

Vorstudie VA

Niklas Liechti

Inhaltsverzeichnis

1. Management Summary	1
2. Antrag	2
2.1. Organigramm.	2
2.2. Projektauftrag	2
2.3. Ausgangsleg / Bisherige Erfahrung	2
2.4. Termin	3
2.5. Budget	3
3. Systemdefinition / Abgrenzungen	4
3.1. Systemgrenzen	4
3.2. Applikation.	5
4. Anforderungen	6
4.1. Anforderungen nach Projektantrag	6
4.2. Anforderungen aus Gespräch mit Thomas Kälin	8
5. Swot.	10
5.1. Strengths	10
5.2. Weaknesses	10
5.3. Opportunities.	10
5.4. Threats	10
6. Ziele	11
6.1. Muss Ziele	11
6.2. Kann Ziele	11
6.3. Abgrenzungen	11
7. Erhebung.	12
7.1. 1) Nullvariante.	12
7.2. 2) Entwicklung einer Individuallösung mit Kubernetes Files hosted on Github	13
7.3. 3) Entwicklung einer Individuallösung basierend auf Helm	14
8. Empfehlung	15
9. Weiteres Vorgehen.	16



1. Management Summary

In dem Projekt geht es darum, eine minimale Management Applikation für die Lehrer der TBZ zu bauen, welche Applikationen auf einen Kubernetes Cluster deployen kann.

Die Applikation müssen bereits in einem Kubernetes "fähigen" Zustand vorliegen.

Dies heisst:

- Es muss ein Dockerimage der Applikation geben
- Es muss bereits kubernetes config file dafür geben.

Die Lehrer sollen dadurch die neu erschaffenen Kubernetes Cluster in den Schulzimmern einfacher nutzen können.

Theoretisch können damit auch Lehrer die kein explizites Kubernetes Wissen haben, eine Applikation für den Unterricht aufsetzen.

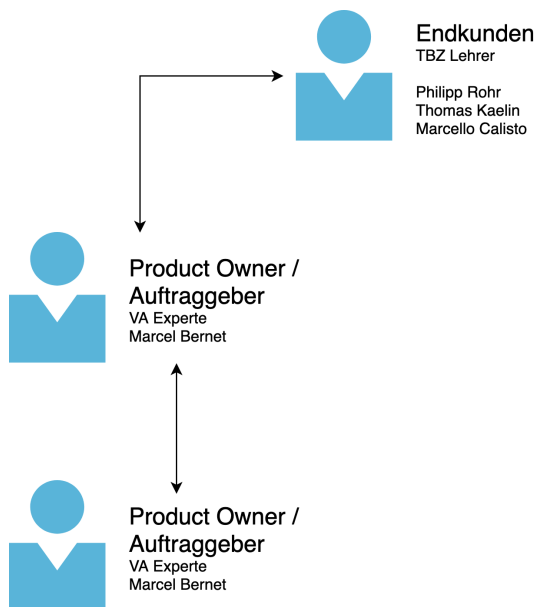


2. Antrag

2.1. Organigramm

Da das Projekt recht klein ist, hält sich auch das Organigramm in Grenzen. Grundsätzlich sind es 5 Beteiligte. Im Kern sind aber nur 3 Personen enger involviert.

- Marcel Bernet (Auftraggeber)
- Niklas Liechti (Projektleitung / Umsetzung)
- Thomas Kälin (Berufsschullehrer)



2.2. Projektauftrag

Die Vorgabe für das Projekt ist das entwickeln eines Usecases für einen Clusterservice auf Kubernetes basierend. Nach dem Initialen Gespräch wurde die Aufgabenstellung sehr konkretisiert.

Es soll nun um ein kleines Management System für die Kubernetes Cluster des TBZ werden.

Die Plattform ist ausdrücklich nur für Lehrer, um die neu angeschafften Kubernetes Cluster für den Unterricht nutzen zu können. Es soll einfach eine mehrzahl von Services zu übungszwecken deployt werden können.

2.3. Ausgangsleg / Bisherige Erfahrung

Beruflich hatte ich bereits einen sehr kleinen Berührungspunkt mit Kubernetes, genauer mit der GKE (Google Kubernetes Engine) in der GCP (Google Cloud Platform). Dieser ausflug war aber nur ein sehr beschränkter POC auf 3-4 Tage.

Da ich Software Engineer bin habe ich auch das nötige wissen und können um diese Applikation zu bauen.



2.4. Termin

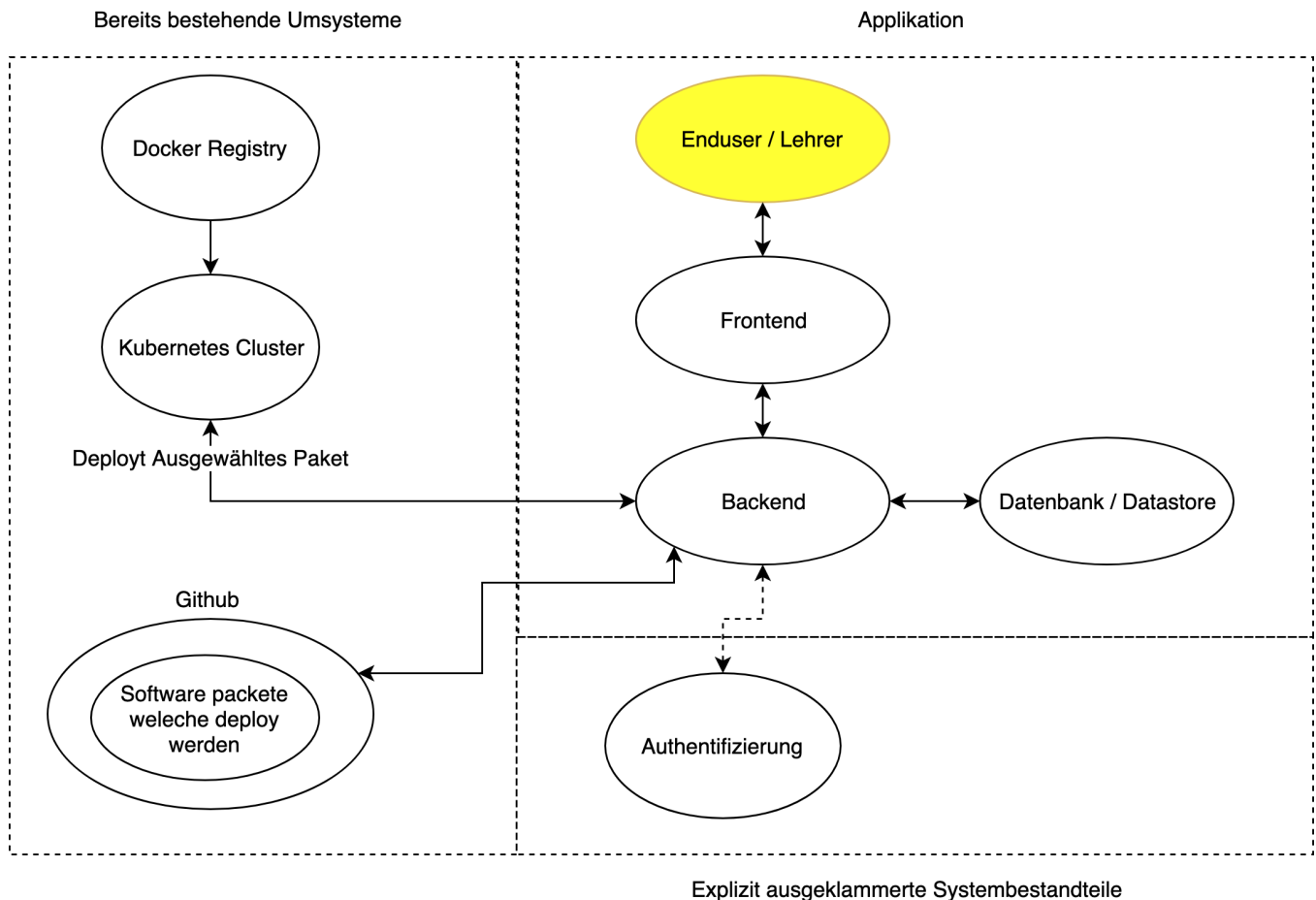
2.5. Budget

Grundsätzlich steht für die Umsetzung der Vertiefungsarbeit kein Budget zu Verfügung. Lediglich ein Zeitbudget von ca. 100 Arbeitsstunden ist vorgesehen.



3. Systemdefinition / Abgrenzungen

3.1. Systemgrenzen



3.1.1. Umsysteme

Kubernetes Cluster

Der Kubernetes cluster wird zum Deployment der Applikationen verwendet. Da der Hauptteil der Applikation die Kommunikation mit der Kubernetes Cluster ist hat er die Wichtigste Schnittstelle zum Projekt.

Github

Github wird zum "hosten" der Kubernetes Config files verwendet.

Bedeutet: Die Konfiguration der zu deployenden Services muss in einem yaml File auf Github gepusht werden, damit der Deployer diese Konfiguration dann von dort wieder einlesen kann.



Docker Registry / Dockerhub

Eine Dockerregistry, im Normalfall das von Docker bereitgestellte Dockerhub beherbergt die Dockerimages der Services. Daher die eigentliche Software die danach auf den Kubernetes Cluster deployt wird.

3.2. Applikation

Die Applikation selber besteht aus Vier Untersystemen.

3.2.1. Enduser

Der Enduser ist in diesem Fall in fast allen Fällen ein Lehrer. Dieser will für seine Schüler mit hilfe der noch zu erstellenden Applikation eine Umgebung bereit stellen.

3.2.2. Frontend

Über das Frontend interagiert der Enduser indirekt mit dem Cluster. Idealerweise muss er nie ein Tool, wie z.B. `kubectl` benutzen.

Das Frontend soll so einfach wie möglich zu bedienen sein, damit kein konfigurationsfehler gemacht werden können.

3.2.3. Backend

Das Backend interagiert mit dem Kubernetes Cluster. Es nimmt die gegebenen Konfigurationen und appliziert diese auf den Cluster.

3.2.4. Datastore

Der Datastore hält möglichst wenige Informationen, da dies in dieser Applikation eigentlich kaum nötig sein sollte. Dennoch müssen z.B. die Hinterlegten Repos irgendwo abgespeichert werden.

3.2.5. Authentifizierung (Abgegrenzt)

Die Authentifizierung gegenüber der Applikation ist explizit abgegrenzt und wird in dieser Arbeit nicht umgesetzt.



4. Anforderungen

4.1. Anforderungen nach Projektantrag

4.1.1. Setup lokale Kubernetes Umgebung

Um überhaupt mit der Arbeit starten zu können muss zuerst eine lokale Kubernetes Umgebung aufgesetzt werden, was ich noch nie gemacht habe.

Aufwand

Klein: Der Aufwand sollte sich hier im Rahmen halten, da ich nicht der erste bin der so etwas machen möchte.

4.1.2. Konzeption Applikation / Aufteilen in Tickets

Die grobe Architektur wurde im initialen Kickoff besprochen. Diese muss aber in einer "Vorstudie" verfeinert werden. Die exakten Tools und Frameworks müssen definiert werden.

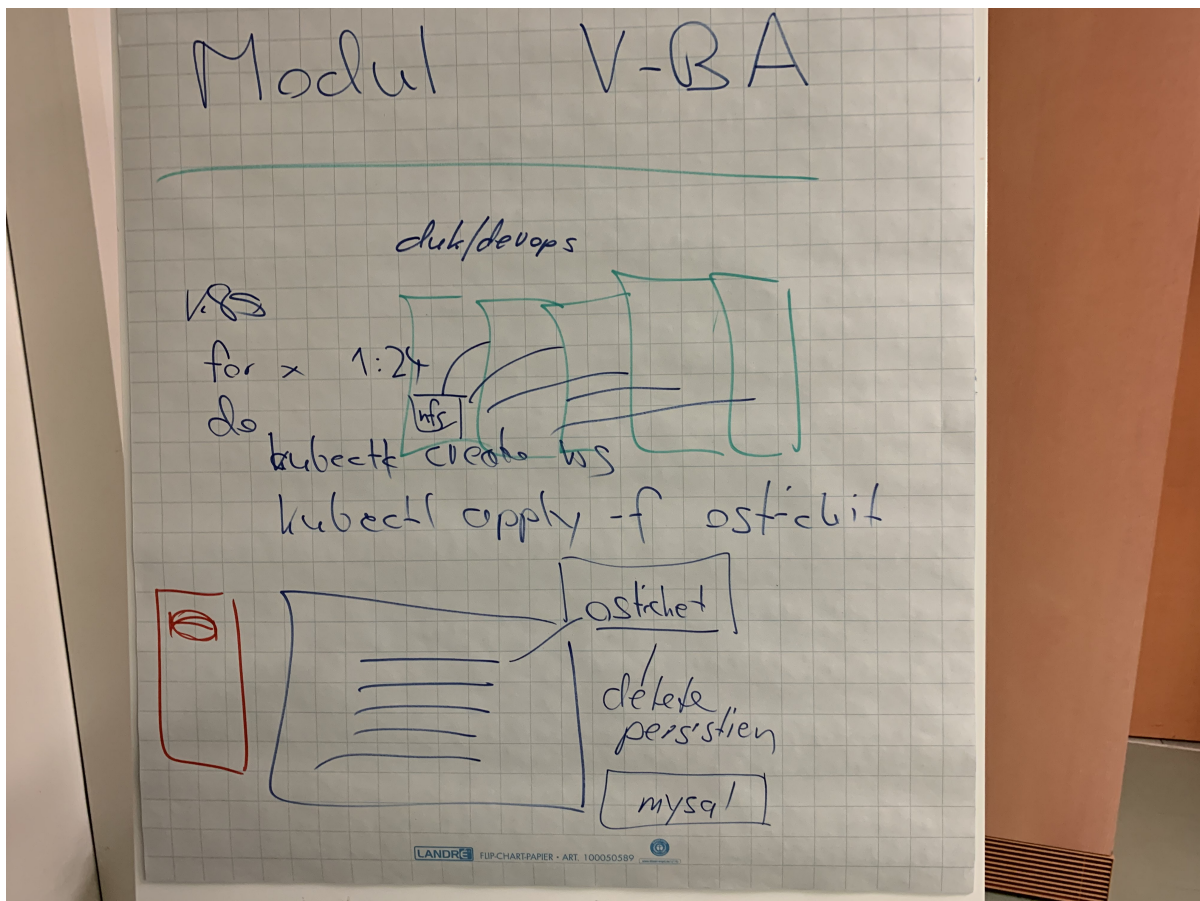


Figure 1. Architektur



Aufwand Mittel: Die Architektur festzulegen und Feedback einzuholen wird meist unterschätzt. Es ist aber ein essenzieller Bestandteil des Projekts.

4.1.3. Entwickeln der Applikation

Dies ist natürlich der Hauptteil der Arbeit, da es um die effektive Umsetzung geht. Der Umfang der Applikation ist nicht gigantisch, muss aber mit vielen Umsystemen kommunizieren und funktionieren.

Aufwand

Gross: Das umsetzen und Integrieren in die Umgebung braucht voraussichtlich am meisten Zeit.

4.1.4. Testen auf TBZ Infrastruktur

Um die Applikation einsetzen zu können muss sie natürlich auf der Tbz Infrastruktur getestet werden. Da sie eigentlich "plattformunabhängig" ist, sollte das eigentlich kein riesiger Aufwand sein, kann aber trotzdem noch Arbeit machen.

Aufwand

Klein - Mittel: Sollte kein all zu grosser Aufwand sein, es können aber doch Dinge schief gehen.



4.2. Anforderungen aus Gespräch mit Thomas Kälin

4.2.1. Wichtige Punkte

Entry Point Login Link generieren

Nach dem Aufsetzen einer Instanz soll eine Liste von Links für jede Instanz generiert werden. Diese soll dem Lehrer dazu dienen die Instanzen den Schülern zu teilen.

Auflistung aller Installationen

Es sollen alle Installationen auf einer Seite aufgelistet werden.

Detail Seite einer Installation

Jede Installation sollte eine Detailseite haben, mit den wichtigsten Informationen drauf:

- Anzahl Pods
- Liveliness probe abfragen

Auch sollten verschiedene Aktionen ausgeführt werden können:

- Löschfunktion
- Start / Stop
- Restart eines einzelnen Pods

Log der Instanz

Wenn möglich sollte das Log der Instanz auf der Detailseite dargestellt werden, damit es bei Problemen schnell überprüft werden kann

4.2.2. Aufsetzen einer Installation

Möglicherweise wäre es noch praktisch die Version der gewählten Installation beim deployen verändern zu können. Dann muss nicht jedes mal das File im Repo angepasst werden

Es soll konfigurierbar sein ob beim runterfahren alles gelöscht werden.

4.2.3. Nice to have



Möglichkeit ende des Tages löschen

Es könnte noch praktisch sein, eine Installation automatisch am Ende des Tages wieder löschen zu können.

Login password automatisch setzten

Ein Service sollte mit verschiedenen Passwörtern für jede Instanz ausgeliefert werden, damit sich nicht alle Schüler auf allen Instanzen einloggen können.

4.2.4. Super nice to have

Eine Visuelle darstellung der deployten Pods mit einer grafiklibrary wäre schön aber funktionsmässig nicht nötig.



5. Swot

5.1. Strengths

- Lehrer können extrem schnell eine Umgebung aufsetzen
- Schüler haben ab dem ersten mal eine Arbeitsumgebung
- Es wird keine Zeit mit dem individuellen setup verloren
- Lehrer hat eine einheitliche datenlage um zu bewerten.

5.2. Weaknesses

- Die installation ist nicht so flexibel
- Bereitstellung benötigt Kubernetes Kenntnisse

5.3. Opportunities

- Lehrer können ihren Unterricht flexibler gestalten als zuvor
- Lehrer brauchen weniger zeit für die Bereitstellung der Umgebung

5.4. Threats

- Die Lehrer brauchen die Applikation nicht, weil die Bereitstellung der Services als Kubernetes config file zu kompliziert ist
- Es kommt eine neue Technologie die Kubernetes ersetzt und die Applikation obsolet macht (unwahrscheinlich)



6. Ziele

6.1. Muss Ziele

1. Die Applikation muss auf der Kubernetes Infrastruktur des TBZ laufen
2. Die Applikation muss auf einer Lokalen Installation laufen
3. Die Applikation muss per WebUI bedienbar sein
4. System (Kann aus mehreren Containerern bestehen)
 - a. Systeme werden von einem Github Projekt mit vorgegebener Ordnerstruktur deploybar gemacht
 - b. Die Anzahl Replikation muss einstellbar sein. Es müssen n Systeme hochgezogen werden können, die je ein Entrypoint haben.
 - c. Die Daten müssen auf dem Cluster persistiert werden können
 - d. Die Deployte Applikation muss entfernt werden können

6.2. Kann Ziele

1. Das Löschen von Applikation und Daten ist getrennt möglich
2. Die aktiven Services sollen für die Lernenden als Web-UI zur Verfügung gestellt werden (nur Anwahl).

6.3. Abgrenzungen

1. Die Applikation wird keinerlei Authentifizierung haben.



7. Erhebung

7.1. 1) Nullvariante

7.1.1. Risiken

1. Hoher Supportaufwand im Betrieb
2. Lehrer benutzen wegen zu hoher Komplexität die Cluster gar nicht
3. Kubernetes wird EOL

7.1.2. Kosten

In der Umsetzung selber gäbe es hier keine Kosten. Wenn diese Variante gewählt würde, müssten aber viele Lehrer im Umgang mit Kubernetes geschult werden. Dieser Aufwand wäre schwierig zu rechtfertigen und mit gewissen Lehrpersonen auch nicht durchführbar.



7.2. 2) Entwicklung einer Individuallösung mit Kubernetes Files hosted on Github

7.2.1. Risiken

1. Die Applikation deckt nicht alle Usecases ab
2. Github steht nicht mehr zur Verfügung
3. Kubernetes wird EOL

7.2.2. Kosten

Die Kosten hier liegen bei der Entwicklung der Applikation, welche im Rahmen der VA gemacht wird. Das Ziel ist dann, dass möglichst wenig Schulung nötig sein wird.



7.3. 3) Entwicklung einer Individuallösung basierend auf Helm

7.3.1. Risiken

1. Wegen zu hoher Komplexität entstehen keine neuen Packages
2. Die Applikation deckt nicht alle Usecases ab
3. Kubernetes wird EOL

7.3.2. Kosten

Diese Möglichkeit wäre womöglich mit leicht weniger Aufwand umzusetzen als Variante 2. Es entsteht aber im nachhinein ein grösserer Aufwand für die Lehrer, welche die Helm Packages erstellen müssen. Dieser würde sich dann auf sehr wenige Lehrpersonen konzentrieren, welche das nötige Knowhow haben.



8. Empfehlung

Die Nullvariante scheidet wegen nicht vorhandenem Knowhow der Lehrer im Kubernetes von anfang an aus.

Die Variante 3 wäre technisch und Nutzertechnisch eigentlich die Beste Variante, welche aber wegen limitiertem Zeitbudget und auch limitiertem Knowhow mit dem erstellen von Helm Packages nicht umgesetzt werden kann.

Daher bleibt eigentlich nur die Variante 2, welche alle Anforderungen zur genüge erfüllt. Diese Version ist auch die Einzige, welche die Technische hürde genug tief legt, um es den meisten Lehrpersonen zu ermöglichen die Kubernetes Cluster zu benutzen.

Empfehlung: Variante 2 "Entwicklung einer Individuallösung mit Kubernetes Files hosted on Github"



9. Weiteres Vorgehen