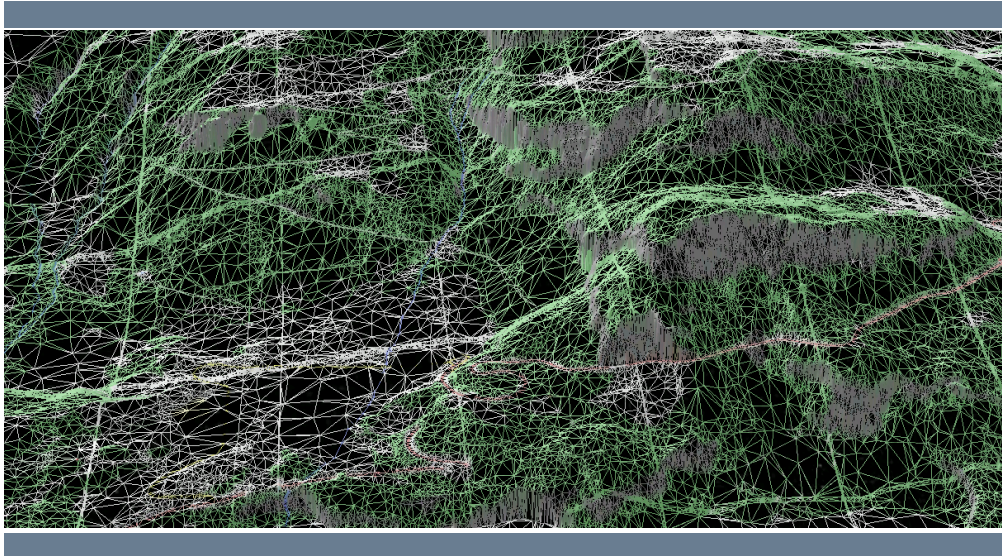




Informatik

Geodeatenprozessierung mit Budget Instanzen (SPOT) auf Amazon EKS

Semesterarbeit



Departement:	Informatik
Kurs:	CAS CLD FS20 – Cloud Computing
Autor:	Tobias Reber
Experte:	Jörg Thomann
Datum:	11. 07. 2020

Perplexity
is the beginning of knowledge.

- Kahlil Gibran

[AD19, p. 33]

Management Summary

In der Kürze liegt die Würze [Doe00]

Inhaltsverzeichnis

1 Vorgehen	1
1.1 Arbeitsmethodik	1
1.2 Arbeitsweise	1
1.3 Projektplan	1
2 Vorarbeiten	2
2.1 AWS Account	2
2.2 AWS Spot Instances	2
2.3 Kubernetes testing	2
3 Ausgangslage	3
3.1 swisstopo bei AWS	3
3.2 Ist-Zustand 3D Prozessierung swisstopo	3
3.2.1 Prozess der Datenpublikation	3
4 Use Case	4
4.1 Beschreibung der Technischen Komponenten	4
4.1.1 SPOT Instanzen	4
4.2 Abgrenzung	4
5 Architektur	5
5.1 Bewertungskriterien	5
5.2 Variante 1: Spot	5
5.3 Variante 2: Spot mit EKS	5
6 Prototyp	6
6.1 Realisierung	6
7 Evaluation	6
7.1 Erfahrungen	6
7.2 Kritische Punkte	6
8 Zusammenfassung	6
9 Ausblick	6
Literaturverzeichnis	7
A Konfigurationen	I
B Fachbegriffe und Abkürzungen	II
C Projektplanung	III
D Testing	III

Abbildungsverzeichnis

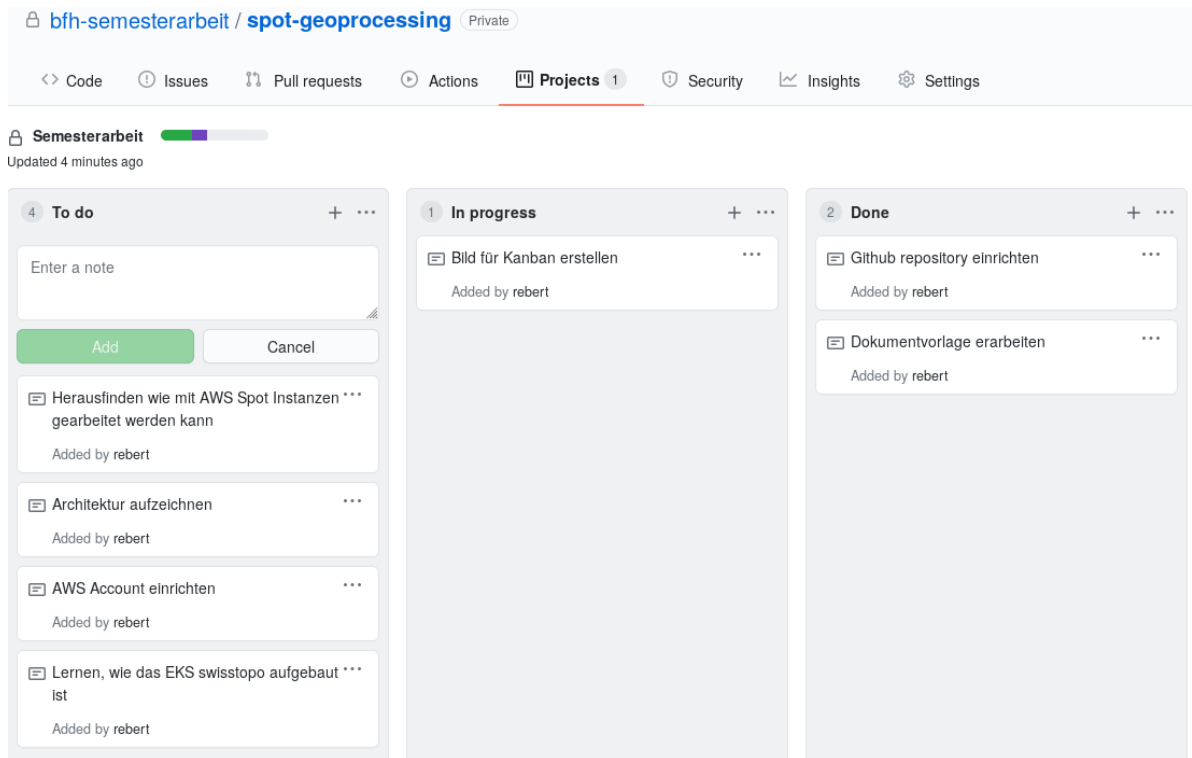
1	Klassisches Kanban auf <i>github.com</i>	1
2	AWS Architektur Spot[Web10]	2
3	Projektplan	III

1 Vorgehen

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

1.1 Arbeitsmethodik

Um die Übersicht nicht zu verlieren, wurde beschlossen, nach Kanban zu arbeiten. Da eigens für die Arbeit ein Repository auf *github.com*¹ angelegt wurde, konnte auch gleich ein Kanban erstellt werden. **TODO: Theorie über Kanban - evtl. mit Fachbegriffen verhängen**



Abbildungsverzeichnis 1: Klassisches Kanban auf *github.com*

1.2 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 ...

1.3 Projektplan

Projektplan²

¹Projekt auf *github.com*

²URL Google Spreadsheet

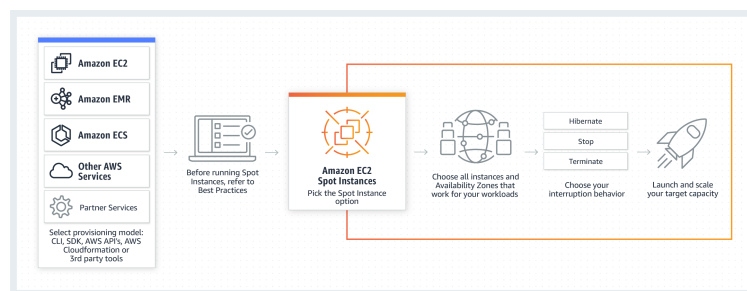
2 Vorarbeiten

2.1 AWS Account

- ▶ Fit werden mit AWS

2.2 AWS Spot Instances

- ▶ Herausfinden wie mit AWS Spot Instanzen gearbeitet werden kann



Abbildungsverzeichnis 2: AWS Architektur Spot[Web10]

2.3 Kubernetes testing

- ▶ Anfangs mit Minicube
- ▶ Eigenem AWS account

3 Ausgangslage

3.1 swisstopo bei AWS

3.2 Ist-Zustand 3D Prozessierung swisstopo

3.2.1 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

4 Use Case

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

4.1 Beschreibung der Technischen Komponenten

4.1.1 SPOT Instanzen

4.2 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.

5 Architektur

Ausfindigmachen von verschiedenen Architekturen

5.1 Bewertungskriterien

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

5.2 Variante 1: Spot

Evtl. lässt sich eine Lösung finden ohne EKS zu verwenden

5.3 Variante 2: Spot mit EKS

Spot mit EKS

6 Prototyp

6.1 Realisierung

7 Evaluation

7.1 Erfahrungen

7.2 Kritische Punkte

8 Zusammenfassung

9 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.
- [Web10] Webpage. (2010). Amazon ec2-spot-instances, [Online]. Available: <https://aws.amazon.com/de/ec2/spot/> (visited on 07/08/2010).

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

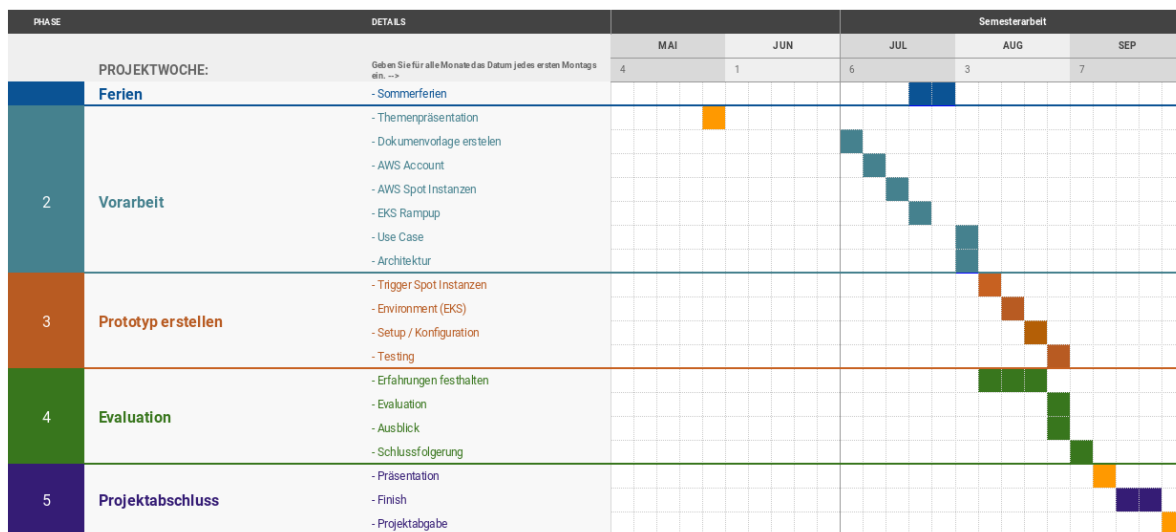
81 82

91 92

101 102

C Projektplanung

PROJEKTTITEL	Semesterarbeit Geodatenprozessierung mit Budget Instanzen
FULL STACK PROJEKTMANAGER	Tobias Reber



Abbildungsverzeichnis 3: 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