

Sem vložte zadání Vaší práce.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Konfigurační management mikroslužeb

Josef Vávra

Katedra softwarového inženýrství

Vedoucí práce: Martin Komárek

10. února 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 10. února 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 Nahraďte seznamem klíčových slov v češtině oddělených čárkou.

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahraďte seznamem klíčových slov v angličtině oddělených čárkou.

Obsah

Úvod	1
1 Best practices konfiguračního managementu v cloud native prostředí	3
2 Cíl práce	5
3 Analýza a návrh	7
4 Realizace	9
Závěr	11
Literatura	13
A Seznam použitých zkratk	15
B Obsah přiloženého CD	17

Seznam obrázků

Úvod

Podle The twelve-factor app[1] je konfigurace aplikace vše co se liší mezi prostředími (testovací, produkční, vývojové prostředí, atd.). To mohou být přístupové údaje k externím službám, přihlašovací údaje k databázím, nebo třeba kanonické názvy hostitelů.

Konfiguraci je možné ukládat jako konstanty v kódu. Je to však vysoce nežádoucí. Jedná se o porušení twelve-factor metodiky. Tato metodika vyžaduje oddělení konfigurace od kódu. I když kód zůstává stejný, konfigurace se většinou mezi nasazeními výrazně odlišuje.

Každá aplikace má alespoň dvě prostředí: testovací a produkční. Každá taková aplikace má pak zhruba čtyři až pět konfiguračních parametrů. Pokud tyto dvě čísla vynásobíme, máme už deset parametrů. Toto číslo roste rychle, podle toho jak se rozšiřuje naše aplikace. Roste ještě rychleji podle toho, jak se zvyšuje počet prostředí (viz. článek o gitOps[2])

Konfiguracím je tedy nutné věnovat pozornost. Podle článku[3] od RedHat jsou nesprávné konfigurace hlavní příčinou bezpečnostních incidentů v kontejnerizovaném Kubernetes-prostředí. Nesprávné konfigurace vedou ke slabému výkonu, nekonzistenci, nefunkčnosti a mají negativní dopad na business logiku a bezpečnost. Pokud jsou nezdokumentované změny prováděny napříč mnoha systémy a aplikacemi, zvyšuje se riziko nestability a prostojů. Manuální řešení této problematiky vyžaduje identifikaci systémů, které potřebují pozornost. Poté rozhodnutí, které kroky je potřeba podniknout k napravení problému a rozhodnout které úkony mají prioritu. Validace správného výsledku je v komplexních systémech příliš komplikovaná. Bez dokumentace kroků, které byly provedeny jsou navíc systémoví administrátoři a softwaroví vývojáři neschopni rozlišit, který server byl zaktualizován a jaké verze aplikací se kde nacházejí.

Tomu se dá však zabránit. Řešením tohoto problému je konfigurační management. Ten umožňuje definovat nastavení systému. Cílem je udržet aplikaci ve stavu, který je sledovatelný a v funkční. Proč je konfigurační management

důležitý, je rozebráno ve článku[4] od Atlassian. Konfigurační management podle textu pomáhá vývojářským týmům vytvořit robustní a stabilní systém, který automaticky řídí a monitoruje aktualizace konfiguračních položek. Jako příklad uvádí mikroservisní architekturu. Každá mikroslužba má vlastní konfigurační parametry, které jsou použity při inicializaci. Jde jak o hardwarové požadavky jako je CPU nebo RAM, tak softwarové specifikace jako jsou hesla a endpointy. Konfigurační management tedy umožňuje vytvořit, aktualizovat a monitorovat jediný "zdroj pravdy" tím, že jsou veškeré informace na jednom místě.

Bohužel konfigurační management není žádná zázračná technika, která zadarmo a v krátkém čase přetvaruje libovolný systém na bezvýpadkový produkt, který má bezproblémový release a každý proces, který se odehraje je striktně kontrolován. Implementaci je nutné věnovat prostředky, jak na začátku vývoje, tak i v jejím průběhu. Jak je uvedeno v článku[5] od UpGuard je vyžadující zaškolení současné zaměstnance v oblasti konfiguračních procesů a vyčlenit někoho na stanovení zmíněných procesů a nastavení potřebných nástrojů.

V teoretické části této práce popíšu best practices konfiguračního managementu v cloud native prostředí a jak přistupovat ke konfiguračním parametrům v aplikacích. Co jsou secrets a jaké jsou možnosti šifrování. Jedna z kapitol se bude věnovat problému několika prostředí, které v rámci práce vytvořím. Paralelně k této práci vytvořím aplikaci, podle mikroslužební architektury definovanou zde[6]. Cílem této aplikace je možnost demonstrovat migraci mezi vytvořenými prostředími na konkrétním příkladu. Celý proces bude v jedné kapitole zdokumentován. K této práci přistupuji jako k Proof of concept[7]. Čtenáře provedu postupem od implementace aplikace až po její úspěšné nasazení a údržbu.

Best practices konfiguračního managementu v cloud native prostředí

Jako v každé jiné softwarové disciplíně i v SCM je nutné dodržovat určité postupy, neboli best practices. Podle textu s názvem BEST-PRACTICE RECOMMENDATIONS CONFIGURATION MANAGEMENT od Sun Services[8] jsou jedněmi z hlavních doporučení tyto: udržování kontroly, Sledování správných konfiguračních položek a vytvoření repozitáře.

Pod pojmem udržování kontroly je myšleno, že je nutné pečlivě hlídat konfigurační položky a udržovat systém aktuální. V momentě, kdy je systém zastaralý, přestává být užitečný.

Jak už bylo zmíněno v předchozím bodě, je další doporučení sledování správných konfiguračních položek. Pokud pracujeme s komplexním systémem, je velice pravděpodobné, že při potřebě mít kontrolu nad každým detailem, budeme zahlceni. Je proto důležité, sledovat ty položky které je potřeba a u kterých je to možné.

Vytvoření repozitáře je nezbytné pro správný chod celého systému. Pokud nemáme kam ukládat informace o konfiguračních položkách, nebude systém udržitelný.

Myšlenku udržování kontroly potvrzuje kniha Configuration management best practices[?]. Autoři v ní tvrdí, že cílem dobrého konfiguračního managementu je, mít jednoduše identifikovatelný veškerý kód, který se dostal do produkčního prostředí a nazývají to konfigurační audit.

Identifikaci stanovuje microfocus[9] ve druhém kroku v procesu při tvorbě produktu. Prvním krokem je podle této stránky plánování, které spočívá ve stanovení rozsahu a rozdělení rolí a delegaci povinností. Poté následuje zmíněná identifikace. Ta spočívá v určení a registraci konfiguračních položek a zaznamenání jejich vzájemných vztahů. Následující krok je řízení a definují

1. BEST PRACTICES KONFIGURAČNÍHO MANAGEMENTU V CLOUD NATIVE PROSTŘEDÍ

se v něm stavy aplikace před modifikací a po modifikaci. Když se pak provede změna, je možné ověřit, zda je aplikace ve správném stavu. Čtvrtým krokem je reportování. Report by měl obsahovat identifikaci položky, její stav, verzi a příslušnou dokumentaci. Musí být přesné a aktuální. Posledním krokem je verifikace a audit. Na něj padá zodpovědnost za ověření, že každá informace v celém procesu je pravdivá a že veškeré potřebné položky jsou identifikovány a sledovány.

Cíl práce

Analýza a návrh

Realizace

Závěr

Literatura

- [1] The twelve-factor app. 2017. Dostupné z: <https://12factor.net/cs/config>
- [2] Cloud-Native Configuration Management by GitOps. 2020. Dostupné z: <https://medium.com/trendyol-tech/cloud-native-configuration-management-by-gitops-6-2401bf47418>
- [3] What is configuration management. 2019. Dostupné z: <https://www.redhat.com/en/topics/automation/what-is-configuration-management>
- [4] Why is configuration management important? Dostupné z: <https://www.atlassian.com/continuous-delivery/principles/configuration-management>
- [5] What Is Configuration Management and Why Is It Important? 2022. Dostupné z: <https://www.upguard.com/blog/5-configuration-management-boss>
- [6] Microservice Architecture. Dostupné z: <https://microservices.io/index.html>
- [7] Proof of concept. Dostupné z: https://en.wikipedia.org/wiki/Proof_of_concept
- [8] Sun Services: *BEST-PRACTICE RECOMMENDATIONS CONFIGURATION MANAGEMENT*. 2007.
- [9] Configuration Management Workflows. Dostupné z: https://docs.microfocus.com/SM/9.61/Hybrid/Content/BestPracticesGuide_PD/ConfigurationManagementBestPractice/Configuration_Management_WorkFlows.htm

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