



# Developer Experience at Zalando

---

HANDELSBLATT  
JAHRESTAGUNG

2019-01-23

HENNING JACOBS  
@try\_except\_



# ZALANDO AT A GLANCE

~ **4.5** billion EUR

revenue 2017

> 15.000

employees in  
Europe

> 70%

of visits via  
mobile devices

> 200  
million

visits  
per  
month

> 24

million  
active customers

> 300.000

product choices

~ 2.000

brands

17

countries

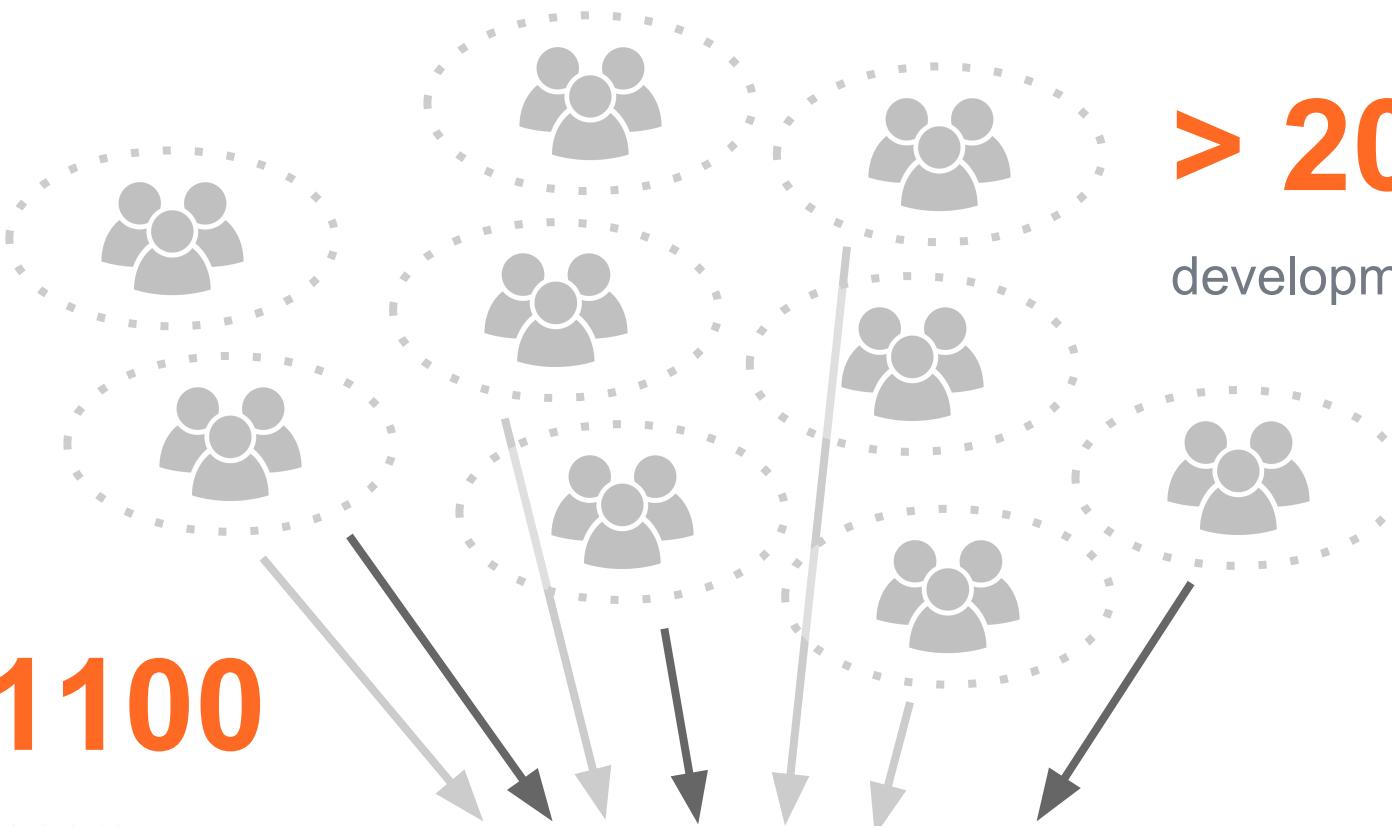
**> 1100**

developers

Platform

**> 200**

development teams



# 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



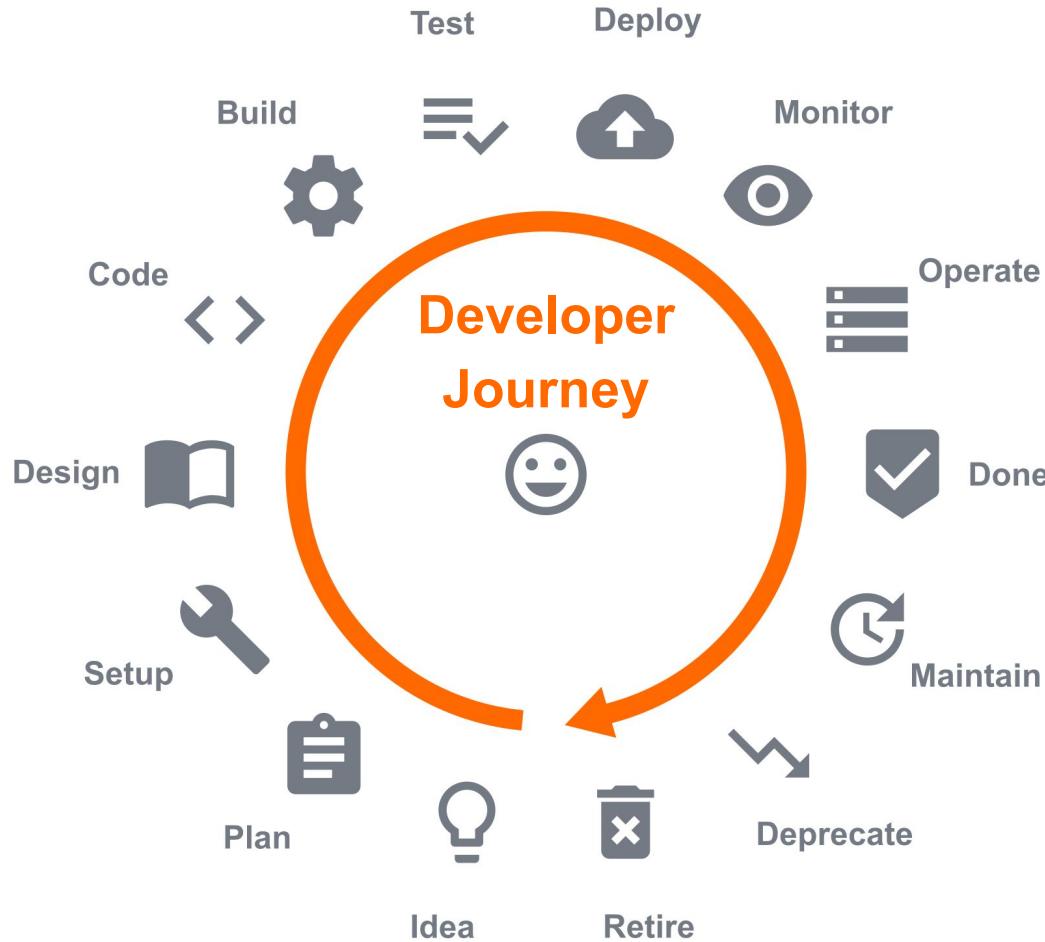
*When things are broken,  
we want people with the best  
context trying to fix things.*

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



## DEVELOPER JOURNEY

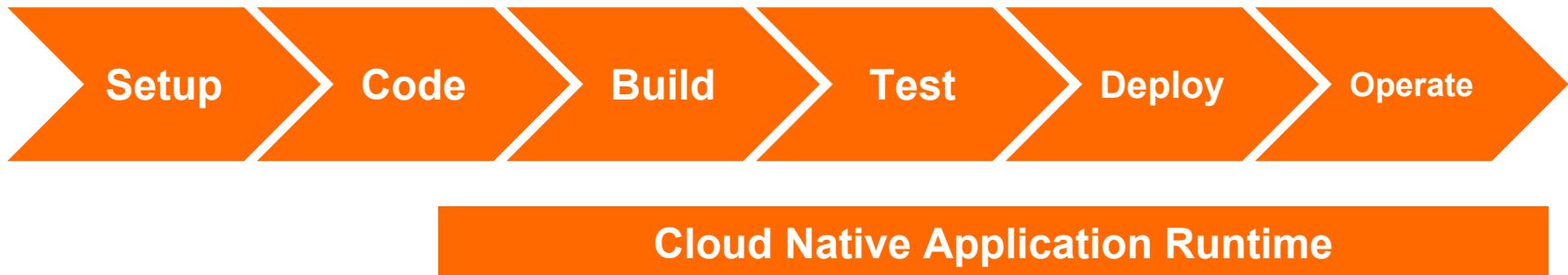
**Consistent story  
that models  
all aspects of SW dev**



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



# DEVELOPER PRODUCTIVITY



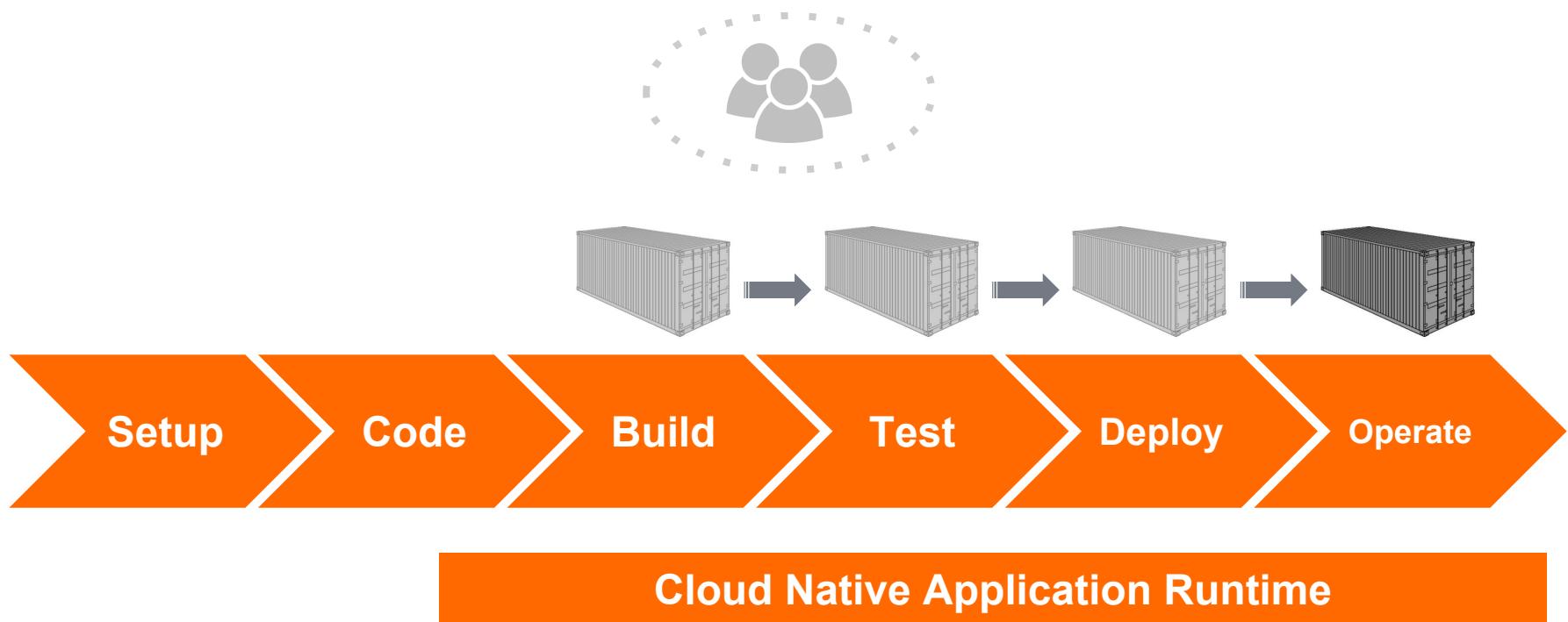
# CLOUD NATIVE

*.. 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.*

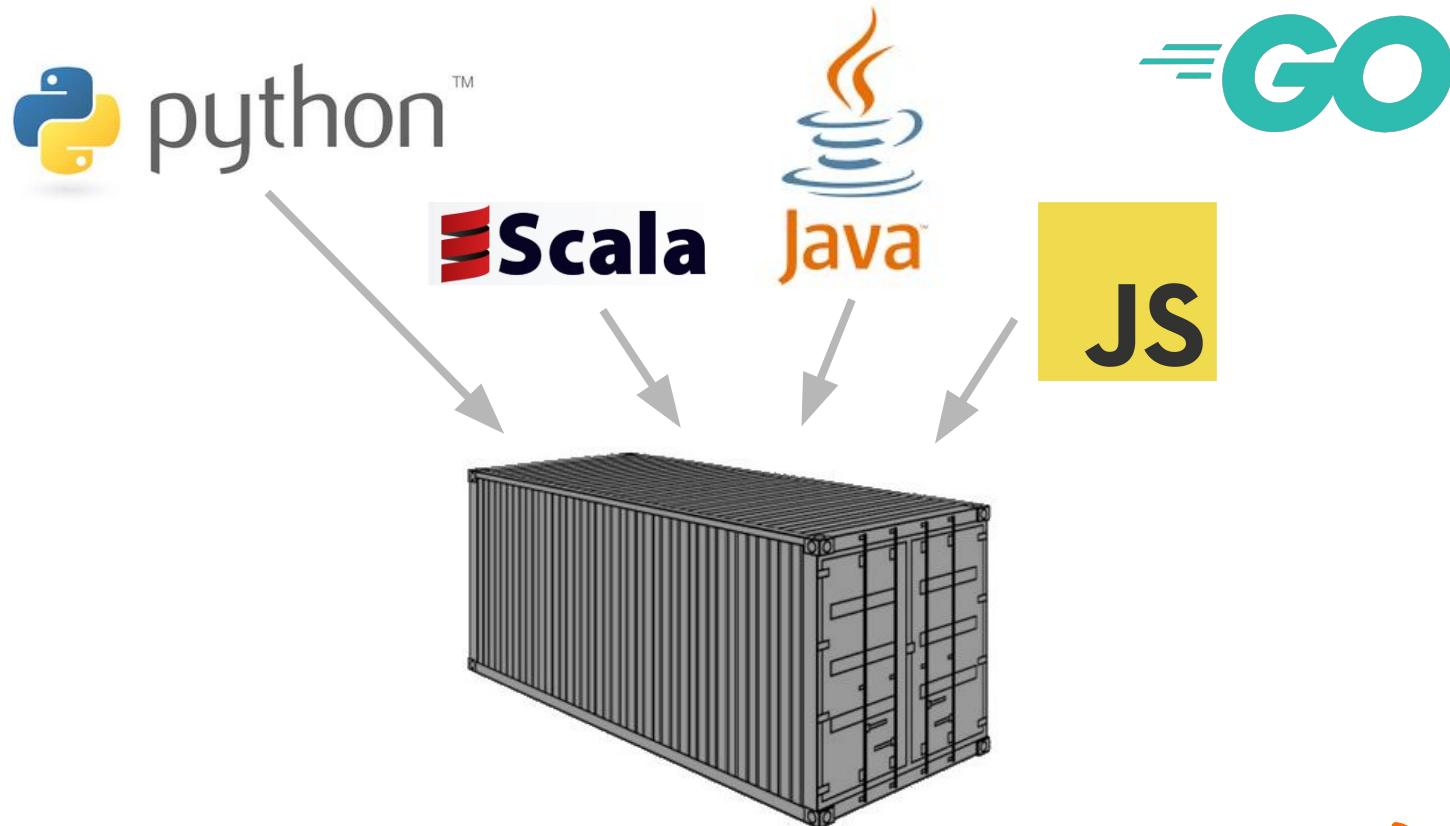
*Cloud native technologies enable software developers to **build great products faster**.*

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

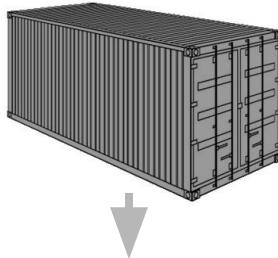
# CONTAINERS END-TO-END



# CONTAINERS



# CONTAINERS



# kubernetes



Google Cloud Platform



## Welcome to the Developer Journey

### APPLICATIONS

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 Idea 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

### TOOLS

#### Deployment

#### Documentation

#### IDE

#### Metrics

#### Monitoring

#### Operability

#### Setup

#### Testing

#### Versioning

#### Workflows

The diagram illustrates the 'The Developer's Journey' as a continuous cycle of 12 stages:

- Plan**: Represented by a clipboard icon.
- Design**: Represented by a book icon.
- Code**: Represented by a code editor icon.
- Test**: Represented by a checkmark icon. This stage is highlighted with an orange glow.
- Deploy**: Represented by a cloud icon.
- Monitor**: Represented by an eye icon.
- Operate**: Represented by a server icon.
- Done**: Represented by a checkmark icon.
- Maintain**: Represented by a clock icon.
- Deprecate**: Represented by a downward arrow icon.
- Retire**: Represented by a trash bin icon.
- Idea**: Represented by a lightbulb icon.

**Test Stage Description:**

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

**Learn more...**

- Rules of Play
- Tech Radar
- Confluence
- Nexus
- Database as a Service
- Open Source One-Stop-Shop
- Continuous Delivery Platform (CDP)

don't show me again close

**Design**

- RESTful API Guidelines
- API Portal
- Zally API Linter
- Fashion Content Platform (AI & Data APIs)

**Test**

**Setup**

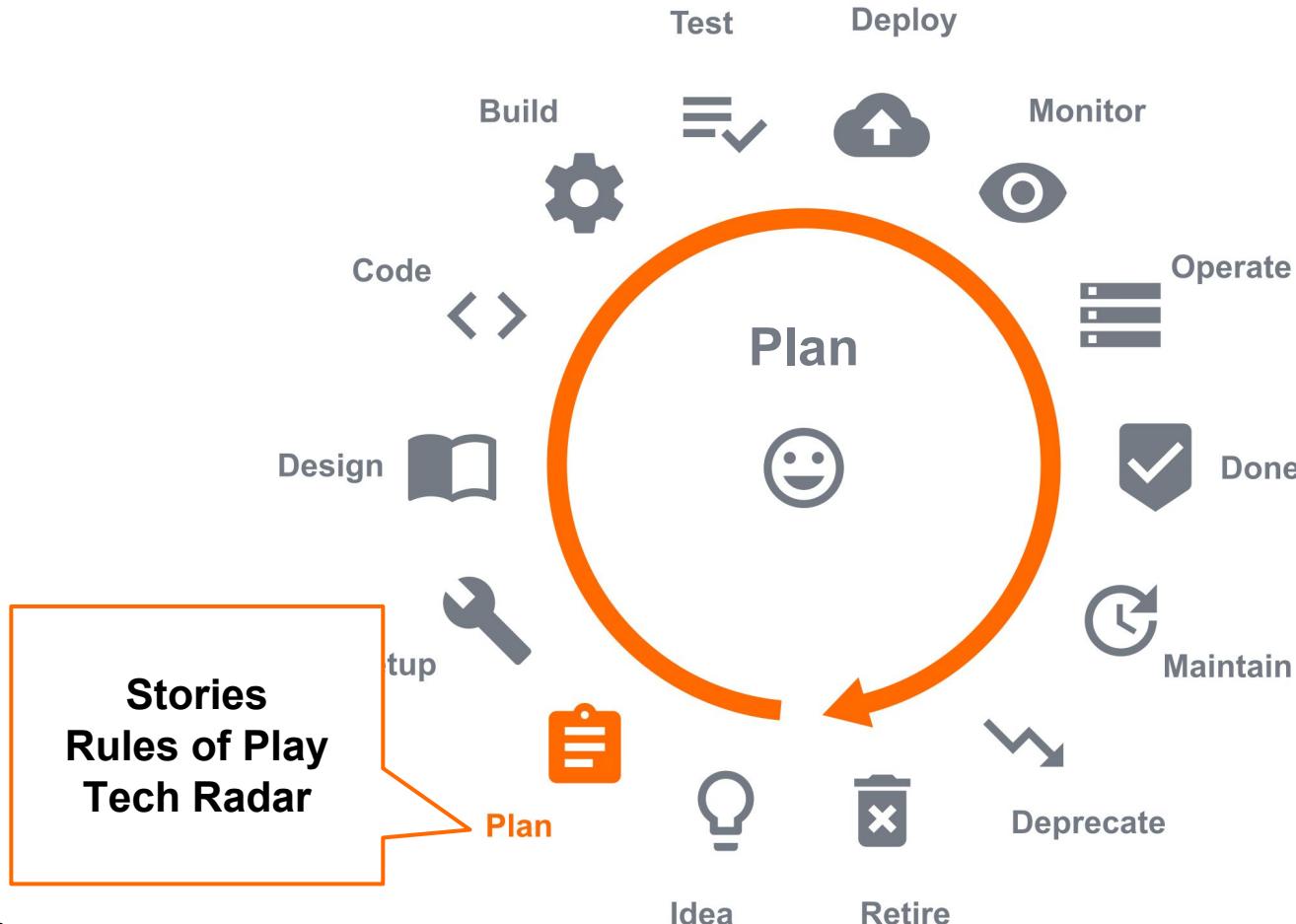
**Build**

**Setup**

- ZACK Role Management
- GitHub Enterprise
- Create Git Repository
- YOUR TURN
- Zappr
- Create database cluster
- AWS Account Chooser



## PLAN & SETUP



# Zalando Tech Radar — 2018.02

## Frameworks

ADOPT	ASSESS
1. Akka (Scala)	15. Akka-Http
2. Node.js	16. Aurelia
3. OpenAPI (Swagger)	17. Ember.js
4. Play (Scala)	18. gRPC
5. React.js	19. Http4s
6. RxJava (Android)	20. JOOQ
7. scikit-learn	21. Redux
8. Spring	22. Vert.x
	23. Vue.js

## TRIAL

HOLD
9. Angular
10. AspectJ
11. Camel
12. Camunda
13. OpenNLP
14. Thymeleaf
24. Activiti
25. AngularJS 1.x
26. BackboneJS
27. Drools
28. Spray

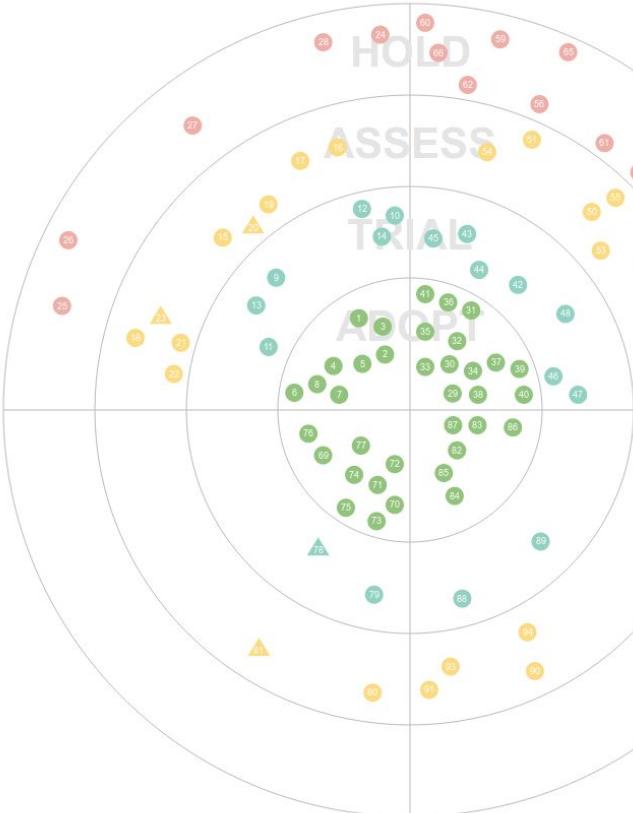
## Infrastructure

ADOPT	ASSESS
69. Docker	80. AWS Lambda
70. HAProxy	81. OpenTracing
71. Hystrix	
72. Jetty	
73. Kubernetes	
74. Nginx	
75. S3 UPS	
76. Tomcat	
77. ZMON	
	HOLD

## TRIAL

78. Failsafe
79. Undertow

▲ moved up ▼ moved down



# 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.

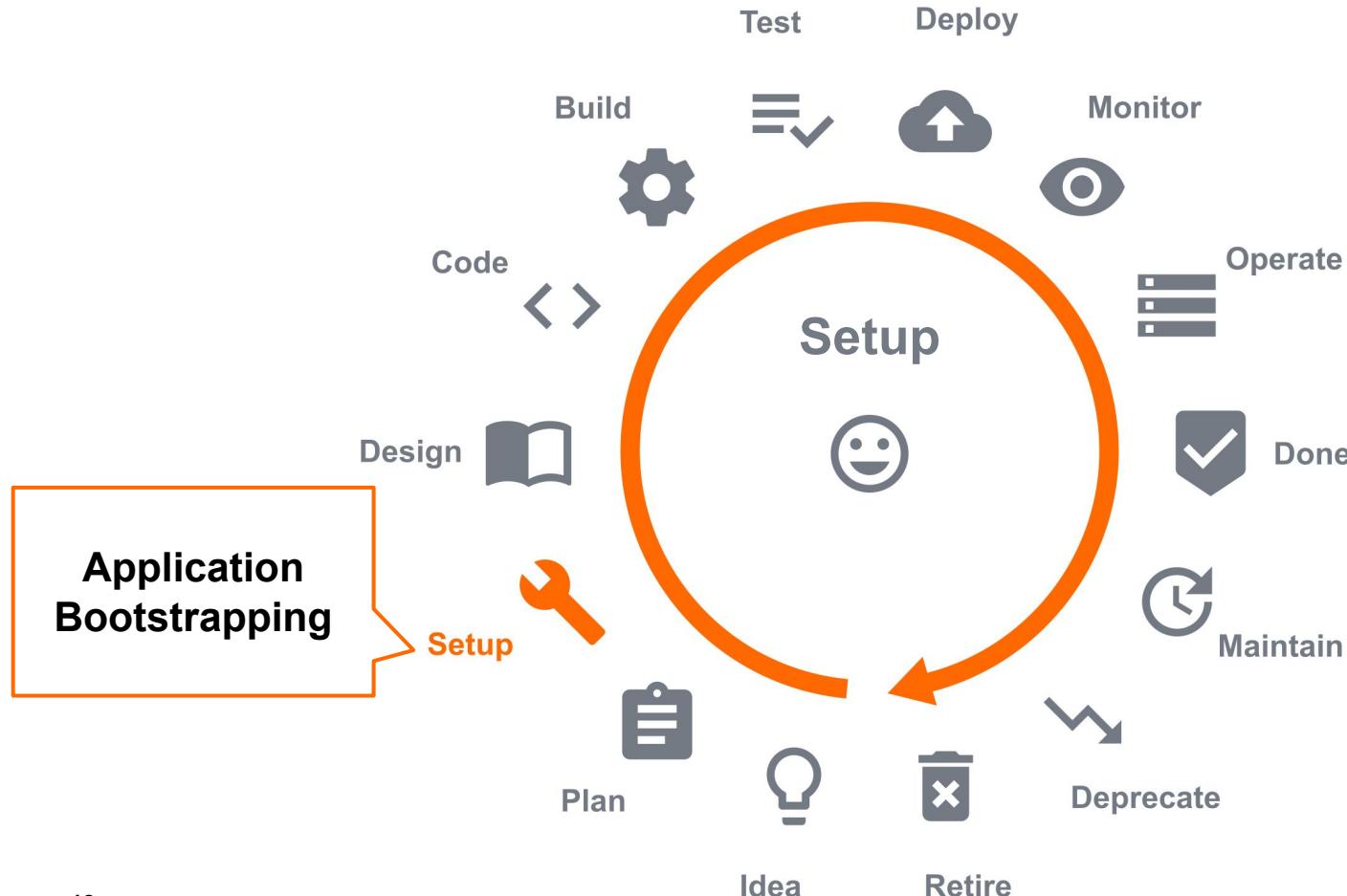
Nonetheless, it supports compilation to [JavaScript](#) and native code via LLVM. More than just compilation, Kotlin makes sure to integrate very well with the respective ecosystem, be it package managers or build tools. Plus, for each additional feature the designers deeply think about its implications 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

- Index
- Changelog
- Common choices
- Data Processing
- Databases
- Frameworks
- Infrastructure
- Languages
  - Go
  - Java
  - JavaScript
  - Python
  - Scala
  - Swift
  - Clojure
  - TypeScript
  - Elm
  - Haskell
  - **Kotlin**
  - R
  - Rust
  - C languages
  - CoffeeScript
  - Erlang
  - Groovy
  - .NET languages
  - Perl
  - PHP
  - Ruby
  - GraphQL
  - JRuby
  - Jython



# Create a new repository

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

hackweek

/ myapp

Description (optional)

Team

Code

Bootstrap: web-java-spring-boot-rest

x

An application bootstrap skeleton will be automatically created based on your stack preference. [Learn more](#)

Add .gitignore: choose template

Create repository

Branch: master ▾

New pull request

Create new file

Upload files

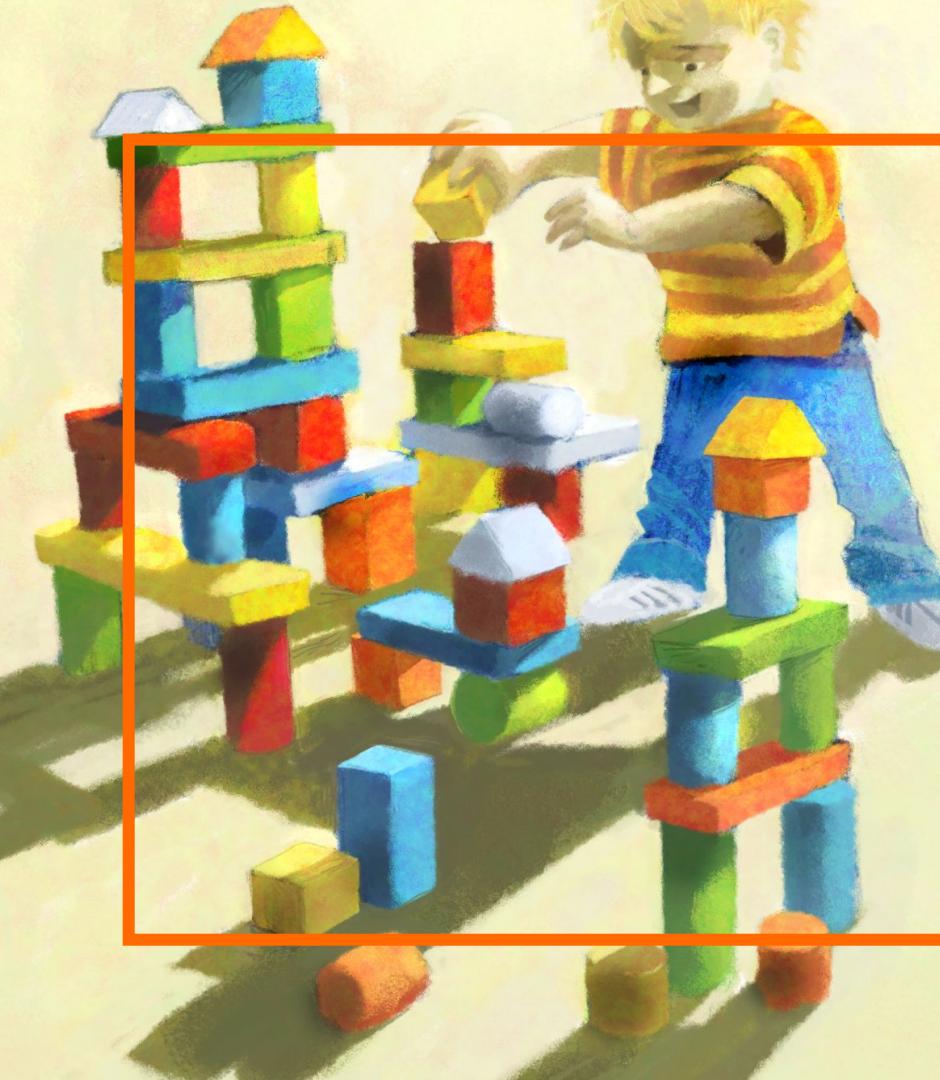
Find file

Clone or download ▾

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

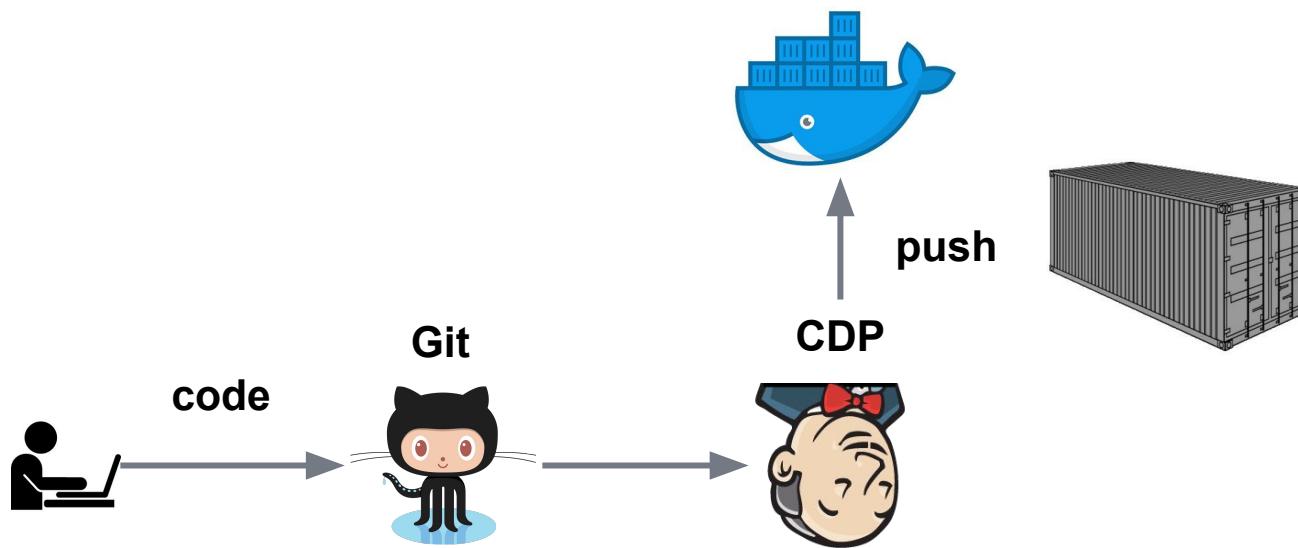
## Java - Spring Boot - REST

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](#)

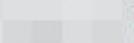
1 pending and 5 successful checks

-  **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.

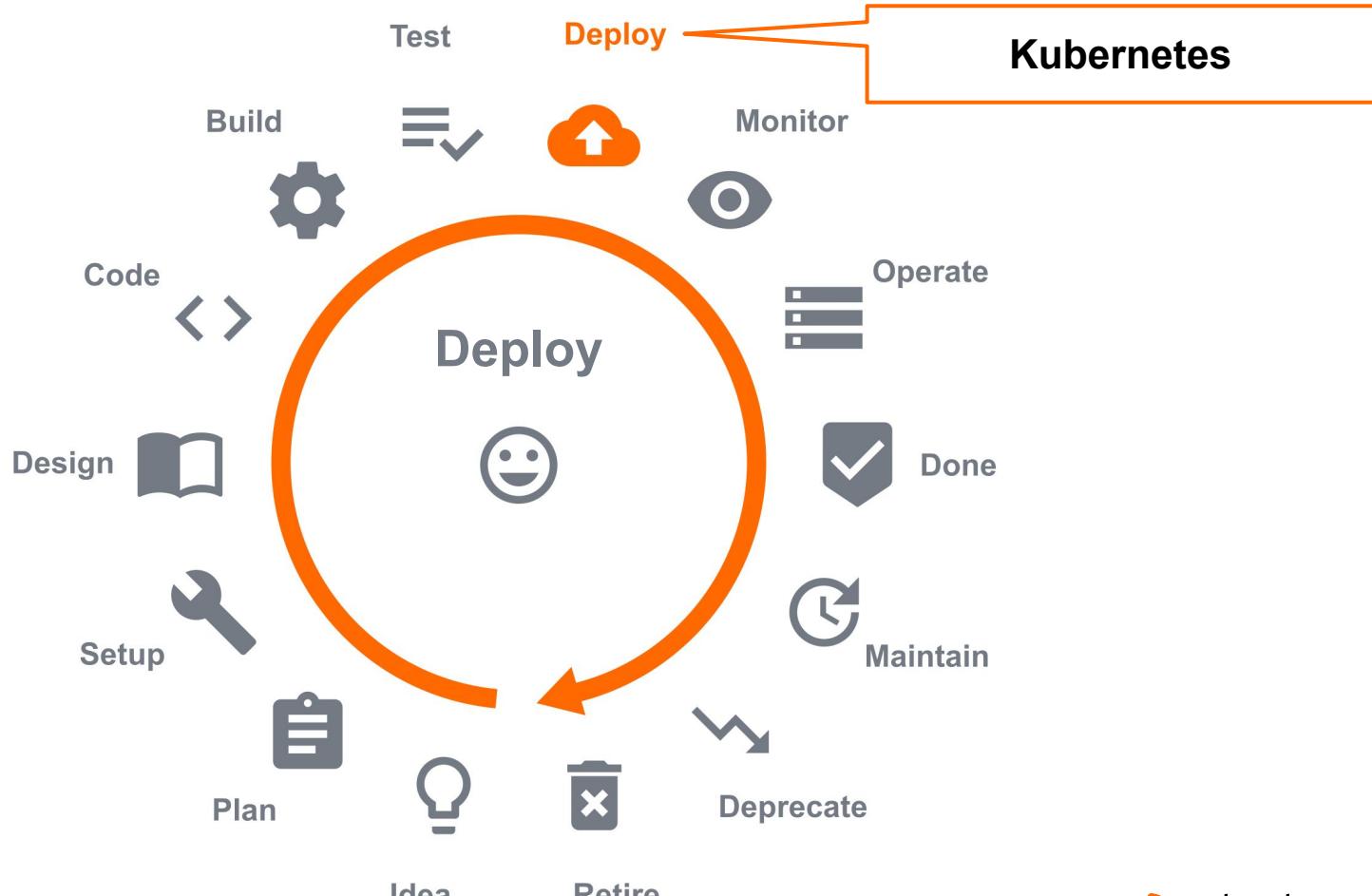
[Squash and merge](#)



or view [command line instructions](#).



# 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
```



# CONTINUOUS DELIVERY PLATFORM

Developer Console  Search Feedback  Support 

 DEPLOYMENT UNITS  RENDERING-ENGINE  Exclude PRs: 

Build Version	Started	Pipeline Runs
master-1090	18h ago	 B...  PE...  DE...  TE...  PE...  DE...  TE...  PE...  P...  P...
master-1089	21h ago	 B...  PE...  DE...  TE...  PE...  DE...  TE...  DE...  P...  P...
master-1088	1d ago	 B...  PE...  DE...  TE...  PE...  DE...  TE...  DE...  P...  P...
master-1087	4d ago	 B...  PE...  DE...  TE...  PE...  DE...  TE...  DE...  P...  P...
master-1086	4d ago	 B...  PE...  DE...  TE...  PE...  DE...  TE...  DE...  P...  P...

# CDP: DEPLOY

The screenshot shows the Zalando CDP Deployment interface. At the top, there are three tabs: "DEPLOYMENT UNITS" (selected), "TRACKING-DEPLOY" (disabled), and "MASTER-91" (disabled). Below the tabs, the deployment status is shown for "TEST", "STAGING", and "PROD" environments. The "TEST" environment is highlighted in green. A summary bar indicates "Ran for 1 m, 16 s".

The main content area displays deployment details for the "merchant-parcels" application across various components:

- 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**

On the right side, there is a feedback form with the heading "Please give us your feedback". It asks "Do you like the new deployment status ?" with "Yes" and "No" radio buttons. There is also a text area for "What could be improved upon?" and a "Submit" button.

# CDP: OPTIONAL APPROVAL

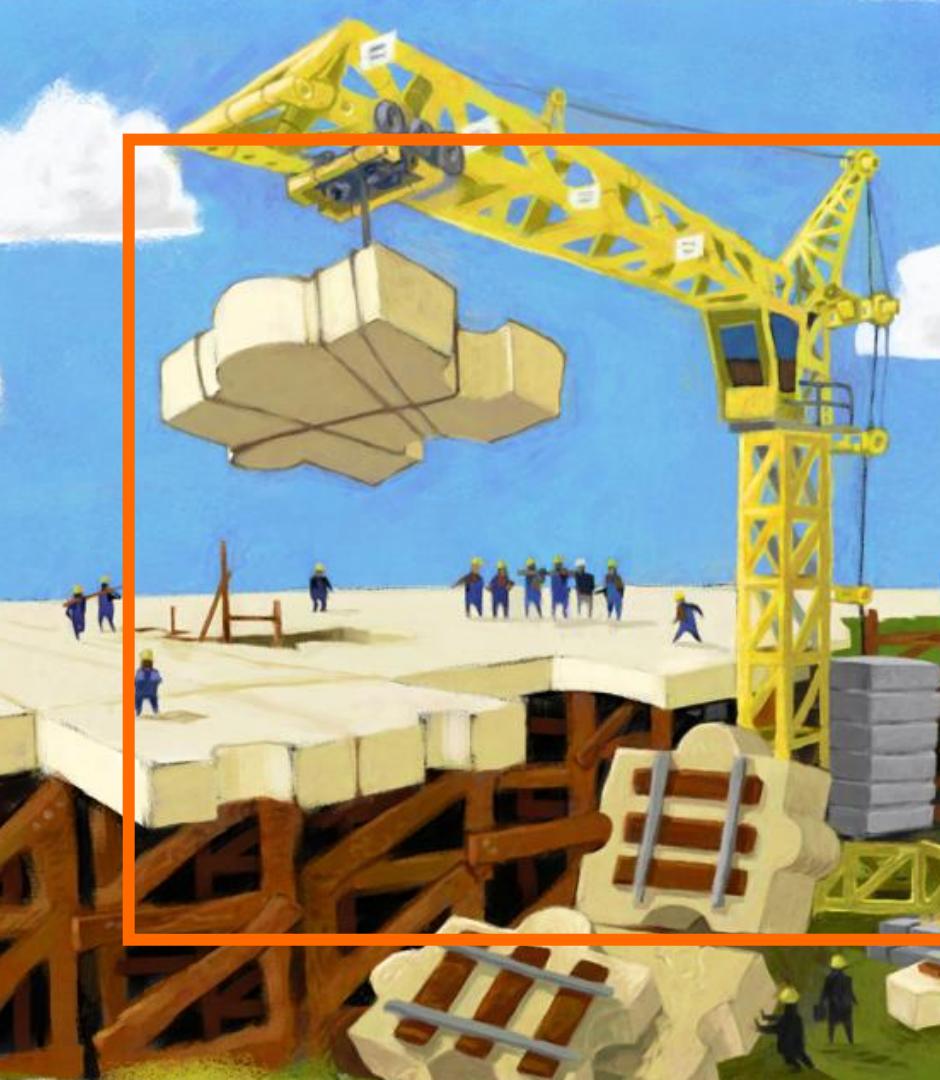
The screenshot shows the Zalando Developer Console interface. At the top, there's a navigation bar with 'Developer Console' on the left, a search bar in the center, and 'Feedback' and 'Support' links on the right. Below the navigation bar, there are tabs for 'DEPLOYMENT UNITS', 'FEED-SERVICE', and 'MASTER-16'.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

The main content area has several sections:

- GENERAL** and **LOGS** tabs are visible at the top of a grey sidebar.
- ACTIONS** section contains two buttons: 'APPROVE' (with a green thumbs-up icon) and 'REJECT' (with a red thumbs-down icon).
- DETAILS** table:

Id	n/a
Type	process
Status	PENDING_APPROVAL
Deployment	deploy-prod
Cluster	[redacted]
Started at	n/a
Finished at	n/a
- ERRORS & WARNINGS** section:

Error	No Error
Warning	No Warnings



# INTEGRATIONS

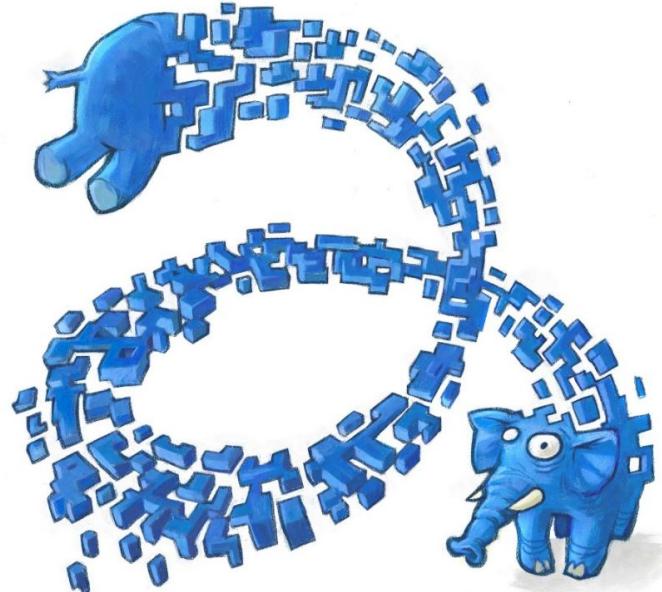


# CLOUD FORMATION VIA CI/CD

*"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
```

# POSTGRES OPERATOR



Application to manage  
PostgreSQL clusters on  
Kubernetes

**>500**

clusters running  
on Kubernetes

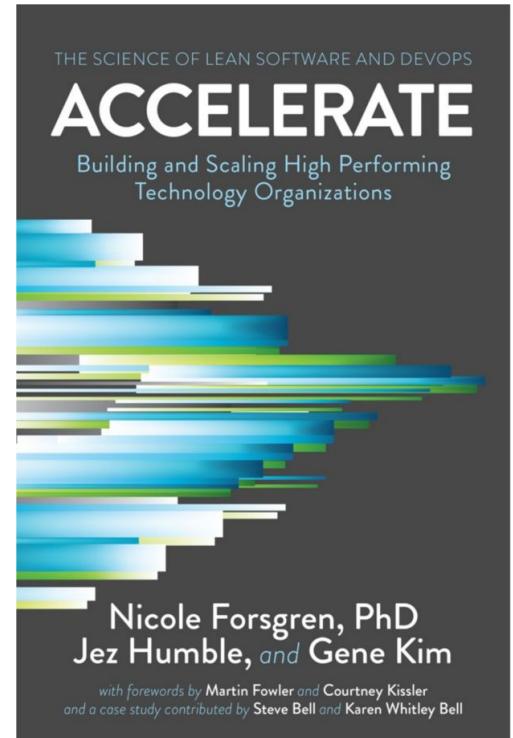
# SUMMARY

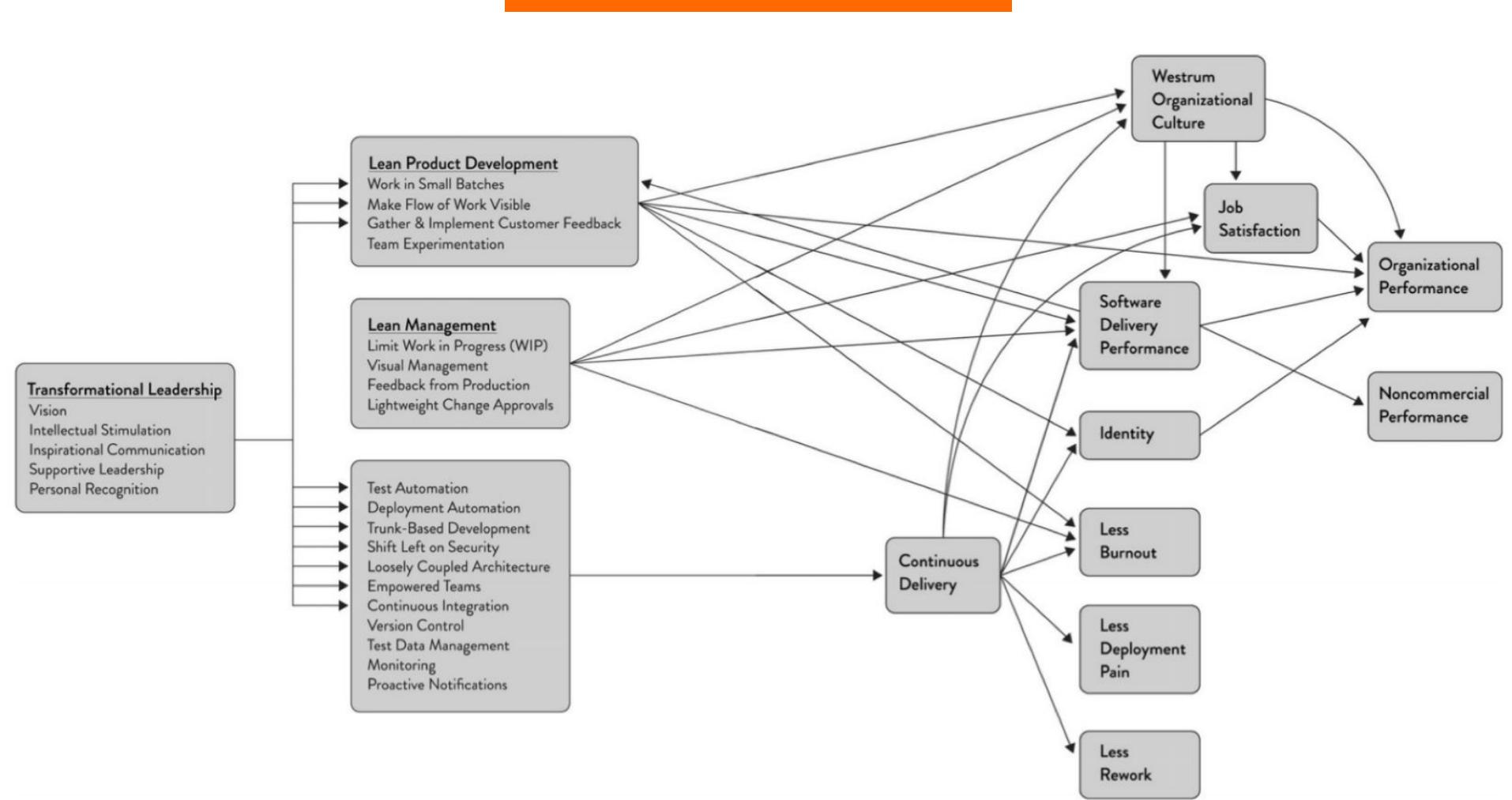
- Application **Bootstrapping**
- **Git** as source of truth and UI
- **4-eyes** principle for master/production
- Extensible **Kubernetes API** as primary interface
  - OAuth/IAM credentials
  - PostgreSQL
- **CloudFormation** for proprietary AWS services



# 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**

## Applying product management to internal platforms

TRIAL

We've s  
to roll o  
access  
find tha  
Applying  
develop  
the pla  
brand  
manag  
the pla  
underu

NEW

*“.. 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*

oking  
vice  
We  
ng.  
read:  
nsure  
e a  
ct  
reating  
gnant,

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



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

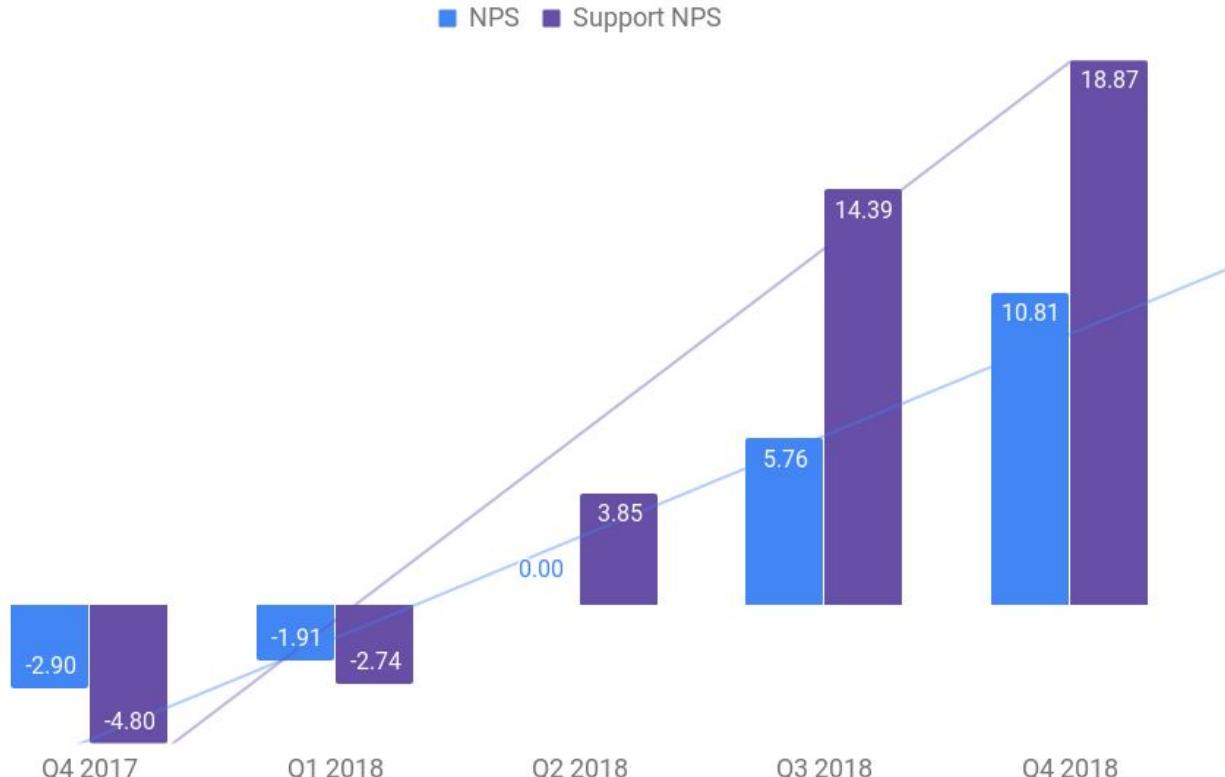


Extremely  
Satisfied

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"*

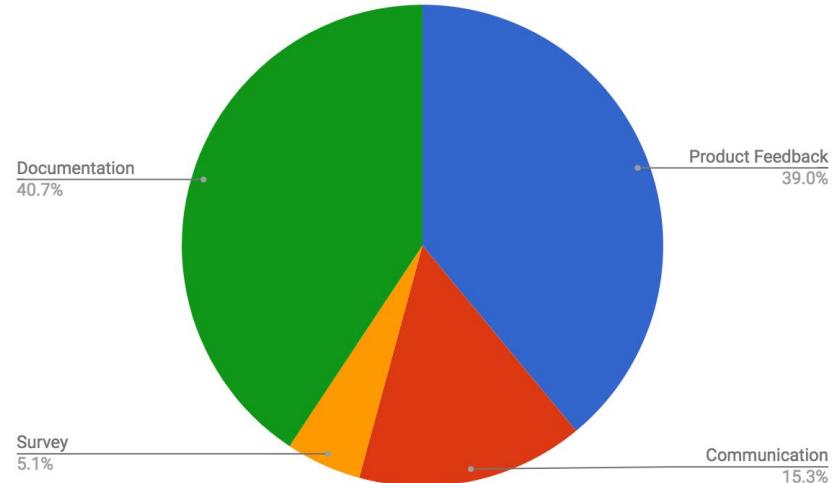
*"Documentation is not comprehensive enough"*

*"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

**PUBLISH****DISCOVERABLE****DOCUMENTATION****Learn more****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  
\\zalan.do Delete a database cluster: zkubectl get postgresql zkubectl delete postgresql

**Rotate PostgreSQL Credentials – Discount Service documentation**[/operations/guides/rotate\\_postgresql\\_keys.html](/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

# NEWSLETTER

"You can now.."

- You can now benefit from the most recent Kubernetes 1.12 features, e.g. ..
- You can now analyse your Kotlin project with SonarQube and upload your Scala code coverage report to SonarQube

# SIGNAL: ISSUE UPVOTES

Uploading things to S3 from CDP builds #63

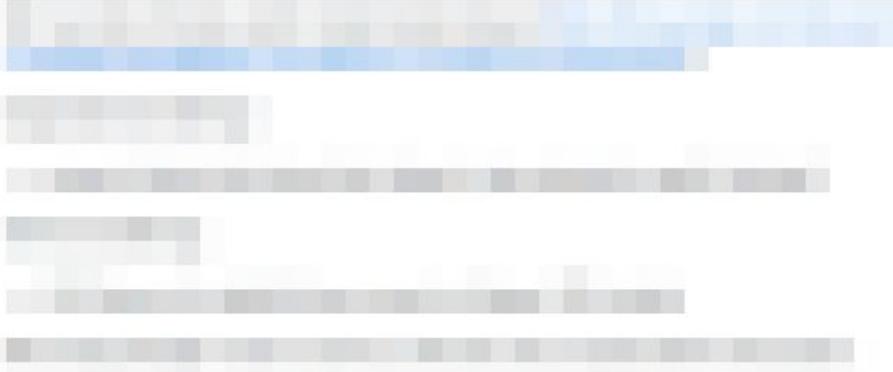
 Closed

[REDACTED] opened this issue on Feb 15, 2018 · 34 comments

commented on Feb 15, 2018 · edited

Member + ...

Users regularly ask to upload build artifacts to S3 from CDP builds (to their teams' S3 buckets).



38 1

# TESTIMONIALS

*“Useful information, good level of details and pleasant to read. It's one of the few newsletters that I took time to read entirely :)"*

*- a reader, July 2018*

# TESTIMONIALS

*“So, thank you, Team Automata, for listening to our community, taking our upvotes in consideration when developing new solutions and building every day  
**'the first CI that doesn't suck'.**”*

*- a user, October 2018*



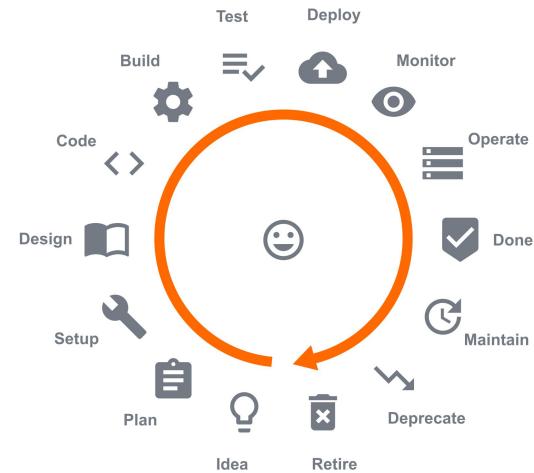
# LEARNINGS FROM THE PAST

# LEARNINGS FROM THE PAST

- Platform as a service vs **governance**
- Platform teams think they are "special",  
but **lack basic customer insights**
- Developers love **autonomy**,  
hard to "go back" once granted
- Developers are **demanding**

# NOTE TO SELF

- Establish a **product mindset**
- **Listen** to your customers
- Exploit **unfair advantages**
- Invest into **docs & support**





# QUESTIONS?

---

HENNING JACOBS  
HEAD OF  
DEVELOPER PRODUCTIVITY

[henning@zalando.de](mailto:henning@zalando.de)

[@try\\_except](https://github.com/try_except)

Illustrations by [@01k](#)

