

# Informatik

## Geodeatenprozessierung mit Budget Instanzen

Semesterarbeit



Abbildung 1: Eine Ausfahrt mit Budget Instanzen [Web10b]

Departement:	Informatik
Kurs:	CAS CLD FS20 – Cloud Computing
Autor:	Tobias Reber
Experte:	Jörg Thomann
Datum:	16. 07. 2020

Perplexity  
is the beginning of knowledge.

- Kahlil Gibran

[AD19, p. 33]

## Management Summary

In der Kürze liegt die Würze [Doe00]

# Inhaltsverzeichnis

<b>1 Einführung</b>	<b>1</b>
1.1 GIS . . . . .	1
<b>2 Vorgehen</b>	<b>3</b>
2.1 Arbeitsmethodik . . . . .	3
2.2 Arbeitsweise . . . . .	3
2.3 Projektplan . . . . .	3
<b>3 Vorarbeiten</b>	<b>4</b>
3.1 AWS Account . . . . .	4
3.2 AWS Spot Instances . . . . .	4
3.3 Kubernetes testing . . . . .	4
<b>4 Ausgangslage</b>	<b>5</b>
4.1 swisstopo bei AWS . . . . .	5
4.2 Ist-Zustand 3D Prozessierung swisstopo . . . . .	5
4.2.1 Prozess der Datenpublikation . . . . .	5
<b>5 Use Case</b>	<b>6</b>
5.1 Beschreibung der Technischen Komponenten . . . . .	6
5.1.1 SPOT Instanzen . . . . .	6
5.2 Abgrenzung . . . . .	6
<b>6 Architektur</b>	<b>7</b>
6.1 Bewertungskriterien . . . . .	7
6.2 Variante 1: Spot . . . . .	7
6.3 Variante 2: Spot mit EKS . . . . .	7
<b>7 Prototyp</b>	<b>8</b>
7.1 Realisierung . . . . .	8
<b>8 Evaluation</b>	<b>8</b>
8.1 Erfahrungen . . . . .	8
8.2 Kritische Punkte . . . . .	8
<b>9 Zusammenfassung</b>	<b>8</b>
<b>10 Ausblick</b>	<b>8</b>
<b>Literaturverzeichnis</b>	<b>9</b>
<b>A Konfigurationen</b>	<b>I</b>
<b>B Fachbegriffe und Abkürzungen</b>	<b>II</b>
<b>C Projektplanung</b>	<b>III</b>
<b>D Testing</b>	<b>III</b>

## Abbildungsverzeichnis

---

1	Eine Ausfahrt mit Budget Instanzen [Web10b] . . . . .	1
2	Internetkarte des Bundes <i>map.geo.admin.ch</i> . Hier ein Ausschnitt der Saane bei Kleinbödingen, das Luftaufnahmen von 1946 zu heute vergleicht. . . . .	1
3	Klassisches Kanban auf <i>github.com</i> . . . . .	3
4	AWS Architektur Spot[Web10a] . . . . .	4
5	Projektplan . . . . .	III

---

# 1 Einführung

Ich arbeite als GIS-Spezialist bei der swisstopo<sup>1</sup>, dem Bundesamt für Landestopografie in Wabern. Wir machen Karten. Mein Team macht Internetkarten - wie Google Maps<sup>2</sup>, jedoch von der Schweiz für die Schweiz; und allen anderen auch. Unsere Internetkarte, der Viewer, erfreut sich relativ grosser Beliebtheit und beinhaltet ca. 800 Themen wie Wanderwege, Solarkataster oder Luftfahrthindernisse. Lieber Leser<sup>3</sup>, falls dir map.geo.admin.ch noch kein Begriff sein sollte, kann ich dir wärmstens empfehlen, dir etwas Zeit zu nehmen, um darin zu schmökern. Es gibt viel zu entdecken und es gratis - ein Service Public.



Abbildung 2: Internetkarte des Bundes *map.geo.admin.ch*. Hier ein Ausschnitt der Saane bei Kleinbödingen, das Luftaufnahmen von 1946 zu heute vergleicht.

## 1.1 GIS

Wie erwähnt, arbeite ich als *GIS-Spezialist*. Wobei mir der Titel *Geo-Informatiker* besser gefällt: weil er *Geografie* und *Informatik* beinhaltet. *Geografie* kommt aus dem Griechischen und bedeutet Erdbeschreibung. *Informatik* ist die Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen.

GIS ist ein Akronym für Geographical Information Systems. GIS bedeutet im engsten Sinn eine Ansammlung von Computerprogrammen, die zur Bearbeitung von Karten und Geodaten verwendet werden. Geodaten sind nichts weiter als Daten mit einem räumlichen Bezug<sup>4</sup>. In einem weiteren Sinn deckt der Begriff GIS ein ganzes Fachgebiet ab, das sich mit Karten und Geodaten auskennt. Es ist also nicht nur ein Werkzeug, sondern ein Fachgebiet, das Kenntnisse über Datensammlung, Speicherung, Analyse und Darstellung innerhalb von vielen verschiedenen Themen mit einem räumlichen Bezug abdeckt.

Es ist zwar weit von der klassischen Geografie zu Zeiten Humboldts entfernt, aber es liegt auf der Hand, dass auch Cloud Computing zu GIS gehört.

Typische Geodaten sind digitale Karten, Inventare und Register von Parzellen, Umweltfaktoren, Grenzen, Entwicklung, Planung etc., die einen räumlichen Bezug haben und dadurch in einem geografischen Zusammenhang

<sup>1</sup>www.swisstopo.ch

<sup>2</sup>maps.google.com

<sup>3</sup>Im vorliegenden Dokument wird durchwegs der männliche Singular (Leser, Benutzer) als Ansprache verwendet. Diese Ansprache bezieht sich auf beide Geschlechter sowie gegebenenfalls mehrere Personen. Sie dient lediglich der leichteren Lesbarkeit der Semesterarbeit

<sup>4</sup>mit Koordinaten, typisch Nord und Ost

analysiert und dargestellt werden können.

Unser Team (IGEB-B) publiziert diese Daten; deren Nachführungszyklus wie auch der Aufwand zur Aufbereitung fürs Web unterschiedlich sind. Einige Daten werden manuell aufbereitet, andere stündlich automatisch nachgeführt.

## 2 Vorgehen

In diesem Kapitel wird das Vorgehen beschrieben, wie die Arbeit geplant und erledigt wurde.

### 2.1 Arbeitsmethodik

Um die Übersicht nicht zu verlieren, wurde beschlossen, nach Kanban zu arbeiten. Da eigens für die Arbeit ein Repository auf *github.com*<sup>5</sup> angelegt wurde, konnte auch gleich ein Kanban erstellt werden. **TODO: Theorie über Kanban - evtl. mit Fachbegriffen verhängen**

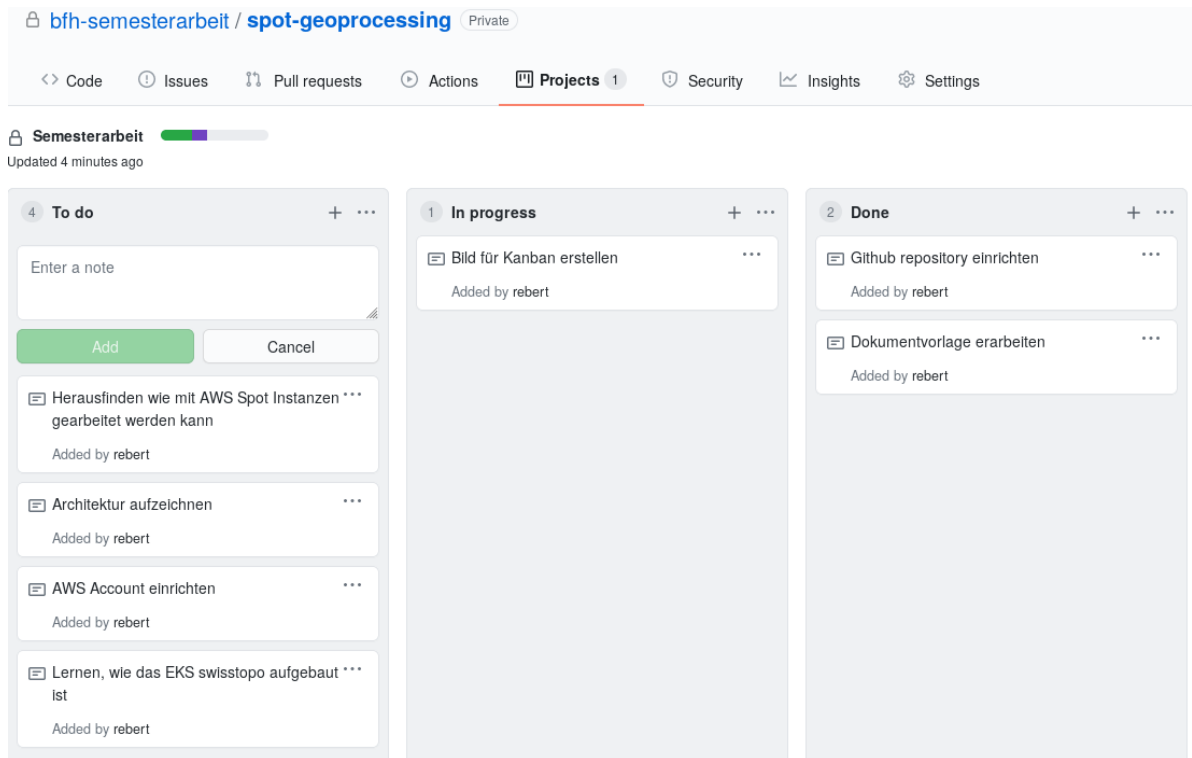


Abbildung 3: Klassisches Kanban auf *github.com*

### 2.2 Arbeitsweise

Der Experte Jörg Thomann begleitet die Arbeit seit unserem ersten Treffen, das vom 14. Juli 2020. Wir sind so verblieben..., dass wir uns dann und wann wieder treffen. Die Arbeit, wie im Projektplan und in den Kanban Tickets definiert abgearbeitet wird... bei Fragen ...

### 2.3 Projektplan

Projektplan<sup>6</sup>

<sup>5</sup>Projekt auf *github.com*

<sup>6</sup>URL Google Spreadsheet



## 3 Vorarbeiten

### 3.1 AWS Account

- Fit werden mit AWS

### 3.2 AWS Spot Instances

- Herausfinden wie mit AWS Spot Instanzen gearbeitet werden kann

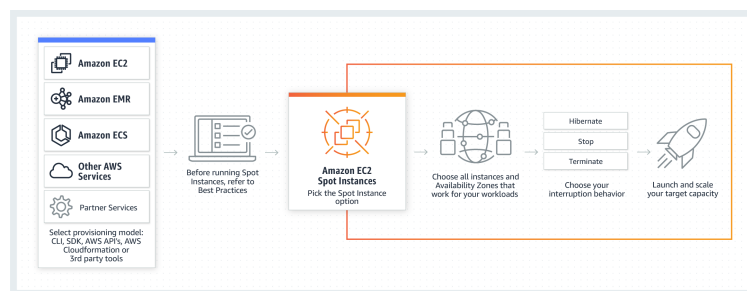


Abbildung 4: AWS Architektur Spot[Web10a]

### 3.3 Kubernetes testing

- Anfangs mit Minicube
- Eigenem AWS account

## 4 Ausgangslage

### 4.1 swisstopo bei AWS

### 4.2 Ist-Zustand 3D Prozessierung swisstopo

#### 4.2.1 Prozess der Datenpublikation

- ▶ Ausfindig machen des Zeitaufwandes pro Update bei uns und bei der IT
- ▶ Topo an Kogis, Kogis an IT, Kogis an Topo ...
- ▶ Bild des Prozesses, der Infra
- ▶ Manuelles Bereitstellen von EC2 Instanzen

## 5 Use Case

- ▶ Automatisierung der Datenprozessierung
- ▶ Testumgebung für TopoKo
- ▶ Automatische Prozessierung via SPOT Instanzen

### 5.1 Beschreibung der Technischen Komponenten

#### 5.1.1 SPOT Instanzen

### 5.2 Abgrenzung

Wie in der Projektskizze beschrieben wird der Fokus auf bereits dockerisierte Prozessierungstools gelegt.

Der Fokus liegt vor allem auf der Automatisierung von SPOT Instanzen im Vergleich zum Manuellen Bereitstellen.

## 6 Architektur

Ausfindigmachen von verschiedenen Architekturen

### 6.1 Bewertungskriterien

- ▶ Muss in AWS Cloud
- ▶ Zugang zu EFS
- ▶ Automatisierbar sein

### 6.2 Variante 1: Spot

Evtl. lässt sich eine Lösung finden ohne EKS zu verwenden

### 6.3 Variante 2: Spot mit EKS

Spot mit EKS

## **7 Prototyp**

### **7.1 Realisierung**

## **8 Evaluation**

### **8.1 Erfahrungen**

### **8.2 Kritische Punkte**

## **9 Zusammenfassung**

## **10 Ausblick**

## Literaturverzeichnis

- [AD19] J. Arundel and J. Domingus, *Cloud native devops with kubernetes*. O'RELLY, 2019.
- [Doe00] J. Doe, *The book without title*. Dummy Publisher, 2100.
- [Web10a] Webpage. (2010). Amazon ec2-spot-instances, [Online]. Available: <https://aws.amazon.com/de/ec2/spot/> (visited on 07/08/2010).
- [Web10b] —, (2010). Keep calm and make your choice..., [Online]. Available: <https://www.grenke-40-one.de/excursions-detail/> (visited on 07/11/2010).

## A Konfigurationen

Listing 1: Sample Code

```
#include <stdio.h>

int main()
{
    // Variable definitions
    char operator;
    double n1, n2;
    double result;

    /* Reading and parsing of equation
     *
     * The format provided in the 'scanf' function
     * tells C how to read the given input and in
     * which variable to store each part of the input
     * %lf := long float aka double
     * %c := char
     */
    printf("Enter equation in the format 1 + 2: ");
    scanf("%lf %c %lf", &n1, &operator, &n2);

    // figure out which operator was used and
    // perform calculation accordingly
    if (operator == '+')
        result = n1 + n2;

    else if (operator == '-')
        result = n1 - n2;

    else if (operator == '*')
        result = n1 * n2;

    else if (operator == '/')
        result = n1 / n2;

    // output result
    printf("Result: %f\n", result);
}
```

## B Fachbegriffe und Abkürzungen

**Paas** Plattform as a Service

21 22

31 32

41 42

51 52

61 62

71 72

81 82

91 92

101 102



## C Projektplanung

PROJEKTTITEL	Semesterarbeit Geodatenprozessierung mit Budget Instanzen
FULL STACK PROJEKTMANAGER	Tobias Reber

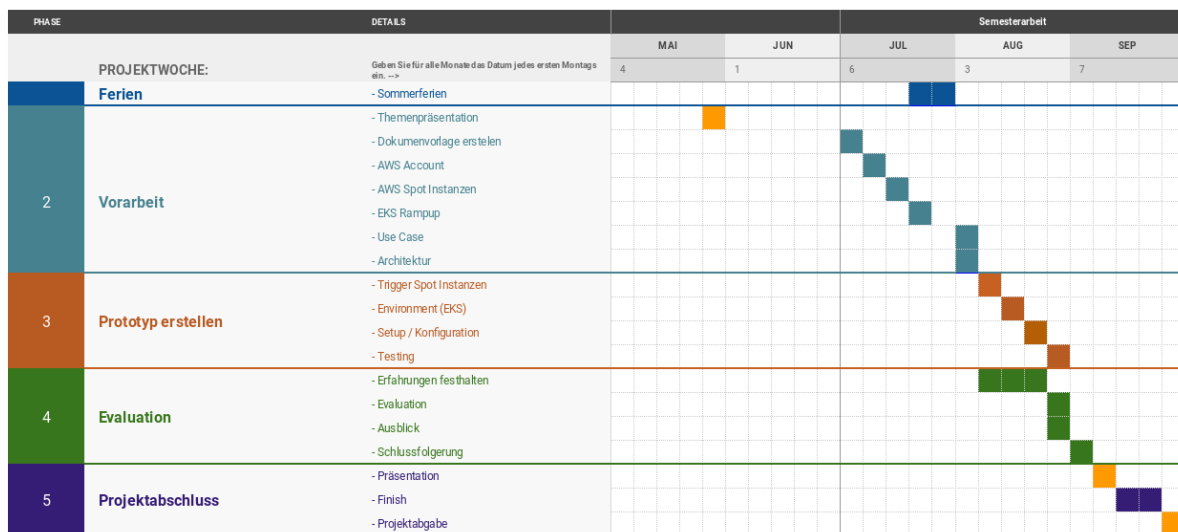


Abbildung 5: Projektplan

## D Testing

bla adsflkj

Listing 2: Sample Code

```
#include <stdio.h>

int main()
{
    // Variable definitions
    char operator;
    double n1, n2;
    double result;

    /* Reading and parsing of equation
     *
     * The format provided in the 'scanf' function
     * tells C how to read the given input and in
     * which variable to store each part of the input
     * %lf := long float aka double
     * %c := char
     */
    printf("Enter equation in the format 1 + 2: ");
    scanf("%lf %c %lf", &n1, &operator, &n2);

    // figure out which operator was used and
    // perform calculation accordingly
    if (operator == '+')
        result = n1 + n2;

    else if (operator == '-')
        result = n1 - n2;

    else if (operator == '*')
        result = n1 * n2;
```

```
else if (operator == '/')  
    result = n1 / n2;  
  
// output result  
printf("Result: %f\n", result);  
}
```

so da