

Sem vložte zadání Vaší práce.



**FAKULTA
INFORMAČNÍCH
TECHNologiÍ
ČVUT V PRAZE**

Bakalářská práce

Konfigurační management mikroslužeb

Josef Vávra

Katedra softwarového inženýrství

Vedoucí práce: Martin Komárek

31. ledna 2022

Poděkování

Doplňte, máte-li komu a za co děkovat. V opačném případě úplně odstraňte tento příkaz.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 31. ledna 2022

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2022 Josef Vávra. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Vávra, Josef. *Konfigurační management mikroslužeb*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

Klíčová slova Nahrad'te seznamem klíčových slov v češtině oddělených čárkou.

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahrad'te seznamem klíčových slov v angličtině oddělených čárkou.

Obsah

Úvod	1
1 Cíl práce	5
2 Analýza a návrh	7
3 Realizace	9
Závěr	11
Literatura	13
A Seznam použitých zkratek	15
B Obsah přiloženého CD	17

Seznam obrázků

Úvod

Konfigurační management(CM) je efektivní strategie určená ke stanovení a udržování konzistence výkonu, funkčních a fyzických atributů produktu s jeho designem, požadavky a provozními informacemi po celou dobu jeho životnosti. Úkony původně zavedené americkou armádou jsou dnes rozšířeny do mnoha rozlišných odvětví. Jedním z těchto odvětví jsou informační technologie. CM je definován v IT Infrastructure Library (ITIL). Jedná se o proces, který stopuje jednotlivé konfigurační položky(CI). To mohou být jednotlivé servery, nebo komplexní clustery v korporátním prostředí obřích rozměrů. Samotná implementace je pak dosažena a pomocí různých nástrojů. CM indentifikuje systémy, které potřebují záplatu, aktualizaci nebo rekonfiguraci a zajistí, aby se dostaly do požadovaného stavu. Manuální úpravy velkých systémů mohou být velmi náročné. Tyto úpravy mohou vyžadovat aktualizaci až mnoha stovek systémů. Pokud tento proces není zautomatizovaný, organizace, která ho má na starost rychle ztratí přehled o změnách které už proběhly a těch, které je teprve nutné vykonat. To vede k prostojům, nepřístupnosti, nestabilitě systému nebo k celkovému selhání. CM definuje automatizaci, která zaručuje konzistenci. Ta je nezbytná při dodávání kvalitních služeb. CM pomáhá zabránit nezdokumentovaným změnám, které mohou způsobit, že je systém nesprávně nakonfigurovaný. Když dojde k selhání a následné nefunkčnosti, téměř vždy se jedná o ztrátu zisku. Tento problém nemusí mít jednoznačné řešení, které by byl jednotlivec schopný vyřešit v čase, který je pro firmu přijatelný. CM tedy podle předem daných definic dovede chybný systém nahradit novým znovu vytvořeným. Zreplikované systémy pak umožňují škálování služby a roz distribuovat zátěž, tak aby nedocházelo výpadkům. Konfiguračním managementem se už zabývalo několik prací.

V roce 2009 na VŠE vznikla práce s názvem Konfigurační management. Jejím autorem je Robert Pelikán. Práce se zabývá analýzou komponent CM a zaměřuje se spíš na rozhodovací proces pro firmy, které se snaží vyhodnotit zda je CM pro jejich podnikání správnou cestou. V době práce se konfigurační

management ukládal ještě do CMDB.

Jie Ma v práci Configuration Management Improvement z roku 2010 hledá nedostatky a mezery v CM. Zaměřuje se na konkrétní strukturu firmy Philips Healthcare, kterou zkoumá.

Práce Marka Bokše z Masarykovy Univerzity byla napsána v roce 2006 a v jejím hledáčku jsou hlavně verzovací systémy a úvod do nasazování verzované aplikace.

Josef Nezbeda vypracoval pro Saimaa University v roce 2011 práci, která se zaměřuje na nalezení ideální údržby Konfiguračního management systému (CMS). Je postavena na základě doporučení z ITIL. Je opět z doby, kdy se pro uložení konfigurací používal CMDB a pohybuje se spíše v teoretické rovině.

Jan Šavel v roce 2017 pro VŠE práci s názvem Redesign Configuration management procesu ve vybrané organizaci, která analyzuje existující CM v konkrétní firmě a v praktické části se zabývá jeho reimplementací.

Azadeh Jahanbanifar ve své práci z roku 2016 prezentuje framework založený na modelech, který umožňuje integraci konfiguračních fragmentů.

Práce, jejímž autorem je JOHAN KRISTENSSON, vznikla také v roce 2016 a zajímá se o optimalizaci procesu. Sleduje jak se zlepšuje kvalita práce po zavedení CM v organizaci.

Tyto práce se zabývají od této práce odlišnou částí konfiguračního managementu.

Moje práce se bude zabývat převážně praktickou částí konfiguračního managementu mikroslužeb. Mikroslužba je architektonický návrh pro vytváření distribuované aplikace. Tato architektura umožňuje rychlé, časté a spolehlivé dodávky změn v softwaru. Kromě toho také může každá služba škálovat nebo být aktualizována bez přerušení ostatních služeb. Ke svojí práci budu potřebovat aplikaci, složenou z několika mikroslužeb. Bude se jednat o eshop, který bude připojen na databázi. Cílem samotné práce pak bude předvést jak migrovat celou aplikaci z lokálního prostředí, přes vývojové, akceptační a nakonec produkční. K této úloze budu využívat hned několik nástrojů. Mikroslužby budou napsané v javě za pomoci Spring frameworku. Tento framework jsem zvolil, protože s ním mám nejvíce zkušeností.

Dále jsem za svou databázi zvolil mongoDB. Rozhodl jsem se na základě těchto článků:

<https://www.integrate.io/blog/which-database/>

zde jsou vyzdvíženy vlastnosti jako je vysoká flexibilita a dobrá spolupráce s webovými aplikacemi

MongoDB má dobrou dokumentaci, takže při práci budu čerpat hlavně z ní.

<https://www.mongodb.com/what-is-mongodb/features>

Mongodb podporuje loadbalancování ve velkém měřítku, což je vlastnost která bude vzhledem k povaze práce důležitá

Svou práci budu vyvíjet nad kontejnery, které mi výrazně ulehčí posuny mezi prostředími.

Kontejner je soubor kódu a jeho závislostí zabalen v lightweight prostředí, které umožňuje umlčet veškeré okolní negativní vlivy na jeho chod. V tomto směru mám zkušenosti pouze s technologií docker, která je však zároveň nejrozsáhlejší.

Ze stejného důvodu si vybírám orchestrační nástroj kubernetes, který má za úkol řízení kontejnerů.

Dokumentace dockeru nemá příliš dobrou reputaci. Určitě se na ní budu při práci odkazovat, ale některé detaily budu dohledávat jinde, například na <https://docker-curriculum.com/>

Dokumentace kubernetes je ve stavu, který je dostatečný pro úspěšnou implementaci této práce.

Na téma kubernetes bylo také napsáno několik prací. Každá z nich však proniká (stejně jako výše zmíněné práce) do této práce jen minimálně.

Práce s názvem Kubernetes as an Availability Manager for Microservice Based Applications od Leila Abdollahi Vayghan se zajímá o kubernetes. Jejím výstupem je komponenta, která integruje kubernetes a má za úkol zlepšení obnovy systému.

Další je Analyzing the Applicability of Kubernetes for the Deployment of an IoT Publish/Subscribe System od Maryam Tavakkoli se zase zabývá pouze IoT světem.

O dockeru byly napsány ve světě tyto práce, které podle mě stojí za zmínku.

Virtualization using Docker Platform od Vladimíra Jurenka je práce z Masarykovy univerzity, která se zabývá spíše virtualizací jako takovou. Docker je pak zkoumán jako nástroj, nikoliv použit jako prostředek.

SOFTWARE CONTAINERIZATION WITH DOCKER od Rubén Cayetano Díaz Alonso z Turku Univerzity je spíše o prozkoumávání virtualizace a porovnávání nástrojů, které ji umožňují.

Práce Dominika Víška z Pardubické Univerzity s názvem Využití technologie Docker pro virtualizaci vývojového prostředí PHP / Python opět porovnává virtualizační nástroje a následně navrhuje virtualizované vývojové prostředí.

Žádná z nich se svým obsahem nepřibližuje k této práci jinak než použitím stejných nástrojů.

Proces posouvání aplikace po prostředích se neobejde bez automatizace. K tomuto úkonu existuje hned několik nástrojů. Jsou jimi například Chef, Puppet nebo Ansible. Nástrojů je více, ale já se rozhodoval hlavně mezi těmito. Ve své práci budu pracovat s Ansible. Narozdíl od ostatních nevyžaduje znalost Ruby. Jeho scripty jsou psané v JSON, což mi značně ulehčí práci.

Při vybírání mi pomohli tyto články:

<https://www.simplilearn.com/configuration-management-tools-article>

<https://opensource.com/resources/what-ansible>

<https://www.devopsgroup.com/blog/puppet-vs-ansible/>

<https://www.simplilearn.com/ansible-vs-chef-differences-article>

O Ansible vznikla na ČVUT práce Automatizace konfigurace síťových prvků pomocí Ansible od Miroslava Hudece, která zkoumá jak nakonfigurovat síťové zařízení MikroTik.

Docker je populární téma závěrečných prací na ČVUT, a proto jich tu vzniklo hned několik.

Práce Application of containers in continuous software development od Le Henrich byla stvořena na FITu. Jedná se o práci, která se zaměřila na psaní pipeline.

Práce s názvem Zajištění škálovatelnosti webových aplikací s využitím architektury mikroslužeb od Lukáše Hamrly z FITu se v některých ohledech této práci podobá. Práce má však za úkol rozebrání škálovatelnosti aplikací. V pozdějších kapitolách se věnuje zkoumání dockeru.

Další prací je Continuous Integration and Continuous Delivery of addictologic web application od Jany Čikelové. Tato práce je zaměřena zejména na monitoring a provoz aplikací při vysokých zátěžích.

Adam Staněk vytvořil práci Systém pro automatizaci nasazování aplikací. Je v ní popsán agilní přístup a nakonec naimplementován CI systém.

Žádná z těchto prací se nijak významným způsobem nepřekrývá s cílem této práce.

Cíl práce

Analýza a návrh

Realizace

Závěr

Literatura

Seznam použitých zkratk

GUI Graphical user interface

XML Extensible markup language

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	exe	adresář se spustitelnou formou implementace
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu L ^A T _E X
	text	text práce
	thesis.pdf	text práce ve formátu PDF
	thesis.ps	text práce ve formátu PS