



CONTINUOUS DELIVERY AT ZALANDO

HU BERLIN GUEST LECTURE

2020-01-29

HENNING JACOBS
@try_except_



ABOUT ME

Henning Jacobs
Senior Principal



- Head of Developer Productivity
- Zalando since 2010
- University Karlsruhe (TH) - now KIT



Blog: srcco.de

Twitter: [@try_except](https://twitter.com/@try_except)



Zalando

ZALANDO AT A GLANCE

~ **5.4** billion EUR

revenue 2018

~ **14,000**

employees in
Europe

> **80%**

of visits via
mobile devices

> **300**
million

visits
per
month

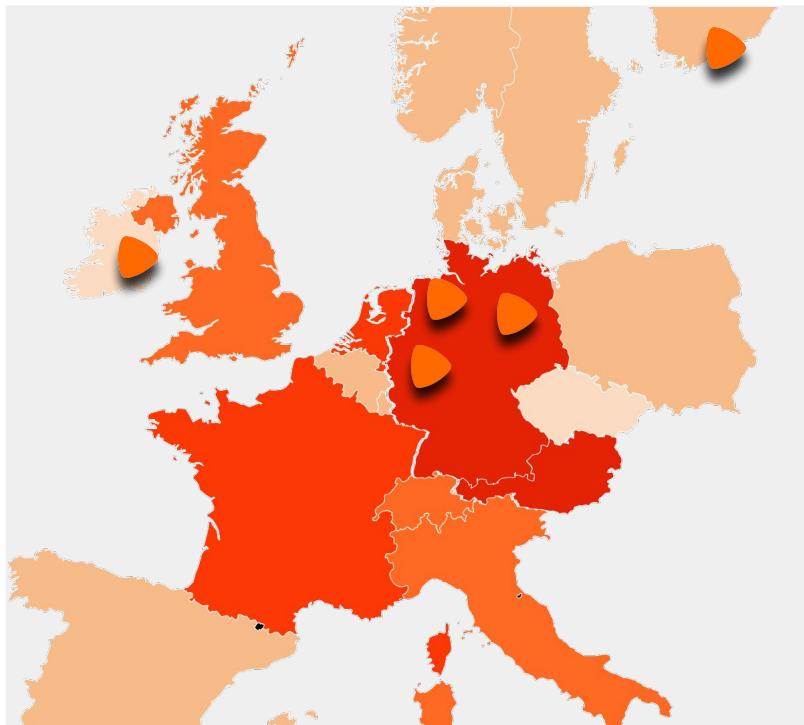
> **28**
million
active customers

> **400,000**
product choices

> **2,000**
brands

17
countries

SOFTWARE PRODUCT DEVELOPMENT AT ZALANDO



**5 tech
hubs**

**> 2,000
employees**

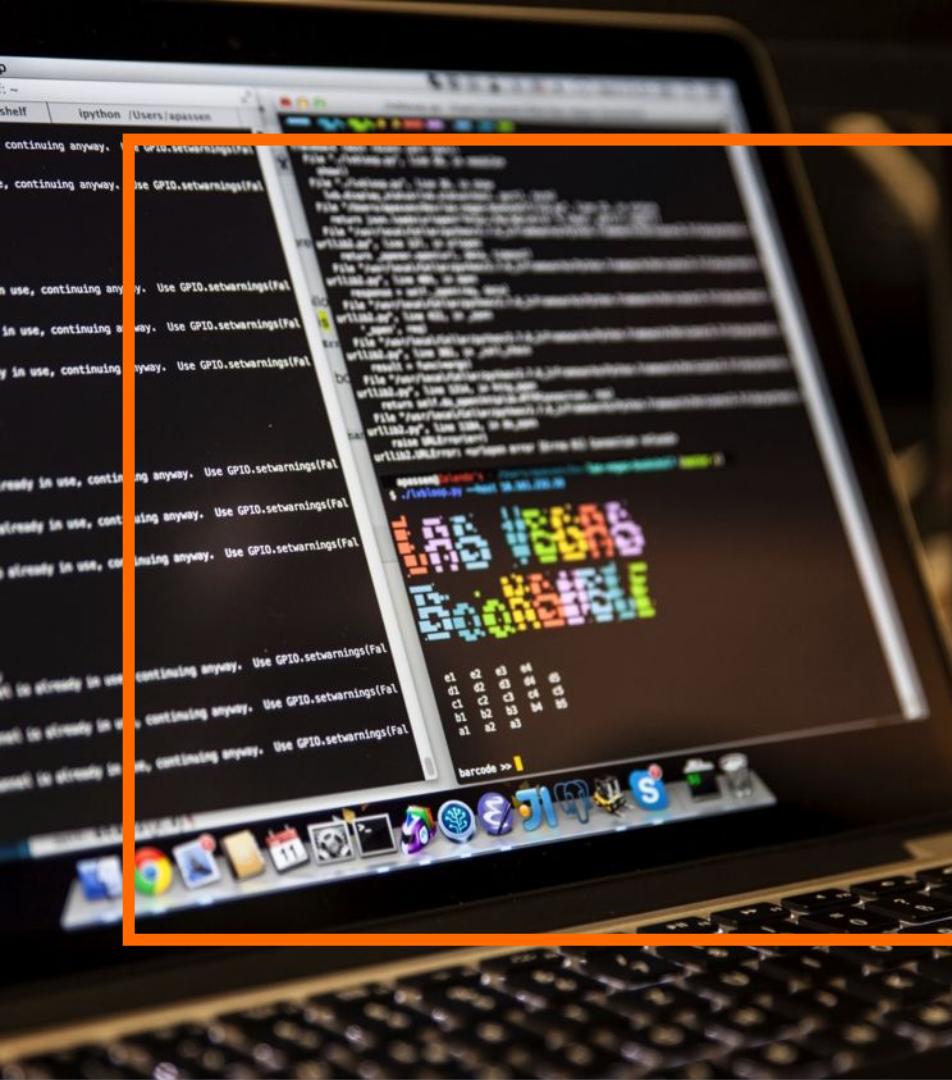
In Tech

From **> 100**
Nations

Across
Europe

> 200

Cross-functional,
Agile Delivery
Teams

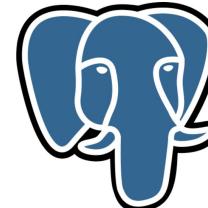


HOME-BREWED SOFTWARE

- >1100 developers
- >200 development teams
- >2000 applications

OSS WE BUILD ON

- Java (OpenJDK)
- Apache Tomcat
- PostgreSQL
- Python
- JS, Scala, Go, ..
- Kubernetes



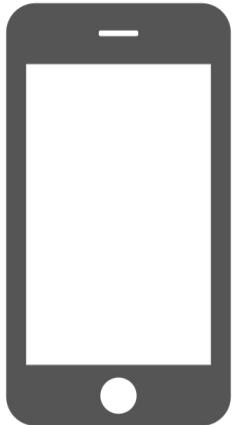
YOU BUILD IT, YOU RUN IT

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](#)

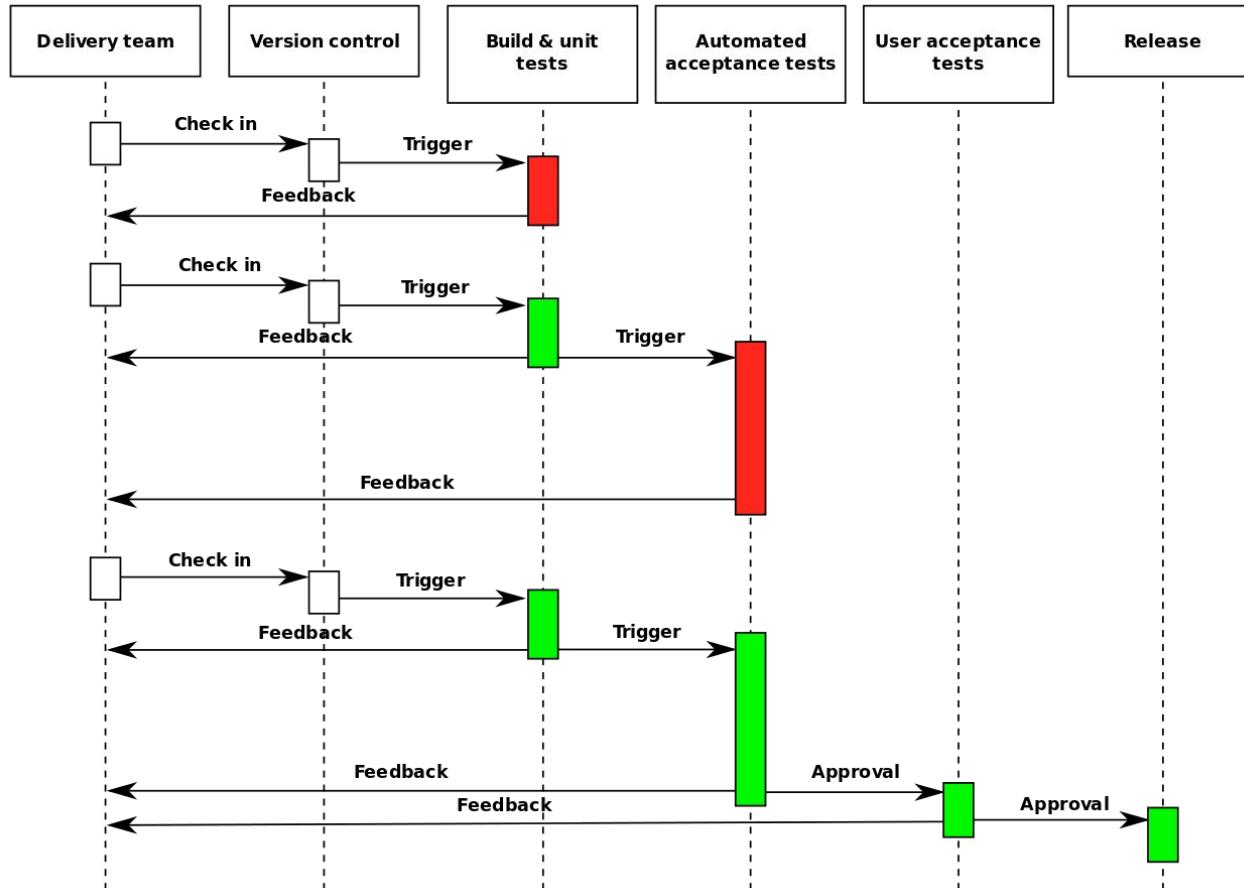
CI/CD

“Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. ”

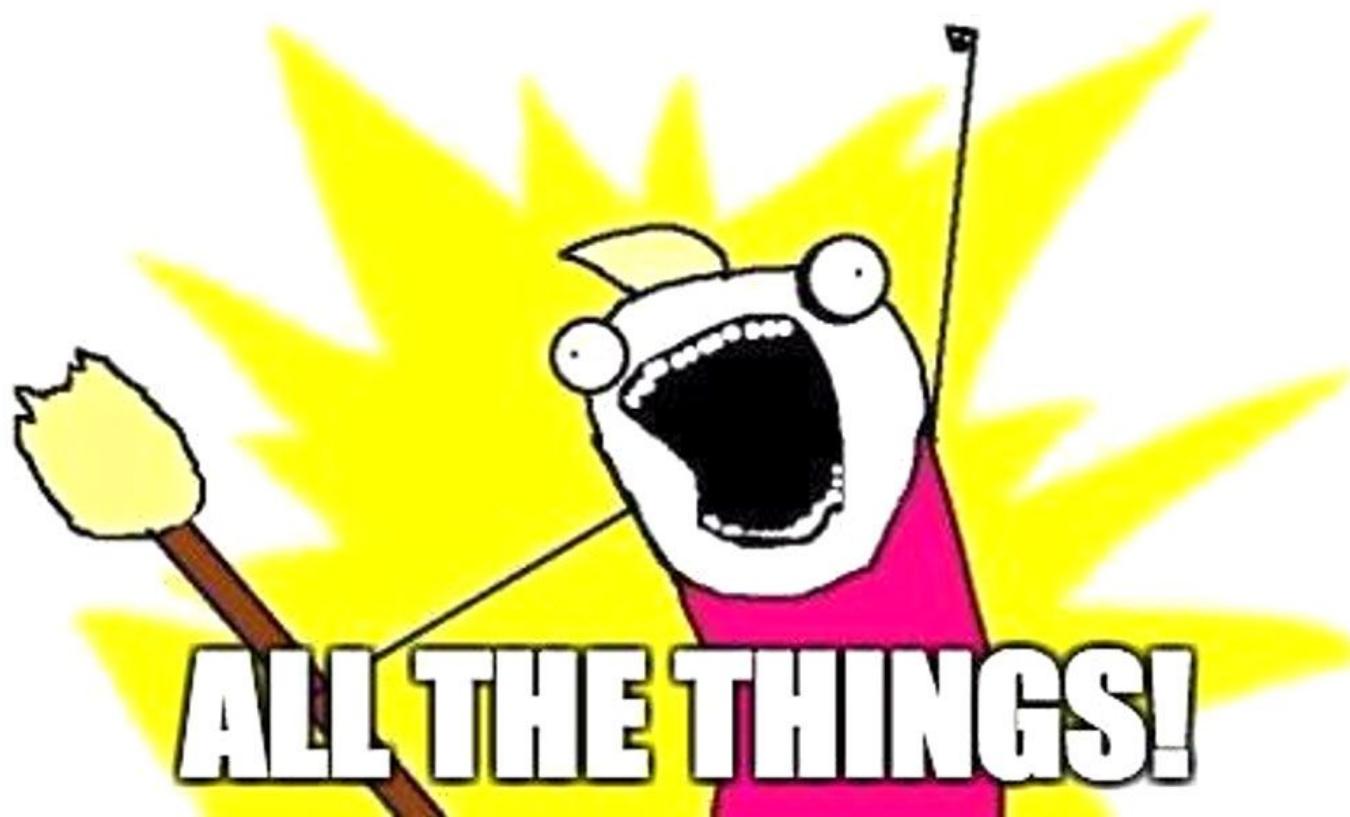
“Continuous Delivery is the ability to get changes of all types—including new features, configuration changes, bug fixes and experiments—into production, or into the hands of users, **safely and quickly in a sustainable way.**”

- Martin Fowler

CONTINUOUS DELIVERY



AUTOMATE

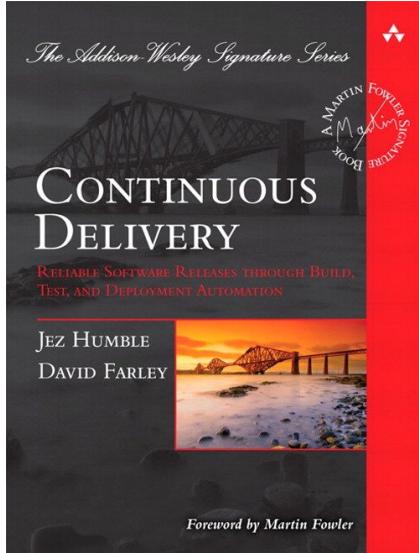


WHY CONTINUOUS DELIVERY?

- **Lower risks:** smaller batches, automation
- Accelerate **Time to Market**
- Build the **Right Product**
- Increase **Productivity and Efficiency**
- Improve **Customer Satisfaction**



CONTINUOUS DELIVERY



If it hurts, do it more frequently, and bring the pain forward.

- Jez Humble

2010

Zalando (Magento-)Shop-Deploy Checkliste

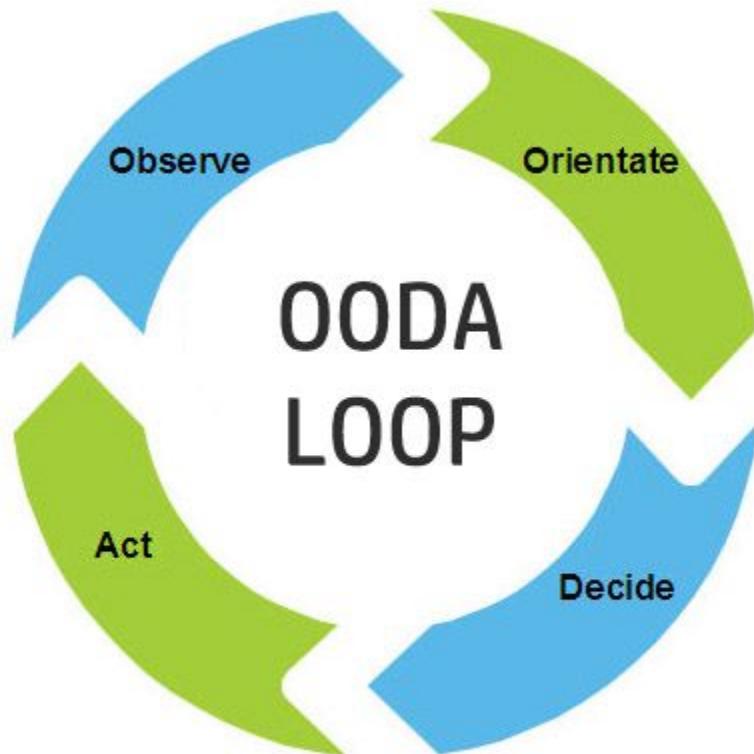
Tag: _____ SysOP: _____ Datum: _____

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

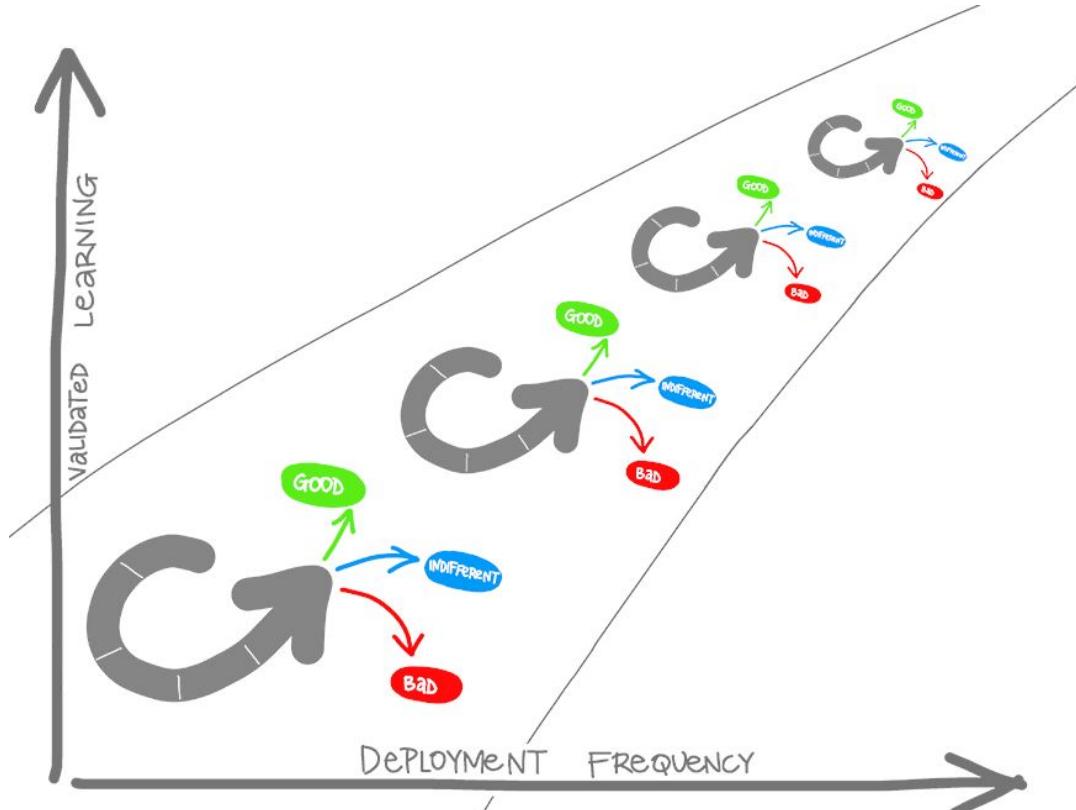
Bei Probleme die **nicht innerhalb von 5 Minuten** behoben werden können folgende Personen Anrufen und informieren!



UNDERSTANDING CYCLE TIME



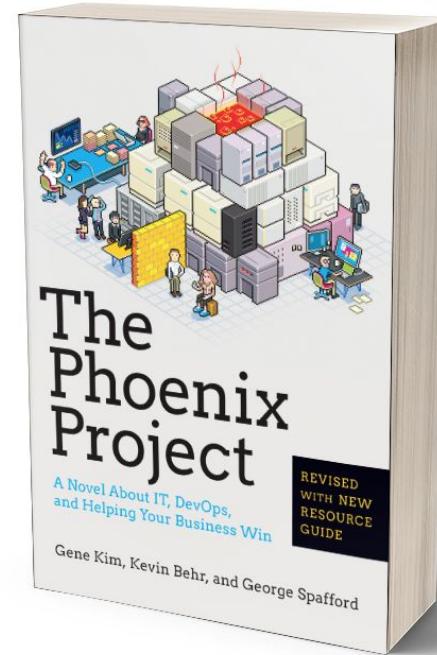
SHORTEN CYCLE TIME



THE PHOENIX PROJECT - 2013

"The Three Ways"

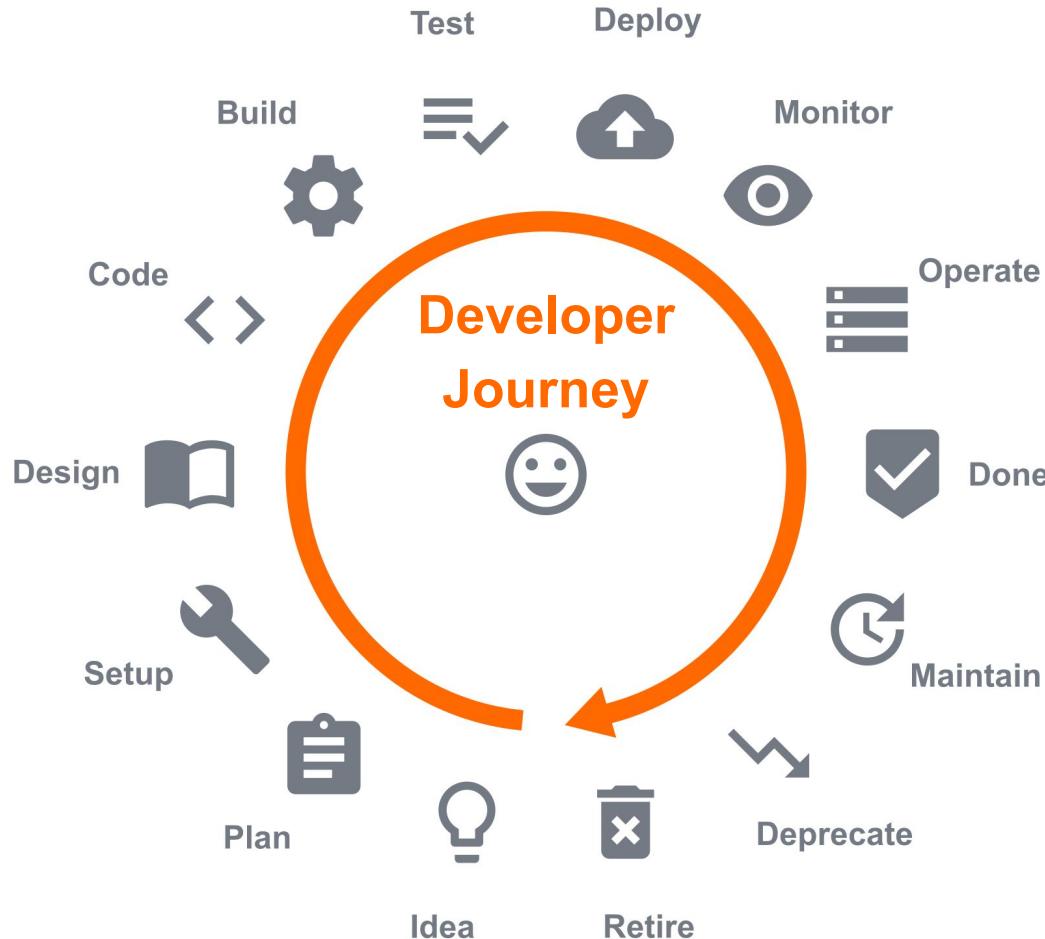
1. Systems Thinking
2. Amplify Feedback Loops
3. Culture of Continual Experimentation And Learning





DEVELOPER JOURNEY

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



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



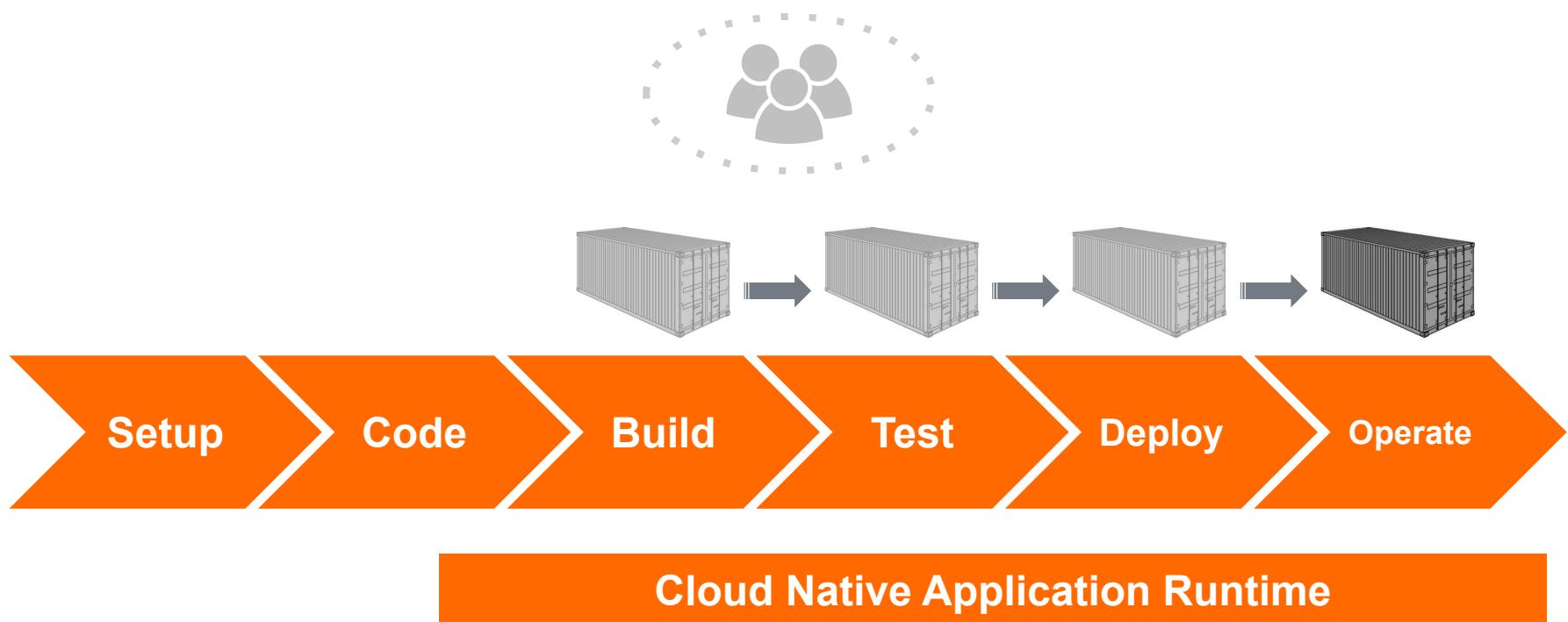
CLOUD NATIVE

*.. 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.*

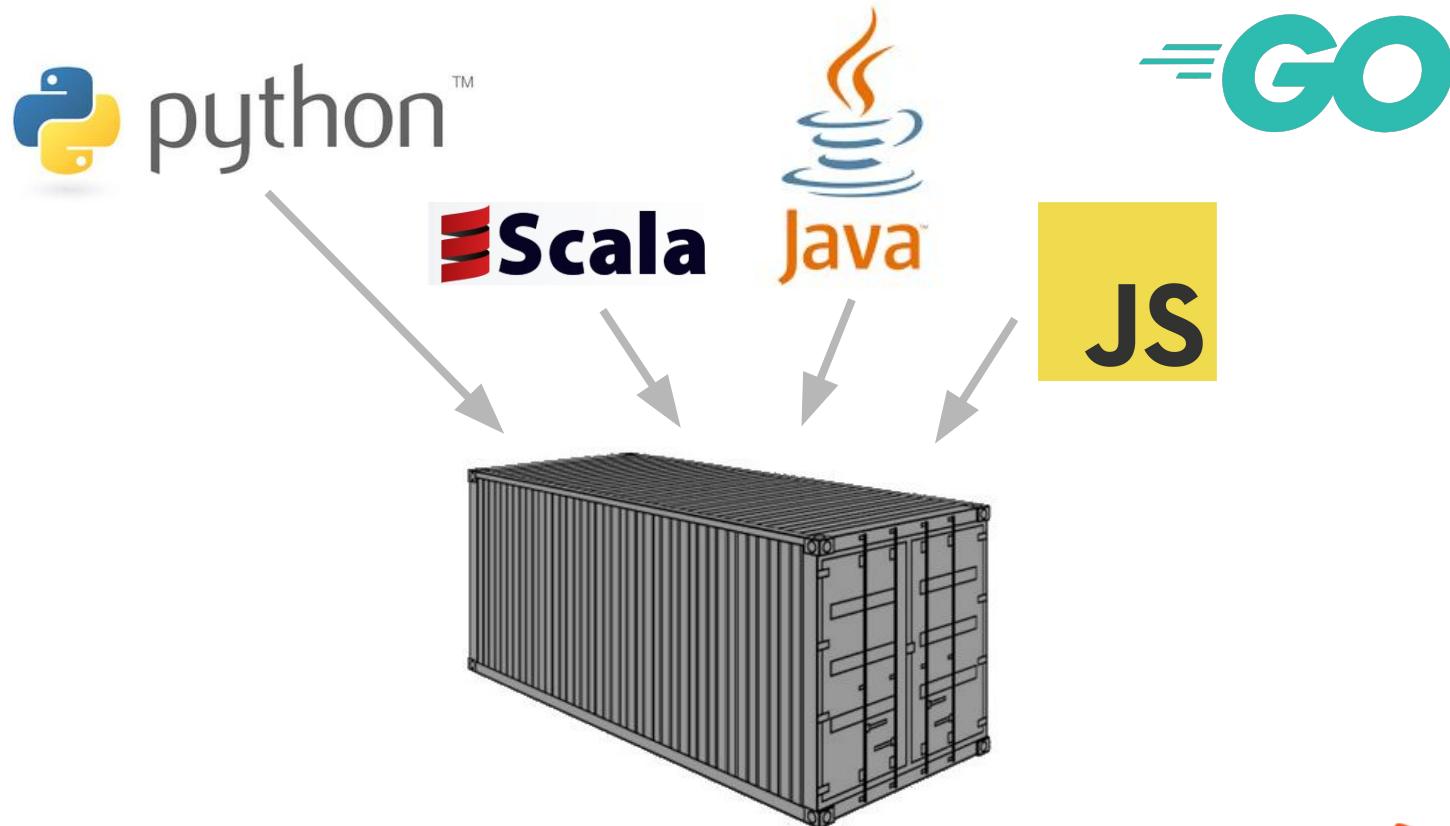
*Cloud native technologies enable software developers to **build great products faster**.*

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

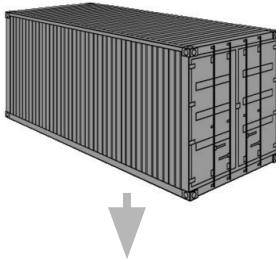
CONTAINERS END-TO-END



CONTAINERS



CONTAINERS



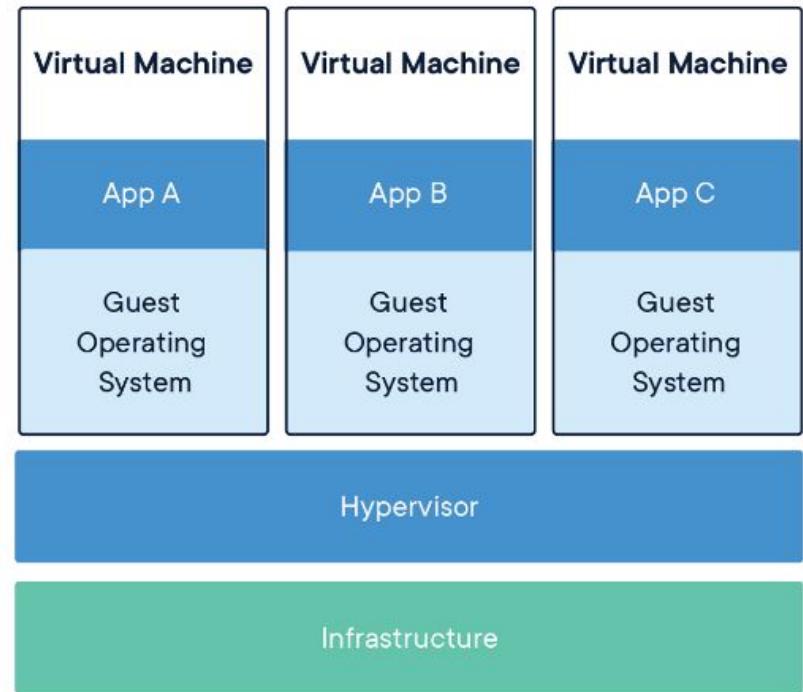
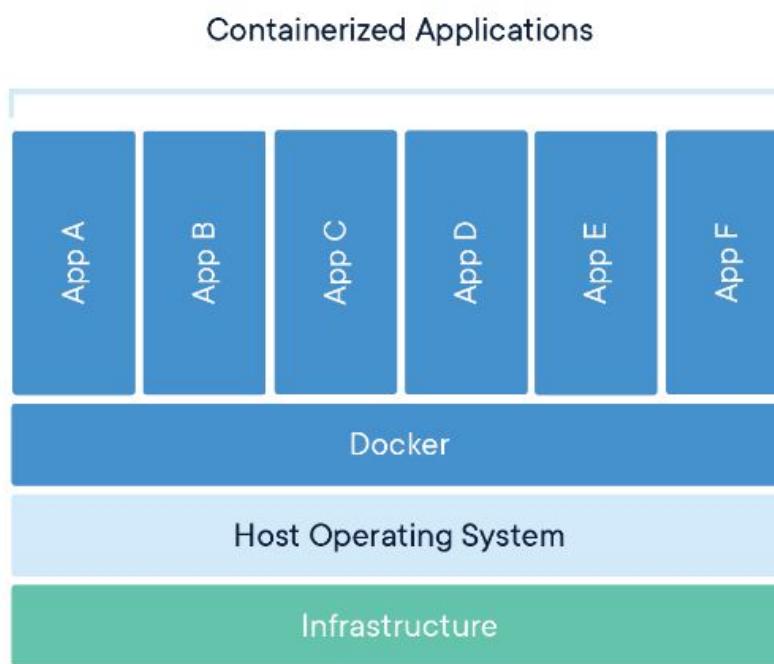
kubernetes



Google Cloud Platform

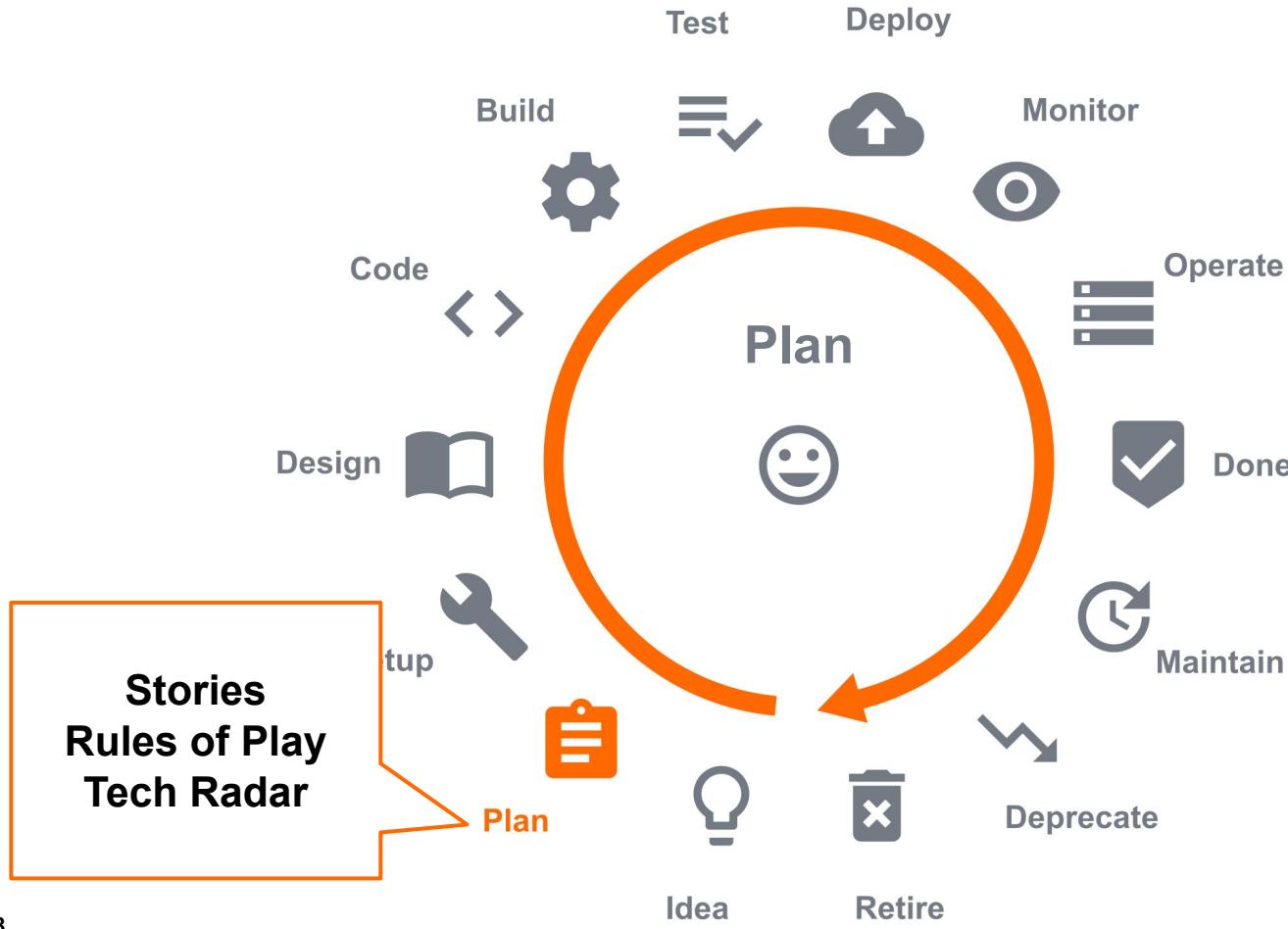


CONTAINERS VS VIRTUAL MACHINES





PLAN & SETUP



Zalando Tech Radar — 2019.09

Frameworks

ADOPT	ASSESS
1. Akka (Scala)	16. Aurelia
2. Node.js	17. Ember.js
3. Play (Scala)	18. gRPC
4. ReactJS	19. Http4s
5. RxJava (Android)	20. JOOQ
6. scikit-learn	21. Redux
7. Spring	22. Vert.x
	23. Vue.js

TRIAL

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

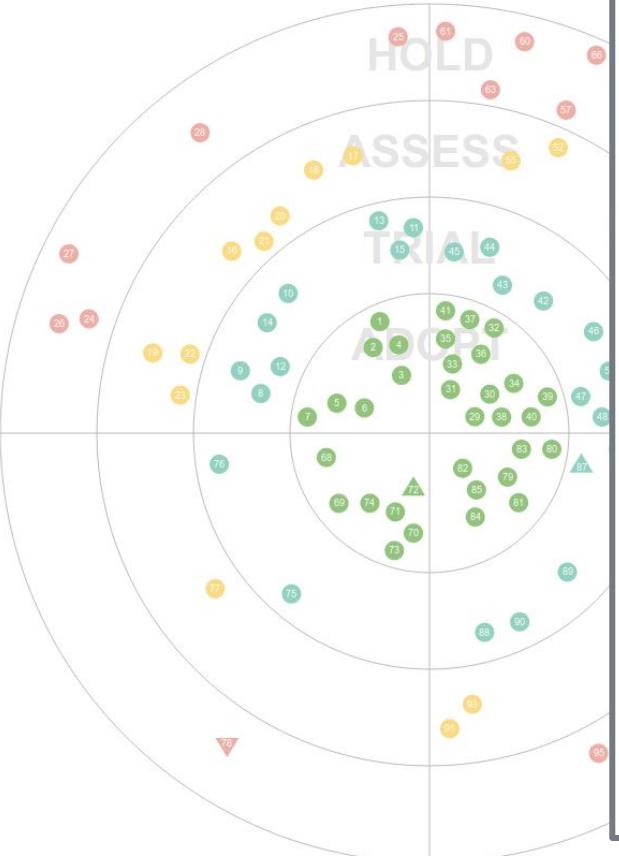
Infrastructure

ADOPT	ASSESS
68. Docker	77. AWS Lambda

HOLD
78. STUPS

TRIAL

75. Failsafe
76. Undertow



ZALANDO TECH RADAR

Kotlin

ASSESS

edit this page

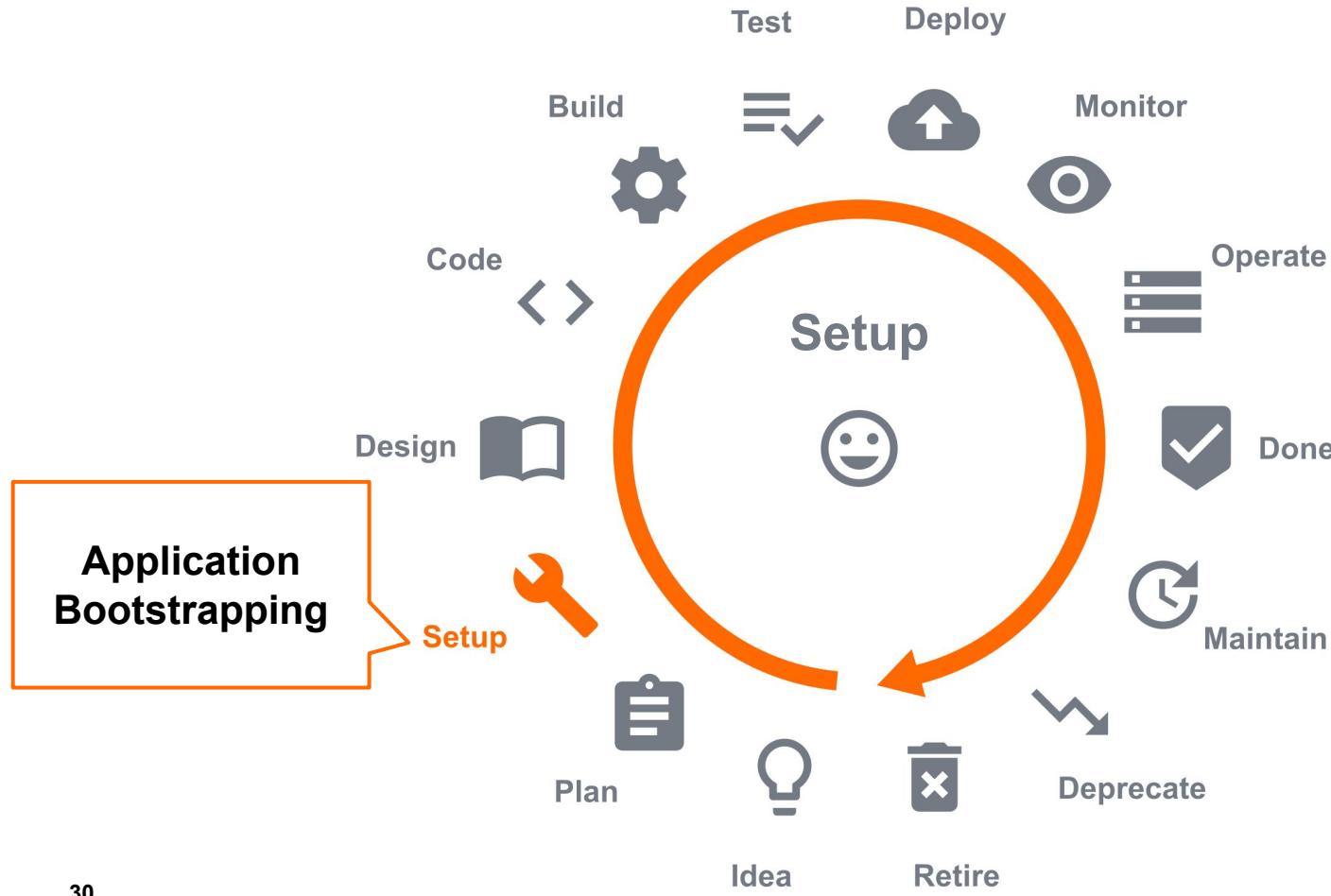
Kotlin is a pragmatic, multi-paradigm, multi-platform, modern, safe, interoperable and toolable programming language by JetBrains. Its primary focus is and has been the JVM ecosystem.

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



Create a new repository

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

Organisation * Repository Name *

Description

Team ID *

Repository type *

Bootstrap with application template

Add .gitignore template

* Mandatory fields

Branch: master ▾

New pull request

Create new file

Upload files

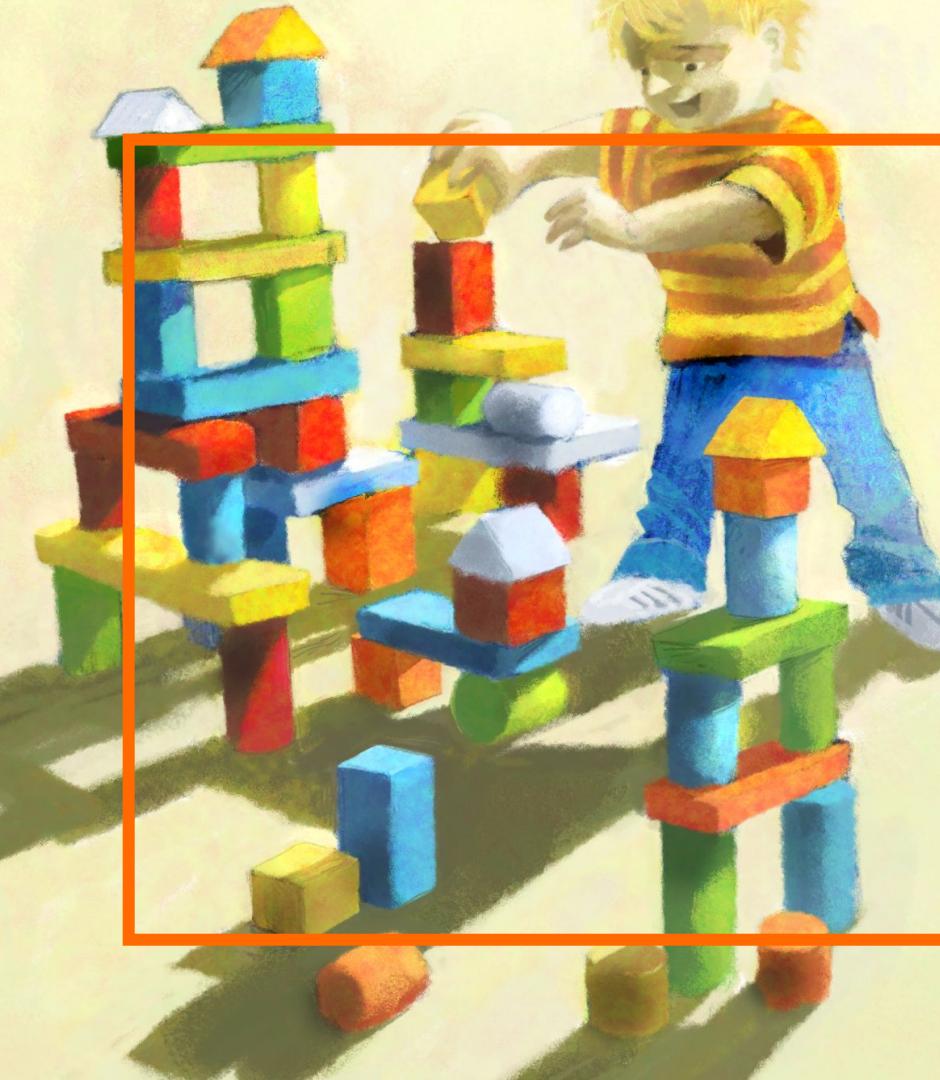
Find file

Clone or download ▾

 [REDACTED]	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
<hr/>  README.md		

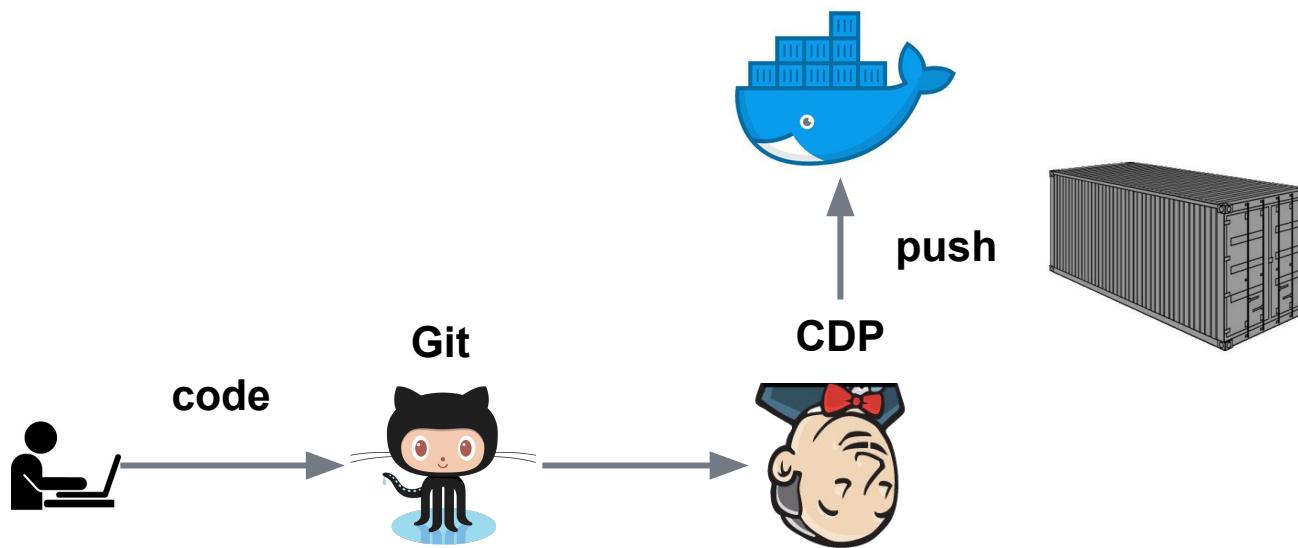
Java - Spring Boot - REST

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

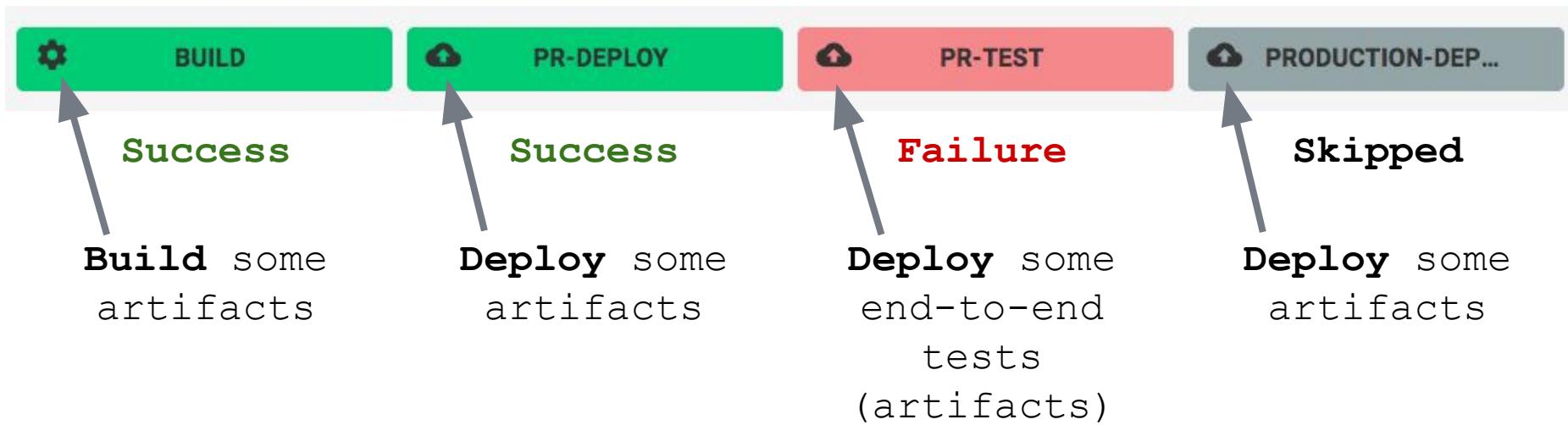
CONTINUOUS DELIVERY PLATFORM: BUILD



A typical pipeline:



A typical pipeline:



Add more commits by pushing to the  branch on 



Some checks haven't completed yet

1 pending and 5 successful checks

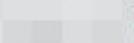
[Hide all checks](#)

-  **zapr** — 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](#).

PULL REQUESTS



- PRs should be open for at least 3 days to review
- PRs should be quick to understand and review
- PRs should ideally have <300 lines of changes

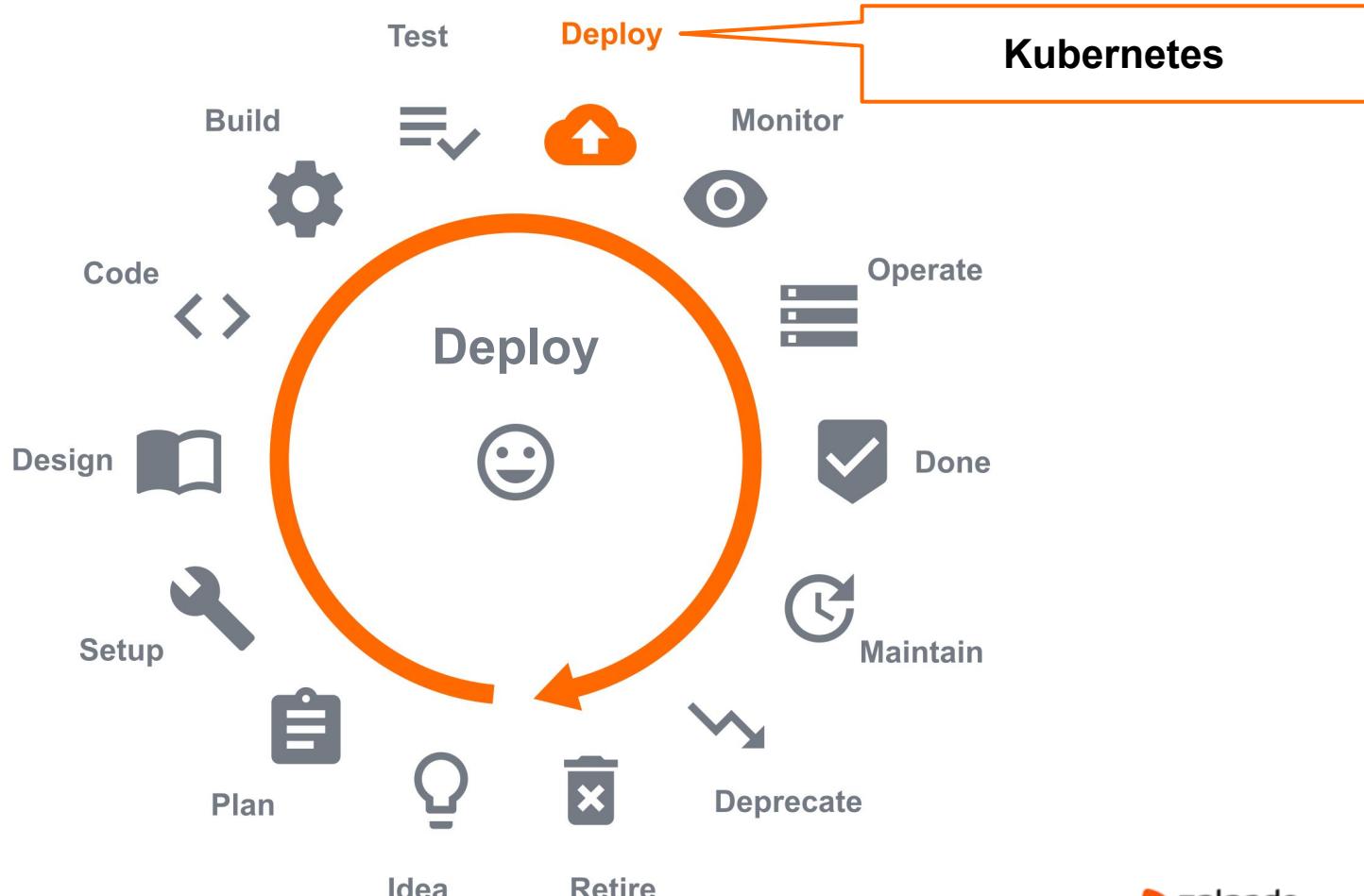
PULL REQUESTS

- ❑ PRs should be open for at least 3 days to review
- ❑ PRs should be quick to understand and review
- ❑ PRs should ideally have <300 lines of changes





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
```



CDP: DEPLOY

DEPLOYMENT UNITS TRACKING-DEPLOY MASTER-91

TRACKING-DEPLOY TEST STAGING PROD

Ran for 1 m, 16 s

GENERAL DEPLOYMENT STATUS LOGS

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 [REDACTED] open

PlatformCredentialsSet [REDACTED]

postgresql [REDACTED]-db

Secret [REDACTED]-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

The screenshot shows the Zalando Developer Console interface. At the top, there's a navigation bar with 'Developer Console' on the left, a search bar in the center, and 'Feedback' and 'Support' links on the right. Below the navigation bar, there are tabs for 'DEPLOYMENT UNITS', 'FEED-SERVICE', and 'MASTER-16'. A large orange circular progress indicator is visible on the right side of the header.

The main content area has several tabs at the top: 'FEED-SERVICE' (selected), 'BUILD' (green), 'DEPLOY-TEST' (green), and 'DEPLOY-PROD' (yellow). Below these tabs, there are two sub-tabs: 'GENERAL' and 'LOGS'.

The 'GENERAL' tab contains sections for 'ACTIONS' and 'ERRORS & WARNINGS'. The 'ACTIONS' section features two buttons: 'APPROVE' (with a green thumbs-up icon) and 'REJECT' (with a red thumbs-down icon). The 'ERRORS & WARNINGS' section shows 'Error' and 'Warning' status indicators, both of which say 'No Error' and 'No Warnings' respectively.

The 'DETAILS' section on the right lists the following information:

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

CONTINUOUS DELIVERY PLATFORM

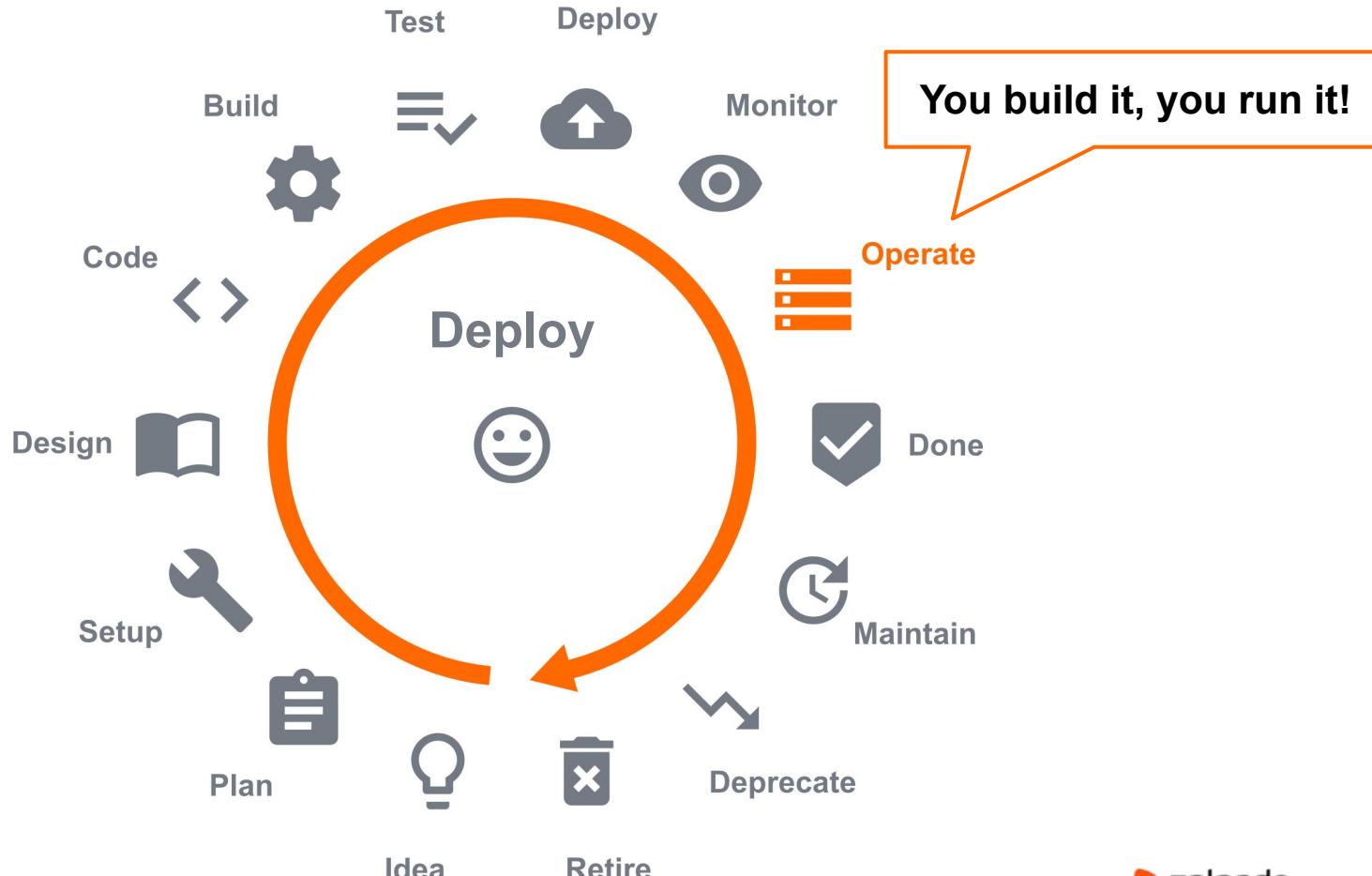
DEPLOYMENT UNITS	RENDERING-ENGINE	Exclude PRs: <input type="checkbox"/>					
Pipeline	Started	Action	Pipeline Runs				
pr-1785-2	3h ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
pr-1785-1	3h ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
Remove `any` usage from our code pr-1784-1	6h ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
master-1188 master-1188	1d ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
Do not bundle node_modules in re package pr-1783-1	1d ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
master-1187 master-1187	1d ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
Document the renderer contribution workflow pr-1775-2	2d ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
master-1186 master-1186	2d ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
pr-1778-1	2d ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS
Document the renderer contribution workflow pr-1775-1	6d ago	i	 BUILD	 PERF	 DEPLOY-BRANCH	 TEST-BRANCH	 DEPLOY-DOCS



CLOUD FORMATION VIA CI/CD

"Infrastructure as Code"

```
└── deploy/apply
    ├── deployment.yaml          # Kubernetes
    ├── cf-iam-role.yaml        # AWS IAM Role
    ├── cf-rds.yaml             # AWS RDS Database
    ├── kube-ingress.yaml
    ├── kube-secret.yaml
    └── kube-service.yaml
└── delivery.yaml            # CI/CD config
```



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-rhh9m	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,deployments,...

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.co **/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.co **/etcd-operator:v0.9.3**
application: deployment-id: d-e8yt17ub9hxt513sr27w66ea environment: staging pipeline-id: l-7bic5kvki6khadtadtqzq5hy3q version: master-7

etcd-operator (Deployment)
/cluster: namespaces/default/deployments/etcd-operator
Created: 2018-10-19 14:13:50 source.zalan.co **/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.co **/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.co **/etcd-operator:v0.9.3**
application: deployment-id: d-so5ukevu2piyw5bdigzxc4gx3 environment: staging version: master-26

CLUSTER RESOURCES

/ all / pods

Pods



Namespaces

Nodes

PersistentVolumes

CONTROLLERS

StackSets

Stacks

Deployments

CronJobs

Jobs

StatefulSets

POD MANAGEMENT

Ingresses

Services

Pods

ConfigMaps

CRDS

PlatformCredentialsSets

postgresqls

META

Resource Types

Events

Pods



Namespace	Name	Application	Component	Ready	Status	Restarts	Age	IP	Node	Nominated Node	CPU Usage	Memory Usage
kube-system	audittrail-adapter-lt2kb	audittrail-adapter		1/1	Running	151	6d9h	172.31.0.70	ip-172-31-0-70.eu-central-1.compute.internal	<none>	1m	20 MiB
kube-system	audittrail-adapter-s6tvf	audittrail-adapter		1/1	Running	23	6d10h	172.31.18.89	ip-172-31-18-89.eu-central-1.compute.internal	<none>	6m	23 MiB
default	devcon-cluster-manager-74746ff998-gg4m9	devcon-cluster-manager		1/1	Running	13	12d	10.2.211.12	ip-172-31-12-12.eu-central-1.compute.internal	<none>	0m	83 MiB
default	devcon-cluster-manager-74746ff998-4zzwn	devcon-cluster-manager		1/1	Running	13	12d	10.2.3.13	ip-172-31-4-240.eu-central-1.compute.internal	<none>	0m	63 MiB
visibility	zmon-sentry-agent-7867b8d48b-rddbg	zmon-sentry-agent		1/1	Running	11	10d	10.2.211.124	ip-172-31-12-12.eu-central-1.compute.internal	<none>	0m	17 MiB
default	[REDACTED]	main		1/1	Running	10	67m	10.2.178.190	ip-172-31-16-111.eu-central-1.compute.internal	<none>	236m	212 MiB
default	cert-manager-ff4d7884d-s6jnm	cert-manager		1/1	Running	8	12d	10.2.220.9	ip-172-31-20-103.eu-central-1.compute.internal	<none>	5m	51 MiB

SUMMARY

- Application **Bootstrapping**
- **Git** as source of truth and UI
- **4-eyes** principle for master/production
- Declarative **Kubernetes API** configuration and **AWS CloudFormation**



DEPLOYING CHANGES



- ❑ Every deployment must be approved by dedicated QA testers
- ❑ Every master commit should go to production
- ❑ Multiple changes should be bundled into one deployment

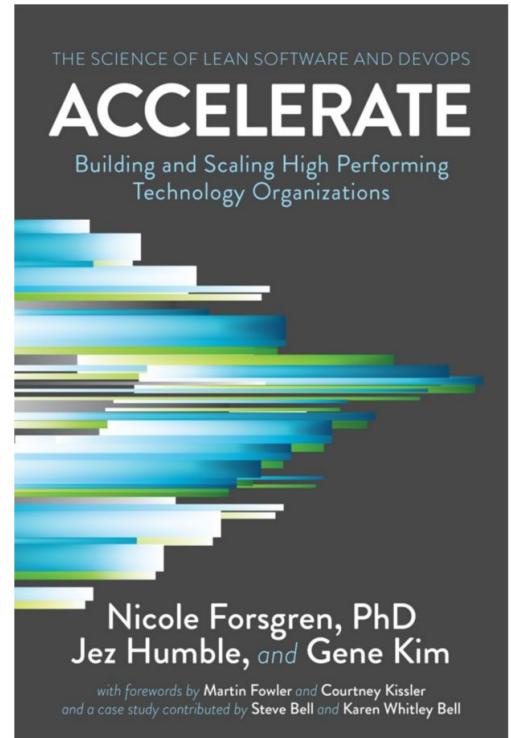
DEPLOYING CHANGES

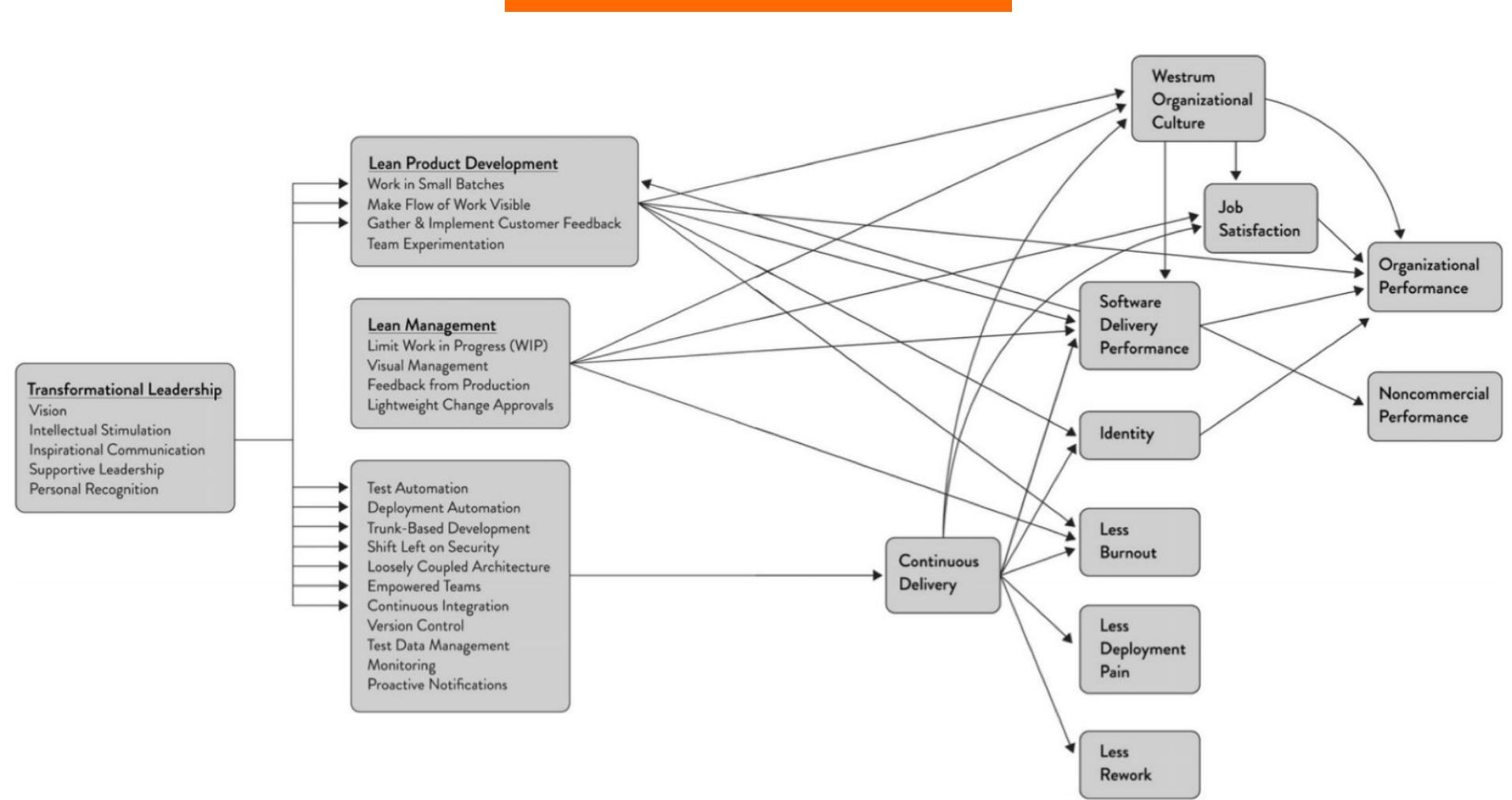


- ❑ Every deployment must be approved by dedicated QA testers
- ❑ Every master commit should go to production
- ❑ Multiple changes should be bundled into one deployment

DELIVERY PERFORMANCE METRICS

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



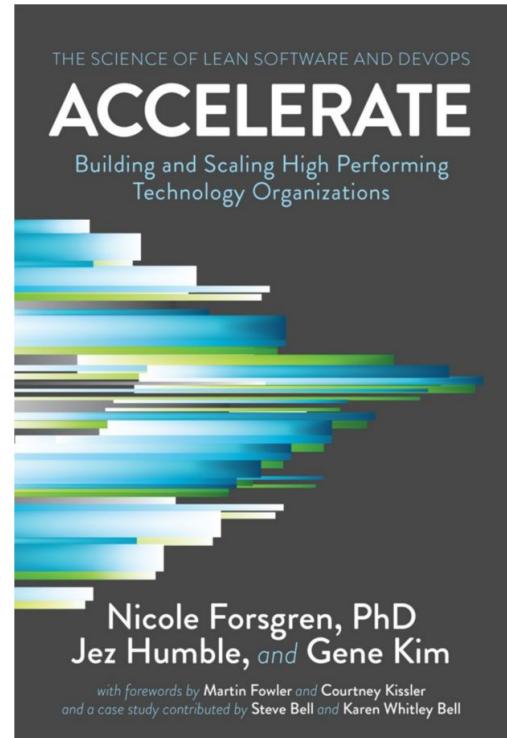


DELIVERY PERFORMANCE METRICS

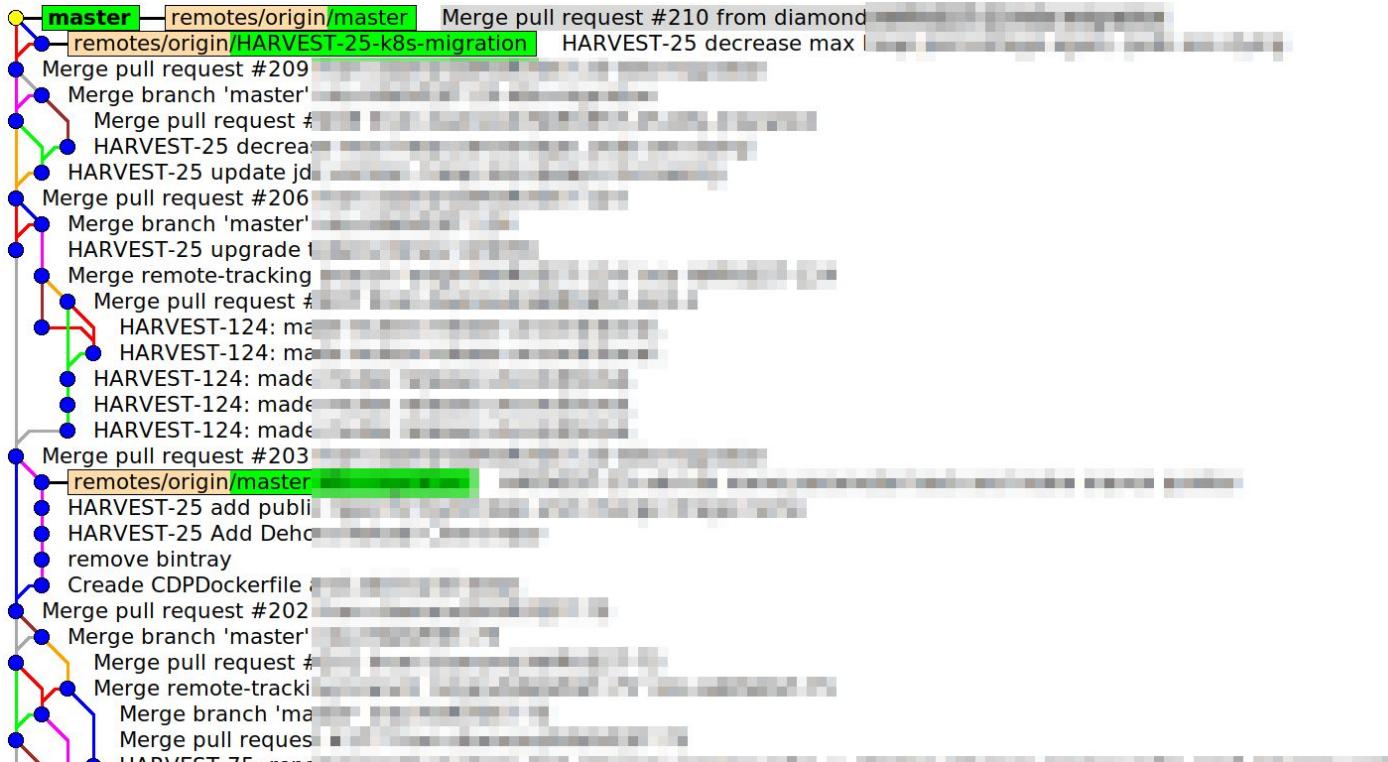
- Lead Time $\hat{=}$ **Commit to Prod**
- Release Frequency $\hat{=}$ **Deploys/week/dev**
- Time to Restore Service $\hat{=}$ **MTRS from incidents**
- Change Fail Rate $\hat{=}$ **n/a**

TRUNK-BASED DEVELOPMENT

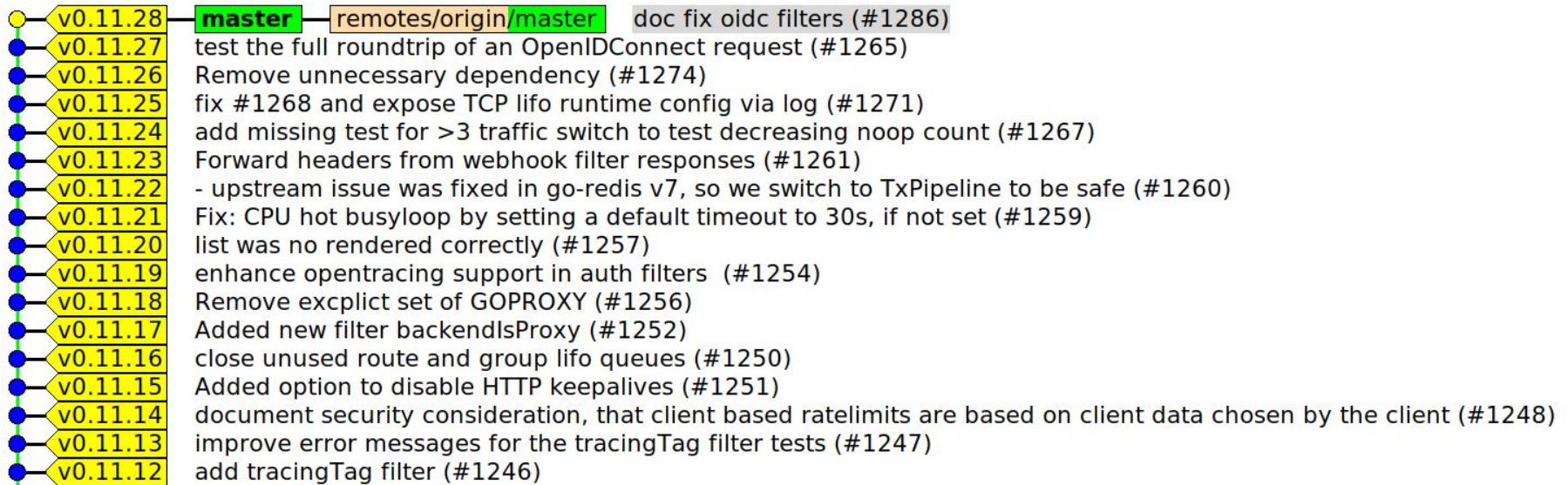
*We .. found that teams using branches that live a short amount of time (integration times **less than a day**) combined with short merging and integration periods (less than a day) do better in terms of software delivery performance than teams using longer-lived branches*



SHORT-LIVED GIT BRANCHES



SQUASHING



PRE-PRODUCTION

TESTING IN PRODUCTION

- UNIT TESTS
- FUNCTIONAL TESTS
- COMPONENT TESTS
- FUZZ TESTS
- STATIC ANALYSIS
- PROPERTY BASED TESTS
- COVERAGE TESTS
- BENCHMARK TESTS
- REGRESSION TESTS
- CONTRACT TESTS
- LINT TESTS
- ACCEPTANCE TESTS
- MUTATION TESTS
- SMOKE TESTS
- UI/UX TESTS
- USABILITY TESTS
- PENETRATION TESTS
- THREAT MODELLING

← DEPLOY →

- INTEGRATION TESTS

- TAP COMPARE

- LOAD TESTS

- SHADOWING

- CONFIG TESTS

← RELEASE →

- CANARYING

- MONITORING

- TRAFFIC SHAPING

- FEATURE FLAGGING

- EXCEPTION TRACKING

← POST RELEASE →

- TEEING

- PROFILING

- LOGS/EVENTS

- CHAOS TESTING

- MONITORING

- A/B TESTS

- TRACING

- DYNAMIC EXPLORATION

- REAL USER MONITORING

- AUDITING

- ONCALL EXPERIENCE

PRE-COMMIT HOOKS

```
~/workspace/kube-downscaler$ git ci -am 'document main function'
Check hooks apply to the repository.....Passed
Check for useless excludes.....Passed
Check Kubernetes manifests.....Passed
Reorder python imports.....Passed
black.....Passed
pydocstyle.....Passed
yamlint.....Passed
mypy.....Passed
Dockerfile linter.....(no files to check)Skipped
Check for added large files.....Passed
Check docstring is first.....Passed
Debug Statements (Python).....Passed
Fix End of Files.....Passed
Flake8.....Passed
Trim Trailing Whitespace.....Passed
Check python ast.....Passed
Check builtin type constructor use.....Passed
Detect Private Key.....Passed
Mixed line ending.....Passed
Tests should end in _test.py.....(no files to check)Skipped
type annotations not comments.....Passed
use logger.warning(.....Passed
check for eval().....Passed
check for not-real mock methods.....Passed
check blanket noqa.....Passed
[pre-commit-hooks d363c9c] document main function
  3 files changed, 8 insertions(+), 10 deletions(-)
```

UNIT TESTS

DEPLOYMENT UNITS

CAMPAIGN-CREATOR

PR-857-7

CAMPAIGN-CREATOR



BUILD



DEPLOY-CAMPAIGN-CREATOR-DEV



DEPLOY-CAMPAIGN-CREATOR-LIVE

GENERAL

ARTIFACTS

TEST RESULTS

LOGS

Ran for 10 m, 46 s

All packages 78.91%

de.zalando.campaign.app 77.54%

de.zalando.campaign.controllers 77.84%

de.zalando [redacted] 64.88%

de.zalando [redacted] 35.05%

de.zalando [redacted] 49.09%

de.zalando [redacted] 92.21%

de.zalando [redacted] 58.46%

de.zalando [redacted] 74.78%

de.zalando [redacted] 75.16%

de.zalando [redacted] 100.00%

de.zalando [redacted] 100.00%

de.zalando [redacted] 97.73%

de.zalando [redacted] 87.37%

de.zalando [redacted] 92.54%

de.zalando [redacted] 64.77%

SCoverage generated at Fri Jan 24 15:44:49 UTC 2020

Lines of code: 8750 Files: 131 Classes: 229 Methods: 77

Lines per file: 66.79 Packages: 19 Classes per package: 12.05 Methods per class: 3.38

Total statements: 10084 Invoked statements: 7957 Total branches: 95 Invoked branches: 59

Ignored statements: 0

Statement coverage: 78.91 % Branch coverage: 62.11 %

Class	Source file	Lines	Methods	Statements	Invoked	Coverage	Branches	Invoked	Coverage
-------	-------------	-------	---------	------------	---------	----------	----------	---------	----------

ActorsConfiguration	ActorsConfiguration.scala	15	1	4	4	100.00 %	0	0	100 %
---------------------	---------------------------	----	---	---	---	----------	---	---	-------

AppLoader	AppLoader.scala	30	1	2	0	0.00 %	0	0	0.0 %
-----------	-----------------	----	---	---	---	--------	---	---	-------

ApplicationComponents	AppLoader.scala	72	1	28	0	0.00 %	0	0	0.0 %
-----------------------	-----------------	----	---	----	---	--------	---	---	-------

ApplicationConfigurationLoader	ApplicationConfigurationLoader.scala	36	1	8	8	100.00 %	0	0	100 %
--------------------------------	--------------------------------------	----	---	---	---	----------	---	---	-------

Binders	Binders.scala	74	4	85	85	100.00 %	0	0	100 %
---------	---------------	----	---	----	----	----------	---	---	-------

STATIC ANALYSIS WITH SONARQUBE

The screenshot displays the SonarQube web interface for the project "shipping-notice-service".

Project Overview: The project is named "shipping-notice-service" and is currently at the "master" branch. The build status is "Passed".

Quality Gate: The Quality Gate is "Passed".

Bugs: 0 Bugs (A)

Vulnerabilities: 0 Vulnerabilities (A)

Leak Period: since master-753 started 3 days ago

Code Smells: 4d (A) Debt, 337 Code Smells, 0 New Debt, 0 New Code Smells

Coverage: 92.1% Coverage, 414 Unit Tests, Coverage on New Code

Duplications: 0.3% Duplications, 2 Duplicated Blocks, Duplications on New Code

Metrics: 8.6k Lines of Code (Java: 8.3k, Kotlin: 339), No tags

Activity: A line graph showing activity over time, with recent builds: master-754, master-753, master-752.

Quality Gate: (Default) SonarQube way

Quality Profiles: (Java) Sonar way, (Kotlin) Detekt way

Navigation: Home, Bug Tracker, Sources

Key: A color-coded legend for the dashboard cards.

VULNERABILITY SCANNING IN GITHUB

 We found a potential security vulnerability in one of your dependencies.
You can see this message because you have been granted [access to security alerts for this repository](#).

[View security alert](#)

Security Alerts

[Automated security updates ▾](#)

[Dismiss all ▾](#)

 1 Open ✓ 0 Closed

[Sort ▾](#)

 org.springframework:spring-webmvc

high severity

 by GitHub  pom.xml  #62

GitHub tracks known security vulnerabilities in some dependency manifest files. [Learn more about security alerts.](#)

FEATURE TOGGLES

```
public boolean shouldUseParcelServiceForReturnLabel(final int appDomainId, final String customerNumber,  
    final ISOCountryCode shippingCountry) {  
    return isCepMigrationEnabled(appDomainId, customerNumber, shippingCountry,  
        CepMigrationFeatureToggle.CEP_RETURN_LABEL);  
}
```

▼ feature.cepMigration.customerPercentage

Type: Percentage

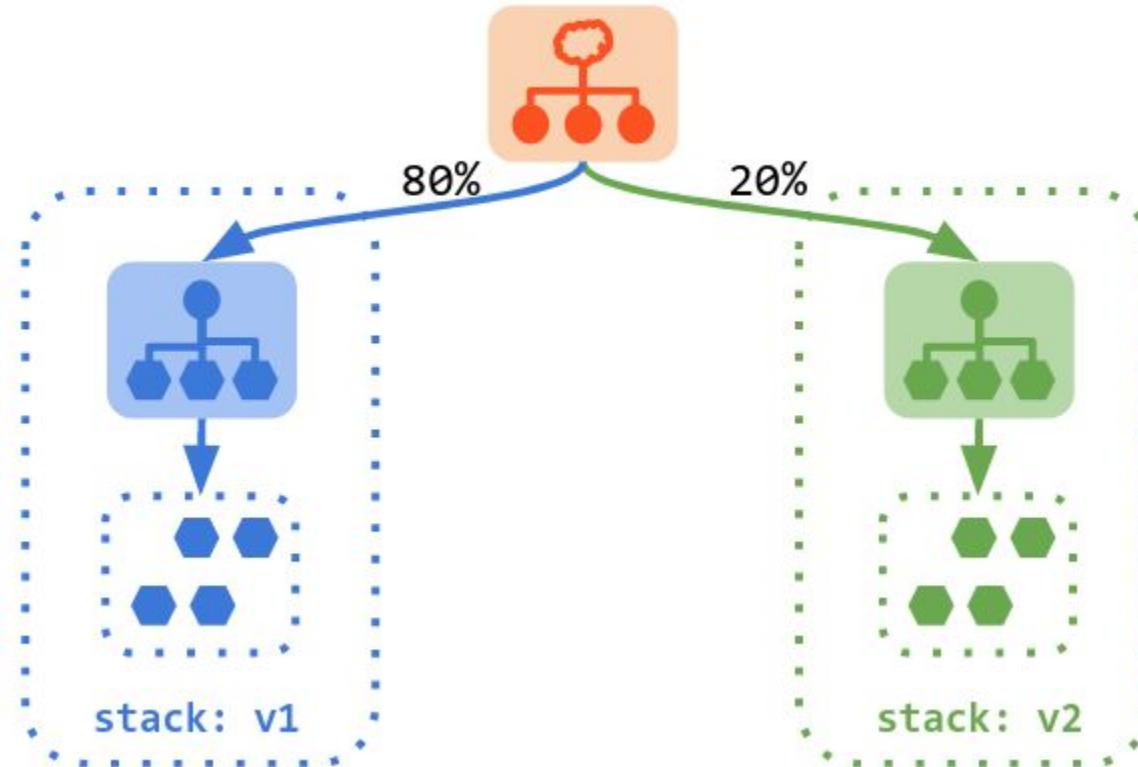
Aliases:

Projects: customer

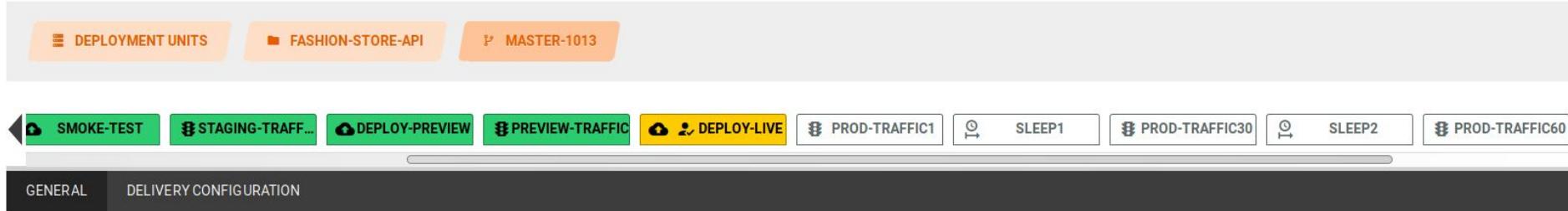
Comment: Percentage of customers that should get CEP Migration. Shipping country and warehouse are required context parameters. 

appDomainId	countryCode	Value	Actions
<input type="text" value="Filter appDomainId"/>	<input type="text" value="Filter countryCode"/>	<input type="text" value="Filter value"/>	  
 1	 DE	100	  
 5	 NL	100	  

STACKSET: TRAFFIC SWITCHING

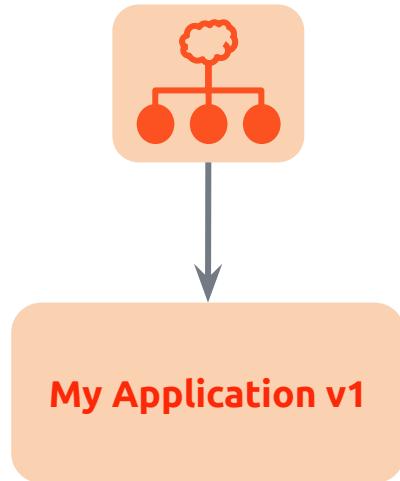


TRAFFIC SWITCHING STEPS IN CDP



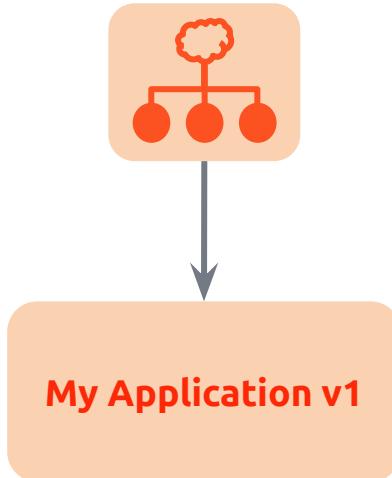
DEPLOYING TO PRODUCTION

<https://my-application.io>

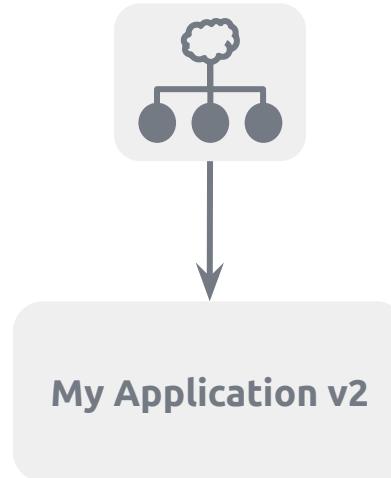


DEPLOYING TO PRODUCTION

<https://my-application.io>

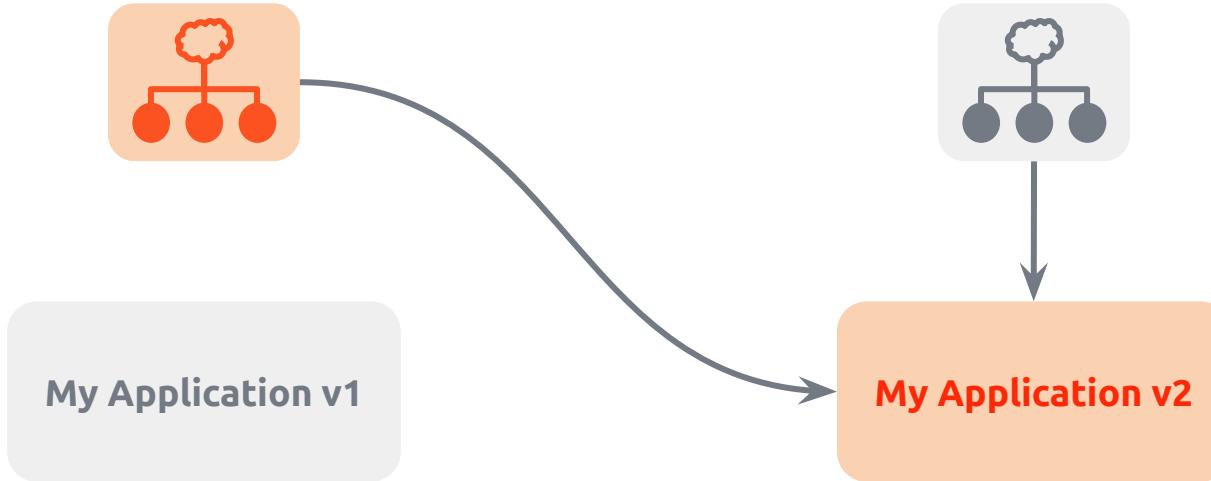


<https://preview-v2.my-application.io>



DEPLOYING TO PRODUCTION (BLUE/GREEN)

<https://my-application.io>

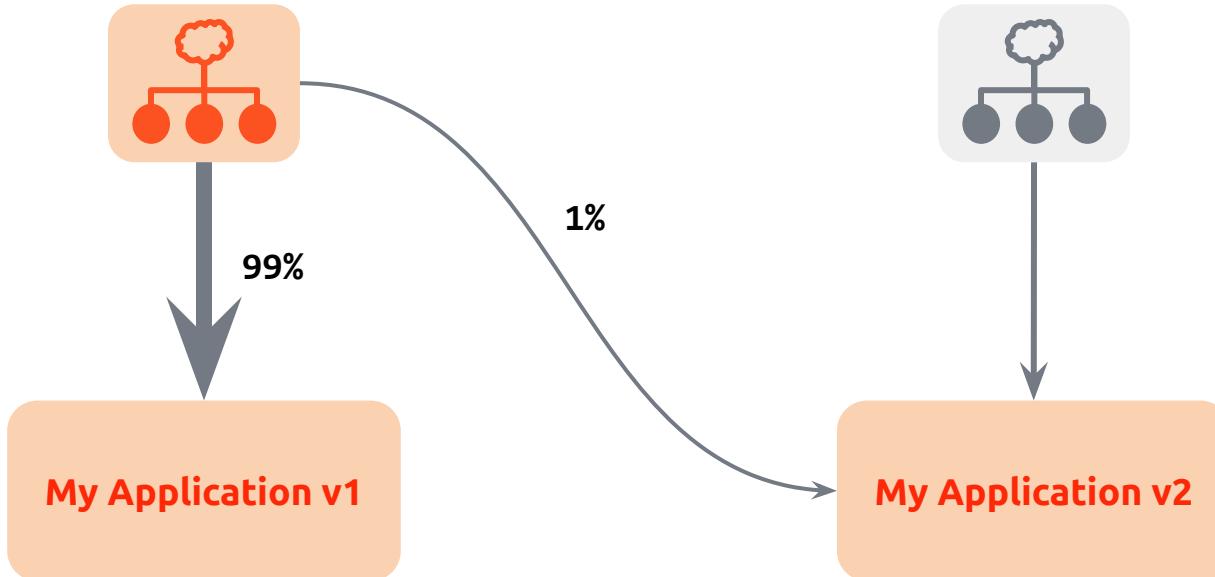


<https://preview-v2.my-application.io>

DEPLOYING TO PRODUCTION (TRAFFIC SWITCHING)

<https://my-application.io>

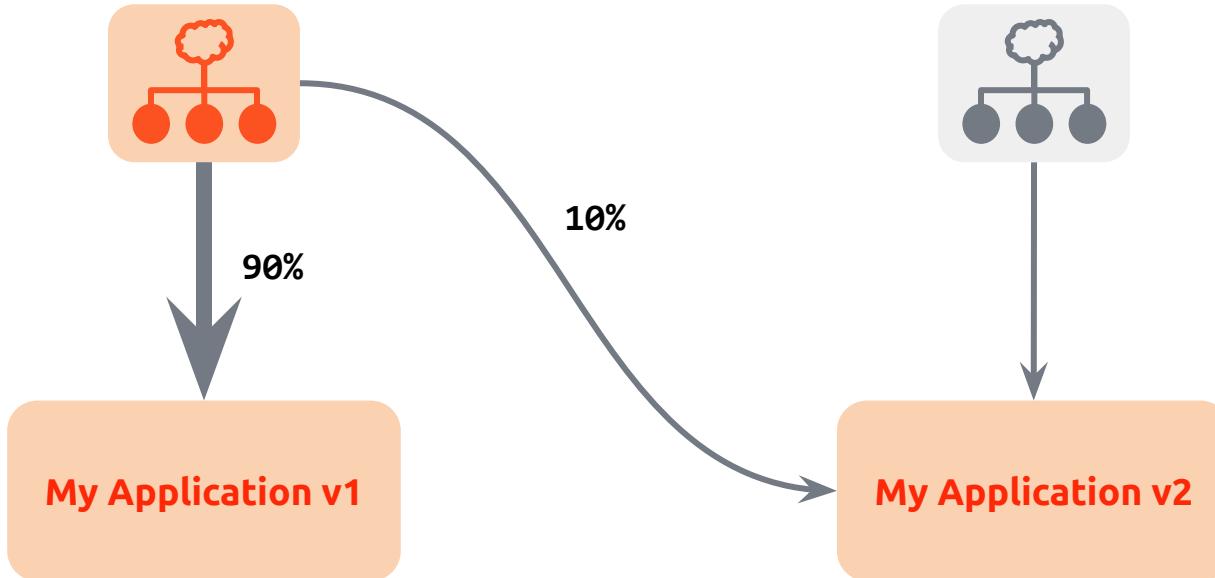
<https://preview-v2.my-application.io>



DEPLOYING TO PRODUCTION (TRAFFIC SWITCHING)

<https://my-application.io>

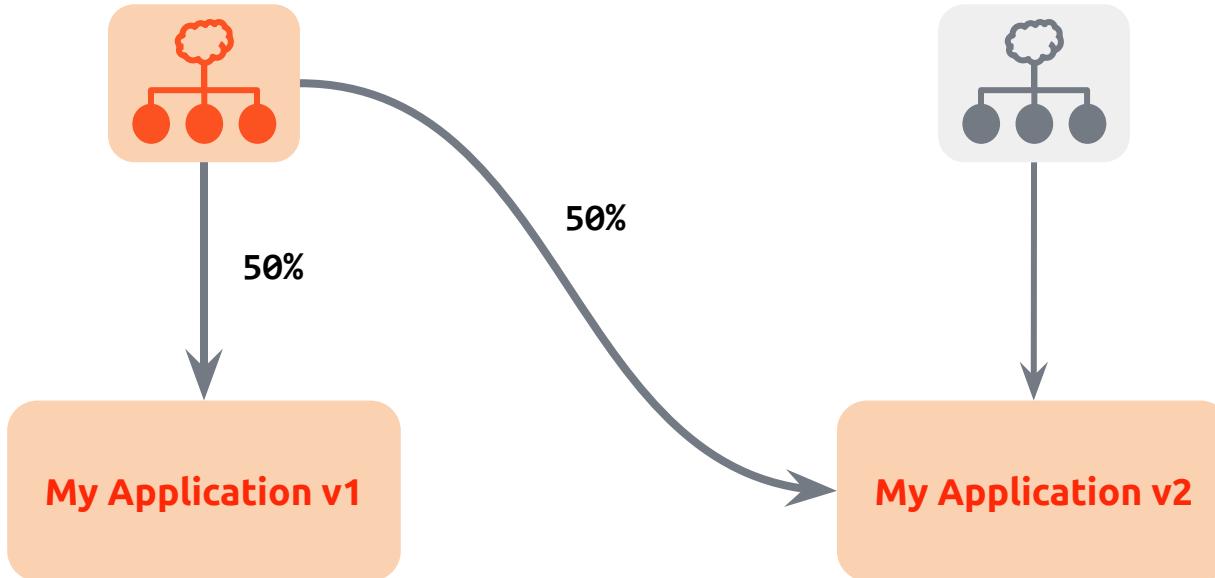
<https://preview-v2.my-application.io>



DEPLOYING TO PRODUCTION (TRAFFIC SWITCHING)

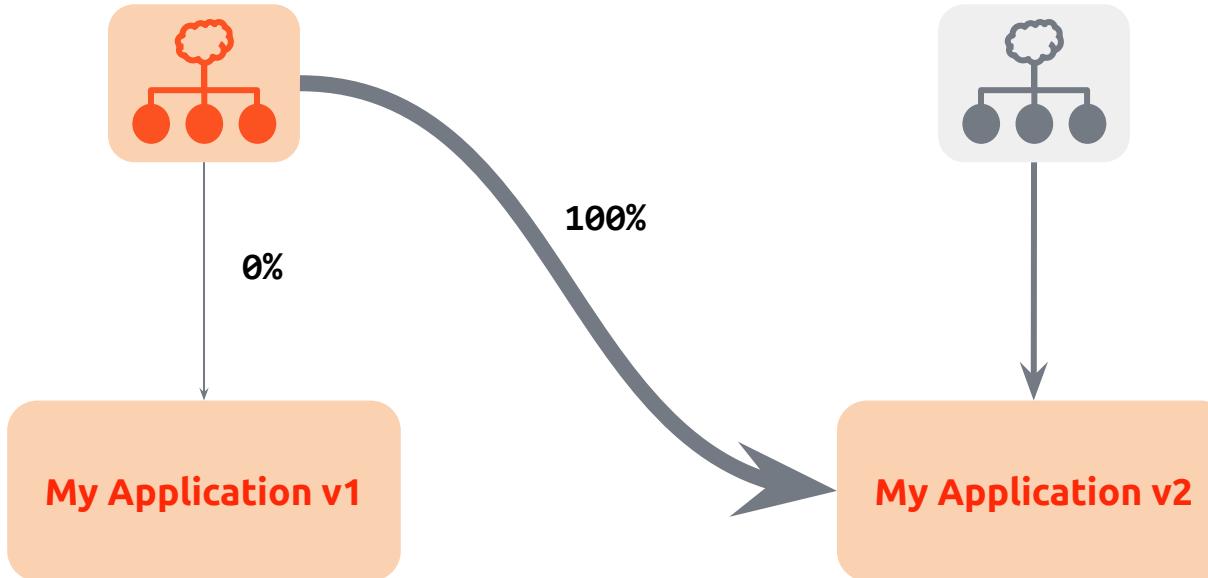
<https://my-application.io>

<https://preview-v2.my-application.io>

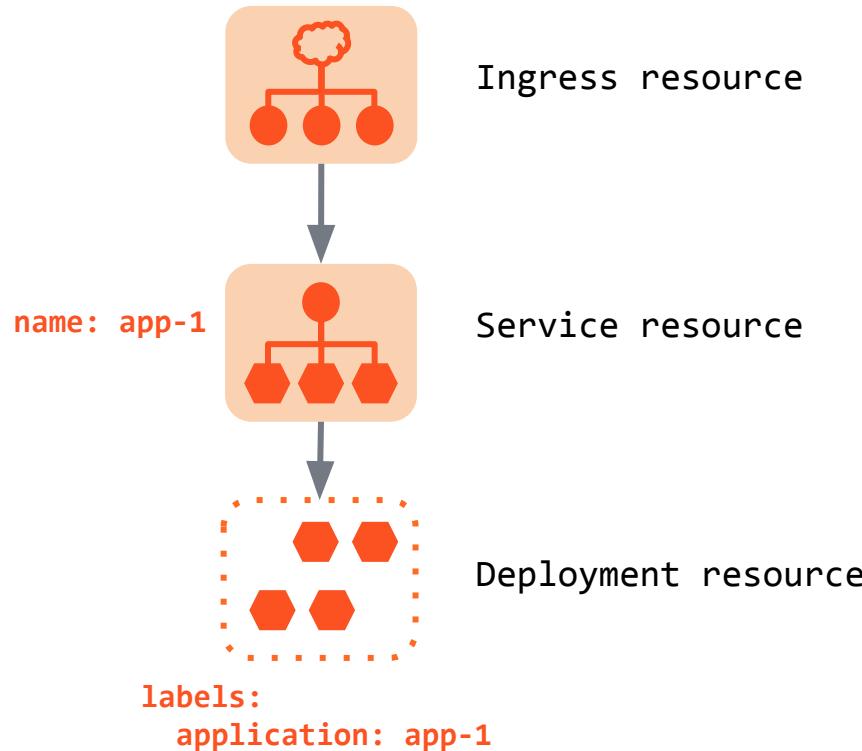


DEPLOYING TO PRODUCTION (TRAFFIC SWITCHING)

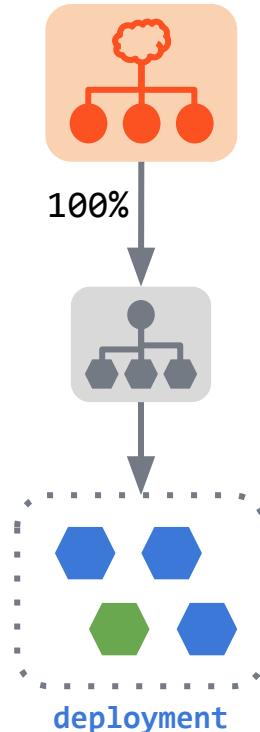
<https://my-application.io>



APPLICATION STACK IN KUBERNETES

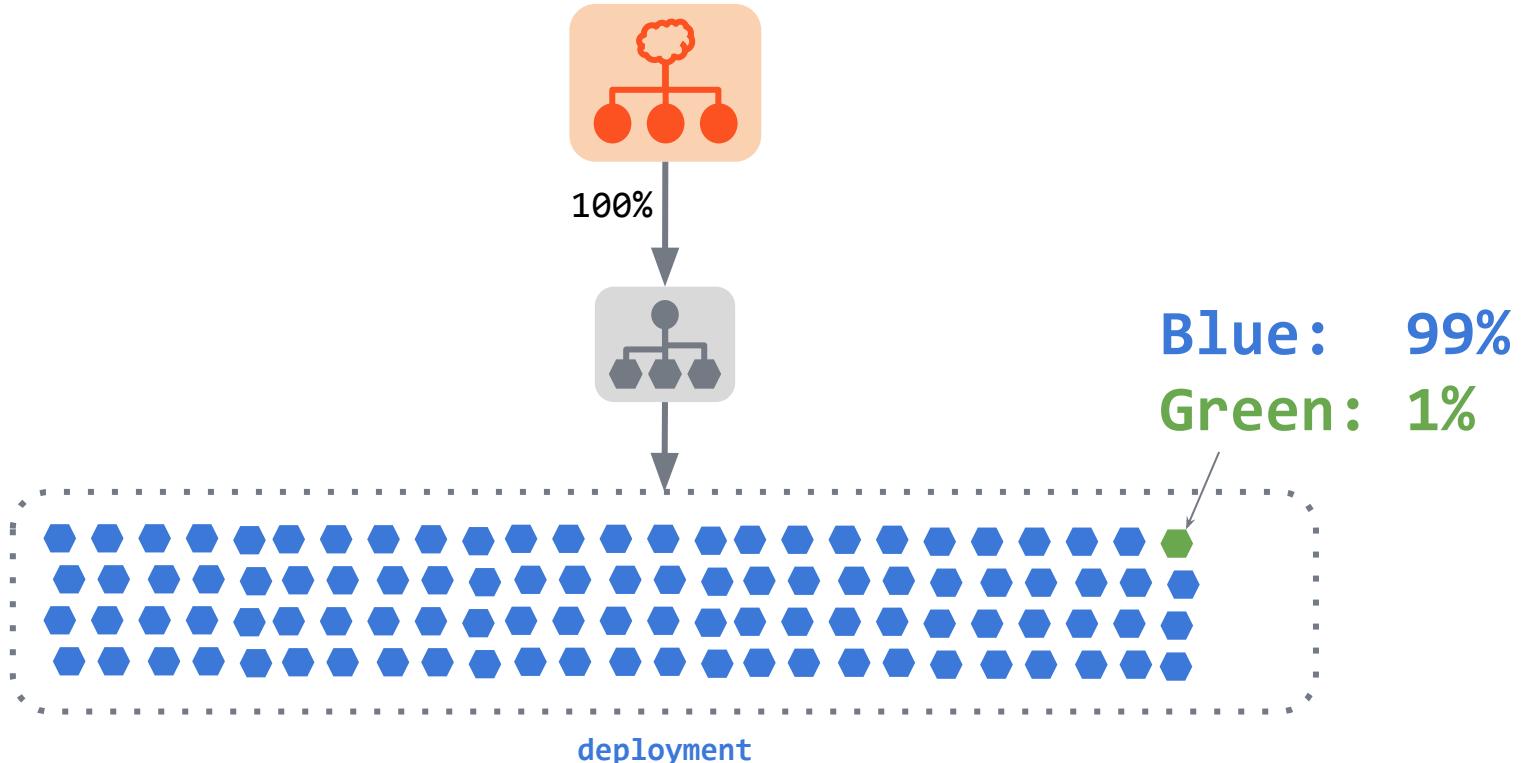


TRAFFIC SWITCHING ROLLING UPDATE DEPLOYMENT



Blue: 75%
Green: 25%

TRAFFIC SWITCHING ROLLING UPDATE DEPLOYMENT



SKIPPER TRAFFIC SUPPORT

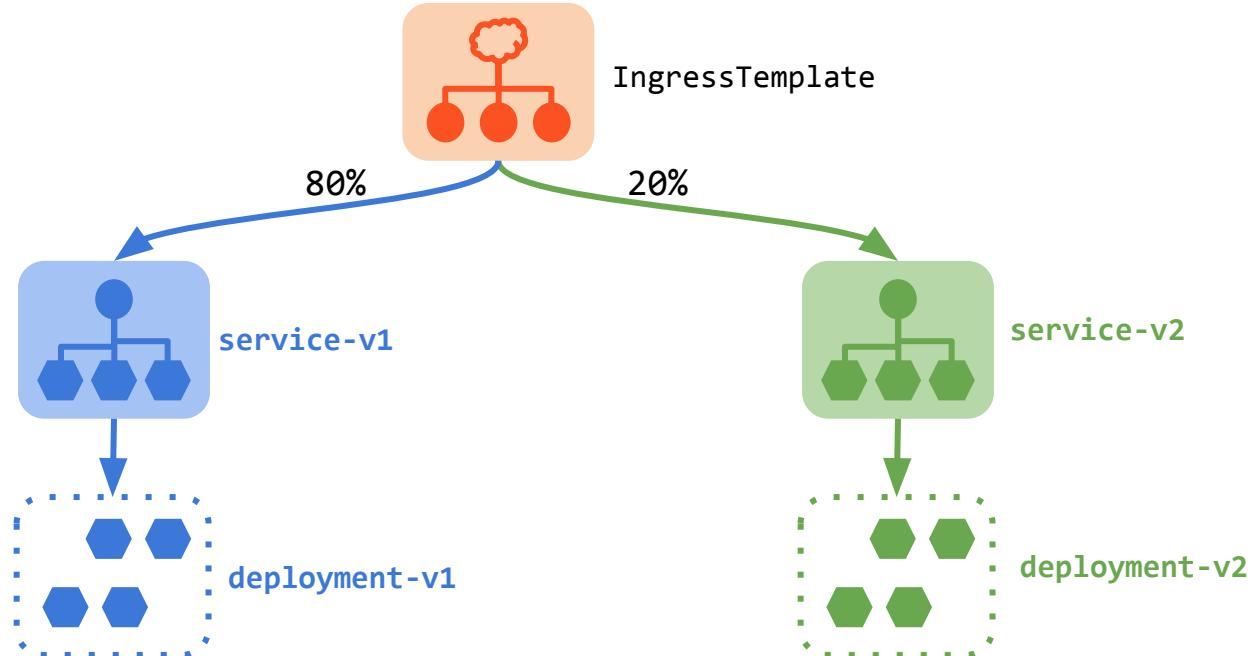
```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-app
  annotations:
    zalando.org/backend-weights: |
      {"my-app-v1": 90, "my-app-v2": 10}
```

```
spec:
  rules:
  - host: my-app.io
    ...
  ...
```



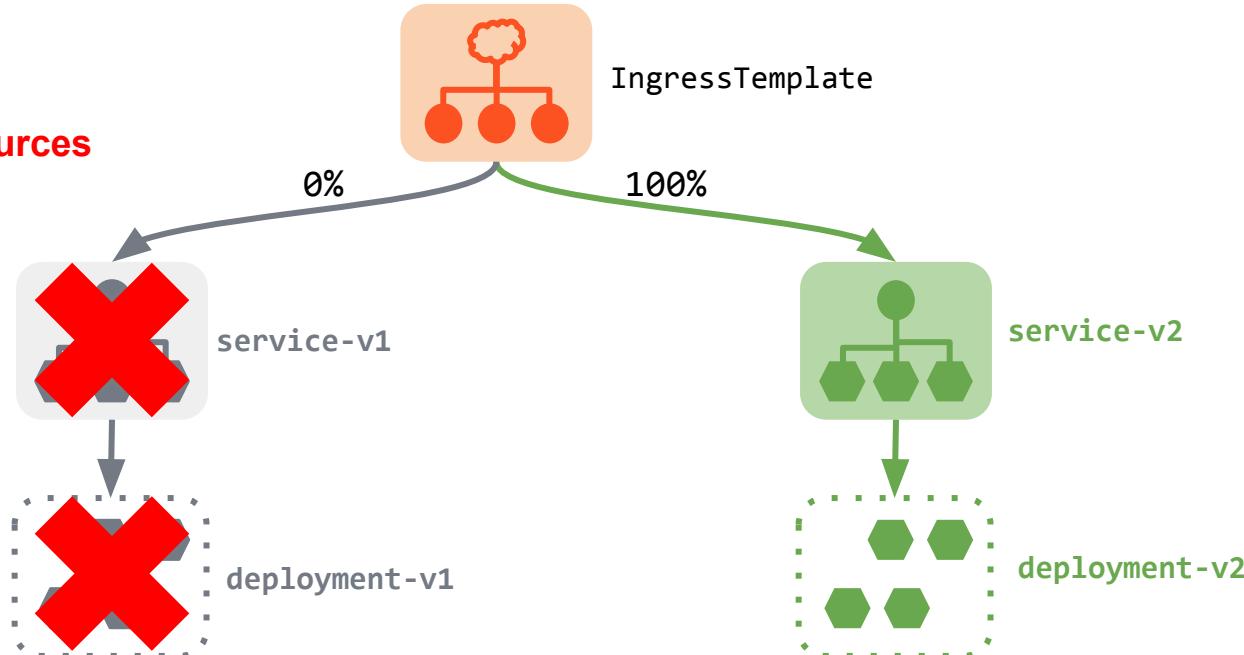
github.com/zalando/skipper

MANUAL TRAFFIC SWITCHING BLUE/GREEN DEPLOYMENT

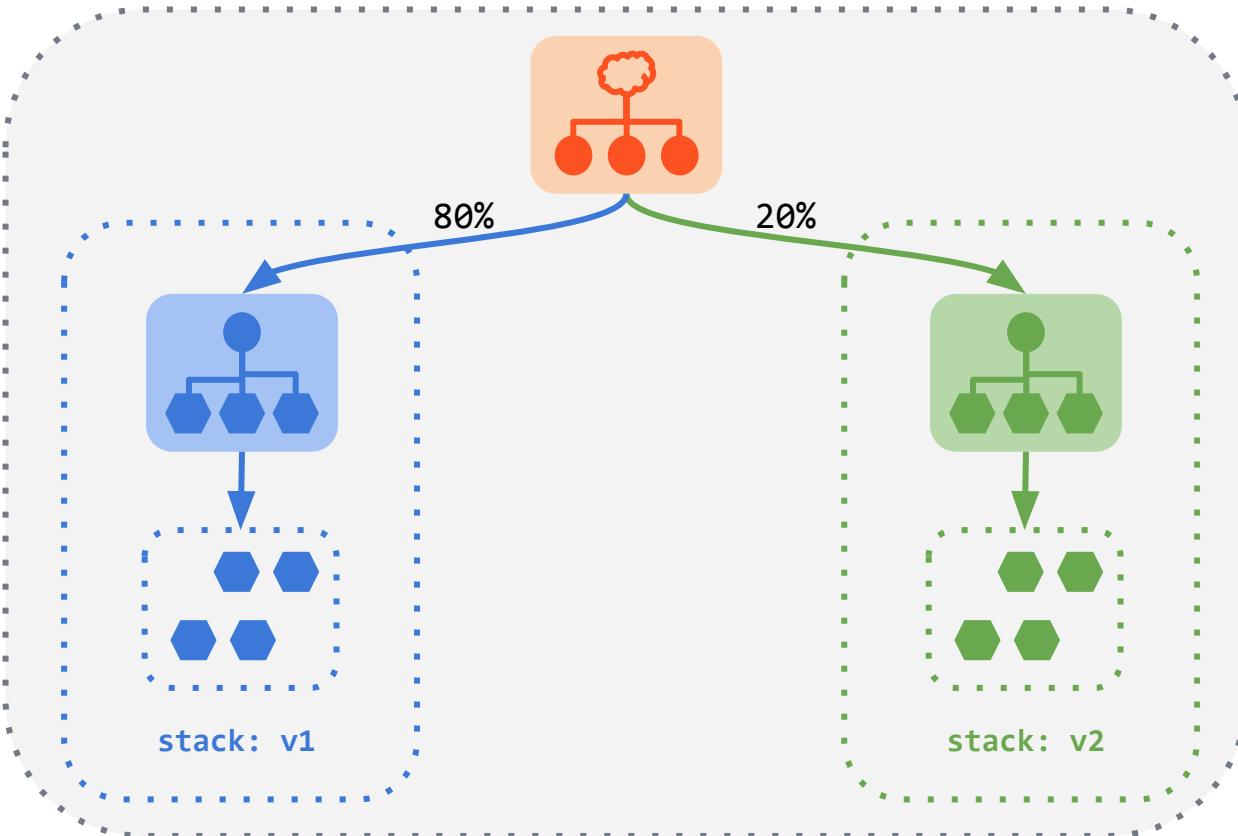


MANUAL TRAFFIC SWITCHING BLUE/GREEN DEPLOYMENT

Need to clean up resources
manually :(



TRAFFIC SWITCHING BLUE/GREEN DEPLOYMENT (StackSet)



STACKSET DEFINITION

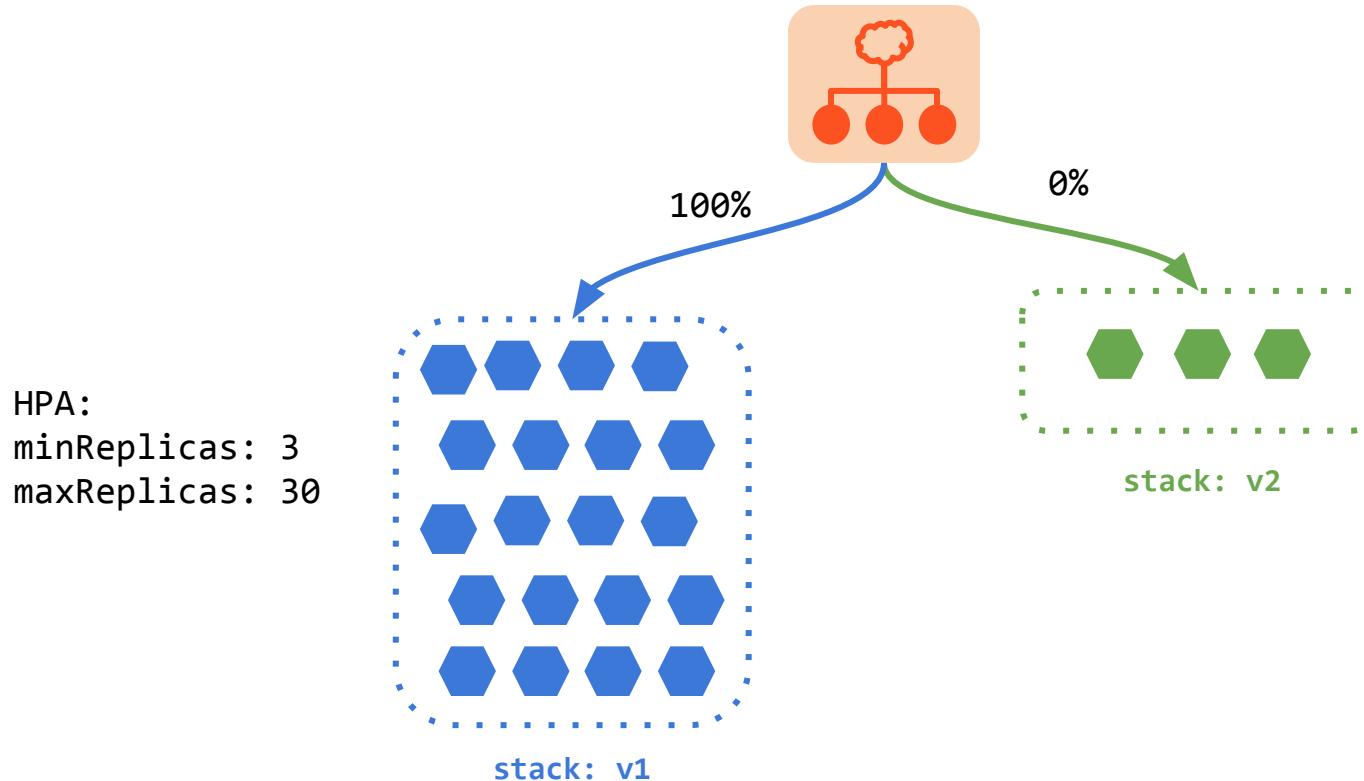
```
apiVersion: zalando.org/v1
kind: StackSet
metadata:
  name: my-app
spec:
  ingress:
    host: [my-application.io]
  stackTemplate:
    spec:
      version: v1
      podTemplate:
        spec:
          containers:
            ...
...
```

github.com/zalando-incubator/stackset-controller

ADDITIONAL FEATURES

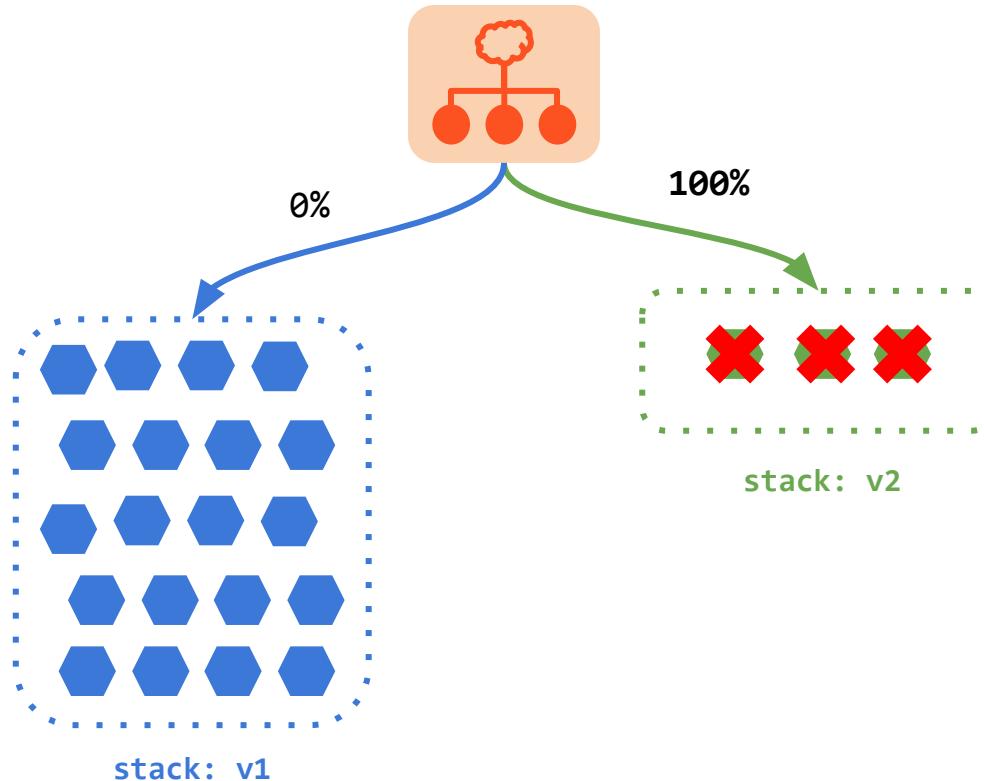


PRESCALE STACKS

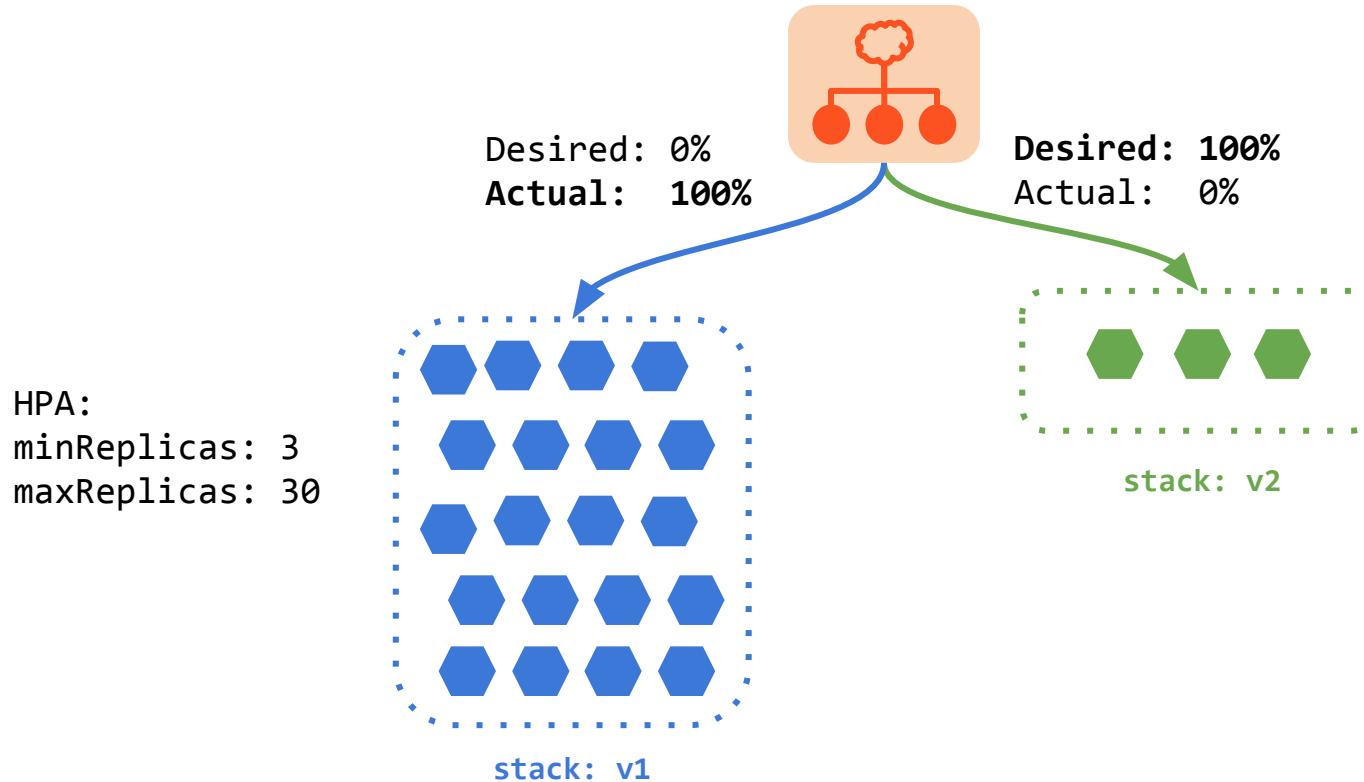


PRESCALE STACKS

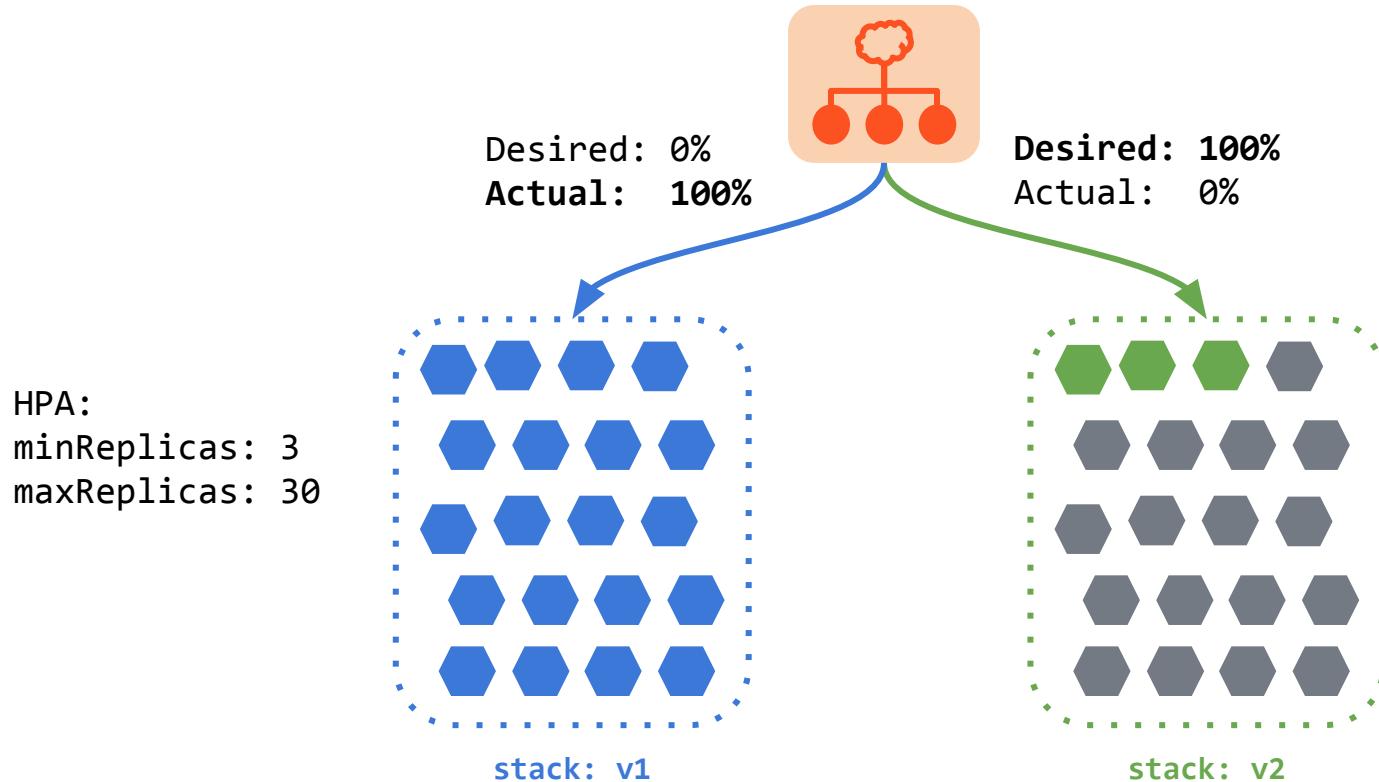
HPA:
minReplicas: 3
maxReplicas: 30



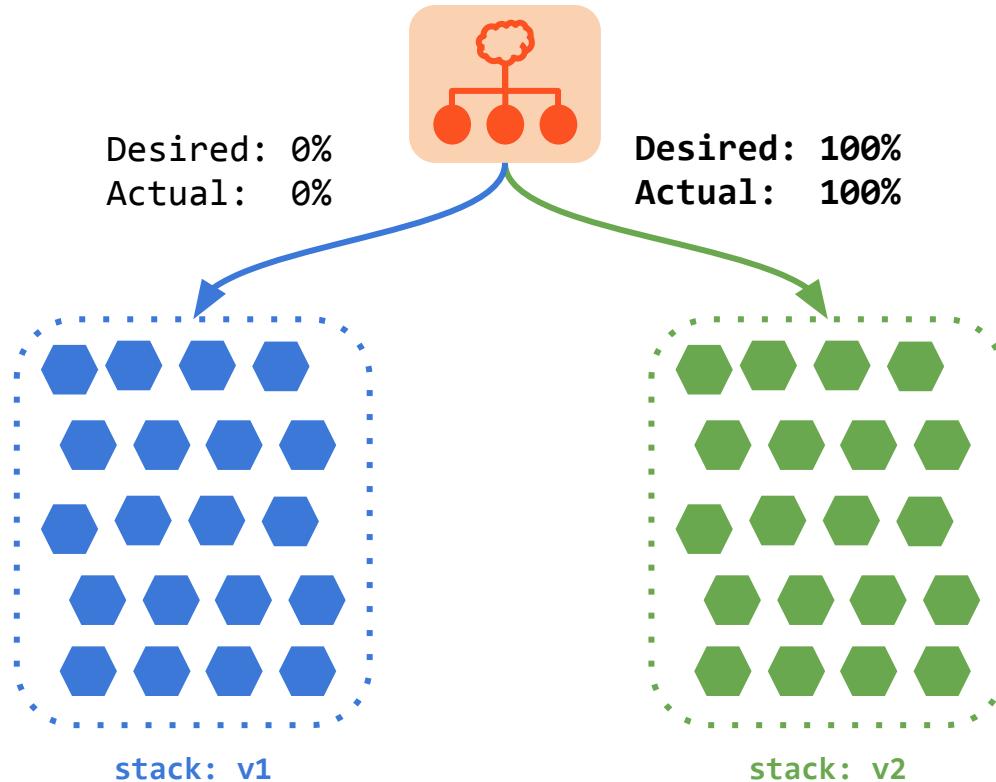
PRESCALE STACKS



PRESCALE STACKS



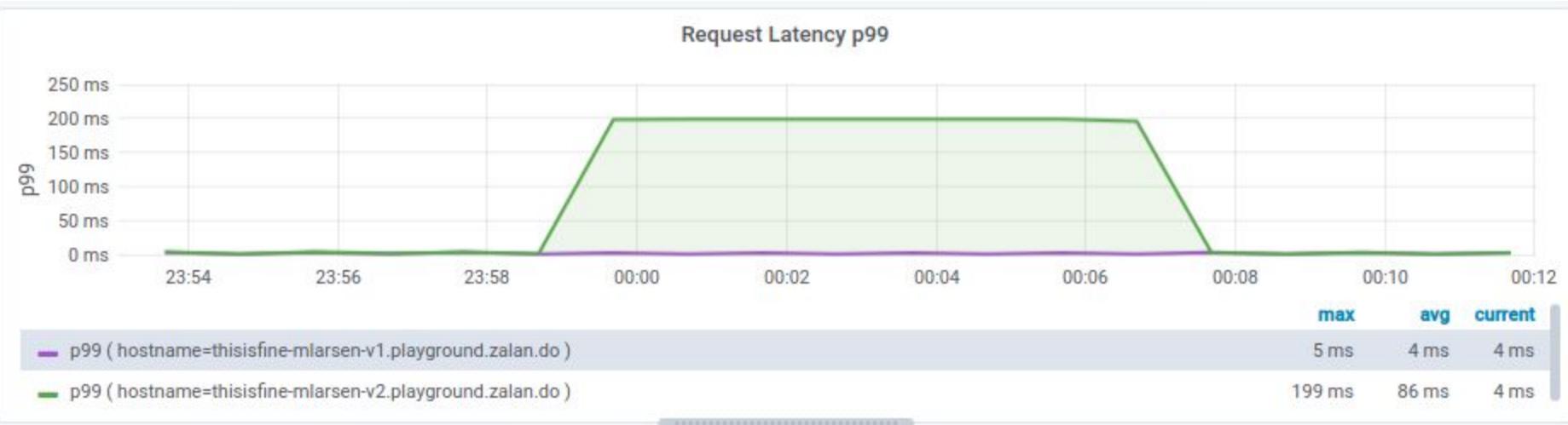
PRESCALE STACKS



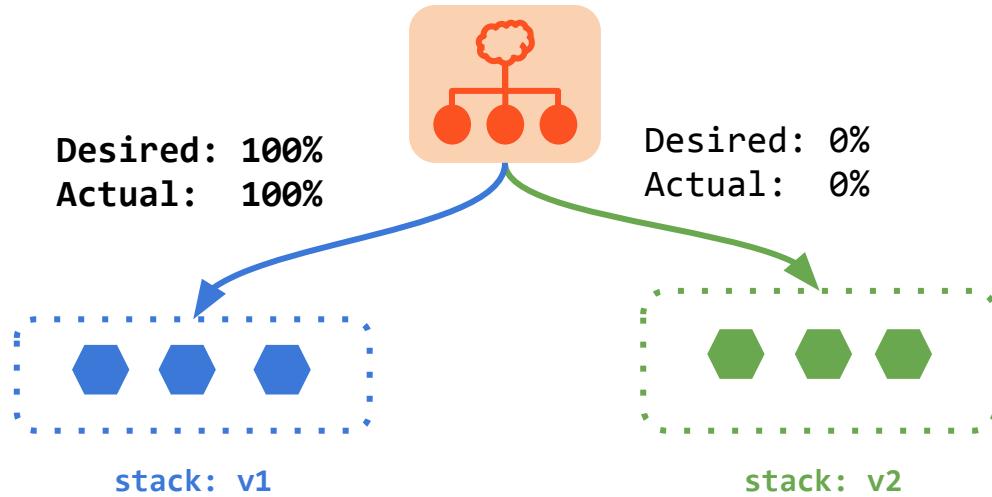


GRADUAL DEPLOYMENTS (AUTOMATIC TRAFFIC SWITCHING)

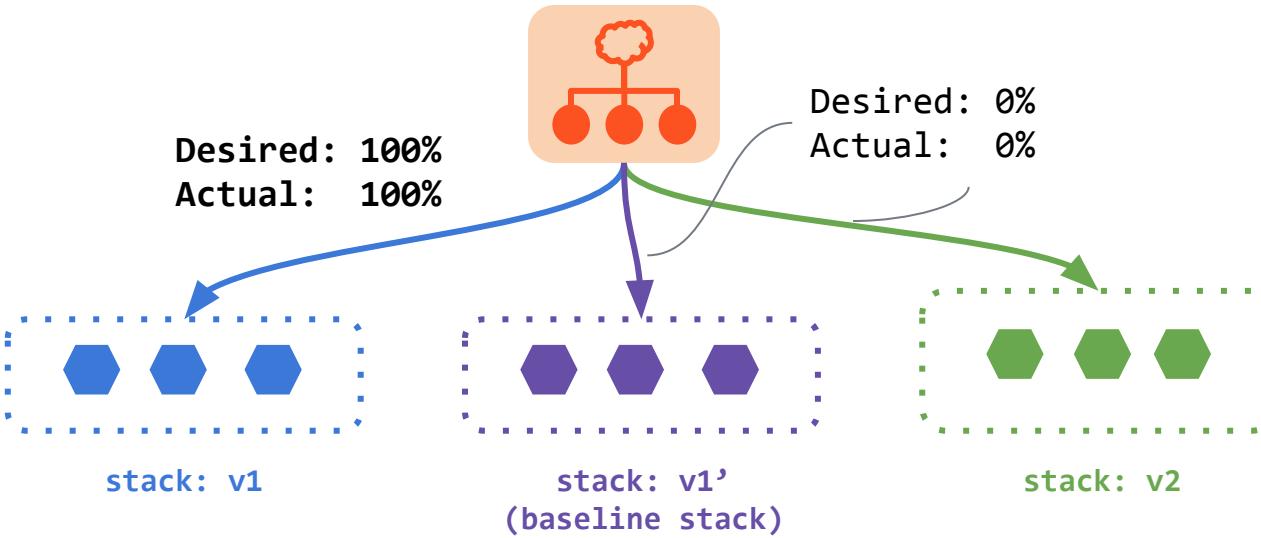
INTEGRATE WITH APPLICATION METRICS



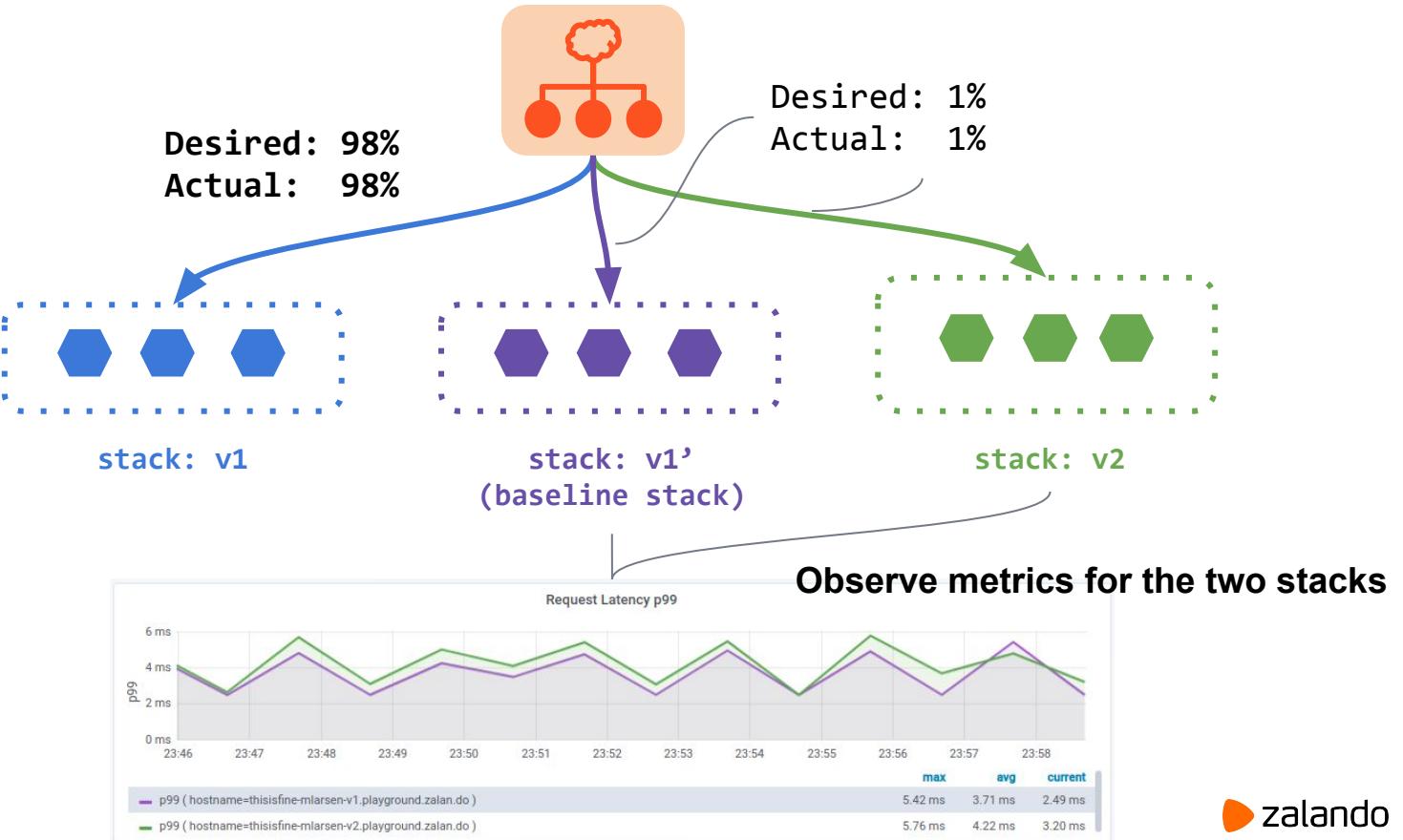
GRADUAL DEPLOYMENTS



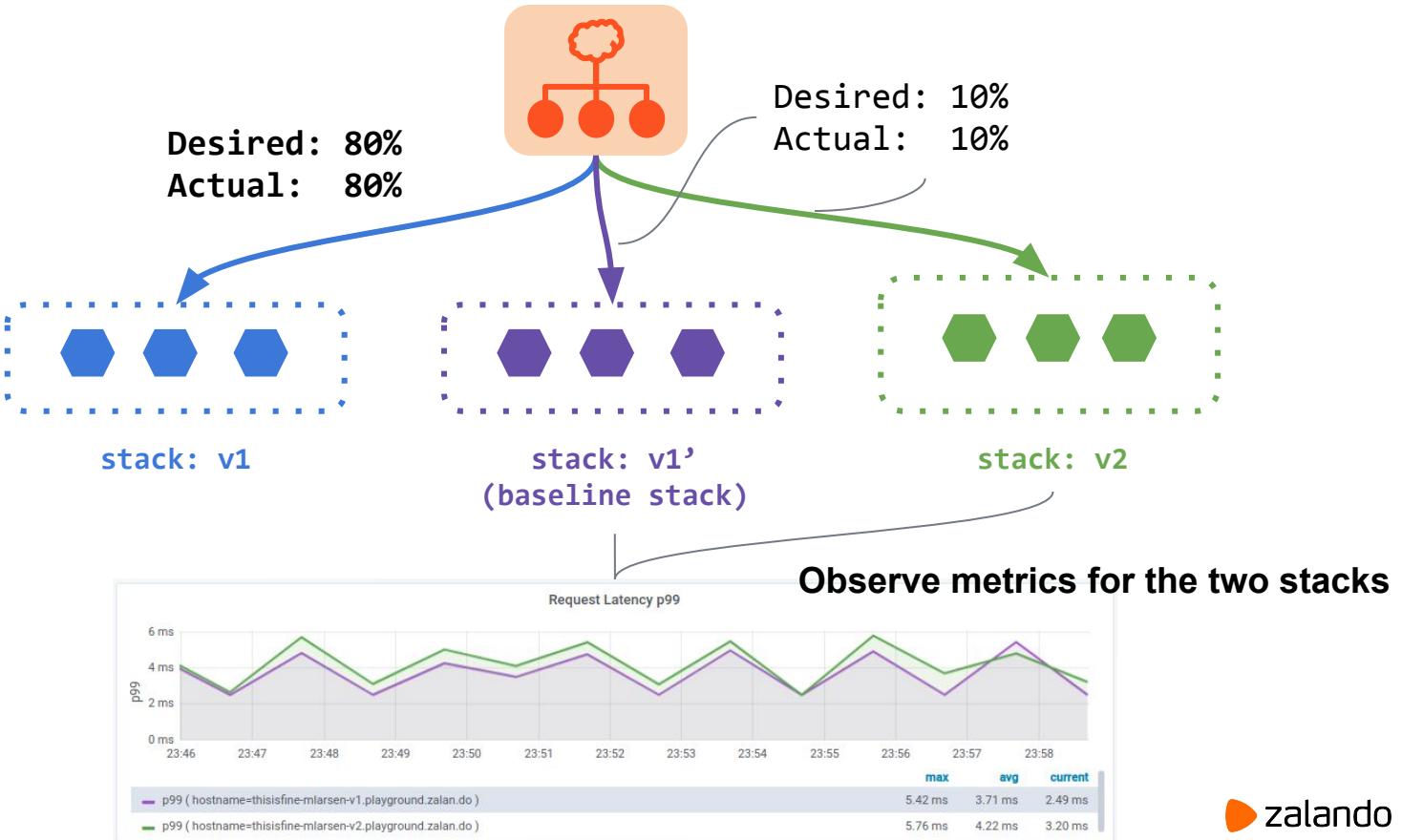
GRADUAL DEPLOYMENTS



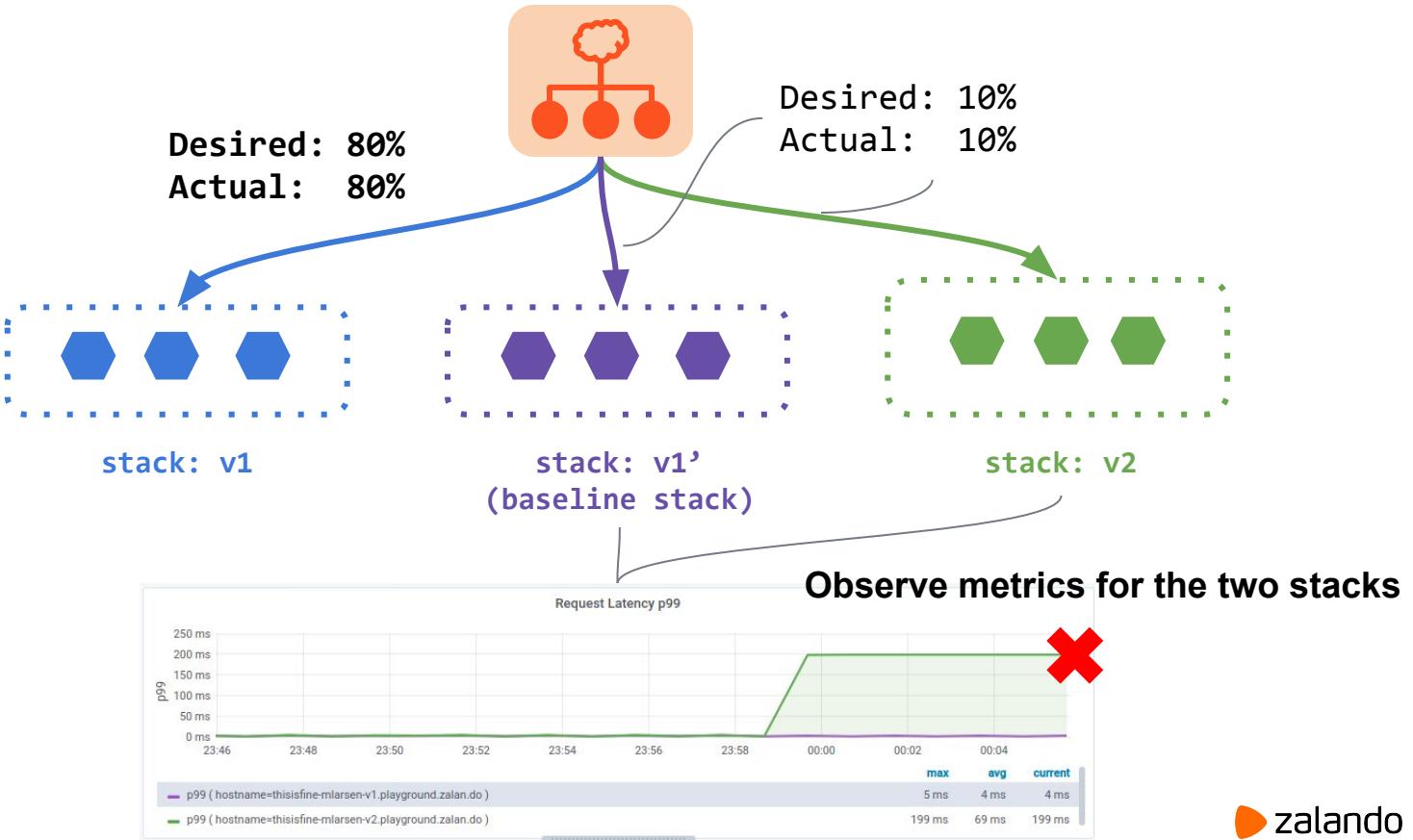
GRADUAL DEPLOYMENTS



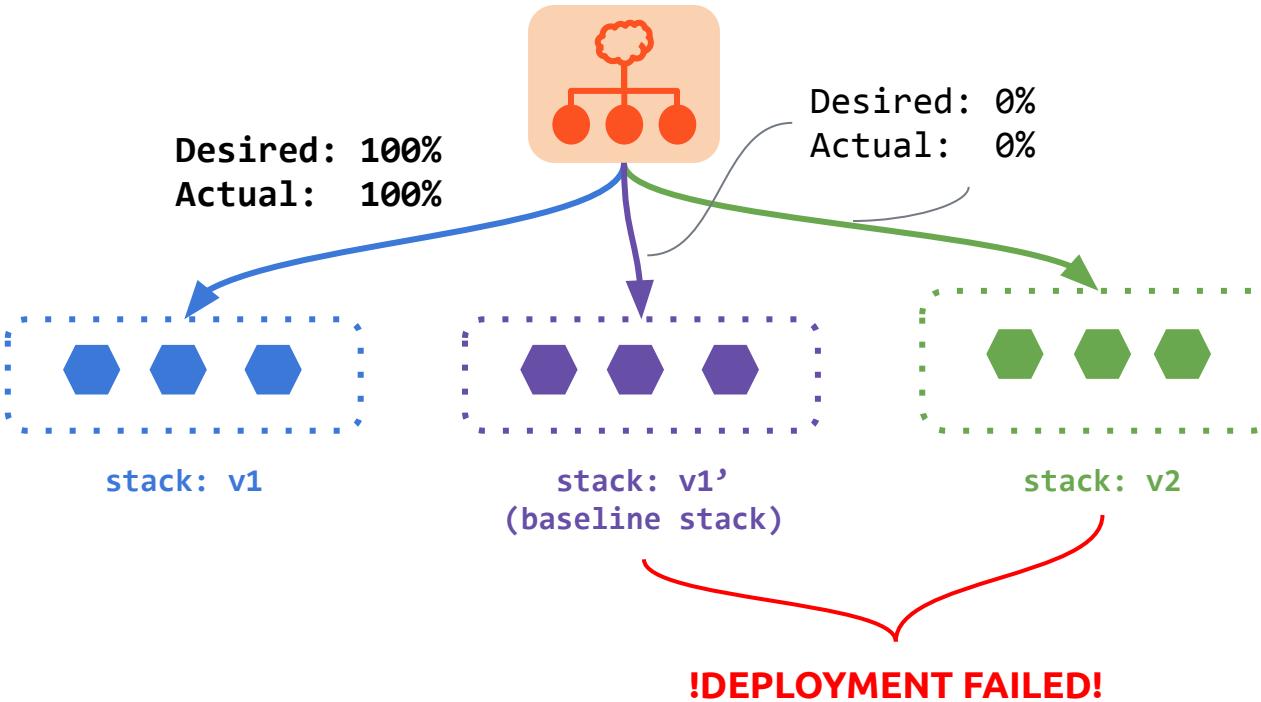
GRADUAL DEPLOYMENTS



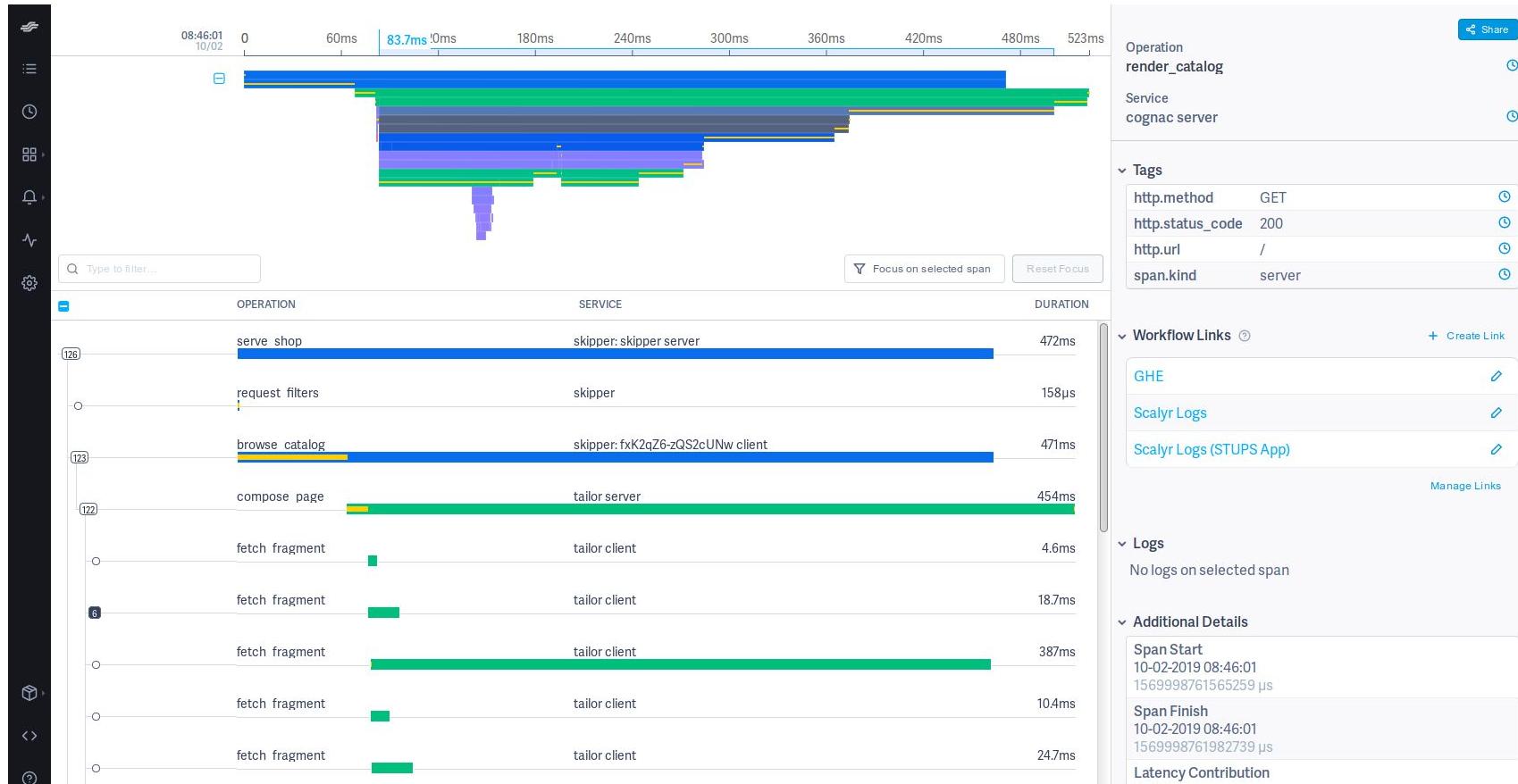
GRADUAL DEPLOYMENTS



GRADUAL DEPLOYMENTS



OPENTRACING



APPLICATION LOGS

SCALYR

LOGS SEARCH DASHBOARDS ALERTS DOCS henning@zalando.de ▾

\$application == "cluster-lifecycle-manager"

LAST 4 HOURS ▾ SEARCH

LOGS TO SEARCH START JUMP TO END ▾ SAVE ▾ SHARE ▾ LIVE TAIL ... DISPLAY HELP

aws: eu-ce Filter source (*=wildcard)

1:12 PM - 5:12 PM [UTC+1] 10,765 MATCHING EVENTS (5 min/bar) EXPAND GRAPH ▾

4k 0 1:15 PM 1:30 PM 1:45 PM 2:00 PM 2:15 PM 2:30 PM 2:45 PM 3:00 PM 3:15 PM 3:30 PM 3:45 PM 4:00 PM 4:15 PM 4:30 PM 4:45 PM 5:00 PM

ALL FIELDS # VALUES

Field	Value	Count
application	'cluster-lifecycle-manager'	1
cluster_environment	'aws:eu-ce'	1
cluster	'cluster-lifecycle-manager'	1
cluster_alias	'cluster-lifecycle-manager'	1
container_id	'cluster-lifecycle-manager'	5
container	'cluster-lifecycle-manager'	2
dataset	'cluster-lifecycle-manager'	1
environment	'cluster-lifecycle-manager'	1
logfile	'cluster-lifecycle-manager'	5
log	'cluster-lifecycle-manager'	2634
namespace	'cluster-lifecycle-manager'	1
node	'cluster-lifecycle-manager'	2
original_file	'cluster-lifecycle-manager'	6
parser	'cluster-lifecycle-manager'	1
pod	'cluster-lifecycle-manager'	4
serverHost	'cluster-lifecycle-manager'	1
serverIP	'cluster-lifecycle-manager'	3
session	'cluster-lifecycle-manager'	26
severity	'cluster-lifecycle-manager'	1
stream	'cluster-lifecycle-manager'	1
threadid	'cluster-lifecycle-manager'	28
threadName	'cluster-lifecycle-manager'	3
time	'cluster-lifecycle-manager'	2632

application='cluster-lifecycle-manager' | log='time="2020-01-27T16:05:23Z" level=info msg="statefulset.apps/skipper-ingress-redis unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:23Z" level=info msg="storageclass.storage.k8s.io/standard unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:24Z" level=info msg="deployment.apps/zmon-agent configured" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:24Z" level=info msg="serviceaccount/zmon unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:24Z" level=info msg="clusterrole.rbac.authorization.k8s.io/system unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:24Z" level=info msg="clusterrolebinding.rbac.authorization.k8s.io/system unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:24Z" level=info msg="verticalpodautoscaler.autoscaling.k8s.io/zmo unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:25Z" level=info msg="awsiamrole.zalando.org/zmon-aws-iam-credentials unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:25Z" level=info msg="deployment.apps/zmon-aws-agent configured" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:25Z" level=info msg="deployment.apps/zmon-redis unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:25Z" level=info msg="service/zmon-redis unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:26Z" level=info msg="deployment.apps/zmon-scheduler configured" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:26Z" level=info msg="deployment.apps/zmon-worker configured" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:27Z" level=info msg="secret/zmon-worker unchanged" cluster=custom cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:27Z" level=error msg="Error from server (NotFound): limitranges \ cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:27Z" level=error msg="warning: deleting cluster-scoped resources, cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:27Z" level=error msg="Error from server (NotFound): customresources \ cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:27Z" level=error msg="Error from server (NotFound): deployments.e cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:27Z" level=error msg="Error from server (NotFound): serviceaccounts \ cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:27Z" level=error msg="warning: deleting cluster-scoped resources, cluster=customer cluster=standardizing-test worker=42' log='time="2020-01-27T16:05:28Z" level=error msg="warning: deleting cluster-scoped resources, cluster=customer cluster=standardizing-test worker=42'

ALERT ON CUSTOMER IMPACT

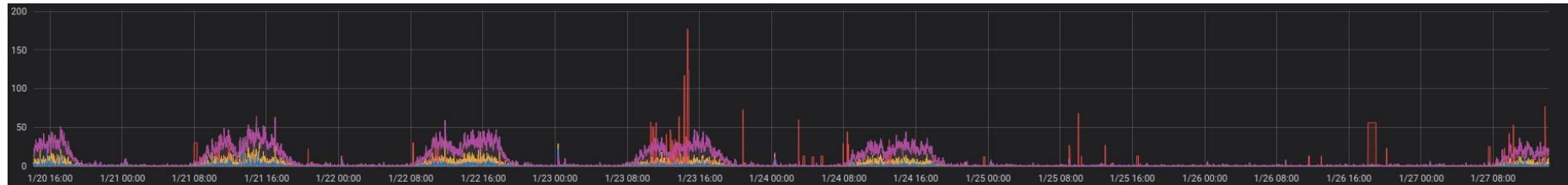
Alert: CDP: Builds waiting more than 5 minutes

Description

Whenever there is a build waiting more than 5 minutes to be started this alert fires.

Condition

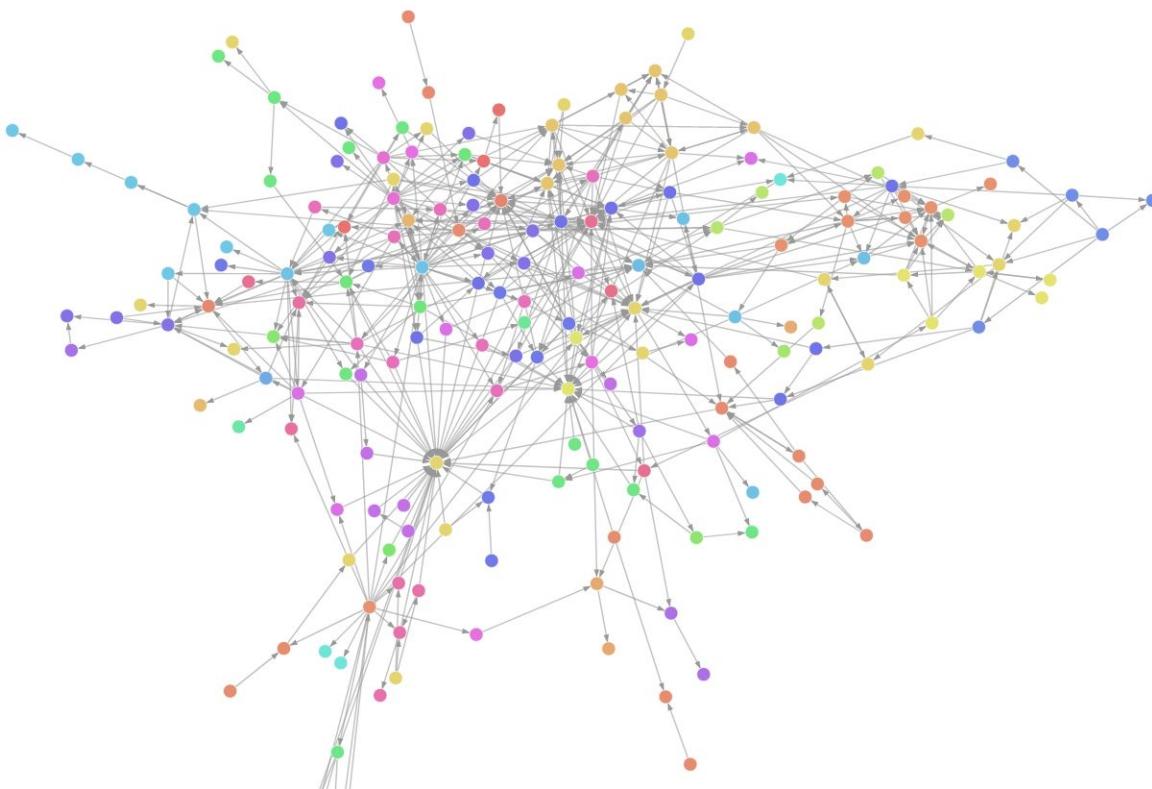
```
value['builds-waiting-for-more-than-5min'] > 0
```



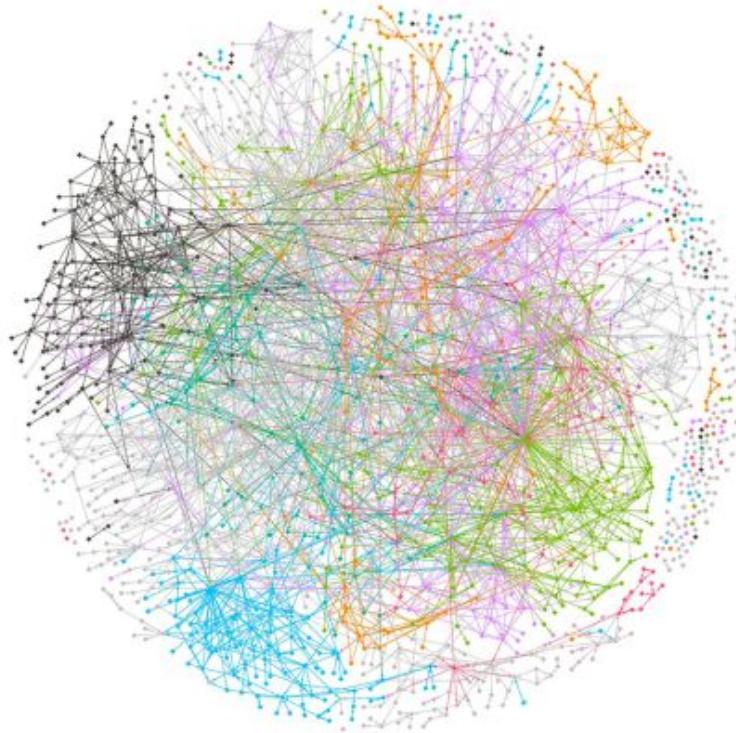
CHALLENGES

- Test coverage vs pipeline speed
- Maintaining test/staging environments
- Detecting (customer) impact quickly
- Database schema changes (DB evolutions)
- Keeping an overview of the microservice landscape

MICROSERVICES



MICROSERVICES



ZALANDO: DEPLOYMENTS/WEEK



- ~300
- ~3000
- ~7000
- ~10000

ZALANDO: DEPLOYMENTS/WEEK



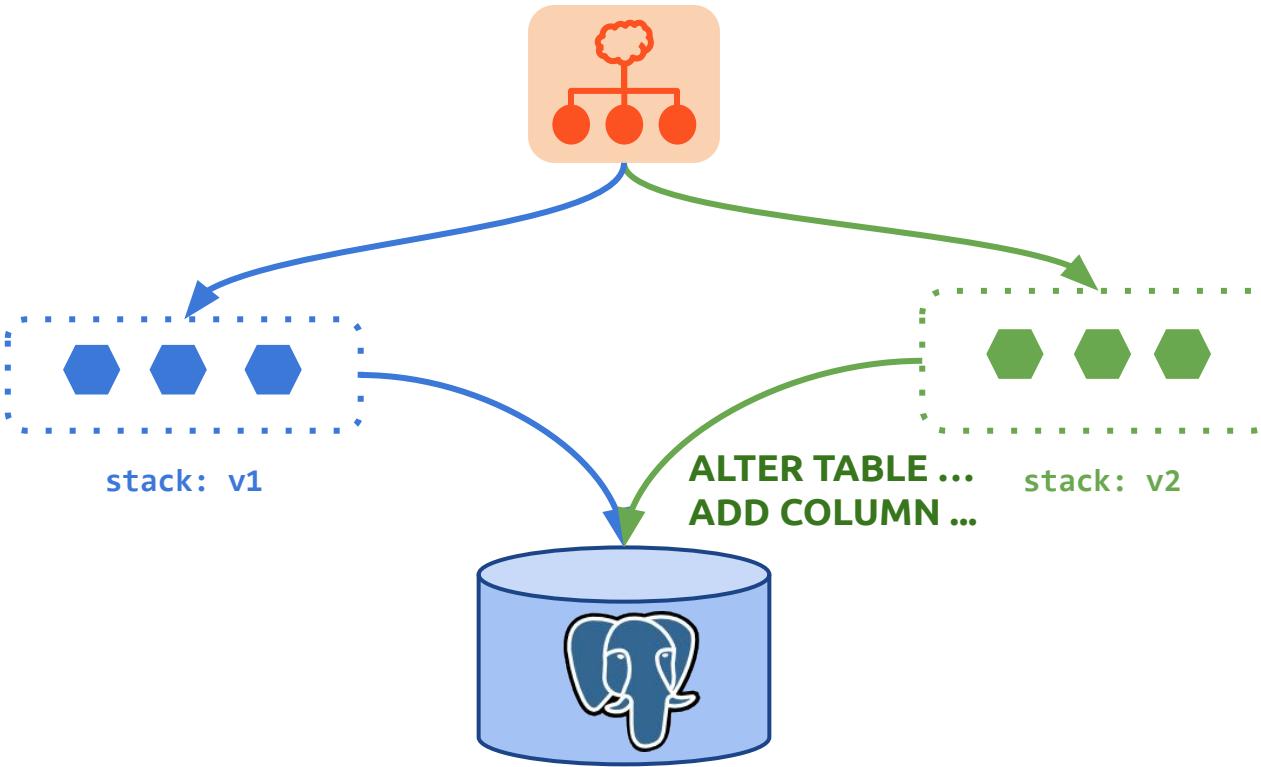
- ~300
- ~3000
- ~7000
- ~10000

CHALLENGES

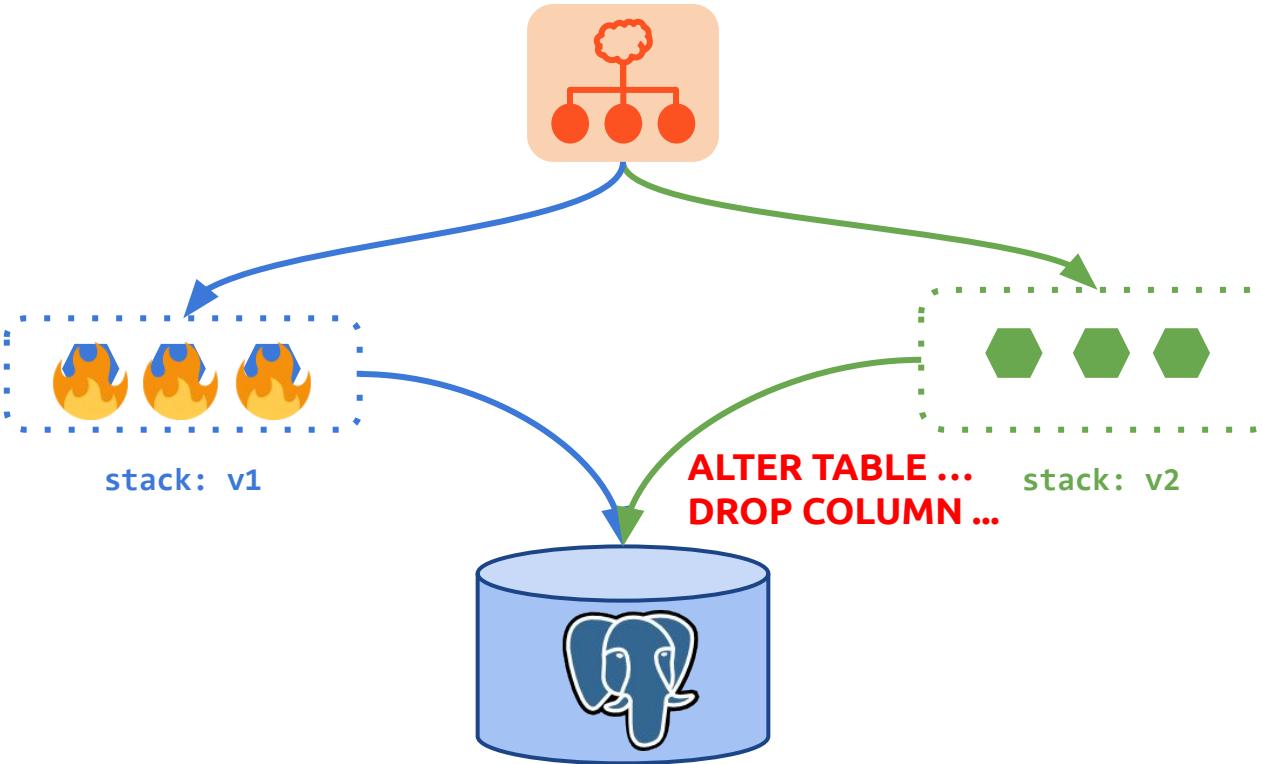
Database schema changes (DB evolutions)

"The outage was caused by a change in the database schema of the service which wasn't detected earlier in the testing and deployment process. It was resolved by switching to the software version that was compatible with the changed database schema."

ADD COLUMN: SAFE WITHOUT DOWNTIME



DROP COLUMN: OUTAGE!



OUTAGES CAUSED BY DEPLOYMENTS



- ❑ Systematic issues (e.g. lack of test coverage) should be addressed
- ❑ It's most important to find the accountable person and teach him/her how to do releases properly
- ❑ There is no single root cause for incidents, multiple contributing factors must be considered
- ❑ Outages should make the team do less deployments

OUTAGES CAUSED BY DEPLOYMENTS



- ❑ Systematic issues (e.g. lack of test coverage) should be addressed
- ❑ It's most important to find the accountable person and teach him/her how to do releases properly
- ❑ There is no single root cause for incidents, multiple contributing factors must be considered
- ❑ Outages should make the team do less deployments

SUMMARY



- **You build it, you run it!**
- Git with **short-lived branches**
- **Continuous Deployment** as default
- **Confidence** via automated tests, gradual rollouts
- **No trade-off** between speed and stability/quality!
- **4 key metrics:** Deployment Frequency, Lead Time, Change Fail Rate, MTTR

OPEN SOURCE & MORE

Open Source at Zalando
opensource.zalando.com

More Zalando Tech Talks
github.com/zalando/public-presentations





QUESTIONS?

HENNING JACOBS
SENIOR PRINCIPAL

henning@zalando.de
[@try_except](https://github.com/try_except)

Illustrations by [@01k](https://www.instagram.com/01k)

