

APEC: How to ensure your application is operated properly?

DevOps Finland meetup 10.10.2023

Uri Savelchev





Agenda

1. The problem and a possible solution
2. Goals we want to achieve
3. What's inside the checklist?
4. APEC History in Zalando
5. Learnings
6. Q & A





Zalando Tech

- **2500+ software engineers**
- **300+ engineering teams**
- **200+ K8s clusters**
- **3500 active applications**

How to maintain operation excellence?



Zalando Death Star

Zalando Micro-Service architecture diagram ~2019 (aka. "Death Star")



Solution: a checklist?

Application Production Engineering Checklist (APEC) is a set of checks that teams should have in place for all applications.

The items on APEC have been identified from system reviews, post-mortem documents, existing production systems, and Zalando requirements.

Together the items on the checklist form a shared understanding of what production readiness is for teams shipping to production.



Outcome is a status (GREEN, AMBER, RED) and a list of what's missing





Operational excellence metrics

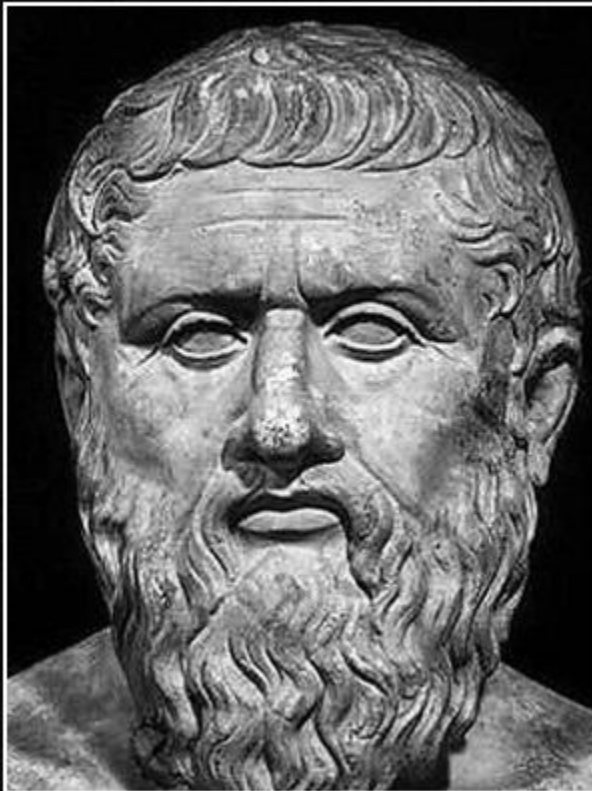
- per team
- per department
- per organization

Teams	Apps	K8s	All Risks	Pending Triage ¹	Valid ¹	In Progress ¹	Closed ¹	CW Apps ²	Needing Review ³	R ⁴	A ⁴	G ⁴
23	75	43	38	3	6	18	11	41	0	0	1	36
17	118	71	24	1	12	5	6	57	0	2	12	41
16	115	82	49	0	12	12	25	43	2	2	1	36
24	191	108	27	0	8	6	13	104	0	7	10	84
4	41	31	9	0	1	2	3	15	1	2	7	5



A tool to find gaps in operations

1. The APEC status
2. A summary of what's missing
3. The review process itself: answering questions helps you to look at your application from a different angle.



The right question is usually more
important than the right answer.

— *Plato* —

AZ QUOTES



APEC is NOT about compliance

- Security,
- Copyright,
- GDPR,
- ...

need separate tools and processes.



What's inside?





APEC Sections (non-complete)

- Application registry information
- Technical stack (and if it is included in the Tech Radar)
- Tier evaluation
- Naming and Code Of Conduct
- Data Grade
- API specification
- Service Level Objectives
- Monitoring and tracing
- Playbooks
- Production Readiness Review





Zalando Tech Radar

2023.02

Databases

ADOPT

1. AWS DynamoDB
2. AWS S3
3. Elasticsearch
4. etcd
5. Exasol
6. PostgreSQL
7. Redis

TRIAL

8. Amazon Feature Store
9. Amazon Redshift
10. Druid
11. HDFS

ASSESS

12. Amazon MemoryDB
13. RocksDB

HOLD

14. Aerospike
15. Apache Cassandra
16. Consul
17. CouchBase
18. Hazelcast
19. HBase
20. Memcached
21. MongoDB
22. MySQL
23. Oracle DB
24. Solr
25. ZooKeeper

Infrastructure

ADOPT

45. Amazon SageMaker
46. AWS CloudFormation
47. AWS CloudFront
48. Docker
49. Kubernetes
50. OpenTracing
51. Skipper

TRIAL

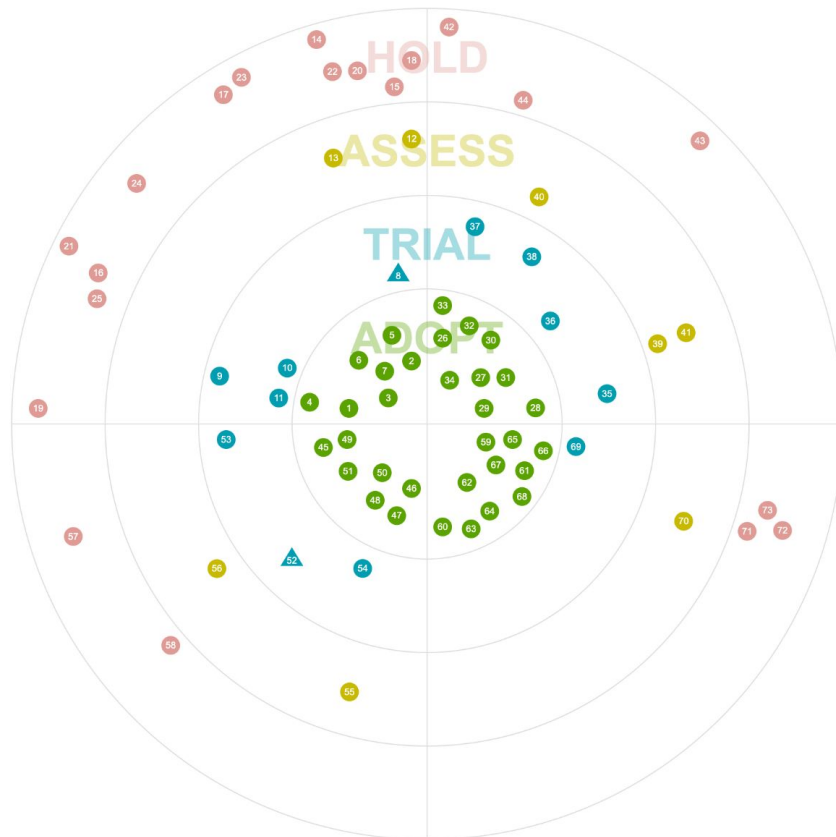
52. AWS Elemental MediaConvert
53. AWS Lambda
54. AWS Step Functions

ASSESS

55. Slurm
56. WebAssembly

HOLD

57. STUPS
58. ZMON



Data Management

ADOPT

26. Airflow
27. AWS EMR
28. AWS Kinesis
29. AWS SNS
30. AWS SQS
31. Databricks
32. Kafka
33. Nakadi
34. Spark

TRIAL

35. Flink
36. Google BigQuery
37. Presto
38. RabbitMQ

ASSESS

39. AWS Athena
40. AWS Glue
41. dbt

HOLD

42. AWS Data Pipeline
43. Hadoop
44. YARN

Languages

ADOPT

59. Go
60. GraphQL
61. Java
62. JavaScript
63. Kotlin
64. OpenAPI (Swagger)
65. Python
66. Scala
67. Swift
68. TypeScript

TRIAL

69. Dart

ASSESS

70. R

HOLD

71. Clojure
72. Haskell
73. Rust

▲ moved up ▼ moved down

<https://opensource.zalando.com/tech-radar/>





Tier evaluation

A way to classify the apps
by their business criticality

Is the application used in production? Does it have access to any other production system(s) or environment and/or stores production data?

Is the application in the direct path of a **critical business operation** [?](#)

Does the application have a fallback strategy to handle errors in dependencies?

Does the application have a retry strategy to handle errors in dependencies?

Do all clients of the application have a fallback strategy?

Do all clients of the application have a retry strategy?

Can the application cause a SEV2 or SEV1 incident (by failing/ being unavailable)?

Will a failure of the application cause a drop in the **sales orders** [?](#) curve?

Will a failure of the application lead to GMV/revenue loss (in the same hour, or over days)?

Will a failure of the application prevent employees from doing their work?

Can a failure of the application lead to contractual penalties/fees?

Will a failure of the application directly affect the customer experience?

Can the application be down for 30 minutes without informing customers/stakeholders?

Is a failure of the application reducing overall redundancy or observability?

Can a failure affect the ability to solve incidents or prolong time-to-repair?


Current criticality (value in Kio)

Tier 3 [?](#)

Calculated criticality

Tier 3 — long-term or low impact to the business,




 API Portal


2037 APIs

NEWSDEVELOPER CONSOLE TEAMAPI GUILDFEEDBACK

Discover great APIs.


The central hub for all APIs in the Zalando universe!

Search for an API




Get started

Short introduction to API Portal.




Publish API

How to publish API specifications.




Improve API design

API Guidelines - rules for high quality APIs.




Lint API online

Use Zally Web UI for checking compliance with guidelines.




Get feedback for API

Feedback quality assurance process for API design.




Register functional component

Central registry for functional names.




Lint API in IntelliJ

Use IntelliJ Plugin for checking compliance with guidelines.




Monitor API Endpoints

Endpoint monitoring of #requests, latencies, and response codes.



Monitor API Clients

Client tracking for your published API.



Monitor Resource Usage

CPU, memory, and network traffic metrics for your containers.



Service Level Objectives

API Specification is a contract,
SLOs are an important appendix to the contract.

```
1  apiVersion: zalando.org/v1alpha1
2  kind: ServiceLevelObjective
3  metadata:
4    name: ...
5  spec:
6    notifications:
7      - channel: email
8        email_address: ...
9      - channel: google_chat
10        webhook_url: https://chat.googleapis.com/v1/spaces/....
11    components:
12      - api_id: db558838-8e19-410b-88ae-a623ce648850
13        name: delivery-api
14        type: http_api
15        paths:
16          /graphql:
17            post:
18              slos:
19                - sli: latency
20                  percentile: p99
21                  max: 150
22                - sli: availability
23                  percentage: 99
```



Monitoring, tracing, playbooks

- Every application running in production must have:
 - A dashboard with metrics
 - A reasonable set of alerts
- Every application running in production *should* be instrumented with tracing (OpenTracing / OpenTelemetry)
- Every application being supported 24x7 must have reliable playbooks



100% ▾ | Normal text ▾ | Roboto ▾ | - 10.5 + | **B** *I* U A |

My wonderful app

Production Readiness Review

Updated: ...
Status: ...
Engineering Team: ...
Contributors: ...

- [Context & Background](#)
- [Production Operations](#)
- [API Traffic & Request Handling](#)
- [Observability](#)
- [Engineering](#)
- [Data Management](#)
- [Release](#)
- [Recommended Action Items](#)
- [References](#)

APEC History





APEC History

1. A plain document
 - Proof of Concept
 - Small set of teams and application to try the approach
2. A very long form
 - Google-form style
 - Painful to fill out
 - A typical review took 20-30 minutes for every app
3. More compact form with expandable sections
 - Autodiscovery (API specs, tracing, monitoring, playbooks,...)
 - Memoization — no need to repeat the same answer twice
 - A repeated review takes 1-5 minutes usually

**OpenTracing**

LightStep

Found 1 service(s): [business-platform-fsa](#) (last seen: 2023-10-09T16:50:38.717873Z)

Congratulations, your application seems to be instrumented with OpenTracing and is reporting to LightStep!

☒ I acknowledge the OpenTracing information and have tickets in my team's backlog to remediate outstanding issues.☐ Not applicable (requires rationale below)**API**

REST APIs

*This application seems to not have published a REST API. See [documentation on how to publish API specifications](#).*If your application provides a REST API, it SHOULD be published on [apis.zalando.net](#).☒ I acknowledge the API information and have tickets in my team's backlog to remediate outstanding issues.☐ Not applicable (requires rationale below)**Tech Stack**

Platforms/Languages

TypeScript ?

Congratulations, your application seems to only use ADOPT technologies from [the Zalando Tech Radar](#)!☒ I acknowledge this information and confirm that the tech stack is aligned with my department/BU.☐ Not applicable (requires rationale below)



Backstage integration

1 Application metadata*

2 Naming and inclusive language*

3 Criticality assessment*

4 Data classification*

5 Deployment information*

6 OpenTracing*

7 API*

8 Tech Stack*

9 Dashboards*

10 Service Level Objectives (SLOs)*

11 Alerts*

12 Playbooks*

13 Load tests*

LightStep

Found 1 service:

- business-platform-fsa (Last seen: 9.10.2023)

✓

Congratulations, your application seems to be instrumented with OpenTracing and is reporting to LightStep!

I acknowledge the OpenTracing information and have tickets in my team's backlog to remediate outstanding issues.

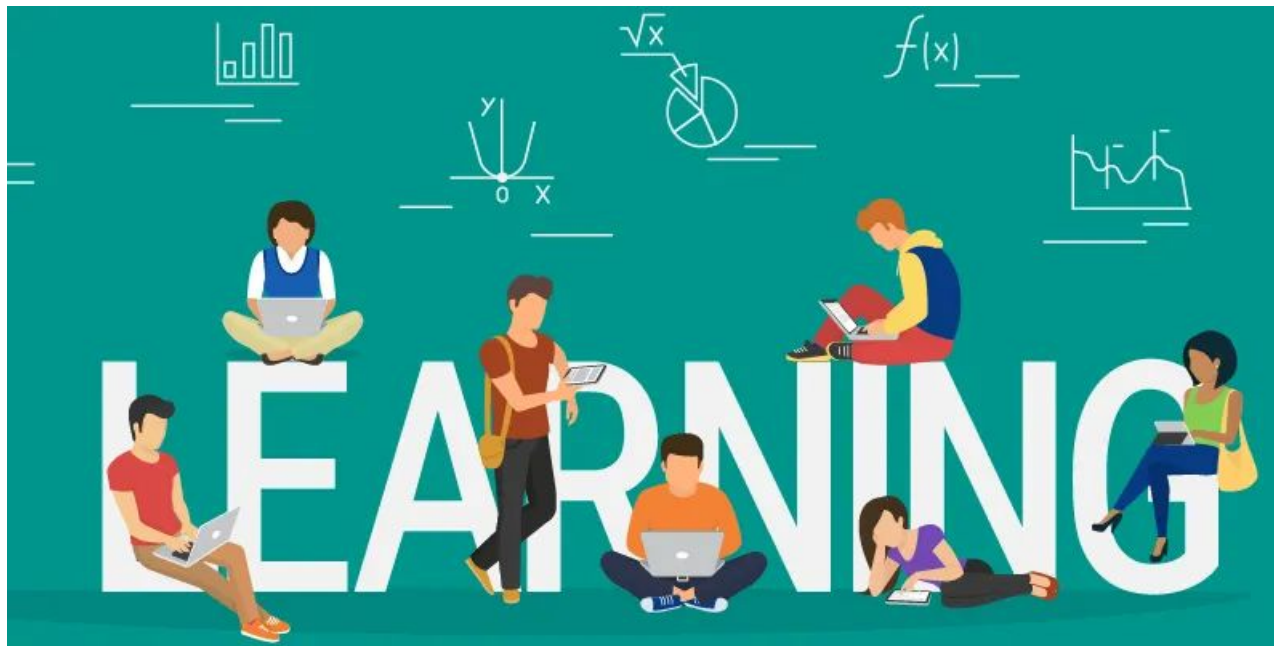
☐ Yes, this is correct.

☐ This is not applicable for this application.

↑ PREVIOUS

↺ RESET

SAVE AND CONTINUE





Learnings

- Engineers: “filling forms is a bureaucracy”
 - The process should be as fast/painless as possible
 - ❑ Memoization (DRY principle)
 - ❑ Autodetect and autofill everything you can
- There are no rules without exceptions
 - Let people to use them
- Comments are important!
(even though they aren't processed automatically)
- The “color” of the APEC should be defined by most critical things



Learn more

- [A nice presentation of Heinrich Hartmann at DevOpsCon Berlin in 2022](#)
- [Henning Jacobs' blog post](#)
- [Zalando engineering blog](#)



