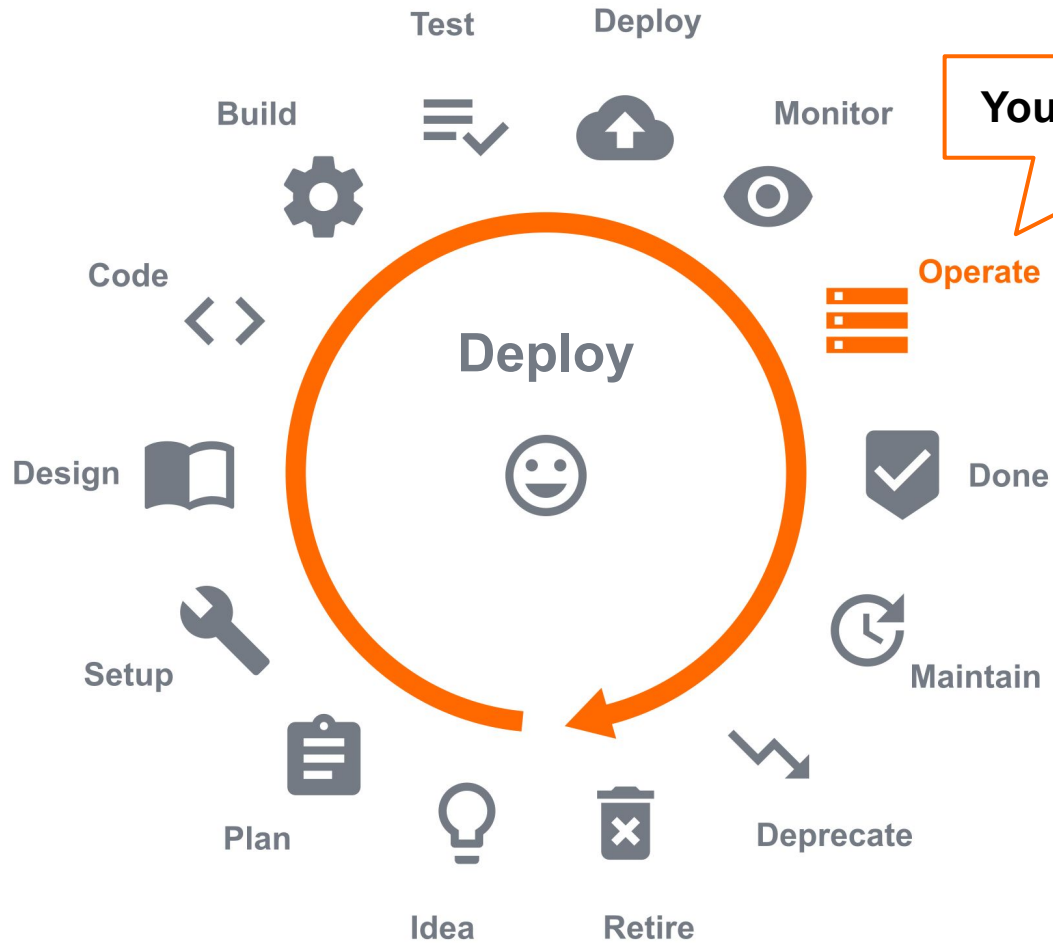


STACKSET CRD

```
kind: StackSet
...
spec:
  ingress:
    hosts: ["foo.example.org"]
    backendPort: 8080
  stackLifecycle:
    scaledownTTLSeconds: 1800
    limit: 5
  stackTemplate:
    spec:
      podTemplate:
        ...
```

TRAFFIC SWITCHING STEPS IN CDP





EMERGENCY ACCESS SERVICE

Emergency access by referencing Incident

```
zkubectl cluster-access request \  
  --emergency -i INC REASON
```



Privileged production access via 4-eyes

```
zkubectl cluster-access request REASON  
zkubectl cluster-access approve USERNAME
```



KUBERNETES WEB VIEW

Clusters

default

Search Kubernetes objects...

CLUSTER RESOURCES

Namespaces

Nodes

PersistentVolumes

CONTROLLERS

StackSets

Stacks

Deployments

CronJobs

Jobs

StatefulSets

POD MANAGEMENT

Ingresses

Services

Pods

ConfigMaps

CRDS

PlatformCredentialsSets

postgresqls

META

Resource Types

Events

default

/ pods,stacks,deployments,services

Pods

Name	Application	Component	Ready	Status	Restarts	Age	IP	Node	Nominated Node	Readiness Gates	CPU Usage	Memory Usage	Created
even-master-33-5db9d68c8d-5srrh	even		1/1	Running	0	5d18h			<none>	<none>	3m	313 MiB	2019-08-29 16:35:03
even-master-33-5db9d68c8d-72px8	even		1/1	Running	0	5d18h			<none>	<none>	3m	297 MiB	2019-08-29 17:05:16
even-master-33-5db9d68c8d-rrh9m	even		1/1	Running	0	5d18h			<none>	<none>	2m	312 MiB	2019-08-29 17:20:13

Stacks

Name	Desired	Current	Up-to-date	Available	Traffic	No-Traffic-Since	Age	Created
even-master-27	3	0	0	0	0	131d	210d	2019-02-05 20:11:56
even-master-29	3	0	0	0	0	131d	140d	2019-04-17 08:30:22
even-master-30	3	0	0	0	0	131d	131d	2019-04-25 12:19:11
even-master-31	3	0	0	0	0	50d	131d	2019-04-25 12:30:11
even-master-32	3	0	0	0	0	64d	64d	2019-07-01 15:19:45
even-master-33	3	3	3	3	100		50d	2019-07-15 12:18:58

Deployments

Name	Ready	Up-to-date	Available	Age	Containers	Images	Selector	Created
------	-------	------------	-----------	-----	------------	--------	----------	---------

kubectl get
pods,stacks,deploys,..

SEARCHING ACROSS 140+ CLUSTERS

Search

Search Text Search!

Resource Types ☒ CronJob ☐ DaemonSet ☒ Deployment ☒ Ingress ☒ Namespace ☐ Node ☐ PlatformCredentialsSet ☐ Pod ☐ ReplicaSet ☒ Service ☒ StackSet ☒ StatefulSet

✕ unselect all

etcd-operator (Deployment)
/cluster: namespaces/default/deployments/etcd-operator
Created: 2018-10-18 13:23:39 source.zalan.c /etcd-operator:v0.9.2-master-2
name: etcd-operator

etcd-operator (Deployment)
/cluster: namespaces/wpi/deployments/etcd-operator
Created: 2019-08-12 12:30:07 e.stups.zalan.dc /etcd-operator:v0.9.3
application: deployment-id: d-e8yt17ub9hxyt513sr27w66ea environment: staging pipeline-id: l-7bic5kvi6khdadtqzq5hy3q version: master-7

etcd-operator (Deployment)
/cluster: namespaces/default/deployments/etcd-operator
Created: 2018-10-19 14:13:50 source.zalan.dc /etcd-operator:v0.9.2-master-3
name: etcd-operator

etcd-operator (Deployment)
/clusters: namespaces/default/deployments/etcd-operator
Created: 2018-05-04 11:01:36 tups.zalan.dc /etcd-operator:v0.6.1-2
app: etcd component: operator

etcd-operator (Deployment)
/clusters: namespaces/incentives/deployments/etcd-operator
Created: 2018-07-03 08:12:51 .zalan.dc /etcd-operator:v0.9.3
application: deployment-id: d-so5ukevu2piyw5bdigzxc4gx3 environment: staging version: master-26

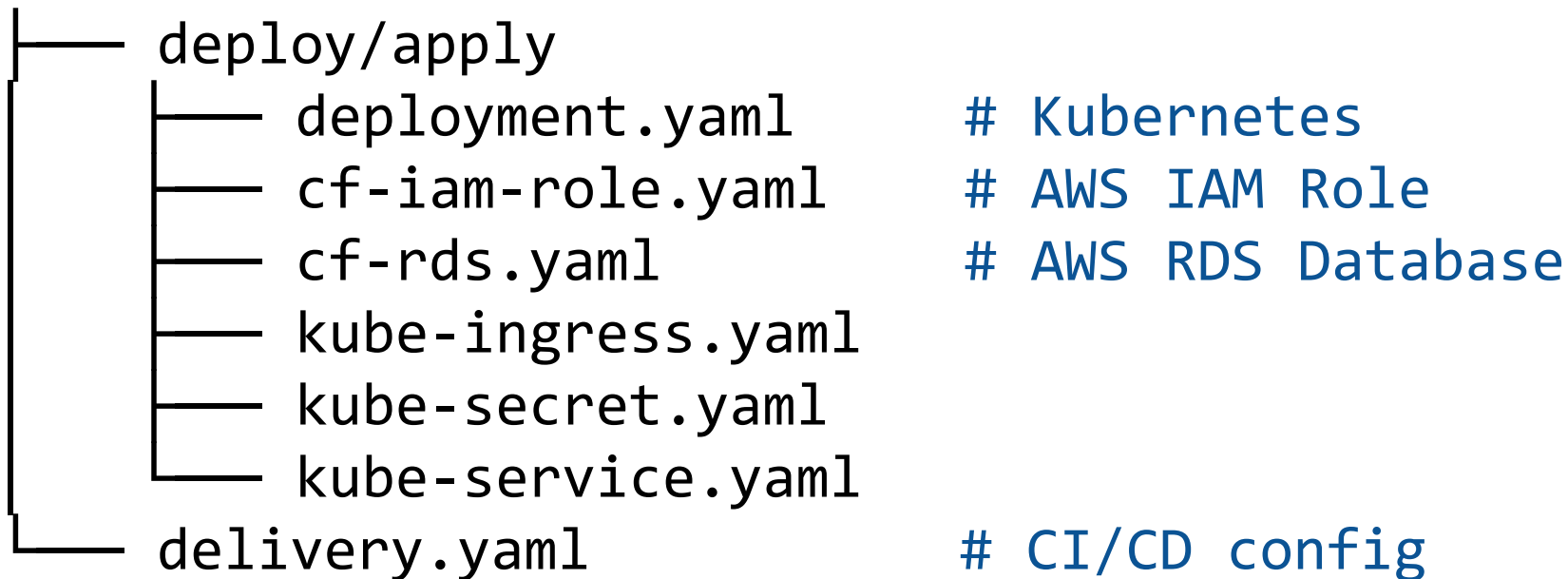


INTEGRATIONS

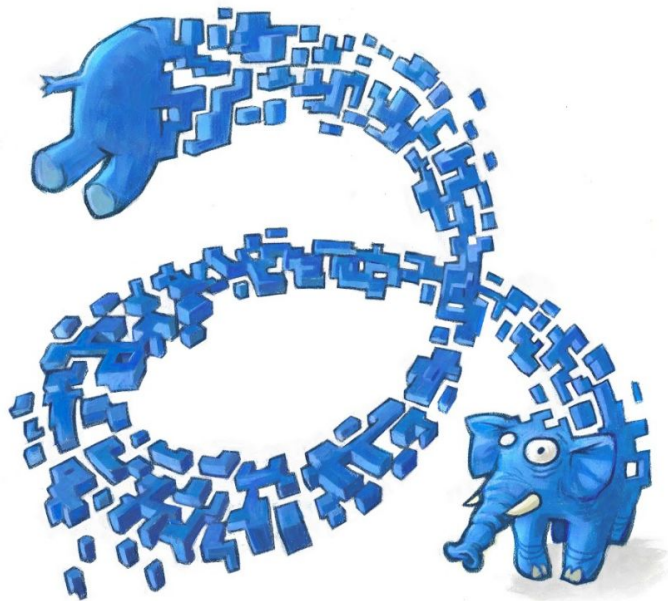


CLOUD FORMATION VIA CI/CD

"Infrastructure as Code"



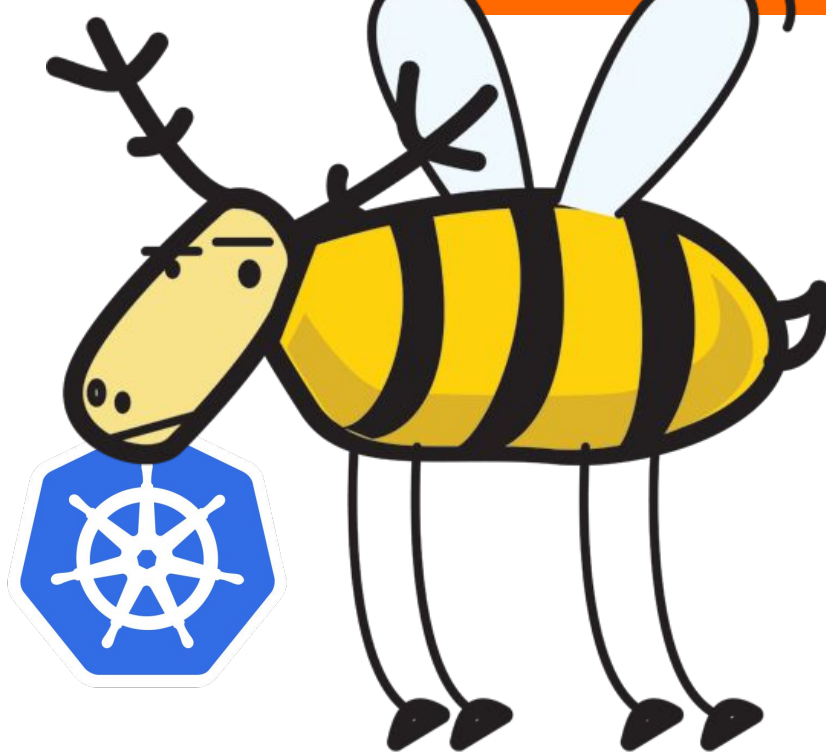
POSTGRES OPERATOR



Application to manage
PostgreSQL clusters on
Kubernetes

>500

clusters running
on Kubernetes



Elasticsearch
2.500 vCPUs
1 TB RAM

Elasticsearch in Kubernetes

github.com/zalando-incubator/es-operator/

SUMMARY

- Application **Bootstrapping**
- **Git** as source of truth and UI
- **4-eyes** principle for master/production
- Extensible **Kubernetes API** as primary interface
 - OAuth/IAM credentials
 - PostgreSQL, Elasticsearch
- **CloudFormation** for proprietary AWS services





MONITORING & COST EFFICIENCY

KUBERNETES RESOURCE REPORT

Overview Clusters Ingresses Teams Applications Pods

Cluster

https://

MASTER NODES

2

WORKER NODES

15

PODS

325

CPU REQUESTS / ALLOCATABLE

55.7 / 60.6

MEMORY REQUESTS / ALLOCATABLE

183.1 GiB / 241.5 GiB

MONTHLY COST

1,687.16 USD

You can potentially save every month by optimizing resource requests and reducing slack.

Price per requested vCPU is per hour and per requested GiB memory is per hour.

Nodes

Name	Role	Instance Type	S?	Version	CC	MC	CPU	Memory (GiB)	Cost
	worker	m4.xlarge	✗	v1.10.5	4	15.7 GiB	0.7 3.7 3.8	3.7 10.8 15.1	70.13
	worker	m4.xlarge	✗	v1.10.5	4	15.7 GiB	1.0 3.4 3.8	7.2 14.1 15.1	70.13
	worker	m4.xlarge	✗	v1.10.5	4	15.7 GiB	0.2 3.8 3.8	3.4 8.0 15.1	70.13
	master	m4.large		v1.10.5	2	7.8 GiB	0.3 1.0 1.8	2.8 1.3 7.3	87.66
	worker	m4.xlarge	✗	v1.10.5	4	15.7 GiB	0.2 2.7 3.8	3.3 9.4 15.1	70.13

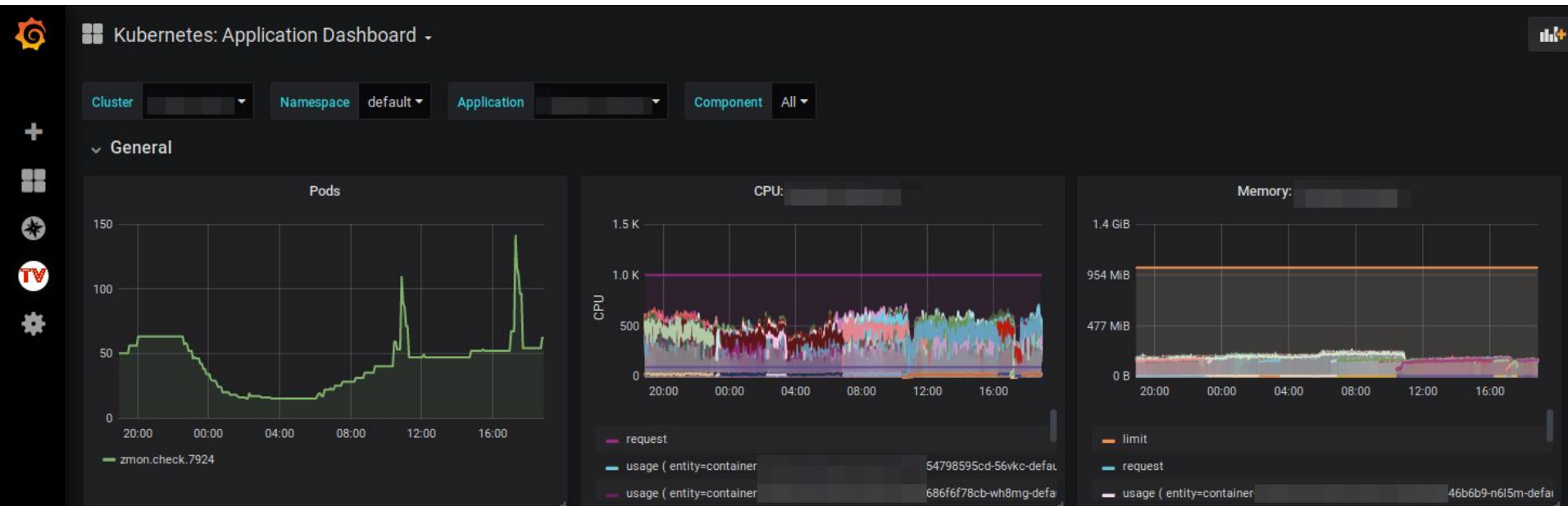
RESOURCE REPORT: TEAMS

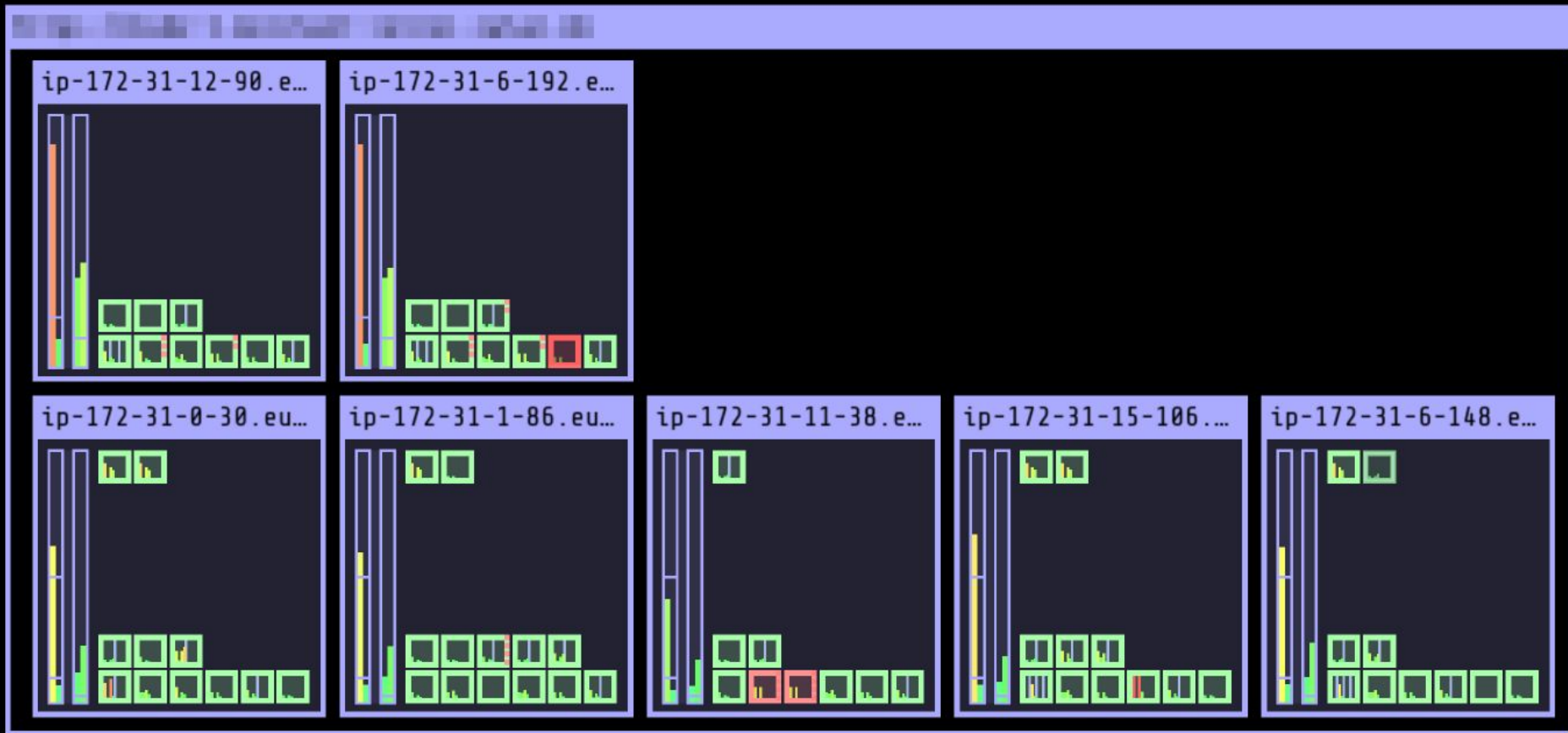
ID	C	A	P	CR	MR	CPU	Memory (MiB)	Cost	Slack Cost
	3	14	114	457.9	1.7 TiB	69.01 / 457.9	1,074,024 / 1,780,420	27,558.50	13,327.47
	1	9	251	428.95	426.2 GiB	137.99 / 428.95	164,613 / 436,384	19,406.82	13,111.24



Sorting teams by
Slack Costs

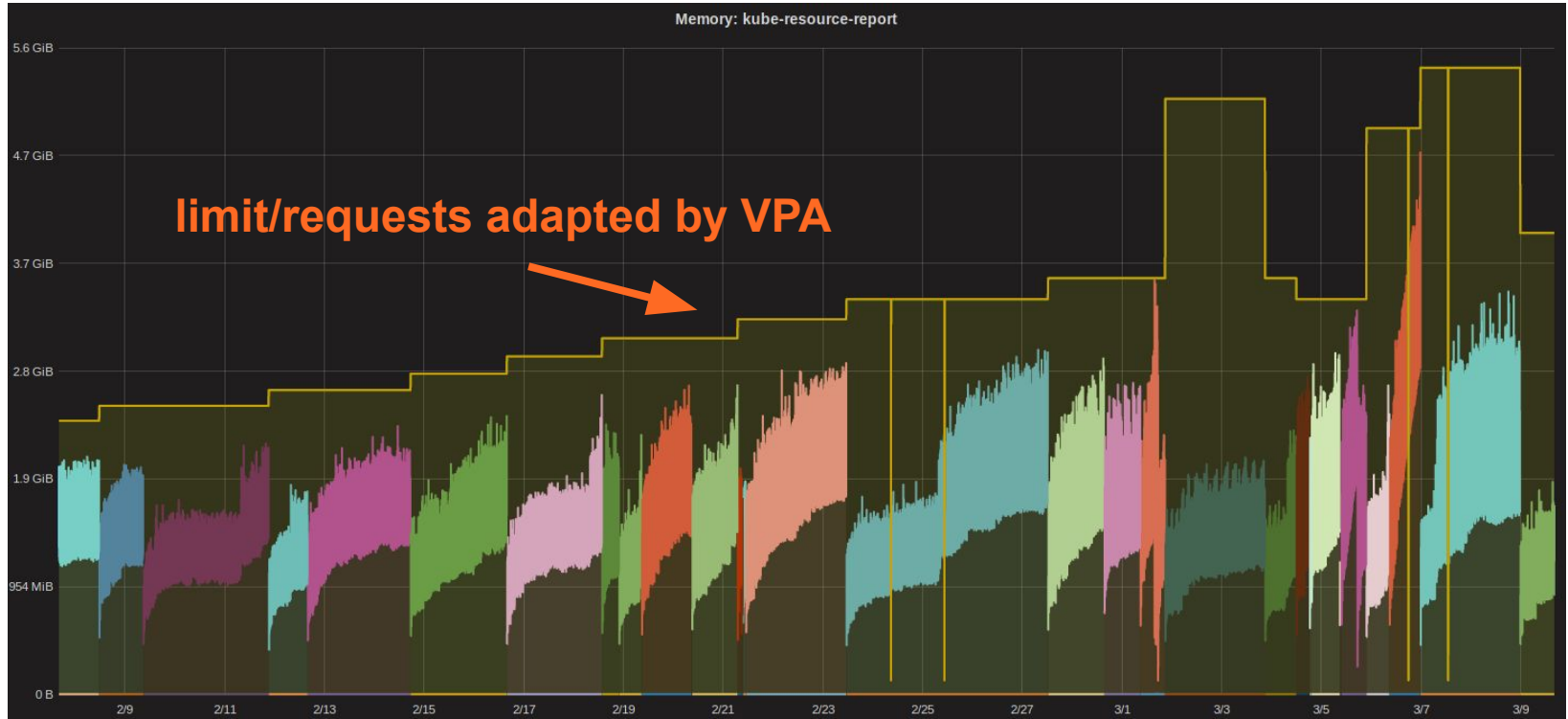
KUBERNETES APPLICATION DASHBOARD



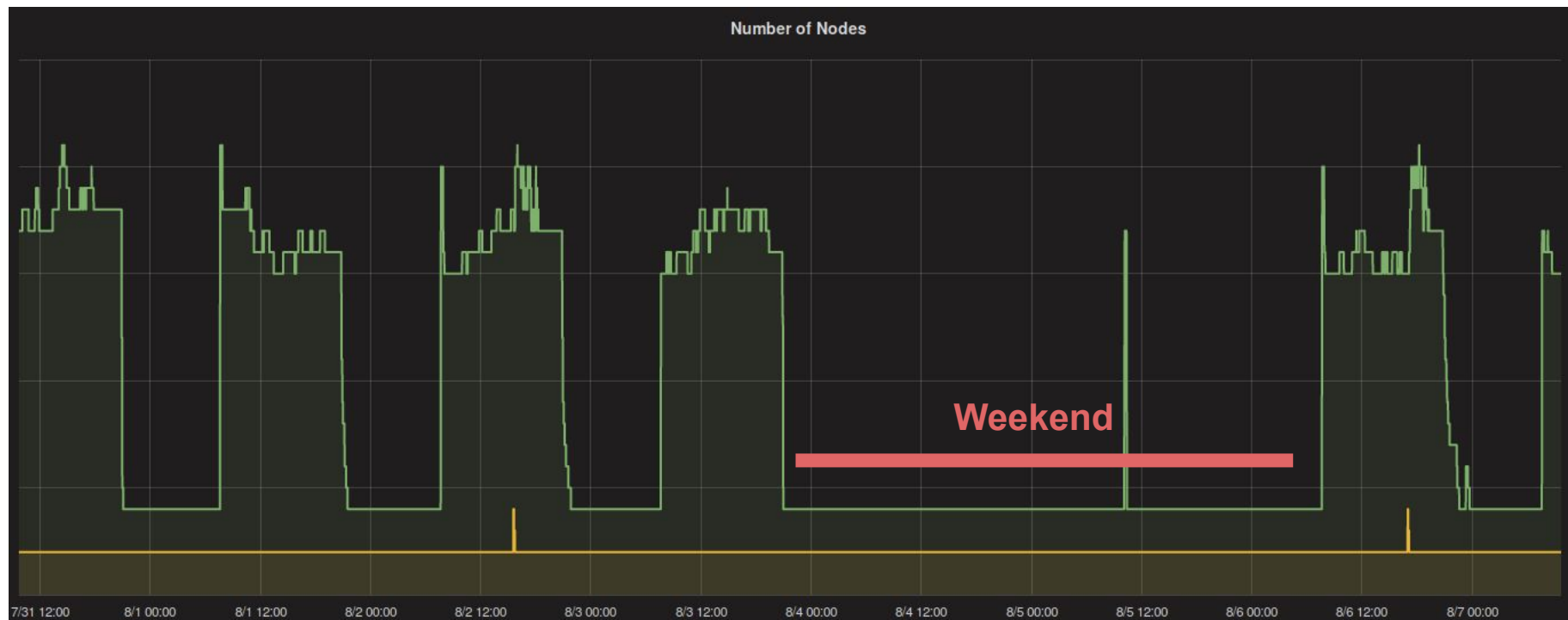


<https://github.com/hjacobs/kube-ops-view>

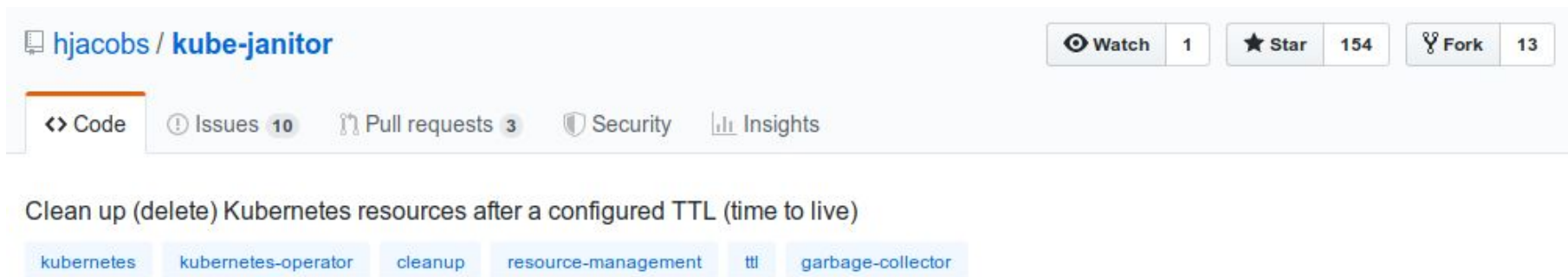
VERTICAL POD AUTOSCALER



DOWNSCALING DURING OFF-HOURS



KUBERNETES JANITOR



- **TTL** and **expiry date** annotations, e.g.
 - set time-to-live for your test deployment
- **Custom rules**, e.g.
 - delete everything without "app" label after 7 days

EC2 SPOT NODES

Role	Instance Type	S?	Version	CC	MC	CPU	Memory (GiB)	Cost
worker	m4.2xlarge		v1.12.5- custom.master-1	8	31.4 GiB	<div><div>1.6</div><div>6.9</div><div>7.8</div></div>	<div><div>8.8</div><div>14.3</div><div>30.9</div></div>	350.64
worker	m4.4xlarge	🟢	v1.12.5- custom.master-1	16	62.9 GiB	<div><div>4.1</div><div>9.0</div><div>15.8</div></div>	<div><div>51.2</div><div>62.3</div><div>62.4</div></div>	193.73

72% savings

STABILITY ↔ EFFICIENCY

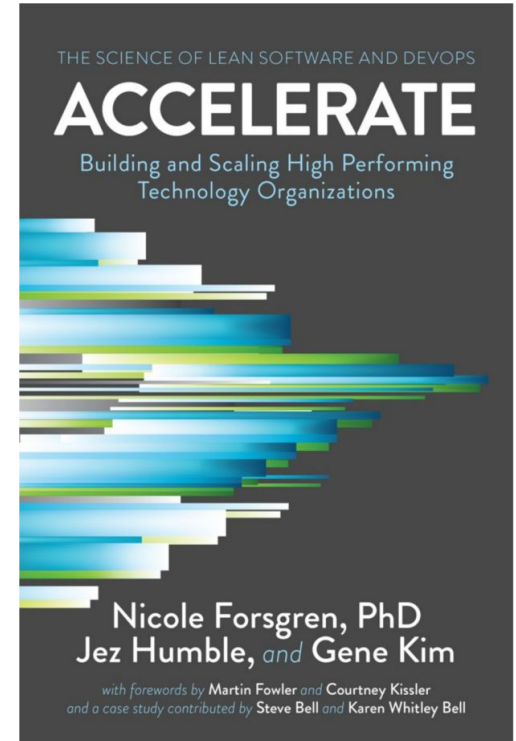
Slack
Autoscaling
Buffer
Disable
Overcommit
Cluster
Overhead

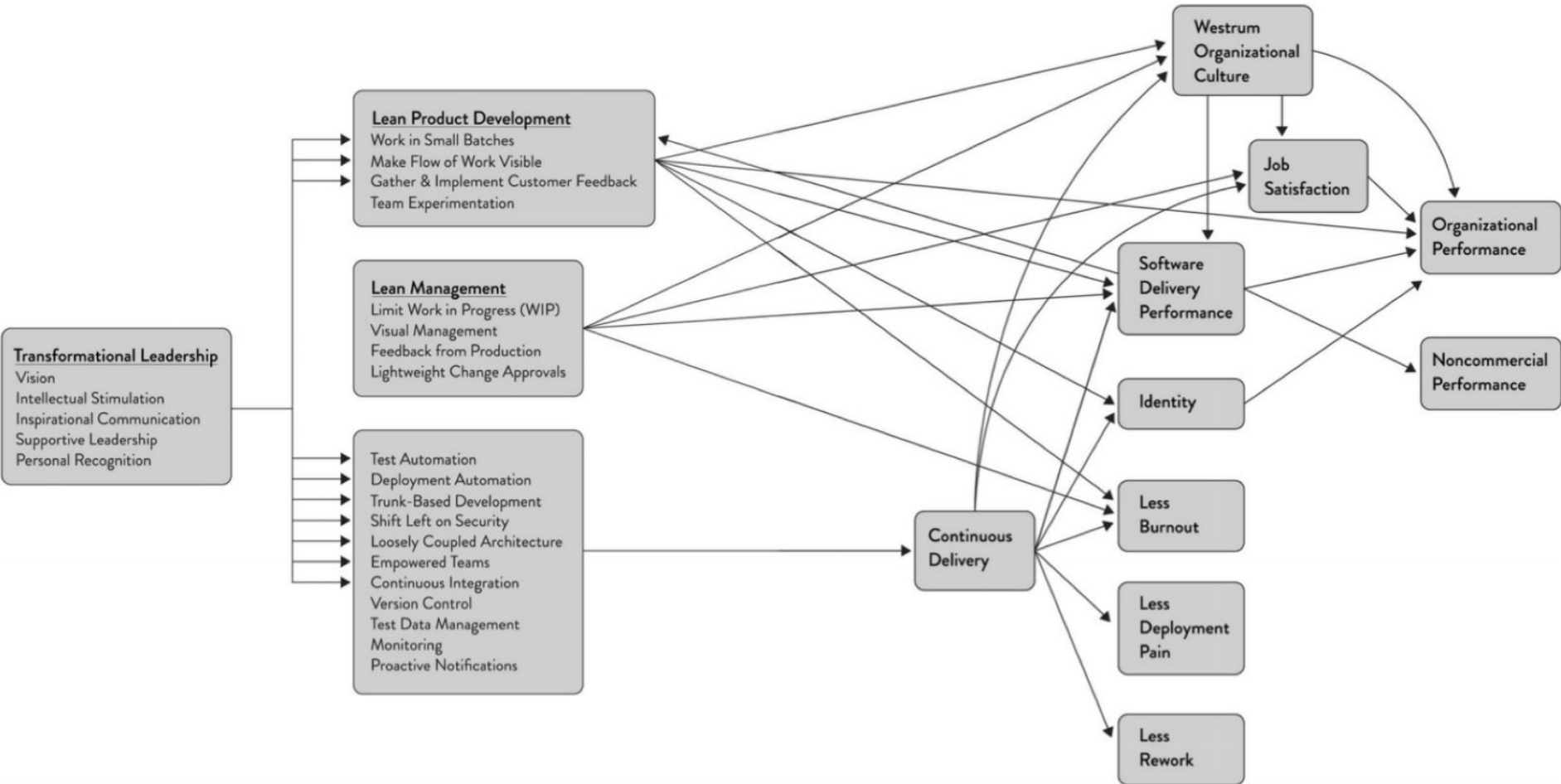


Resource
Report
HPA
VPA
Downscaler
Janitor
EC2 Spot

DELIVERY PERFORMANCE METRICS

- Lead Time
- Release Frequency
- Time to Restore Service
- Change Fail Rate





From "Accelerate: The Science of Lean Software and DevOps"

DELIVERY PERFORMANCE METRICS

- Lead Time $\hat{=}$ **Commit to Prod**
- Release Frequency $\hat{=}$ **Deploys/week/dev**
- Time to Restore Service $\hat{=}$ **MTRS from incidents**
- Change Fail Rate $\hat{=}$ **n/a**

Applying product management to internal platforms

*“.. means **establishing empathy with internal consumers (read: developers)** and collaborating with them on the design. Platform product managers establish roadmaps and ensure the platform delivers value to the business and enhances the developer experience.”*

- [ThoughtWorks Technology Radar](#)

Do you use any software development tools as part of your daily work? *

☐ Yes

☐ No

Overall, how satisfied or dissatisfied are you with your Zalando developer experience? *

	1	2	3	4	5	6	7	8	9	10	
Extremely Dissatisfied	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely Satisfied

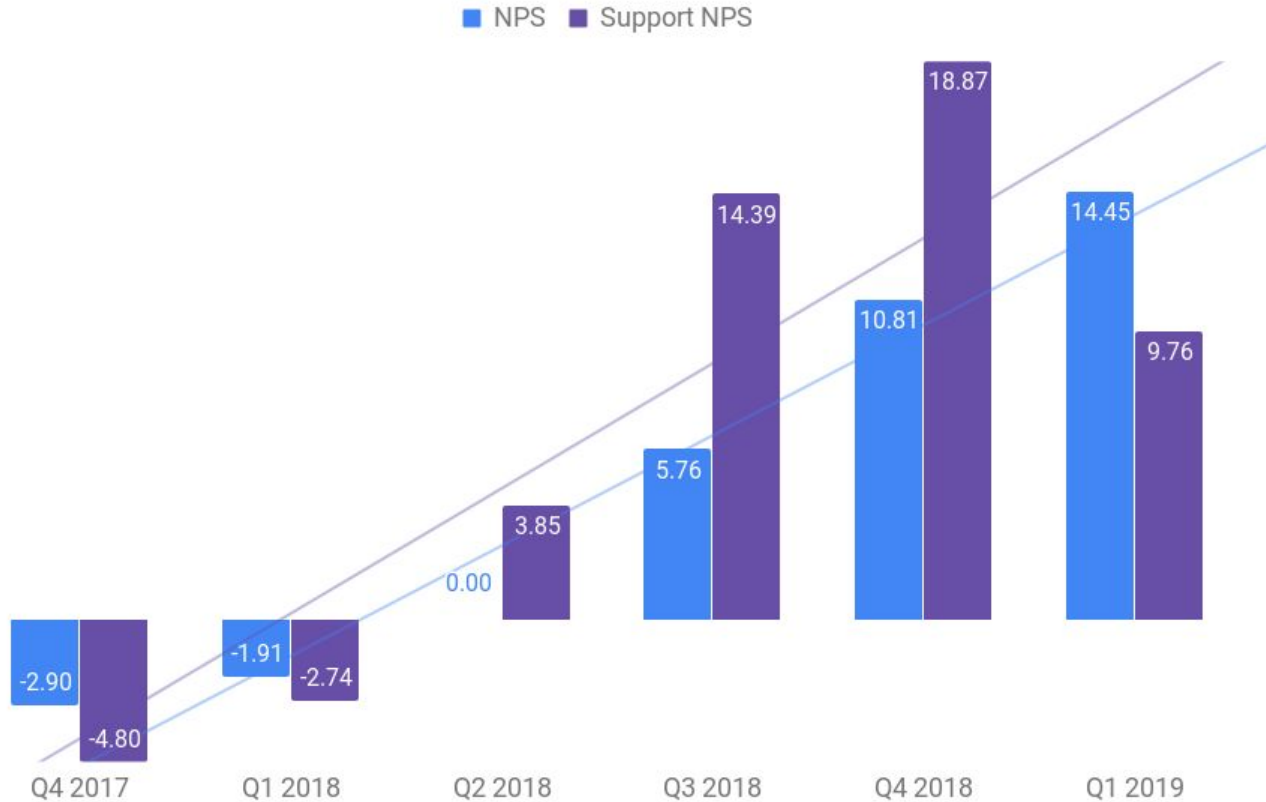
If you needed support for the developer tools you use, how satisfied or dissatisfied are you with the support you received? *

	0	1	2	3	4	5	6	7	8	9	10	
Not Applicable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely Satisfied

What could we do from your perspective to make the developer experience at Zalando even better?

Your answer

DEVELOPER SATISFACTION



DOCUMENTATION

"Documentation is hard to find"

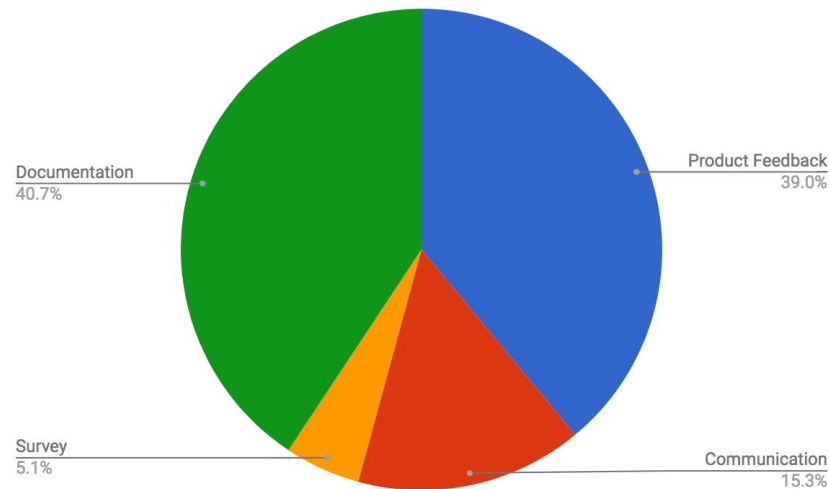
"Documentation is not comprehensive enough"

"Remove unnecessary complexity and obstacles."

"Get the documentation up to date and prepare use cases"

"More and more clear documentation"

"More detailed docs, example repos with more complicated deployments."



DOCUMENTATION



- Restructure following <https://www.divio.com/en/blog/documentation/>
 - Concepts
 - How Tos
 - Tutorials
 - Reference
- Global Search
- Weekly Health Check: Support → Documentation

Documentation (how much did we improve? emphasis on support)



Yellow



Red



Yellow



Yellow

APPLICATIONS

Pipelines

TOOLS

Repositories

Clusters

Kube Resource Report

Search Results

Search finished, found 123 item(s) matching the search query

Databases - Cloud Native Application Runtime

/reference/using-databases/

Using Databases from the Cluster It is common for an application to depend on a PostgreSQL database

The first option is to deploy a PostgreSQL cluster into your account using the PostgreSQL operator.

to deploy a PostgreSQL cluster to your K8S cluster.

Just deploy a PostgreSQL manifest to your account (via zkubectl or CDP). kind: " postgresql " apiVersion

\zalando Delete a database cluster: zkubectl get postgresql zkubectl delete postgresql

Rotate PostgreSQL Credentials — Discount Service documentation

/operations/guides/rotate_postgresql_keys.html

Rotate PostgreSQL Credentials For security reasons, all PostgreSQL credentials must be rotated every

Postgresql · GitBook

/postgresql-24x7.html

PostgresShort descriptionPostgres is used in Team

Migration from Stups - Cloud Native Application Runtime

/reference/migration/

The application has the following components: RDS PostgreSQL database Docker container ELB load balancer

Kubernetes Best Practices - Cloud Native Application Runtime

/concepts/kubernetes-best-practices/

this is just a redeployment with a different target namespace, but for stateful applications, like a postgresql



PUBLISH

DISCOVERABLE

DOCUMENTATION

Learn more



WHY KUBERNETES?

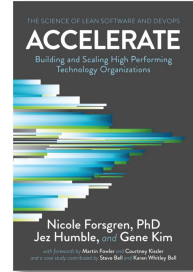
WHY KUBERNETES?

- provides enough **abstractions** (StatefulSet, CronJob, ..)
- provides **consistency** (API spec/status)
- is **extensible** (annotations, CRDs, API aggreg.)
- certain **compatibility** guarantee (versioning)
- widely **adopted** (all cloud providers)
- works across environments and implementations

WHY KUBERNETES?

(for Zalando)

- Efficiency
- Common Operational Model
- Developer Experience
- Cloud Provider Independent
- Compliance and Security
- Talent



KUBERNETES FAILURE STORIES



- Learning about production pitfalls!
- Availability bias?



zerkms commented 14 days ago



@hjacobs while you're here I wanted to thank you and tell that this is the most important repository to follow for those who run their kubernetes clusters :-)



1

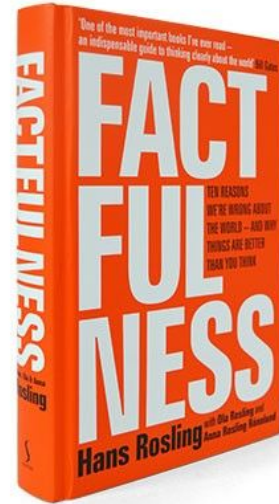


1

FACTFULNESS

Things can be both better and bad!

How would failure stories for
your non-K8s infra look like?



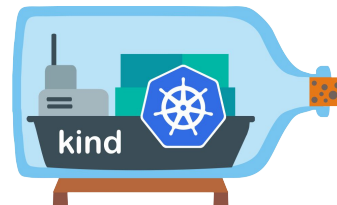
COMPLEXITY FOR GOOGLE-SCALE INFRA?

- Managed DO cluster: 4 minutes
- K3s single node: 2 minutes



install.sh 838B

```
1 #!/bin/bash
2
3 # Install k3s
4 curl -sfL https://get.k3s.io | sh -
```



DE-FACTO STANDARD, EXTENSIBLE API



Kelsey Hightower 

@kelseyhightower

Kubernetes is not the kernel; it's systemd.

4:57 nachm. · 25. Jan. 2019 · [Twitter for iPhone](#)

201 Retweets **969** „Gefällt mir“-Angaben



Corey Quinn

@QuinnyPig

Follow



Nuclear hot take: nobody will care about Kubernetes in five years.

CZnative @ home @pczarkowski

Replying to @tmclaughbos @iteration1 @behemphi

As I keep telling people, if you have a kubernetes strategy you've already failed. Kubernetes should be an implementation detail at the tactical level to deal with the strategic imperative of solving the problems that are halting the flow of money.

6:32 PM - 6 Feb 2019

97 Retweets 439 Likes



41



97



439



MAYBE THAT'S GOOD?



Kuberkus 1.16.0 is a Wrap

@fuzzychef



Antwort an [@QuinnyPig](#)

Speaking as a kubernetes dev, that's a victory condition. It means that Kube becomes so ubiquitous, and so easy, that's it's ignorable.

[Tweet übersetzen](#)

7:06 nachm. · 7. Feb. 2019 · [Twitter Web Client](#)

1 Retweet 18 „Gefällt mir“-Angaben

OPEN SOURCE & MORE

Kubernetes on AWS

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

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

More Zalando Tech Talks

github.com/zalando/public-presentations





QUESTIONS?



HENNING JACOBS
SENIOR PRINCIPAL



henning@zalando.de

[@try_except](#)

Illustrations by [@01k](#)

