<0110/> hackathon@thalia in Münster 2019

Neue Technologie, die Dinge spielerisch einfach macht! Neuer Inhalt, der zählt! Neues Modell, das uns und unsere Prozesse ein bisschen besser macht. Neue Idee, die du entdecken möchtest! Welche Themen fallen dir ein? Dieser Hackathon kennt keine Grenzen.

Für den diesjährigen Hackathon haben wir uns bewusst für ein offenes Format entschieden und damit getreu dem agilen Prinzip ‚inspect and adapt' das Feedback zur Themeneinschränkung beim letzten Mal berücksichtigt.

Die Beteiligung sprach für sich: 18 motivierte Menschen, 5 interessante Pitch-Vorschläge und 1 Tag Zeit.

Pitch der Ideen

Der Preis!

Das Ergebnis: 3 aus 5. Die Gruppen hatten sich schnell formiert und ihren Weg jeweils nach Bullerbü, Springfield oder Entenhausen in der Meeting-Area in Speicher 6 gefunden.

Die Türen in Winterfell waren zu jeder Zeit für den uneingeschränkten Zugriff auf die eiskalten Getränke weit geöffnet. Ebenso der Blick auf den Siegerpreis, der in den Katakomben unterhalb der Burg … genug davon, ich schweife ab.

Kommen wir zu den
Themen:

https://www.okd.i

Projekt: OKD – Openshift Kubernetes Distribution

Worum ging es?

Docker – auch bei uns ein Thema. Wie können wir die Administration
und Orchestrierung unserer wachsenden Container-Infrastruktur besser und effizienter

managen? Dieser Fragestellung will das Team mit 8 Leuten auf den Grund gehen und schaut sich das OpenSourceprodukt OKD genauer an.

OKD ist ein Upstream-Projekt für Openshift aus der Produktpalette von Red Hat und setzt auf Kubernetes auf.

Ergebnisse

oc cluster up – mit diesem Kommando ist das Team in die
Evaluierung gestartet. Das hat schon mal gut und vor allem schnell funktioniert. Ein SingleNode Cluster wird mittels Minishift und VirtualBox aufgebaut. In einem Gitlab-Projekt wurde eine Testapplikation mit Webfrontend und ein Dockerfile
angelegt und deployed. Parallel haben Teammitglieder versucht, Minishift auf einem MacOs-Gerät zum Laufen zu bekommen. Nach anfänglichen Konflikten mit der
installierten VirtualBox gelang auch dies.

Während der Abschlusspräsentation konnte eine schlanke Webanwendung (Login-UI) gezeigt, eine neue Version gebaut und über eine eigene Jenkins-Pipeline deployed werden. Et voilà – es funktioniert.

Während man mit Kubernetes alles bauen kann, was man so haben möchte (Load Balancer, Network Policies, Benutzerverwaltung, …), bringt OKD alle diese Features „out of the box“ mit. OKD ist somit eine gute open source Alternative zu OpenShift.

https://developers.google.com/awareness

Projekt: Google Awareness API

Worum ging es?

Unser eCommerce-Standort in Münster befindet sich in der Speicherstadt, wir arbeiten und treffen uns in unterschiedlichen Gebäuden. Es gibt zur Mittagszeit eine Kantine auf dem Campus, die ‚Hofrunde' ist ein allseits anerkanntes Kommunikationsinstrument und der Parkplatz ist ebenfalls auf dem Gelände. Das Team wollte analysieren, ob die Google Awareness API für die Erfassung von Zeiten genutzt werden kann, um somit anhand von Standorten zwischen Freizeit und Arbeit zu unterscheiden. Im ersten Schritt sollte dies auf Basis der geographischen Positionen von 2 Speichergebäuden erfolgen.

Ergebnisse

Die Zeit war eng bemessen. Ein Teil des Teams hat die App installiert und ein anderes die REST-Schnittstellen zum Sammeln und Anzeigen der Tracking-Daten entwickelt. Erkenntnis nach 6 Stunden: Die Anwendung wirkte instabil. Der Energieverbrauch durch fortlaufendes, aktives Ansprechen der App war hoch, d.h. durch fehlende Automatisierung wurden die genutzten Geräte stärker beansprucht. Eventuell war auch der gedachte Anwendungsfall hier einfach nicht der richtige. An dieser Stelle unkritisch – bei Thalia gibt es Vertrauensarbeitszeit:-).

Einige Java-Entwickler waren Teil des Teams und konnten so die genutzte Programmiersprache Kotlin kennenlernen und im Zuge der REST-Schnittstellen sofort damit entwickeln. Viele Vorteile wurden im direkten Vergleich mit Java gesehen – insbesondere die Null-Safety-Fähigkeit und ‚Lines of Code-Reduktion' konnte begeistern.

Das Team hat bei der abschließenden Präsentation alles gegeben und im Rahmen einer Live-Performance Tracking-Daten gesammelt. Sehr sportlich!

https://www.testcontainers.org/

Projekt: Testcontainers

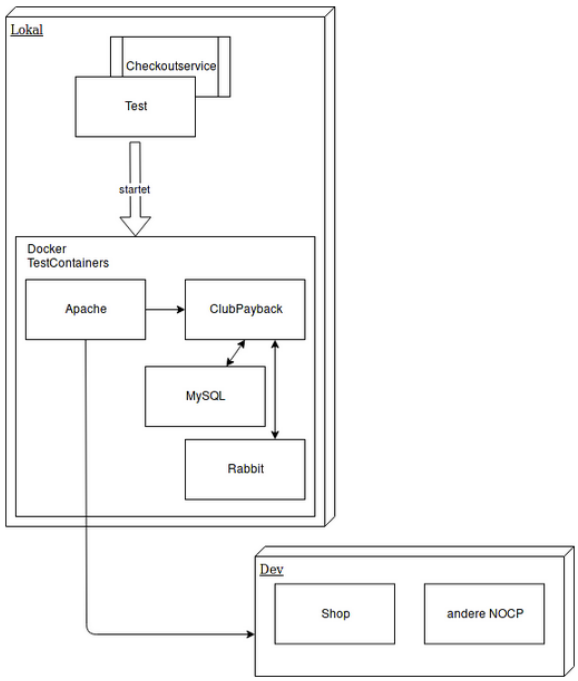
Worum ging es?

Mal wieder die Dev-Umgebung kaputt oder der Test schlägt erst auf der Integrationsumgebung nach dem Deployment fehl? Dieser Zustand kostet extrem viel Zeit. In unserer Microservice-Architektur haben unsere Produktteams zunehmend die Herausforderung, Test in Abhängigkeit zu SCS-Services anderer Teams auszuführen.

In unserem Thalia-Club-Programm schließt der Kunde die Mitgliedschaft im Rahmen eines Checkout-Prozesses ab. In diesem Prozess wird eine Payback-Kontenverknüpfung durchgeführt. Dieser Prozessbestandteil liegt in der Verantwortung eines separaten Produktteams. Wie kann nun das ‚Club-Checkout-Team' seine End2End-Tests (Front- und/oder Backend) mit möglichst hoher Unabhängigkeit zum ‚Payback-Team' gestalten? Beide Services wurden auf der Docker-Infrastruktur aufgesetzt.

Testcontainers ist ein Framework, um Docker-Container innerhalb von JUnit-Tests zu verwenden (Infrastructure-as-Code). Das Team möchte prüfen, ob hierdurch die Tests auf der Dev- oder Integrations-Umgebung reduziert werden und die Ausführung von Tests ohne Deployment auf die Teststages möglich ist.

Ergebnisse



Darstellung der beteiligten Services

Der PoC zeigt, wie von einander abhängige Services aus unserer Produktentwicklung im Test auf der lokalen Umgebung zu starten und auszuführen sind, dass wir vor dem Deployment auf den jeweiligen Umgebungen Fehler feststellen können. Und eben nicht hinterher;-).

Wichtig ist hierbei der Blick auf die transitiven Abhängigkeiten. Je mehr Kaskaden an inkludierten Testcontainern es gibt, desto komplizierter kann es werden. In einem Folgeschritt sollten noch die Laufzeiten angeschaut werden, die sich aufgrund der Start/Stop-Prozesse der Container erhöht haben. Das Team sieht in der Nutzung von Testcontainers eine Menge Potential. Klasse, dass sich hier Leute aus unterschiedlichen Produktentwicklungen zusammengefunden haben. So konnte schon der erste ‚Reality-Check' erfolgen.

Andreas, Christian, Benjamin und Julian haben die Zuschauer-Jury überzeugt.

Die Ergebnisse und die Präsentation haben dann auch am Schluss des Hackathons die Zuschauer-Jury überzeugt. Nach einer Stichwahl stand das Team als Gewinner fest. Herzlichen Glückwunsch!

Fazit

Ist es Zufall, dass sich 2 von 3 Themen rund um die Docker-Infrastruktur konzentrieren? Wahrscheinlich nicht – wir beschäftigen uns im Moment sehr mit dem Thema und wollen eine steile Lernkurve über Einsatz, Handling und Nutzen dieser Infrastruktur erreichen. Die Erfahrungen aus dem Hackathon werden uns hierbei unterstützen. Well done!

Im letzten Jahr haben wir die App-Entwicklung von Thalia von Berlin nach Münster umgezogen. Jetzt sitzen die Kollegen/innen im Büro nebenan. Macht

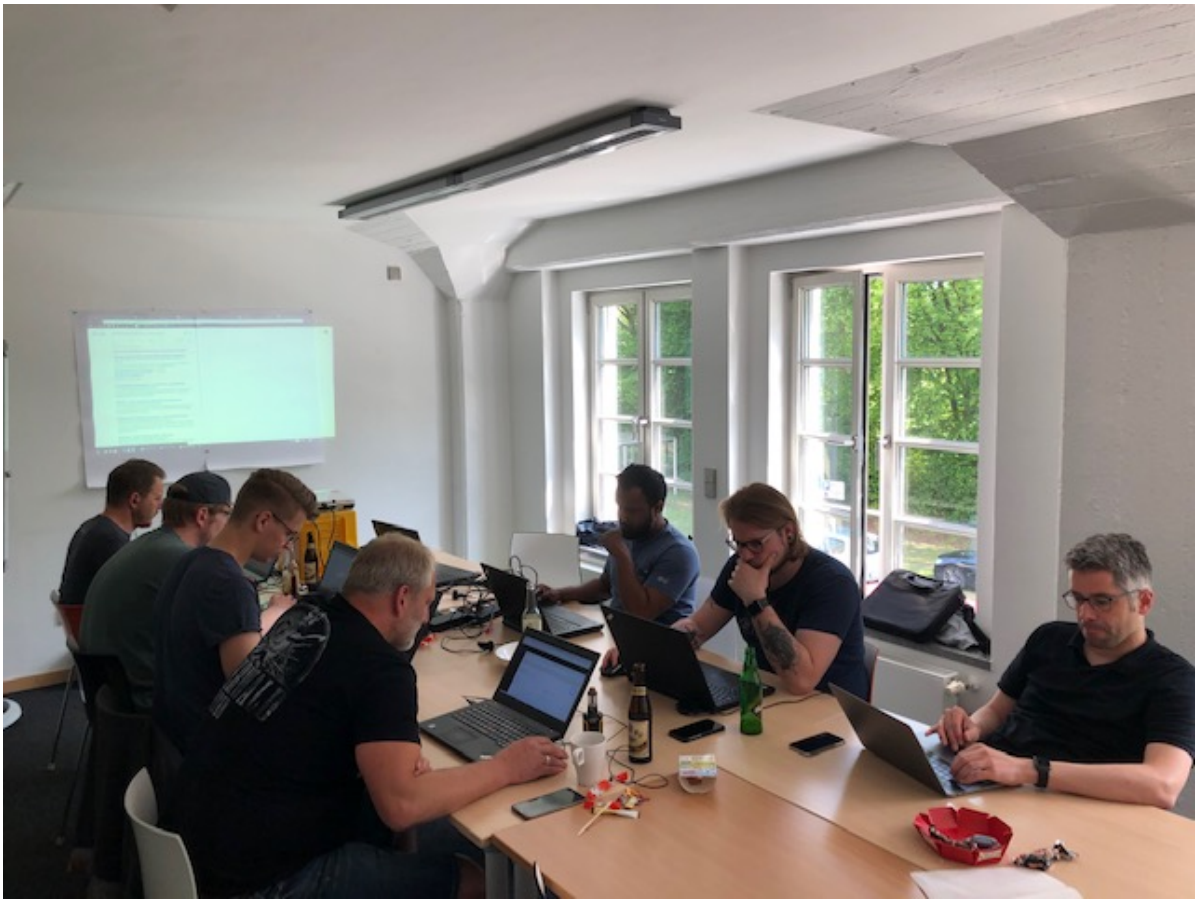
richtig Spaß, diese teamübergreifende Zusammenarbeit zwischen App-Experten und Java-Spezialisten. Kotlin wird aktuell bei uns in der App-Entwicklung genutzt. Mal sehen, was jetzt in den ‚Java-Teams' passiert:-).

Mit folgenden Stimmungsbildern verabschieden wir uns und schließen mit den Worten:

Das machen wir wieder – see you in 05/2020.

- 

- 
VELTINS
flaschenpost.de
- 

- 
<#0110/>
<hackathon@thalia/>
Willkommen beim
hackathon 2019
raum1:
winterfell
raum2:
bullerbue
raum3:
entenhausen
raum4:
springfield
okd
Google Awareness API
Testcontainers
- 

- 
- 

-