



Redesigning and Implementing the Bootstrap of Large Scale Kubernetes Enterprise Infrastructure through Automated Self Contained CLI

Project 2b (T3_2000)

in the context of the examination for the
Bachelor of Science (B.Sc.)

in Computer Science (Informatik)

at the Cooperative State University Baden-Württemberg Karlsruhe

by

Yannik Schiebelhut

Submission Date:	September 19, 2022
Processing period:	July 4, 2022 - September 19, 2022
Matriculation number, Course:	3354235, TINF20B1
Training institution:	SAP SE Dietmar-Hopp-Allee 16 69190 Walldorf, Deutschland
Supervisor of the training institution:	Samed Guener
Appraiser of the Cooperative State University:	Prof. Dr. Johannes Freudenmann

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Project 2b (T3_2000) mit dem Thema:

*Redesigning and Implementing the Bootstrap of Large Scale Kubernetes Enterprise
Infrastructure through Automated Self Contained CLI*

gemäß § 5 der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. September 2017 selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, den 2. August 2022

Schiebelhut, Yannik

Sperrvermerk

Die nachfolgende Arbeit enthält vertrauliche Daten der:

SAP SE
Dietmar-Hopp-Allee 16
69190 Walldorf, Deutschland

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anderslautende Genehmigung vom Dualen Partner vorliegt.

Abstract

- Deutsch -

Um unternehmensübergreifende Prozesse auf ein neues Niveau zu heben, arbeitet SAP an der Entwicklung von *Cross-Company Workflow Collaboration*. Die dort anfallenden, geteilten Prozessdaten sollen in *Signavio Process Intelligence* importiert werden, um sie dort mittels Process Mining zu analysieren und in Key Performance Indicators darzustellen. Da bei Cross-Company Workflow die *SAP Blockchain* zur Datenhaltung verwendet wird, stellt die Aufbereitung der Daten für eine Analyse in Signavio jedoch eine Herausforderung dar.

Im Rahmen dieser Projektarbeit werden zunächst diese Schwierigkeiten herausgearbeitet. Anschließend wird sich mit der Konzeptionierung eines Datenintegrationsprozesses befasst. Hierzu wird zum einen erörtert, wo die Transformation der Daten aus Cross-Company Workflow in das angestrebte Format durchgeführt werden sollte. Andererseits werden auch verschiedene Transportwege beleuchtet, über die die aufbereiteten Daten in Signavio Process Intelligence bereitgestellt werden sollen. Auf Basis des erarbeiteten Konzepts wird ein Prototyp der Datenintegration implementiert.

Abstract

- English -

In order to raise cross company processes to a new level SAP is working on the development of Cross-Company Workflow. The shared process data is to be imported into *Signavio Process Intelligence* to be analyzed with process mining and to be presented in Key Performance Indicators. Since Cross-Company Workflow uses the *SAP Blockchain* to store data the preparation of the data for analysis within Signavio is a challenge.

In the context of this report, these difficulties will first be discussed. It then deals with the conceptual design of a data integration process. For this purpose, on the one hand, it is discussed where the conversion of the data from Cross-Company Workflow into the target format should be carried out. On the other hand, different transport routes are discussed, through which the processed data should be made available in Signavio Process Intelligence. Based on the developed concept, a prototype of the data integration is implemented.

Contents

List of Abbreviations	VI
List of Figures	VII
List of Code Listings	VIII
1 Introduction	1
2 Fundamentals	2
2.1 Go	2
2.2 Cobra	2
2.3 Terraform	3
2.4 Hyperscalers	3
2.5 Kubernetes	3
2.6 Vault	3
2.7 Bootstrap	3
3 Evaluation and Future Work	4
Literaturverzeichnis	IX

List of Abbreviations

CLI command-line interface

List of Figures

List of Code Listings

1 Introduction

2 Fundamentals

2.1 Go

Go (also known as Golang) is an open source programming language that was started at Google in 2007 and initially launched in 2009. The language was designed to face engineering challenges at Google with the goal to make it “easy to build simple, reliable and efficient software”. [1, 2] By now the compiled and statically typed language [3] is widely used and the way it approaches network concurrency and software engineering has influenced other languages to a noticeable extent. [1] Through its structure go supports programming on various levels of abstraction. For instance, one can embed Assembler or C code into a Go program or on the other hand combine groups of components into bigger, more complex components to realize abstract design patterns. [4] Nowadays, Go is a popular choice for everything related to DevOps and therefor also for the development of command line tools. [5]

- toolchain (uniform formatter, included test suite, source code based generation of documentation)
- third party modules repo / go get
- extensive standard library
- (cite from golang.org tutorial)

2.2 Cobra

Cobra is an open source library for Go. Its aim is to provide developers with an easy way to create modern command-line interface (CLI) applications. The cobra library is being used by noticeable projects like the CLIs for Kubernetes or for GitHub. The idea behind Cobra’s intended command schema is that commands of a well constructed CLI should read like sentences. This way, new users that are familiar with CLIs in general

quickly feel native because interacting with the CLI feels more natural. In this approach, a command represents a certain action that the CLI can perform. This action then takes arguments and flags to further specify on which objects and in which way the command should take action. With Cobra, one can also easily create nested subcommands. This means that a before mentioned command can also be divided into multiple sub-actions to enable detailed handling of complex actions. Further benefits of Cobra are for example the automated generation of autocomplete for the most common shells as well as the ability to automatically create man pages. [6, 7]

2.3 Terraform

2.4 Hyperscalers

2.4.1 AWS

2.5 Kubernetes

2.6 Vault

2.7 Bootstrap

3 Evaluation and Future Work

Literaturverzeichnis

- [1] Pike, R. *Using Go at Google - The Go Programming Language*. 08/2020. URL: <https://go.dev/solutions/google/> (visited on 08/02/2022).
- [2] *The Go programming language — GitHub*. URL: <https://github.com/golang/go> (visited on 08/02/2022).
- [3] Chris, K. *What is Go? Golang Programming Language Meaning Explained*. 10/2021. URL: <https://www.freecodecamp.org/news/what-is-go-programming-language/> (visited on 08/02/2022).
- [4] Maurer, C. *Objektbasierte Programmierung mit Go*. Springer Vieweg, 2021.
- [5] Van Sickle, M. *Go: Getting Started — Pluralsight*. 01/2020. URL: <https://app.pluralsight.com/library/courses/getting-started-with-go/table-of-contents> (visited on 08/02/2022).
- [6] *A Commander for modern Go CLI interactions — GitHub*. URL: <https://github.com/spf13/cobra> (visited on 08/02/2022).
- [7] *Cobra. Dev*. URL: <https://cobra.dev/> (visited on 08/02/2022).