

Monitoring & Steuerung K8s Online-Banking

Studiengang : MAS Information Technology

Im Rahmen einer Technologieablösung wird das Online-Banking Frontend von PostFinance in verschiedene Services zerlegt und in Kubernetes überführt. Dadurch entstehen neue Anforderungen für Monitoring und Steuerung.

Umfeld

Im Zuge einer Technologieablösung wird das Frontend des Online-Banking (FIPO) von PostFinance derzeit vollständig überarbeitet. Dabei wird die, bisher eher monolithisch strukturierte Applikation, in verschiedene Komponenten aufgeteilt und in den On-Premise Kubernetes Cluster der PostFinance AG verschoben.

Anforderungen

Um in dieser neuen Umgebung eine zentrale Oberfläche für Monitoring und Steuerung der einzelnen Applikationsbestandteile zu haben, wurde im Rahmen dieser Arbeit eine entsprechende Lösung umgesetzt. Die initialen Anforderungen an dieses 'Cockpit' umfassten:

- Übersicht über aktive Deployments in einem Namespace
- Übersicht über aktuell laufende Pods pro Deployment bzw. Namespace
- Detailinformationen pro Pod (Sizing, Status, wo vorhanden Health-Checks und Statusinformationen des beinhalteten Services)
- Skalierung der Deployments
- Restart beliebiger deployter Pods

Im Zuge der detaillierten Anforderungsanalyse wurde ausserdem die Anbindung von Spring Actuator Endpoints der deployten Services Zwecks Informationsammlung sowie Steuerung aufgenommen. Aufgrund der sehr breiten Anforderungen und des Wunsches, alles auf einer Oberfläche vereinen zu können, wurde entschieden, die entsprechenden Funktionalitäten in das bereits vorgesehene Admin-Tool aufzunehmen.

Lösung

Entsprechend dieser Ausgangslage wurde in der vorliegenden Arbeit, auf Basis des parallel entstandenen Admin-Tool, die Monitoring- und Steuerungsoberfläche Cockpit entwickelt (vgl. Grafik). Diese besteht aus einem Frontend, welches auf Basis von Angular entwickelt wurde, und aus einem Backend auf Basis von Spring Boot. Das Backend bietet REST APIs für die

verschiedenen UIs des Frontend an und kommuniziert seinerseits via REST mit dem Kubernetes API Server des jeweiligen Cluster. Für gewisse Features werden ausserdem HTTP Calls direkt auf die entsprechenden Pods der eigentlichen Applikation ausgeführt.

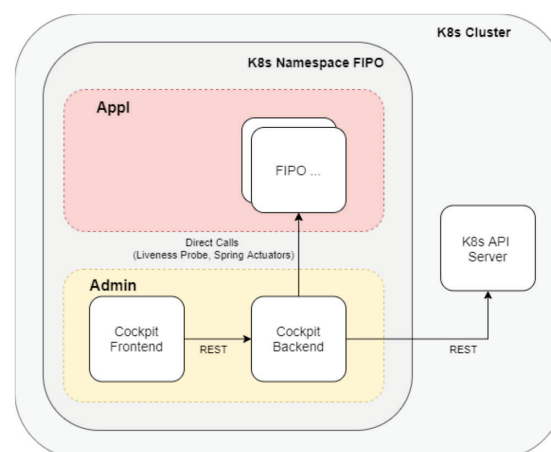
Für die Kommunikation mit dem Kubernetes API Server wird hierbei die frei verfügbare Java Client Bibliothek von Fabric8 eingesetzt. Auf Seiten der Applikations-Pods werden Spring Actuator Endpoints angeboten, welche dann wiederum vom Cockpit Backend konsumiert werden.

Methodik

Um dem Projektumfeld als Teil des grösseren Migrationsprojektes gerecht zu werden, wurde die Arbeit in einer agilen Vorgehensweise durchgeführt. Mit vier vierwöchigen Sprints konnte optimal auf fehlende technische Voraussetzungen sowie anfangs unklare und/oder sich ändernde Anforderungen reagiert werden.



Florian Fankhauser



Systemübersicht 'Cockpit' vereinfacht