Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Diplomová práce

Automatizace správy vybraných IT procesů

Lukáš Koucký

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Lukáš Koucký

Informatika

Název práce

Automatizace správy vybraných IT procesů

Název anglicky

Automation of management of selected IT processes

Cíle práce

Hlavním cílem práce je automatizace zvolených procesů ve vybrané instituci na základě analýzy prostředí s využitím nástroje Ansible v souladu s ITIL.
Dílčí cíle:
Analyzovat stávající prostředí a procesy v konkrétní vybrané instituci a identifikovat oblasti, které je možné automatizovat.
Navrhnout řešení pro automatizaci správy operačních systémů pomocí nástroje Ansible v souladu s procesy ITIL.
Implementovat navržené řešení a otestovat jeho funkčnost, zhodnotit výsledky a přínosy zavedení automatizace správy operačních systémů v souladu s ITIL.

Metodika

V teoretické části diplomové práce bude provedena studie a analýza odborných a vědeckých zdrojů týkajících se automatizace procesů. Bude vytvořen přehled řešené problematiky v oblasti automatizace IT procesů a jejich optimalizace s využitím moderních nástrojů. Součástí teoretické části bude také charakteristika metodiky ITIL, metodiky ISO 207001 a její vztah k metodice ITIL. Následně bude charakterizován a hodnocen nástroj Ansible, který se využívá pro automatizaci IT procesů a zároveň bude hlavním nástrojem samotné praktické části.

V praktické části diplomové práce se provede analýza prostředí instituce, která bude předmětem automatizace. V rámci inventarizace bude zjištěno, jaké jsou zdroje dat a jaké procesy jsou v současné době používány. Bude provedena také analýza dostupných nástrojů a technologií pro automatizaci IT procesů. Následně se vybere proces vhodný pro automatizaci a určí se jeho klíčové body. Zváží se, zda je proces vhodný pro plnou automatizaci nebo zda je potřeba částečná automatizace. Následně bude automatizován vybraný proces, případně jeho klíčové body a v praxi otestován.

Na základě syntézy poznatků teoretické části a vyhodnocení výsledků praktické části budou formulovány závěry práce. V této části bude poskytnuto doporučení pro další kroky automatizace v instituci. Bude zvažováno, jaké další procesy je vhodné automatizovat a jak lze procesy optimalizovat pomocí automatizace. Bude také zvažováno, jaké další nástroje a technologie by mohly být použity pro automatizaci IT procesů v budoucnu.

Doporučený rozsah práce

60 – 70 stran

Klíčová slova

automatizace, Ansible, státní správa, ITIL, IT proces

Doporučené zdroje informací

HABIBI, Shahryar. Building Automation and Digital Technologies. Cambridge, MA: WoodHead Publishing, 2022. ISBN 9780128221297.

HEAP, Michael. Ansible: From Beginner to Pro. Imprint: Apress, 2016. ISBN 9781484216590.

ITIL® Foundation, ITIL 4 edition, 2019. ISBN: 978-0-11-331607-6

SESTO, Vincent. Practical Ansible: Configuration Management from Start to Finish: Apress, 2021. ISBN 9781484264850

Předběžný termín obhajoby

2023/24 LS – PEF

Vedoucí práce

doc. Ing. Jiří Vaněk, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 10. 7. 2023

doc. Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 3. 11. 2023

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 30. 03. 2024

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Automatizace správy vybraných IT procesů" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30. března 2024 _____

Poděkování

Rád bych touto cestou poděkoval doc. Ing. Jiří Vaňkovi, Ph.D. za cenné rady a motivaci během tvorby mé práce.

Automatizace správy vybraných IT procesů

Abstrakt

Cílem diplomové práce je automatizace zvolených IT procesů ve vybrané instituci na základě analýzy prostředí s využitím nástroje Ansible v souladu s Information Technology Infrastructure Library. Práce se skládá z teoretické a vlastní části práce. Teoretická část je zpracována na základě studia odborné literatury a internetových zdrojů v oblasti automatizace a moderních trendů v oblasti automatizace. Následně je charakterizována metodika Information Technology Infrastructure Library podle níž se IT oddělení v organizaci řídí a veškeré postupy jsou dle ní provozovány. V závěru teoretické části je rozebrán nástroj Ansible, jeho playbook, a role, které se staly důležitou součástí vlastní práce. Ve vlastní části práce je nejdříve provedena analýza prostředí organizace, ze které jsou zjištěny zdroje dat (konfigurace linuxových serverů), pro následné kroky. Dále je provedena analýza IT procesů organizace, ze které jsou vybrány tři hlavní procesy, u kterých bylo rozhodnuto o automatizaci v rámci této diplomové práce. Jedná se o správu repozitářů, serverů a administrátorských uživatelů. Nejdříve je rozebrána správa repozitářů, která vychází z analýzy prostředí. Poté je provedena analýza dostupných nástrojů, které lze využít společně s Ansible. Z této analýzy vyšel nejlépe nástroj Foreman, který je využit. Následně se autor zaměřuje na správu serverů. Nejprve na skupiny hostů, které jsou důležité pro rozdělní serverů dle podobnosti a zároveň pro parametry využívané v jednotlivých rolích a celkově v celém pojetí konfiguračního managmentu. Poté jsou ukázány způsoby přidávání serverů do nástroje Foreman. Jako poslední oblastí je správa administrátorských uživatelů operačního systému Linux, kde vznikla Ansible role pro vytváření, rušení a aktualizaci uživatelů na serverech. Na závěr práce je představen současný stav a budoucí vývoj automatizace v organizaci.

Klíčová slova: automatizace, Ansible, státní správa, ITIL, IT proces, Foreman

Automation of management of selected IT processes

Abstract

The goal of this diploma thesis is the automation of selected IT processes within the chosen institution, based on analysis of the environment using the tool Ansible, consistent with the Information Technology Infrastructure Library. This paper consists of a theoretical part and the practical part of the work. The theoretical part is prepared based on a study of specialized literature and internet resources in the field of automation and modern trends in this area. Following that, the Information Technology Infrastructure Library methodology is described, according to which the IT department in the organization is working and all procedures are operated. At the conclusion of the theoretical part the tool Ansible is described, its playbook and role which was the important part of my paper. In the practical part of the work is firstly done an analysis environment of the organization from which are found data sources for following steps. The analysis of IT processes is then made, from which are chosen three main processes which were decided to be automated within this work. Those ended up being management of repositories, servers and users. Initially, the management of repositories, which is based on an analysis of the environment, is described. Then an analysis of the available tools that can be used together with Ansible, is performed. From this analysis, the tool Foreman came out as the best and is used. Then the author focuses on the server management. Firstly, group of hosts are examined as they are important. for the partitioning of servers which is done according toto the similarity, as well as defining of prameters used in individual roles further in overall concept of management configuration. Then are demonstrated ways of adding servers into the Foreman tool. The final part is the administration of users, for which the Ansible role was created for forming, terminated and updating user accounts on servers. In conclusion of this thesis, the current status and future development of automation within the organization is introduce.

Keywords: automation, Ansible, state administration, ITIL, IT process, Foreman

Obsah

1 Úvod

V oblasti informačních technologií zaujímá automatizace důležité místo. V současné velmi dynamické době se stále více organizací snaží hledat různá efektivní řešení, jak zlepšovat a optimalizovat veškeré své procesy a činnosti. Většina firem k tomu využívá právě automatizaci. Tato práce se zabývá tématem automatizace vybraných IT procesů, které se v dané organizaci řídí metodikou ITIL, která ukládá jeden ze základních rámců v oblasti řízení IT služeb.

Teoretická část práce se nejdříve zaměřuje na samotný pojem automatizace, její význam v aktuálním světě IT. Dále jsou popsány moderní trendy v oblasti automatizace IT procesů. Metodika ITIL dále hraje klíčovou roli v řízení IT procesů, a proto je podrobně rozebrána. Zároveň se organizace, ve které byla vlastní část práce implementována řídí podle této metodiky a získala v tomto směru certifikaci.

Dalším důležitým bodem teoretické části je správa konfigurací a nástroj Ansible, který slouží k automatizaci IT procesů. Popsány jsou základní komponenty, jako playbook a role v rámci Ansible, které se staly klíčovými prvky při následné tvorbě praktické části.

Praktická část práce se nejdříve zabývá nalézáním dalších možných nástrojů, které společně s Ansible lze využít pro automatizaci IT procesů. Následuje analýza prostředí a zároveň zdrojů dat, které jsou v práci využita. Z analýzy jsou vybrány tři hlavní IT procesy, které jsou určeny pro samotnou automatizaci: správa repozitářů, správa serverů a správa administrátorských uživatelů operačního systému Linux.

Následně už je popsána samotná implementace, u které je nejdříve nutné určit architekturu. Následuje instalace nástroje Foreman, který se stal klíčovým nástrojem pro správu repozitářů, serverů a administrátorských uživatelů operačního systému Linux. Praktická část obsahuje také popis struktury host group a proces automatické registrace serverů pomocí rolí Ansible. V poslední části vlastní práce je implementována automatizace zakládání administrátorských uživatelů operačního systému Linux.

Celkově lze konstatovat, že se práce zaměřuje na problematiku automatizace IT procesů a její implementace ve vybrané organizaci. Zároveň se jedná o organizaci, která spravuje systém kritické infrastruktury, proto autor práce nemohl ukazovat, jakékoliv ukázky ze samotného nástroje Foreman.

Autor práce si toto téma vybral z důvodu zájmu o oblast automatizace. Diplomová práce řeší jednu z částí širšího projektu automatizace správy IT v organizaci, a to správu vybraných linuxových serverů.

V současné době je systém plně implementován na testovacím prostředí organizace. Na produkční prostředí bude však systém postupně implementován. Po vyřešení několika počátečních problémů je řešení plně funkční a postupně je využíváno v každodenním provozu.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavním cílem práce je automatizace zvolených procesů ve vybrané instituci na základě analýzy prostředí s využitím nástroje Ansible v souladu s ITIL.

Dílčími cíli jsou:

1. Analyzovat stávající prostředí a procesy v konkrétní vybrané instituci a identifikovat oblasti, které je možné automatizovat.
2. Navrhnout řešení pro automatizaci správy operačních systémů pomocí nástroje Ansible v souladu s procesy ITIL.
3. Implementovat navržené řešení a otestovat jeho funkčnost, zhodnotit výsledky a přínosy zavedení automatizace správy operačních systémů v souladu s ITIL.

2.2 Metodika

V teoretické části diplomové práce bude provedena studie a analýza odborných a vědeckých zdrojů týkajících se automatizace procesů. Bude provedena literární rešerše v oblasti automatizace IT procesů a jejich optimalizace s využitím moderních nástrojů. Tato část bude zaměřena na teoretický základ automatizace procesů, seznámení s principy automatizace, definicí klíčových pojmů a výzkum současných trendů a postupů v oblasti automatizace.

Součástí teoretické části diplomové práce bude popis metodiky ITIL, která je v současnosti jednou z nejrozšířenějších metodik pro správu IT služeb. Bude popsán celý cyklus služby ITIL, tedy strategie, návrh, provoz a kontinuální zlepšování služby. Dále bude popsána metodika ISO 27001 a její vztah k ITIL. Budou popsány klíčové prvky obou metodik, jako jsou procesy, funkce a role a jejich vztah k automatizaci procesů. Dále bude popsán nástroj Ansible, který bude použit pro automatizaci procesů. Bude popsána jeho architektura, funkcionality a způsob práce s tímto nástrojem. Budou vysvětleny základní pojmy jako jsou inventář, playbook a modul a jaký mají význam při automatizaci IT procesů

V praktické části diplomové práce se provede analýza prostředí instituce, která bude předmětem automatizace. V rámci inventarizace bude zjištěno, jaké jsou zdroje dat (konfigurace linuxových serverů) a jaké procesy jsou v současné době používány.

Bude provedena také analýza dostupných nástrojů a technologií pro automatizaci IT procesů. Následně bude vybrán proces vhodný pro automatizaci a určí se jeho klíčové body. Zváží se, zda je proces vhodný pro plnou automatizaci nebo zda je potřeba částečné automatizace. Dále bude automatizován vybraný proces, případně jeho klíčové body a v praxi otestován.

Na základě syntézy poznatků teoretické části a vyhodnocení výsledků praktické části budou formulovány závěry práce. V této části bude poskytnuto doporučení pro další kroky automatizace ve firmě. Bude zvažováno, jaké další procesy je vhodné automatizovat a jak lze procesy optimalizovat pomocí automatizace. Bude také zvažováno, jaké další nástroje a technologie by mohly být použity pro automatizaci IT procesů v budoucnu.

3 Teoretická východiska

3.1 Automatizace

Pojmem automatizace označujeme důležité procesy, které slouží pro zvýšení produktivity práce, kvality výroby produktů a pomáhají organizacím zvýšit svou konkurenceschopnost. Automatizace dále může pomáhat v oblasti zabezpečení větší bezpečnosti. Další oblastí, kde je její nasazování velice přínosné, jsou provozy s těžkými manuálními pracemi. Tam postupně z části nahrazují tuto práci stroje. Automatizace se však postupně rozšiřuje do i do dalších odvětví. Lze konstatovat, že v současné době se s ní setkáváme skoro všech odvětvích. (Kolektiv autorů, 2012)

Zahrnuje soubor aktivit, které se zabývají navrhováním a implementací opatření, které umožňují automatické provádění úkonů. Tyto úkony se nacházejí v průběhu celého výrobního procesu například při ovládání strojů, výpočtech v průběhu řízení provozních parametrů strojů, optimalizaci provozu strojů a jejich zastavení. Stroje zde nahrazují, usnadňují, urychlují a zpřesňují lidskou práci. Avšak pokud se stroje používají hlavně k odstranění namáhavých a opakujících se úkolů, hovoříme pouze o mechanizaci. (Kolektiv autorů, 2012)

Termínem řízení rozumíme záměrné zásahy, které s cílem dosáhnout požadovaného výsledku ovlivňují události a procesy. Speciálním typem je automatické řízení. Je jím myšleno řízení, které funguje bez nutného zásahu lidí. (Balátě, 2003)

Dalším pojmem, se kterým se v automatizaci setkáváme, je řídicí algoritmus. Ten představuje univerzální popis metody, jak efektivně řešit danou řídicí úlohu přes postupné používání základních operací či kroků. Aby bylo dosaženo správného postupu musí se zajistit proveditelnost a bezpečnost řídicího cíle. Dnes se nejčastěji řídicí algoritmy dávají do programů řídicích jednotek automatů, čímž se dosahuje velkého stupně variability. Tento typ automatizace je označován jako flexibilní automatizace. Výhodou flexibilní automatizace je, že lze snadno, rychle a za vynaložení nejnižších nákladů vyměnit program automatu. Opakem je nepružná automatizace, která označuje technická řešení, kdy je program v automatu statický a jeho výměna je obtížná a značně nákladná. Obvykle to vyžaduje zásahy do vnitřního uspořádání a konfigurace zařízení (jako například výměna mechanických součástek nebo nastavení mechanických parametrů). Dalším rozlišením je dělení na částečnou automatizaci, kdy jsou automatizovány pouze konkrétní dílčí úkony

řízení, a pak komplexní automatizaci, při které je celý řízený proces plně automatizován a člověk do procesu zasahuje pouze v rámci strategického rozhodování. (Balátě, 2003)

Efektivní řízení procesů samozřejmě vyžaduje schopnost nejen technicky provést změny v rámci daného procesu, ale také dovednost získat a analyzovat informace o důsledcích těchto kroků. Tyto informace je nutné dále zhodnotit a případně se musí navrhnout změny v řídicím algoritmu. (Kolektiv autorů, 2012)

Vynucená automatizace nastává v situacích, kdy je náhrada lidské práce stroji či automaty způsobena určitými podmínkami. Důvody, proč je vhodné, aby lidská účast v těchto procesech byla nahrazena automatizací, jsou například:

1. Extrémní nebezpeční pro člověka. Mnohdy při těchto činnostech může docházet i k smrtelným úrazům. Například se jedná o práci s radioaktivními materiály, o práce v extrémních hloubkách, či v jakýkoliv extrémních podmínkách.
2. Pravděpodobnost vážných lidských chyb, při kterých může dojít k ohrožení lidských životů. Například u letecké dopravy, kde je využívána automatická navigace při složitých povětrnostních podmínkách.
3. Při nedostatečné efektivitě práce lidí a zároveň v případě, kdy je kvalita lidské práce nižší než práce strojů. Často k této situaci dochází, když je práce lidí příliš pomalá a nepřesná. Například výroba mikročipů, lakování karoserie aut.
4. Málo lidských sil v daném odvětví, proto je nutné daný proces automatizovat, aby byl zachován jeho plynulý chod. Příkladem je automatizace prodeje jízdenek na veřejnou dopravu (Kolektiv autorů, 2012)

Ekonomické důvody automatizace:

1. Použití automatických zařízení znamená snížení nákladů ve srovnání s manuální výrobou. To zahrnuje úspory na mzdách (lidská práce je drahá) a materiálech (přesná výroba generuje méně odpadu).
2. Snižuje náklady spojené s prostory, energetickou spotřebou a administrativními pracemi.
3. Přechod na automatizované řešení může vést k větší produktivitě práce, a tudíž větší efektivitě. Za stejný čas je schopen stroj vytvořit více výrobků než člověk.
4. Může poskytovat konkurenční výhodu oproti ostatním organizacím. Jedná se hlavně o rychlost k přístupu k informacím, trhu a potřebám zákazníků.

Další důvody automatizace:

1. Faktor prestiže, kdy organizace ukazuje svou vyspělost. Například zapojení robotů ve výrobě.
2. Zvyšování komfortu zákazníka. Například možnost automatických aktualizací několika serverů z jednoho centrálního serveru. (Kolektiv autorů, 2012)

Procesy IT jsou souborem činností, úkolů a postupů, které provádějí oddělení IT za účelem správy a podpory technologií používaných v organizaci. Zahrnuje úkoly, jako jsou požadavky na podporu IT, řešení incidentů, prosazování bezpečnosti a další. Mnoho těchto činností je těžko automatizovatelných i přesto však existuje mnoho nástrojů, jak většinu automatizovat. Automatizace IT procesů zahrnuje využití technologií k automatizaci provádění opakujících se a časově náročných úkolů a procesů v oddělení IT. Může jít o úkoly, jako je zajišťování a nasazování nových systémů, monitorování a údržba stávajících systémů, vykazování a analýza dat. (Kissflow, 2023)

3.1.1 Moderní trendy automatizace

Umělá inteligence a strojové učení je jedním z rychle se rozvíjejících oblastí. Adaptivní umělá inteligence v poslední době získává na popularitě díky službě Chat GPT. Cílem tohoto sytému je průběžně přeškolovat modely nebo používat jiné mechanismy k adaptaci a učení v rámci běhového a vývojového prostředí. Její výhoda spočívá ve schopnosti rychle vyvíjet, nasazovat, přizpůsobovat a udržovat umělou inteligenci v různých prostředích podniku. Díky tomu se neustále učí a přizpůsobuje prostředí, ve kterém je využívána. (Şimşek, 2023)

Automatizace bez kódu či s nízkým kódem umožňuje netechnickým uživatelům vytvářet aplikace a automatizované procesy bez rozsáhlých znalostí programování. Platformy pro nízko kódovou automatizaci poskytují vizuální rozhraní a předem připravené komponenty, které uživatelům umožňují jednoduše vytvářet aplikace. K takové automatizaci se také využívá umělá inteligence a strojové učení. Vývoj aplikace, která toto dokáže je však složitý a časově náročný. Vyžaduje zároveň speciální dovednosti a schopnosti. Na druhou stranu výhodou je pak šetření nákladů a času. (Comidor, 2023)

U hyperautomatizace jde o způsob, jak centralizovat lidi, procesy a informace na jednom místě. Hyperautomatizovaný software využívá umělou inteligenci, strojové učení a robotickou automatizaci procesů. Může automatizovat rutinní, opakující se procesy,

poskytovat různým subjektům data prostřednictvím pokročilé analytiky, která byla dříve nedosažitelná, a tím podporovat růst celého podniku od řízení projektů přes marketingové procesy až po podnikovou architekturu. Tento technologický přístup je v souladu s digitální transformací, která probíhá ve všech odvětvích. Cílem je eliminovat potřebu lidí trávit čas drobnými každodenními úkoly, které místo toho může automaticky zvládnout robot. Cílem je, aby organizace co nejvíce ušetřila čas zaměstnanců při řešení drobností a ti se následně mohli soustředit na složitější úkoly. (Şimşek, 2023)

Inteligentní zpracování dokumentů je automatizovaná technologie založená na umělé inteligenci, která zpracovává dokumenty z různých formátů a zdrojů. Využívá kontroly jazyka textu, rozpoznávání obrazu, předpovídání tvorby vět a hluboké učení při organizaci nestrukturovaných dat. Pokročilá technologie systémů umožňuje organizacím snadno a rychle organizovat obrázky, slova, dokumenty, online formuláře a další. Klíčovou schopností je, že dokáže napodobit lidské schopnosti při třídění, roztřiďování a zpracování dokumentů. Díky implementaci strojového učení dokáže zobrazovat kontext jakéhokoliv dokumentu a tím ušetřit lidem hodně času. (Comidor, 2023)

3.2 Metodika ITIL

ITIL neboli IT Infrastructure Library je sada knižních publikací, které popisují způsob řízení ICT služeb a ICT infrastrukturu. Nebo lze konstatovat, že se jedná o ucelenou metodiku tohoto řízení. Základem je, že k řízení služeb informačních technologií využívá procesně orientovaný přístup. Touto definicí je myšleno, že všechny aktivity v daném procesu musí mít přidanou hodnotu pro uživatelské služby. Hlavními ukazateli jsou kvalita, spolehlivost a stabilita IT služeb. Tato metodika vychází z nejlepších zkušeností (best practies). Zaměřuje se komplexně na IT služby a jejich neustálé zlepšování kvality. (Procházka, a další, 2011) Samozřejmě, že kvalita samotné implementace je závislá na míře znalosti prostředí dané organizace. Je důležité si uvědomit, že ITIL neřeší konkrétní podobu organizační struktury ani podobu a obsah pracovních postupů. Tyto body musí být řešeny až při samotné realizaci projektu přímo pro danou organizaci. Z tohoto důvodu neexistují dvě organizace, které by měly procesy řízení ICT služeb dle ITIL naimplementovány naprosto stejně. (Danel, 2011)

Základními pojmy metodiky ITIL jsou:

1. Služba – umožňuje generovat výstupy, které mají hodnotu pro klienta, a pomáhá mu dosáhnout jeho vlastních cílů. Klíčové je, že konkrétní rizika a náklady jsou neseny poskytovatelem služby místo zákazníka.
2. Poskytoval služby (service provider) – Společnost, která poskytuje služby jednomu nebo více externím zákazníkům.
3. Hodnota (value) – Přínosy, užitečnost a význam věcí pro zákazníka.
4. Nabídka služby (Service Offering) - Formální popis služeb, které jsou specificky navrženy tak, aby splňovaly potřeby konkrétní zákaznické skupiny.
5. Servisní vztah (Service relationship) - Jedná se o kooperaci mezi zákazníkem s organizací, která službu zprostředkovává. (Alvao, 2023)

Metodika ITIL vznikla v 80. letech 20. století za účelem zkvalitnění IT služeb ve Velké Británii. Úkolem, bylo vytvořit soubor standardních postupů, které by mohly více sjednotit IT systémy veřejného i soukromého sektoru. Tohoto úkolu se zhostila Centrální počítačová a telekomunikační agentura (CCTA), později přejmenovaná na Úřad pro vládní obchod (OGC). Dala si za cíl vytvořit efektivnější rámec a finančně výhodnější způsob využívání zdrojů IT. (ILX Marketing Team, 2018)

Důležitým rokem pro tuto metodiku byl rok 2000, kdy přijala společnost Microsoft ITIL jako základ pro vývoj svého Microsoft Operations Framework (MOF) a zároveň došlo k první významné změně této metodiky, jejímž výsledkem byl ITIL v2. Nová verze se zaměřila na jeho zpřístupnění pro více lidí a byla uspořádána do třiceti svazkového rámce s devíti souvisejícími kategoriemi. Pak se již za několik následujících let ITIL stal standardem pro osvědčené postupy v oblasti IT a také nejrozšířenějším nástrojem pro řízení služeb IT na světě. (ILX Marketing Team, 2018)

Třetí verze této metodiky byla vydána v roce 2007. Zaměřila se na integraci IT s podnikem. Jejím základem bylo řízení životního cyklu služby.

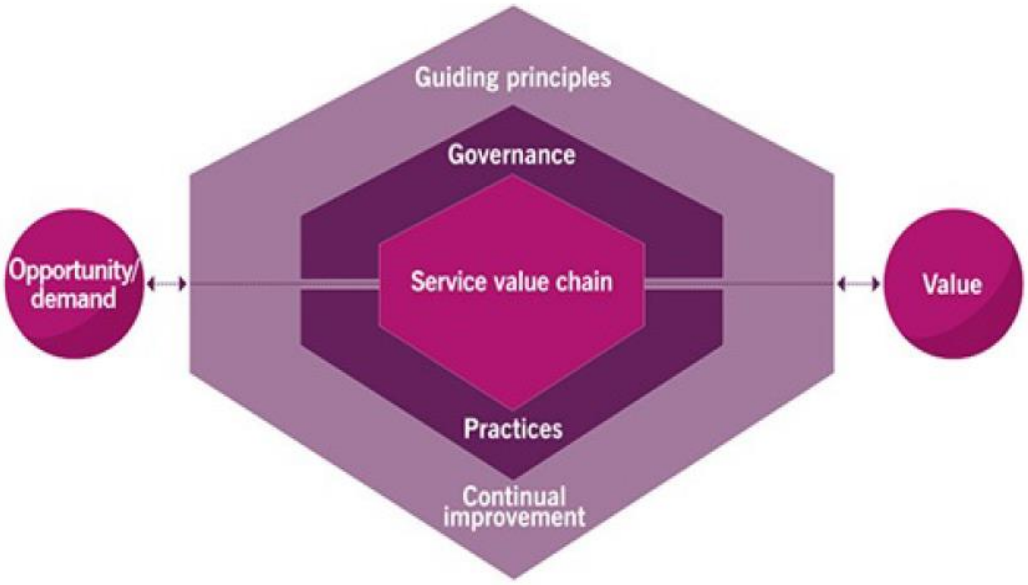
V roce 2011 vydala společnost AXELOS revizi ITIL, která vyřešila chyby a nesrovnalosti z verze tři. V této aktualizované verzi z roku 2011 tvoří katalog služeb 5 svazků: ITIL Service Strategy, ITIL Service Design, ITIL Service Transition, ITIL Service Operation a ITIL Continual Service Improvement. Těchto pět svazků tvoří nyní základ všech osvědčených postupů ITIL na celém světě.

Od roku 2013 je vlastníkem ITIL společnost AXELOS Ltd. - společný podnik Capita Plc a britského vládního úřadu Cabinet Office.

Aktuální verze ITIL byla vydána v roce 2019. Obsahuje více praktických pokynů, jak tuto metodiku používat, zejména v prostředí spolupráce. (ILX Marketing Team, 2018)

3.2.1 ITIL SVS (systém hodnot služeb)

ITIL SVS (systém hodnot služeb) je jádrem celého ITIL verze čtyři. Jedná se o způsob vytváření hodnoty prostřednictvím služeb s využitím IT, jakým různé složky a činnosti organizace spolupracují. Jednotlivé služby můžeme flexibilně kombinovat. To však vyžaduje integraci a koordinaci, tak aby byla organizace konzistentní. ITIL SVS tuto integraci a koordinaci usnadňuje a poskytuje organizaci silný, jednotný směr pro vytvoření hodnoty na základě požadavku.



Obrázek 1: Systém hodnot služeb (ITIL® Foundation, 2019)

Hlavními komponentami jsou:

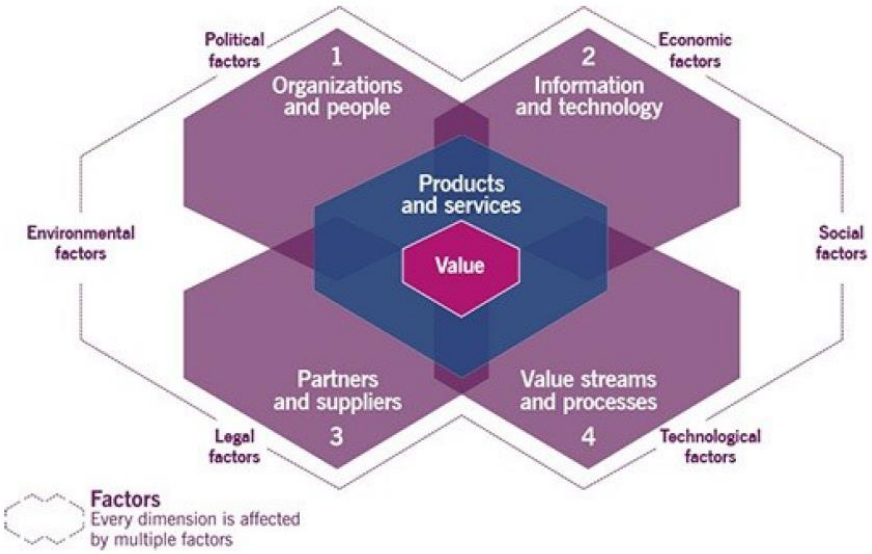
1. Řídící principy (Guilding principles) – Všeobecné doporučení, kterým by se firma měla za všech okolnostní řídit nehledě na typ, strategii a organizační strukturu firmy.
2. Správa (Governance) – Jedná se o prostředky, kterými je firma kontrolována a řízena. Pomáhá držet strategii IT pod kontrolou a v souladu se strategií firmy.
3. Hodnotový řetězec (Service value chain) – Jedná se o propojené činnosti, které firma provádí, tak aby dodala kompletní produkty nebo služby zákazníkům a dokázala tak vytvořit hodnoty.

4. Praktiky (Practices) – Hodnotový řetězec nabízí strukturovaný soubor činností, které jsou navrženy za účelem dosažení konkrétního cíle. Obsahuje sestavy šablon, souborů a postupů, které jsou určené pro návrh, vývoj, implementaci a optimalizaci IT, a také pro plnění stanovených cílů. Díky tomu lze snadno nastavit aktivity obsažené v service value chain podle osvědčených postupů (best practice).
5. Kontinuální zlepšování – Neustále se opakující aktivity napříč organizací, jejichž cílem je neustálé zlepšování výkonu organizace. Je nedílnou součást ITIL frameworku. Zajišťují efektivitu opakujících se činností na všech úrovních organizace a pomáhají naplnit očekávání zainteresovaných stran. (Alvao, 2023)

3.2.2 Čtyři dimenze řízení služeb

Čtyřmi dimenzemi propojených služeb je myšlen soubor aktivit, které přeměňují vstupy na výstupy. Pro zdárné provedení projektu musí být použity a zohledněny všechny čtyři dimenze společně.

Organizace se bohužel často zaměřují pouze na jednu oblast. Například vylepšování procesů může být naplánováno bez ohledu na zapojené uživatele, partnery a technologie nebo mohou být technologická řešení implementována bez péče o procesy nebo uživatele, které mají podporovat. Aby se však dosáhlo, co nejefektivněji všech požadovaných výsledků musí se zvážit všechny aspekty chování. Protože řízení služeb má více stránek a pokud jsou posuzovány odděleně, žádná z nich nestačí k dosažení požadovaných výsledků. (Alvao, 2023)



Obrázek 2: Čtyři dimenze řízení služeb (ITIL® Foundation, 2019)

Proto ITIL definuje čtyři dimenze, které tvoří společně efektivní a účinné zprostředkování služeb a produktů pro zákazníky a další zainteresované strany. Těmito dimenzemi jsou:

1. Organizace a lidé (Organizations and people) – Tato dimenze zahrnuje role, odpovědnosti, formální organizační struktury, kulturu a kompetence, které souvisejí s vytvářením, poskytováním a zlepšováním služeb. Když velikost organizace postupně roste je důležité zajistit, aby byly správně nastaveny styly, kterými je organizace řízená a strukturována. Dále se musí zajistit její role, odpovědnosti, pravomoci a komunikace, aby správně podporovaly její celkovou strategii. Klíčovým prvkem ve vztahu ke službě jsou lidé, kteří jsou v jakémkoli postavení vůči službě. Například zákazníci, zaměstnanci dodavatelů, zaměstnanci poskytovatele služeb. Zaměřit se musí pozornost nejen na dovednosti a kompetence týmů nebo jednotlivých členů, ale také na styly řízení, vedení, schopnosti komunikace a spolupráce. Se stálým vývojem jednotlivých postupů se musí také lidé učit novým dovednostem. Stále více platí, že lidé musí rozumět nejen svému zaměření, ale i přesahu do jiné specializace a role. Toto platí nejen u vlastní organizace, ale i u organizace spolupracující. Tím se zajistí správná úroveň spolupráce a koordinace. Každý člověk v organizaci by měl mít jasnou představu o tom, jak přispívá k vytváření hodnoty pro organizaci. (Knowledgehut, 2023)

2. Informace a technologie (Information and technology) – Jsou kritickými komponentami při aplikaci systému hodnoty služeb. Zahrnují vztahy mezi různými složkami, které mohou být na vstupu i výstupu. Obsahuje technologie, které podporují řízení služeb. Patří k nim mimo jiné systémy pro řízení pracovních postupů, znalostní báze, inventární systémy, komunikační systémy a analytické nástroje atd. Všechny tyto prostředky jsou důležité a klíčové pro rozvoj vlastního potenciálu manažerů projektů. Vývojem technologií se stále více vylepšuje řízení služeb. S tím souvisí i to, že se v poslední době zde začíná také uplatňovat umělá inteligence a strojové učení, které se mohou využívat na všech úrovních projektu. Další oblasti, které se stále více rozšiřují mezi poskytovateli služeb, jsou mobilní platformy, cloudová řešení, nástroje pro vzdálenou spolupráci a automatizovaná testování. Samozřejmě konkrétní informace a technologie závisí na povaze poskytovaných služeb a obvykle zahrnují všechny úrovně architektury IT. Mezi které patří i aplikace, databáze, komunikační systémy a jejich integrace. Při využívání nejnovějších technologických prvků v IT službách, je těmito službami poskytována vyšší šance na efektivnější využití a současně získávají konkurenční výhodu. Tato výhoda se nejvíce projeví v odvětvích s vysokou konkurencí.

   Při správě této dimenze metodika ITIL říká, že by každá organizace měla mít vždy odpověď na tři související otázky, kterými jsou:

   - Jaké informace služby spravují?

   - Jaké podpůrné informace a znalosti jsou potřeba k poskytování a řízení služeb?

   - Jak budou informace a znalosti chráněny, spravovány, archivovány a likvidovány?

   Správa informací je pro mnoho služeb hlavním prostředkem, který umožňuje vytvářet hodnotu pro zákazníka. Jako příklad můžeme uvést službu správy sítě, která vytváří hodnotu pro své uživatele tím, že udržuje a poskytuje přesné informace o aktivních síťových připojeních organizace, což jí umožňuje přizpůsobit kapacitu šířky pásma sítě. Informace jsou obecně klíčovým výstupem většiny IT služeb, které jsou organizacemi využívány.

   Dalším důležitým aspektem je metoda výměny informací mezi různými službami. Musí být zabezpečena vhodná propojenost a optimalizace

informační architektury různých služeb. Zde jsou hlavními kritérii dostupnost, spolehlivost, přístupnost, přesnost a relevance.

Služby jsou v současné době většinou založeny na informačních technologiích a jsou na nich silně závislé. Principy fungování organizace mívají velký vliv na to, jaké technologie budou používány. Samozřejmě jsou některé organizace, které mohou mít větší zájem využívat moderní technologie. A na druhé straně existují stále organizace, kde tyto moderní technologie nevyužívají. Jedna společnost může mít zájem využívat umělou inteligenci, zatímco jiná nemusí chtít využívat ani základní výpočetní techniku. Dalším faktorem, který ovlivňuje to, jaká technologie se využívá, je samotná podstata organizace. Například organizace, která spravuje systémy kritické infrastruktury, může mít z bezpečnostních důvodů omezení pro používání některých technologií. Omezení se samozřejmě týkají i některých dalších průmyslových odvětví. (ITIL® Foundation, 2019) Musí se však zvážit správné technologické komponenty, které jsou kompatibilní se současným technologickým prostředím, vyhovují potřebám s předpisy, stanovují potřeby informační bezpečnosti, škálovatelnost, automatizaci, umožňují komunikaci a spolupráci. (Knowledgehut, 2023)

3. Partneři a dodavatelé (Partners and suppliers) – Tato dimenze se zaměřuje na vztahy organizace s dalšími subjekty, které se podílejí na návrhu, vývoji, zavádění, poskytování, podpoře a neustálém zlepšování služeb. Řeší také smluvní vztahy mezi organizací a jejími partnery nebo dodavateli. Tyto vztahy mohou zahrnovat různé úrovně integrace a formálnosti. Obsahují na jedné straně oblast formálních smluv s jasným rozdělením odpovědností a na druhé straně i flexibilní partnerství, kde strany sdílejí společné cíle, rizika a spolupracují na dosažení požadovaných výsledků. Strategie a cíle organizace, která vystupuje v pozici poskytovatele služeb, se budou lišit ve vztazích se zákazníky od organizace, která vystupuje jako spotřebitel služeb. U využívání partnerů a dodavatelů by měla strategie organizace vycházet z jejich cílů, kultury a podnikatelského prostředí. Samozřejmě existují i organizace, kde je vliv externího subjektu minimální. V těchto případech jsou většinou dodavatelé využíváni pouze pro přesně danou činnost

a organizace má zaměření hlavně na své vlastní zdroje. Naopak jiné organizace se mohou spoléhat primárně na své dodavatele. (ITIL® Foundation, 2019)

4. Hodnotové toky a procesy (Value Streams and processes) – Dimenze hodnotových toků a procesů se zabývá tím, jak různé části organizace pracují, aby umožnily vytváření hodnoty prostřednictvím produktů a služeb. Dimenze se zaměřuje na to, jaké činnosti organizace vykonává a jak jsou organizovány. Dále se zabývá také tím, že zajišťuje v organizaci to, aby umožňovala účinně a efektivně vytvářet hodnoty pro všechny zúčastněné strany. Hodnotovým tokem rozumíme řadu kroků, které jsou v rámci jedné organizace použity k vytváření a poskytování produktů a služeb jinému subjektu. Hodnotový tok je kombinací jednotlivých činností hodnotového řetězce organizace. Samotný tok hodnot je řada kroků, které organizace dělá, aby vytvořila, dodala výrobky a služby dalšímu subjektu. Identifikace a pochopení různých hodnotových toků organizace je zásadní pro zlepšení její celkové výkonnosti. Po uspořádávání činností organizace do podoby hodnotových toků má organizace jasnou představu o tom, co a jak poskytuje, a zároveň dokáže neustále vylepšovat svoje služby a produkty. (ITIL® Foundation, 2019)

   Procesem je v této souvislosti myšlen soubor vzájemně propojených nebo vzájemně působících činností, které přeměňují vstupy na výstupy. Procesy definují posloupnost činností a jejich závislosti. (Alvao, 2023)

Tyto čtyři dimenze představují základní hlediska, která jsou významná pro celý proces poskytování služeb. Tato hlediska bývají často omezena nebo ovlivněna různými vnějšími faktory, které jsou často mimo kontrolu organizace.

Nebudou-li všechny čtyři rozměry správně řešeny, může to vést k tomu, že služby nebudou splňovat očekávání v oblasti kvality či efektivity. Například nezohlednění dimenze hodnotových toků a procesů může vést k plýtvání nebo ještě hůře k práci, která je v rozporu s jinými činnostmi organizace. Stejně tak ignorování dimenze partnerů a dodavatelů může znamenat, že služby prováděné externím subjektem nejsou v souladu s potřebami organizace. Jednotlivé dimenze nemají ostré hranice, proto se mohou vzájemně překrývat. Někdy se mohou ovlivňovat nepředvídatelným způsobem v závislosti na úrovni složitosti a podmínek ve kterých se organizace pohybuje.

Čtyři dimenze řízení služeb jsou velmi důležité pro všechny řízené služby. Je proto nezbytné mít stále na paměti, aby všechny čtyři dimenze byly využity u všech služeb a zároveň na všech úrovních řízení služeb. (Alvao, 2023)

3.2.3 Řízení změn

Účelem řízení změn je maximalizovat počet úspěšných změn služeb a produktů tím, že se zajistí řádné posouzení rizik, schválení a řízení harmonogramu. Samotná změna je úprava čehokoli, co by mohlo mít přímý nebo nepřímý vliv na služby.

Každá organizace řadí do této oblasti jiné úpravy. Obvykle však zahrnuje veškerou IT infrastrukturu, aplikace, dokumentaci, procesy, vztahy s dodavateli a vše ostatní, co by mohlo přímo či nepřímo ovlivnit produkt nebo službu. Řízení změn se obvykle soustřeďuje na změny produktů a služeb. Změny by měly být prováděny tak, aby při nich docházelo k pozitivnímu vlivu na organizaci. Vždy by měla být určena osoba, která dokáže určit, zdali daná změna byla či nebyla prospěšná. Existují tři základní typy změn. Každý z těchto typů má jiné řízení.

1. Standartní změny – Jde většinou o změny s nízkým rizikem. Jsou delší dobu připravovány a díky tomu jsou tyto změny připraveny dobře. Často vznikají na základě servisního požadavku uživatelů. Pokud dochází k úpravě způsobů, jakým je změna prováděna, je vhodné, aby byla přehodnocena všechna rizika, které s danou změnou souvisí.
2. Běžné změny – Jde o změny, které je potřeba naplánovat, posoudit a schválit. Tyto změny mají obvykle též nižší míru rizika. Jsou však pro organizaci mnohem více důležité.
3. Nouzové změny – Jde o změny, které jsou nutné udělat co nejdříve. Například při řešení bezpečnostních incidentů je nutné situaci okamžitě řešit. Obvykle nejsou tyto změny plánovány a jejich schvalování je tudíž ve výrazně rychlejším režimu. Pro tyto mimořádné změny je vhodné mít v organizaci směrnice, které udávají, jakým způsobem zaměstnanci mají v případě přijímání nouzových změn postupovat.

V ideálním případě by změny měly být plánovány, a tudíž i dobře připraveny. (ITIL® Foundation, 2019)

3.2.4 Monitorování a správa událostí

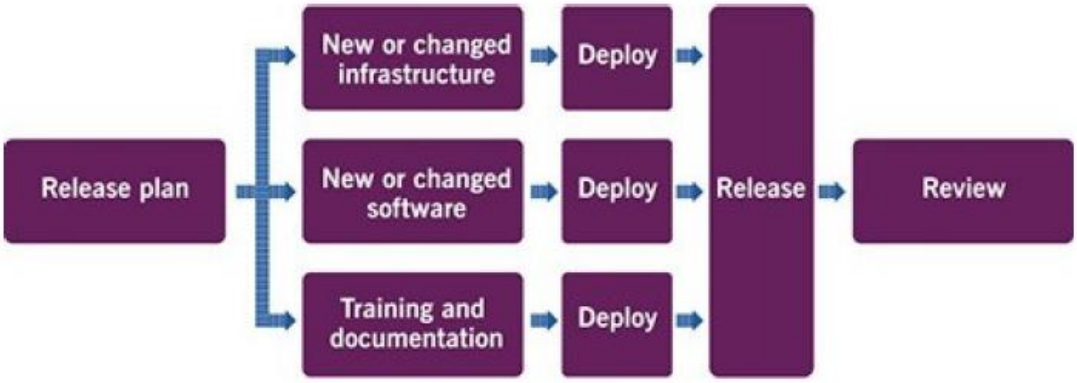
Hlavním cílem monitorování a řízení událostí je sledovat služby, komponenty služeb, zaznamenávat a hlásit vybrané změny stavu, které jsou označovány jako události. Toto slouží k určení vhodných reakcí na události včetně reakce na stavy, které by mohly vést k vážným problémům. Událost je jakákoliv změna stavu, která má význam pro správu služeb nebo produktů. Jsou obecně zjišťovány pomocí oznámení, která jsou vytvořena IT službou nebo monitorovacím nástrojem. Monitorování a správa událostí zpracovává události v průběhu jejich životního cyklu s cílem zabránit, minimalizovat nebo eliminovat jejich negativní dopad na podnikání.

Monitorovací část se zaměřuje na sledování služeb a produktů. Hlavním cílem je odhalení případných problémů. Monitorování by mělo být v ideálním případě zcela automatizované. Zároveň souvisí s řízením událostí, které vznikají při odhalení abnormálního chování. Ne všechny události mají pro organizaci stejný význam. Zároveň nevyžadují stejnou reakci. Některé události mohou být čistě informativní a mohou se řešit až po nějaké době. Jiné mohou být pro organizaci zcela kritické, jako například náhlá nefunkčnost. U takových problémů je potřeba naopak situaci okamžitě řešit. (Servicenow, 2023)

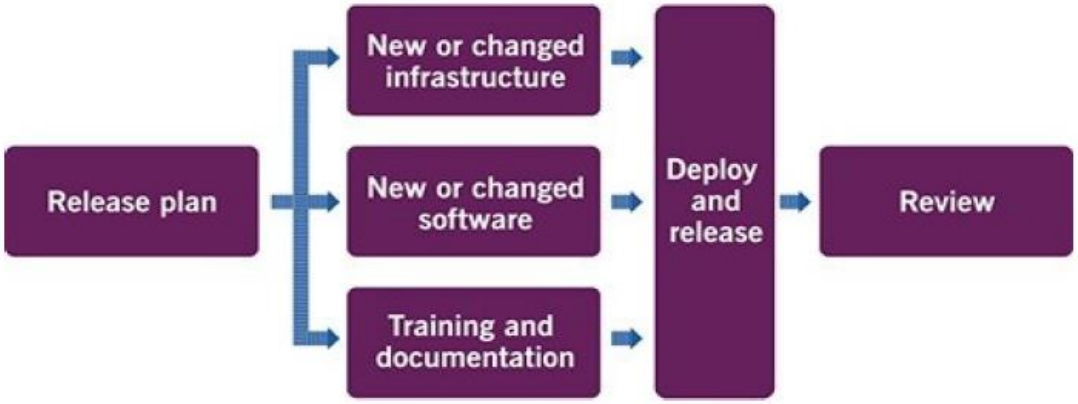
I když je v dané oblasti použita vysoká míra automatizace, bez lidského faktoru se neobejde. Je tomu tak nejen z důvodu správného nastavení monitorovacích nástrojů, jako například kritéria pro hodnocení anomálií nebo strategie monitorování, ale i při vzniku události musí většinou problém řešit některý z pracovníků organizace. (ITIL® Foundation, 2019)

3.2.5 Řízení verzí

Cílem správy verzí je zpřístupnit nové a změněné služby a funkce. Release je verze služby, konfigurace nebo soubor konfigurací, které jsou připravené pro použití. Verze mohou obsahovat různé aplikační a infrastrukturní komponenty, které poskytují nové nebo změněné funkce. Velikost verzí může být různá, od velmi malých, které obsahují pouze jednu drobnou změnu, až po velmi rozsáhlé, které zahrnují mnoho součástí poskytujících zcela novou službu. V obou případech jsou tvořeny plány aktualizací, ve kterých je uvedeno, které komponenty a kdy budou instalovány a nasazovány. (Manageengine, 2023)



Obrázek 3:Tradiční způsob nasazení (ITIL® Foundation, 2019)



Obrázek 4: Agilní způsob nasazení (ITIL® Foundation, 2019)

Existují dva základní přístupy k tvoření nových verzí. První, tradiční je založen na způsobu, kdy se různé komponenty nasazují v jednu chvíli. U druhého způsobu se jednotlivé komponenty nasazují nezávisle na sobě. Z těchto komponent se pak vytvoří jedna verze, která se následně vydá. Verze se nejdříve nasazují na testovací prostředí, pokud v organizaci je. V případě, že se během testování neprojeví žádné chyby je možné verze nainstalovat i na produkční prostředí. (ITIL® Foundation, 2019)

3.2.6 Správa katalogu služeb

Účelem zřízení katalogu služeb je poskytnout jediný zdroj konzistentních informací o všech poskytovaných službách a nabídkách služeb. Zároveň je potřeba zajistit, aby byly

k dispozici dané skupině osob. Seznam katalogu služeb představuje služby, které jsou aktuálně k dispozici. Služeb tedy může organizace nabízet více. Ty, které však nejsou aktuálně k dispozici, nepatří do katalogu služeb. Správa katalogu služeb zajišťuje, že popisy služeb a produktů jsou jasně vyjádřeny pro danou skupinu zákazníků, tak aby katalog podpořil zapojení všech stran. Katalog služeb může mít mnoho podob. Nejčastěji je však ve formě dokumentu, a to jak v tištěné, tak elektronické podobě. Poskytuje přehled o tom, jaké služby jsou k dispozici a za jakých podmínek. Katalog obsahuje role, jako je vlastník služby. Také určuje odpovědné osoby za správu, editaci a aktualizaci seznamu dostupných služeb při jejich zavádění, změnách nebo vyřazování. (ITIL® Foundation, 2019)

3.2.7 Service Desk

Účelem service desk je zápis řešení incidentů a požadavků na služby. Měl by být také vstupním bodem a jediným kontaktním místem poskytovatele služeb se všemi jeho uživateli. Poskytuje uživatelům způsob, jak nahlásit problémy, dotazy a požadavky. Zároveň jsou zde zaznamenávány všechny změny požadavků. Způsob a forma může být různá, a to podle místa, kde je využíván. Funkce však zůstává vždy stejná. Pracovník servisního oddělení nemusí být vysoce technicky zdatný. I když je práce na servisním oddělení poměrně jednoduchá, hraje stále důležitou roli při poskytování služeb a musí být aktivně podporována. Má zásadní vliv na uživatelskou zkušenost a na to, jak je poskytovatel služeb vnímán uživateli. S vývojem moderních technologií se servisní oddělení mohou přesouvat k poskytování více samoobslužné formy service desku za využití online portálů, vědomostních databází a mobilních aplikací. I přesto se v mnoha organizacích stále využívá spíše osobní forma, kdy se o všechny příchozí požadavky starají pracovníci service desku. Proto je mnohdy pracovní doba tohoto oddělení omezená. (ITIL® Foundation, 2019)

3.2.8 Ověření a testování služeb

Účelem ověřování a testování služeb je zajistit, aby nové nebo změněné produkty a služby splňovaly definované požadavky. Hodnota služby vychází ze vstupních informací od zákazníků, obchodních cílů a regulačních požadavků a je dokumentována jako součást analýzy, návrhu řešení a jeho realizace. Tyto požadavky se používají ke stanovení kvality a ukazatele výkonnosti, které podporují správné určení kritérií pro schválení a požadavků na testování.

Schválení služby se zaměřuje na stanovení kritérií pro přijatelnost nasazení produktu do produkce. Celý proces se pak ověřuje testováním. Akceptační kritéria mohou být zaměřena buď na hodnotu, spolehlivost, kvalitu nebo jejich kombinaci stanovenou na začátku projektu. Akceptační kritéria jsou vždy tvořena v součinnosti se všemi zúčastněnými subjekty. Schválení těchto kritérií funguje, jako základ pro testování.

Strategie testování stanovuje celkový přístup k testování. Může se vztahovat na prostředí, platformu, sadu služeb nebo jednotlivé služby. Testování by mělo být prováděno stejně důkladně, jak u systémů vyvinutých vlastními silami, tak u externě vyvinutých řešení. Strategie testování vychází z akceptačních kritérií. (ITIL® Foundation, 2019).

3.2.9 Správa nasazení

Tímto pojmem označujeme koordinaci přesunu veškerého hardware, software, dokumentace, či jakékoliv části služby do produkce. Můžeme ji použít i pro přesun pouze některých komponent do jiného prostředí. A úzce souvisí s řízením vydávání a změn. Pro nasazování změn a verzí můžeme použít několik různých strategií.

1. Postupné nasazování – To znamená, že nové komponenty jsou do produkčního prostředí nasazovány postupně. Po ověření, že nasazovaná část funguje bez problémů, se pokračuje na dalších části. Tento postup se opakuje, dokud není nová komponenta nasazena všude. Samozřejmě to platí pouze pokud se nejedná o hardwarovou úpravu.
2. Kontinuální nasazovaní – Jedná se o strategii, kdy jsou nové komponenty vydávány ve chvíli kdy je potřeba.
3. Celoorganizační nasazení – Jedná se o největší zásah ze všech strategií. Jde o naimplementování nové komponenty na všechna zařízení, pro která je daná komponenta určená, najednou. Tento postup není úplně doporučovaný. Využívá se většinou v případě, kdy by vynechání některého ze zařízení znamenalo nefunkčnost některé ze služeb.

Komponenty, které jsou k dispozici pro nasazení, by měly být uchovávány na jednom nebo více zabezpečených místech, aby se zajistilo, že nebudou před nasazením upraveny. Pro tuto správu existuje mnoho nástrojů, které pomáhají udržet vše na jednom místě. Tyto nástroje často sdružují několik bodů ITIL najednou. (ITIL® Foundation, 2019)

3.3 Metodika ISO 27001

Certifikace ISO 27001 dokládá, že organizace úspěšně implementovala systém řízení bezpečnosti informací a tím pádem splňuje požadavky této normy. Základním pojmem normy jsou rizika. Tato metodika se zabývá primárně jejich identifikací, vyhodnocením a zároveň prevencí proti nim. Jedná se o rizika celé organizace a oblasti ochrany dat. Na základě veškerých analýz a hodnocení, jsou prováděny kroky, které by měly vést ke sledování a zlepšování v oblasti bezpečnosti informací. Hlavním cílem je udělit doporučení, jak zavádět, implementovat, pozorovat, monitorovat, přezkoumávat, udržovat a zlepšovat systém řízení bezpečnosti informací. (Svobodová, 2023)

Dalšími důležitými pojmy jsou důvěrnost, integrita a dostupnost. Důvěrnost představuje informace, ke kterým mají přístup pouze určené osoby. Integrita představuje určené osoby, které mohou provádět změny informací. Dostupností se rozumí nepřetržitý a včasný přístup určených osob k informacím. Zavádění této normy obnáší čtyři hlavní body: plánování, realizaci, ověření a zlepšování řízení informační bezpečnosti v organizaci. (RVX, 2023)

Na tuto metodiku lze nahlížet, jako na soubor ověřených postupů v oblasti zabezpečení dat. Nejedná se o návod, jak dosáhnout dokonalého zabezpečení. Jde spíše o rady, na základě, kterých by organizace měla dokázat zkvalitnit zabezpečení dat a zároveň tím zmenšit rizika spojená se ztrátou dat a informací. Všechny změny a zvolené postupy vychází z vlastní analýzy rizik. Analýza rizik obsahuje seznam všech aktivit, kterými se společnost zabývá. Každé riziko má určenou osobu, která je za dané riziko odpovědná. Po analýze aktivit je nutné identifikovat veškerá rizika, která mohou nastat. Dále se u těchto rizik určuje vážnost vlivu na organizaci v případě, že by riziko nastalo. Vyberou se ty nejpravděpodobnější a zároveň ty nejkritičtější. Na základě tohoto výběru se vyberou postupy, které vedou k lepší ochraně dat a informací spojené s danými riziky. V případě, že jsou připravena preventivní opatření, je na závěr potřeba určit kontrolní mechanismus, který bude dané opatření kontrolovat. Rizika je nutné pravidelně aktualizovat a kontrolovat. (Svobodová, 2023)

3.4 Správa konfigurací

Správa konfigurací umožňuje sledovat jednotlivé komponenty v infrastruktuře. Udržovat přehled o tom, co dělají, jak mají vypadat a co jsou potřebné kroky pro jejich

nasazení od prvotního vytvoření až po jejich kompletní implementaci. Typy informací, které by měly být zahrnuty, jsou spravované služby, verze aplikací a systémů. Zároveň by měly být dokumentovány způsoby jejich konfigurace a umístění v síti.

Efektivní řešení správy konfigurací může být v mnoha odhledech velmi užitečné.

1. Úspora času: Pomáhá zkrátit čas potřebný ke správě aktualizací, změn a opakujících se úloh v různých prostředích a sítích. Servery mohou být centralizovány a následně po otestování aktualizovány všechny najednou. Díky dobrému systému správy konfigurací je možno automatizovat běžné procesy a tím ušetřit mnoho času s implementacemi na všechny zařízení.
2. Zlepšení dostupnosti – Správa konfigurací by měla také sloužit pro rychlejší řešení problémů s dostupností. Toho lze docílit díky tomu, že je jasně uvedeno, kdy, co a na jakých zařízení bude implementováno. Díky jasnému plánování vzniká lepší povědomí o plánovaných a provedených změnách napříč systémy.
3. Snížení rizika – Na základě podrobné evidence změn snižuje pravděpodobnost chyby na produkčním prostředí. Díky tomu, když se na testovacím prostředí objeví chyba, tak na to lze reagovat. Následně se tato chyba odstraní, a tudíž se na produkční prostředí naimplementuje pouze funkční řešení.
4. Zlepšení kontroly – Díky správě konfigurace je jasně daný postup, jakým se v organizaci implementují změny. Proto je zajištěna větší konzistentnost a tím i větší kontrola nad konfiguracemi jednotlivých zařízení.

Je dobré si uvědomit, že celková správa konfigurací je jedním z důležitých kroků pro celkovou automatizaci. (Sesto, 2021)

3.5 Nástroj Ansible

Ansible je jednoduchý, flexibilní a velmi výkonný nástroj, který umožňuje automatizovat běžné úlohy infrastruktury a spouštět příkazy na více zařízeních najednou. Přestože pomocí Ansible lze spouštět příkazy na několika hostitelích souběžně, největší výhoda spočívá v jejich správě pomocí playbook a rolí. (Shah, 2015)

Ansible původně vyvinul Michael DeHaan, který byl rovněž autorem softwaru Cobbler, jenž byl vyvinut v době, kdy pracoval pro společnost Red Hat. Dne 23.února 2012 vydal první veřejnou verzi nástroje Ansible. V roce 2013 se projekt rozšířil a byla založena

společnost Ansible, Inc., která již nabízela podporu svým uživatelům, kteří nástroj využívali pro správu virtuálních, fyzických nebo hostovaných cloudových serverů. Společnost následně získala 6 milionů dolarů v rámci financování série A. Díky těmto penězům vznikl komerční Ansible Tower. Ten slouží, jako webová aplikace, ke které mohou uživatelé připojit své servery a následně na nich využívat hromadnou správu pomocí Ansible, primárně na základě rolí. V roce 2015 byla firma odkoupena společností Red Hat. V současnosti je Ansible jedním z nejpoužívanějších automatizačních nástrojů pro správu vzdálených zařízení. (Heap, 2016)

Ansible má několik základních principů, které jsou od vzniku zachovány. Veškeré využívání nástroje by mělo být bez nutnosti instalace speciálního agenta, ani žádného speciálního software. Vše by mělo být spravováno pomocí démona SSH nebo WinRM v případě Windows systémů. Celý proces zprovozňování by měl být, co možná nejjednodušší. (McKendrick, 2018)

3.5.1 Yaml

Jedná se o jazyk, který se využívá pro psaní Ansible playbook a rolí. Lze ho však využívat v jiných aplikacích, jako například Kubernetes, Docker a mnoho dalších. Výhodou je jeho jednoduchost a dobrá čitelnost. Je tedy relativně snadný na naučení. To je jeden z dalších důvodů, proč je Ansible tak oblíbený. Uživatelé se nemusejí učit žádný složitý programovací jazyk. YAML má pár základních bodů, které je nutné dodržovat ke správnému fungování.

1. První řádek playbooku by měl začínat znakem "--- " (tři pomlčky), který označuje začátek dokumentu YAML. Každý příkaz musí mít na začátku řádku "- " za kterou je mezera.
2. Jeden playbook může mít několik za sebou jdoucích příkazů. Důležité jsou mezery v případě vynechání Ansible zahlásí chybu.
3. Všechny příkazy musí udržovat stejnou úroveň odsazení.
4. Každý playbook také obsahuje dvojici klíč-hodnota. Vždy jsou oddělené znakem ":". V tomto případě se jedná o určení proměnných, vzdálených strojů, rolí, úloh.

YAML je jednoduchý jazyk je však potřeba si dát pozor na syntaxi. Jakákoliv chybějící mezera nebo naopak mezera navíc může způsobit nefunkčnost celého playbooku či role.

3.5.2 Playbook

Obvykle se v IT pod pojmem playbook rozumí soubor pokynů, které někdo spustí. Jedná o různé postupy, od konfigurací nových serverů až po způsob nasazování nových aktualizací kódu a řešení problémů, které mohou nastat. V tradičním slova smyslu je playbook obvykle souborem skriptů nebo pokynů, které má uživatel dodržovat, a přestože mají být pro všechny systémy stejné, tak se to většinou nepodaří. (Heap, 2016)

Spouštění jednotlivých příkazů v příkazovém řádku však není nejefektivnějším způsobem, jak tyto změny nasadit do firemního prostředí. Proto je velmi dobré využít nástroj Ansible. Pomocí playbooku v podstatě lze konstatovat, že tyto změny a příkazy budou použity na hostitele, místo toho, aby se některý z pracovníků musel přihlašovat na každý stroj zvlášť a vše tam spouštěl ručně.

Pro správné fungování musí být nejdříve vytvořen inventář ve kterém jsou zaznamenány všechny stanice, které daný playbook má ovládat. V inventáři je možné dělat skupiny, které mají názvy na prvním řádku a jsou ohraničené [ ]. Playbook lze spouštět na všechny hostitele nebo lze vybrat pouze danou skupinu, či jeden samostatný server. Jeden hostitel může být obsažen ve více skupinách. V takovém případě se proměnné slučují a spouštějí se na základě priorit. Při dokončení inventáře je vhodné ověřit funkčnost připojení pomocí příkazu Ansible -m ping all. (Shah, 2015)

Vzdálený uživatel (remote_user) je uživatel, který provádí úkoly. Uživatel může být kdokoliv kdo má přístup na spravovaných serverech. Vzdálený uživatel může být definován v každé úloze, ale i na začátku playbook. Může nastat situace, kdy bude potřeba použít jiného uživatele pro přístup k vzdálenému systému, a zároveň ho bude potřeba změnit pro spouštění konkrétní úlohy (become_user). Úkoly (Tasks) definují operace, které playbook na vzdáleném zařízení provádí. Vždy musí mít na prvním řádku jméno (- name), kde by měl být vždy popis toho, co následující příkaz udělá. (Sesto, 2021)

3.5.3 Role

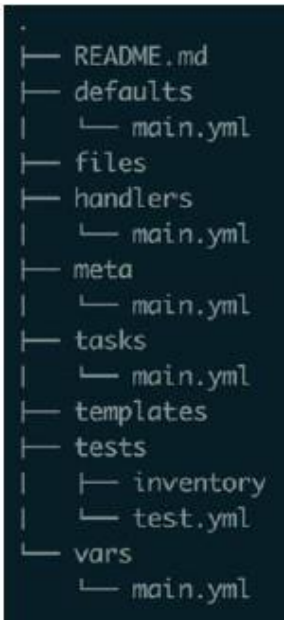
Playbook a role jsou si velmi podobné, ale zároveň velmi odlišné. Playbook je samostatný soubor, který může administrátor spustit. Zároveň obsahuje veškeré informace potřebné k nastavení stavu zařízení. Role se dají představit jako skript, který je rozdělen do několika různých souborů. Místo jednoho souboru, který by obsahoval vše, co je potřeba, existuje jeden soubor pro proměnné, jeden pro úlohy a další pro další části. Role však nelze

spustit samostatně. Musí k ní existovat playbook, který obsahuje informace o tom, na kterých hostitelích má být spuštěn, případně s jakými parametry. Role je mechanismus, který se využívá k vytváření balíčků úloh, obslužných rutin a všech dalších potřebných komponent, které se následně spojí do jednoho balíčku tím, že jsou zahrnuty do jednoho playbooku.

Role jsou obecným konceptem systému Ansible. Ve skutečnosti plní zásadní funkci. Z toho důvodu vzniklo uložiště a dodatečný nástroj, který se nazývá Ansible Galaxy. Jedná se o webovou stránku, na kterou mohou lidé nahrávat role, které vytvořili. Stahovat a nahrávat na tento portál může kdokoliv. Lze zde najít i některé playbook, ale primárně je tento portál určen pro sdílení rolí. Roli před využitím je nutné vždy zkontrolovat, zdali se v ní neskrývá něco nežádoucího. Tím, že se jedná o otevřené uložiště, tak i zde existují dobré, ale i špatné role, které mohou dělat úplně něco jiného, než je původně zamýšleno.

Při instalaci role je možné ji buď nainstalovat přímo do zařízení nebo do vybraného projektu. Stahování rolí se dělá pomocí příkazu ansible-galaxy a cestu k dané roli -p, aby se role nainstalovala do složky s názvem roles. Příkaz ansible-galaxy by se měl spouštět ve stejném adresáři, kde se nachází soubor playbooku, který danou roli následně bude spuštět. (Heap, 2016)

Většina vytvořených složek rolí je volitelná, na obrázku níže je vidět struktura prázdné role. Každá složka a obsah v ni má svou specifickou funkci.



Obrázek 5: Prázdná role (Heap, 2016)

Každá role by měla začínat souborem README. V něm by měl být popsán účel role, proč byla vytvořena a všechny proměnné, které bude možné v roli upravovat.

1. Defaults/main.yml představuje konfigurační soubor, který lze používat k definování výchozích hodnot proměnných využívaných v roli.
2. V souboru vars/main.yml lze také definovat proměnné, které přepíšou cokoli, co je definováno v souboru defaults/main.yml, je tomu tak z důvodu vyšší priority. Například proměnné umístěné v souboru vars/main.yml budou mít přednost před proměnnými definovanými při prvotním definování role, ale proměnné umístěné v souboru defaults/main.yml nikoli.
3. Files je místo, kam jsou umisťovány soubory potřebné pro využívání role. Může se jednat o jakékoliv typy souborů. Tyto soubory však nelze upravovat. Jedinou možností práce s nimi je kopírování.
4. Handlers/main.yml je místo, kde jsou vloženy pomocné aktivity, jako je například restart určité vybrané služby. Shromáždění všech dostupných pomocných aktivit na jednom místě usnadňuje používání celé role. Zde je jednoduše vidět, co vše lze v roli spustit. Pomocné aktivity lze spouštět ve stejné roli, z jiných rolí a také z playbooku.
5. Meta/main.yml je soubor s metadaty role. Díky tomuto souboru jsou definována metadata, která Ansible Galaxy využívá, pokud se majitel role rozhodne zveřejňovat. Je zde také možnost definování všech závislostí, které daná role má. Například minimální verze či podporované platformy.
6. Tasks/main.yml - Jedná se o úkoly, které by byly normálně psány v playbooku. Veškeré akce, které jsou definovány v tomto souboru bude Ansible provádět při spuštění role.
7. Templates (šablony) obsahují všechny soubory, které zpracovává šablonovací jazyk jinja2, tak aby bylo možné všechny šablony správně použít.
8. Tests je adresář, ve kterém je dobré vytvářet playbook, který testuje danou roli. Většinou se používá, pokud se využívá automatické testování role. (Heap, 2016)

4 Vlastní práce

4.1 Úvodní pojednání

V rámci praktické části diplomové práce jsou aplikovány poznatky a teoretická východiska z předchozí části práce. Všechny organizační postupy jsou systematicky řízeny podle metodiky ITIL, přičemž celá organizace dosáhla certifikace dle této metodiky. Vzhledem k charakteru organizace, která spravuje systémy kritické infrastruktury státu, bylo nezbytné zachovat diskrétnost ohledně některých specifických údajů týkajících se implementace.

Celý projekt vznikl z důvodu rostoucího objemu pracovního zatížení při zachování stále stejného počtu pracovníků. Prvním krokem byla analýza existujících procesů a dostupných zdrojů dat. Následovala analýza možností automatizace vybraných činností pomocí dostupných nástrojů. Tento proces vedl k rozhodnutí o implementaci vybraného řešení.

Pro dosažení automatizace byl využíván systém Foreman, který byl pečlivě vybrán jako nejvhodnější na základě provedené analýzy. Paralelně s tím je využíván nástroj Ansible, jehož pomocí probíhá automatizace konkrétních procesů. Autor práce se věnuje i dokumentaci chyb a různých druhů postupů, které se objevily během implementace.

Tímto přístupem je zajištěno efektivnější a optimalizovanější řízení vybraných IT procesů organizace, což vede ke zvýšení produktivity a snížení pracovní zátěže pracovníků v těchto oblastech. Implementace nástroje pro automatizaci se ukázala jako vhodný krok vzhledem k modernizaci, optimalizaci a efektivnějším práci v rámci IT.

Sám autor v rámci organizace celý projekt vede. A zároveň dělal veškeré činnosti a analýzy v této části práce. Ve vlastní části práce jsou z pozdější analýzy vybrány tři hlavní oblasti automatizace, kterým se autor věnoval. Správa repozitářů, správa serverů a správa administrátorských uživatelů operačního systému Linux.

4.2 Dostupné nástroje

Microsoft System Center je platforma pro správu podnikového IT, která pomáhá spravovat infrastrukturu včetně serverů, stolních počítačů, aplikací a sítí. Poskytuje řadu nástrojů a funkcí pro automatizaci úloh, sledování výkonu a řešení problémů. Jedná se o výkonný nástroj, který pomáhá ušetřit čas, peníze, zvýšit efektivitu a spolehlivost IT

prostředí. Je doporučován primárně při velkém zastoupení strojů s operačním systémem Windows.

Ansible Automation Platform je platforma, která umožňuje automatizovat úlohy a procesy IT. Používá deklarativní přístup, což znamená, že se definuje požadovaný stav IT prostředí a za pomoci Ansible se pak využívá znalosti o infrastruktuře k provedení potřebných automatizačních kroků. Ansible je oblíbenou volbou pro automatizaci úloh, jako je poskytování nových serverů, nasazování aplikací a konfigurace síťových zařízení. U tohoto nástroje poptána cenová nabídka. Ta však byla vyšší, než by organizace byla ochotna akceptovat.

Oracle Automation je sada nástrojů, které lze použít k automatizaci různých úloh IT, včetně spuštění, správy konfigurace a nasazení. Je součástí Oracle Cloud Infrastructure a lze ji použít k automatizaci úloh v lokálních i cloudových prostředích. Oracle Automation je výkonný nástroj, který pomáhá zjednodušit provoz IT a zvýšit efektivitu. Tento nástroj je však též drahý a nenabízí o moc více funkcí než konečné řešení.

Foreman je open-source nástroj, který pomáhá správcům systému spravovat servery po celou dobu jejich životním cyklu, od spuštění, konfigurace až po monitorování a údržbu. Podpora spouštění umožňuje snadnou kontrolu nad nastavením nových serverů a pomocí správy konfigurace lze snadno automatizovat opakující se úlohy. S nástrojem Foreman lze rychle nasazovat aplikace a proaktivně spravovat změny, a to jak v lokálním prostředí s virtuálními servery, fyzickými servery, tak v cloudu. Foreman lze dobře škálovat na více lokalit (kanceláře, datová centra atd.) a více organizací, což umožňuje růst, aniž by byla ztracena jakékoliv data. Tento nástroj byl nakonec vybrán, jako nejvhodnější pro dané prostředí organizace. Umožňuje většinu funkcí, jako placené nástroje výše, s výhodou nulových pořizovacích nákladů. Zároveň je primárně dělaný, pro servery s operačním systémem Red Hat Enterprise Linux, kterých má organizace nejvíce. Tento nástroj byl již v rámci organizace dříve použit, ovšem pouze pro zprostředkování repositářů virtuálních stanic s operačním systémem Red Hat v Oracle virtuálním prostředí. Z tohoto důvodu se přistoupilo k úplně nové implementaci. Konečný výběr nástroje nebyl čistě na autorovi práce, ale konečné rozhodnutí bylo na vedení odboru.

4.3 Procesy a zdroje dat

V rámci prvotní analýzy byl vytvořen seznam činností (tabulka níže), které jsou v rámci organizace obstarávány odborem informačních technologií, oddělením provoz IT.

Vznikl na základě dokumentace organizace, autorových vlastních zkušeností a komunikací s dalšími zaměstnanci odboru. Na základě toho byl zpracován návrh IT procesů, který byl následně diskutován. Cílem diskuse bylo vybrat procesy, které by byly nejvhodnější pro automatizaci. K výběru došlo, jak z hlediska úspory času pro pracovníky, tak logického hlediska vzhledem k možnostem dalšího rozšiřování funkcí. Hlavními procesy, které byly zvoleny pro automatizaci, a tím i tuto práci, byla správa repozitářů, serverů a administrátorských uživatelů operačního systému Linux. Před začátkem projektu bylo nutné konfigurovat a spravovat každý server zvlášť. Tudíž pokud administrátor chtěl aktualizovat větší množství serverů, tak se musel manuálně na každý z nich přihlásit a aktualizaci spustit. Stejně tomu tak bylo i u jakýkoliv jiných činností, které bylo potřeba udělat na více serverech najednou. Celkově postup zabíral velké množství času. Proto hlavním požadavkem projektu bylo implementovat systém, který by umožňoval správu serverů dělat vzdáleně, hromadně a z jednoho rozhraní nebo serveru. Stejný problém byl v rámci správy administrátorů operačního systému Linux. Zde byl požadavek, aby konečné řešení dokázalo hromadně účet založit, zakázat, přidat do různých skupin a zároveň šlo provést mnoho dalších nastavení. Běžné uživatelské účty jsou řešeny pomocí Active directory (adresářová služba od firmy Microsoft). Také bylo požadavkem, aby mohl být systém do budoucna rozšířen o mnoho dalších funkcionalit, které mohou pomoci s automatizací i v dalších procesech. Celkový výběr procesů pro automatizaci byl schvalován na vyšší úrovni vedení. Procesy jsou v rámci vlastní části práce chápány jako činnosti IT oddělení vybrané organizace.

Tabulka 1: Vybrané činnosti IT oddělení

Inventarizace a evidence majetku
Monitoring
Podpora uživatelů – Helpdesk/ServiceDesk
Přidělování a správa licencí
Správa aplikací
Správa certifikační autority
Správa cloudových aplikací (Azure, Oracle)
Správa databázových systémů

Správa dodavatelů
Správa docházkového a přístupového systému
Správa interních repozitářů
Správa komunikačních systémů (email, webex, jabber)
Správa LAN
Správa perimetru
Správa pracovních stanic
Správa serverů
Správa tiskáren – tiskové služby
Správa uložišť a diskových polí
Správa uživatelů
Správa virtuálních prostředí VMware a OVM
Správa VOIP ústředny
Správa Wi-Fi sítě
Správa WWW
Školení uživatelů
Zálohování

Co se týče zdrojů dat (konfigurace linuxových serverů), tak v organizaci je přes 450 serverů (virtuální plus fyzické). Do projektu byly zapojeny pouze ty, které splňovaly podmínky určité verze operačního systému. Jednalo se o servery s Red Hat Enterprise Linux 7 a novější, Ubuntu 20.04 a novější.

Vyřazeny byly servery CentOS, se kterými se původně počítalo. Od nich se nakonec vzhledem k velkému množství repozitářů ustoupilo a bylo rozhodnuto, že tyto servery budou postupně převedeny na systém Red Hat Enterprise Linux. Další větší vyřazenou skupinou byly historické servery. U těchto serverů byla zakázána jakákoliv manipulace. Poslední početnou a nyní vyřazenou skupinou byly Windows servery. U těch se sice do budoucna s automatizací počítá. Nyní se však nejednalo o primární zaměření diplomové práce vzhledem k jejich počtu.

Po tomto výběru tak celkový počet serverů pro automatizaci skončil na přibližně 250 serverech. Na těchto serverech byl proveden scan s využitím Ansible na základě,

kterého byly určeny veškeré unikátní repozitáře, které se napříč celou organizací využívaly. Výstupy následně sloužily pro určení veškerých dat, který bude muset implementovaný systém obsahovat.

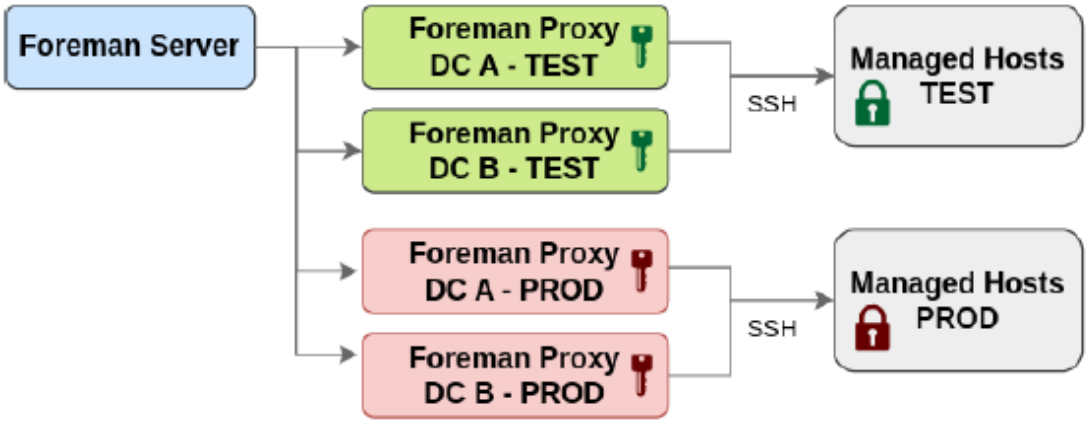
4.4 Instalace Foreman

Pro fungování Foreman se v rámci organizace využívá pět serverů. Foreman server neboli také managment server, je hlavní server, na kterém je nainstalovaný managment Foreman. Zbylé čtyři servery jsou Foreman proxy neboli také smart proxy, pomocí kterých se distribuuje obsah na jednotlivé servery. Čtyři proxy servery vznikly z důvodu „čtyř prostředí“: DCA/TEST, DCA/PROD, DCB/TEST a DCB/PROD. Tato prostředí jsou oddělena. Každé prostředí má přiřazen jeden proxy server. Níže v tabulce jsou požadavky na oba typy serverů. Zároveň na obrázku níže je schéma propojení Foreman server a Foreman proxy.

Tabulka 2: Požadavky na Foreman

Systém	Požadavky	
Foreman Server (1x)	OS	Red Hat Enterprise Linux 8.6 a novější 64-bit
	RAM	20 GB* (povýšeno na 32 GB)
	CPU	8
	OS Partitions	30 GB - / 4 GB SWAP
	Data Partitions	50 GB - /var/lib/pgsql 600 GB - /var/lib/pulp
Foreman Proxy (4x)	OS	Red Hat Enterprise Linux 8.6 a novější 64-bit
	RAM	20 GB
	CPU	8
	OS Partitions	30 GB - / 4 GB SWAP
	Data Partitions	50 GB - /var/lib/pgsql 600 GB - /var/lib/pulp

* Při používání bylo zjištěno, že 20 GB RAM u Foreman serveru je nedostatečné. Hlavně kvůli synchronizaci balíčků. Proto byla RAM povýšena na 32 GB.



Obrázek 6: Schéma propojení v rámci Foreman

4.4.1 Foreman server

Po přípravě operačních systémů serverů je nutné provést instalaci Foreman serveru. Před začátkem je zapotřebí vypnout veškeré repozitáře, které na serveru jsou již nainstalovány. Pro tento úkon byl využit příkaz subscription-manager repos --disable "*". Následně došlo k mazání metadat. Poté byly instalovány tři základní balíčky. Byly jimi foreman-release, katello-repos-latest a puppet7. Pro katello a puppet byly zvoleny moduly. Moduly fungují od verze Red Hat Enterprise Linux 8. Slouží u repozitářů pro odkrývání a skrývání různých balíků uvnitř celého repozitáře. V tomto případě byl odkryt například balíček pro postgresql 12. Následně byla provedena aktualizace balíčků. Poté probíhala instalace repozitáře foreman-installer-katello, který instaluje veškeré závislosti mezi jednotlivými balíčky.

```
dnf install https://yum.theforeman.org/releases/3.7/el8/x86_64/foreman-release.rpm
dnf install https://yum.theforeman.org/katello/4.9/katello/el8/x86_64/katello-repos-latest.rpm
dnf install https://yum.puppet.com/puppet7-release-el-8.noarch.rpm
dnf module enable katello:el8 pulpcore:el8
dnf update
dnf install foreman-installer-katello
```

Obrázek 7: Instalace balíčku a spuštění foreman-installer-katello

Po nainstalování potřebných balíčků bylo nutné vytvořit základní konfigurační soubor, který zároveň fungoval jako skript. Tento soubor upravoval parametry puppetu. Tyto parametry jsou většinou při upravení a znovu spuštění přepsatelné. Vše se tvoří pomocí foreman-installer --scenario katello, který má mnoho parametrů viz obrázek níže. Celkově je možné mít mnohem více parametrů. Vzhledem ke složitostem některých funkcí bylo rozhodnuto, že bude udělána spíše menší instalace. Jakékoliv zakázané možnosti je

v budoucnu možno povolit. Naopak většinu z těchto možností je v případě potřeby možno zakázat.

```
foreman-installer --scenario katello \
--foreman-initial-organization [redacted] \
--foreman-initial-location [redacted] \
--foreman-initial-admin-username [redacted] \
--foreman-initial-admin-password [redacted] \
--enable-foreman-cli-ansible \
--enable-foreman-plugin-ansible \
--enable-foreman-proxy-plugin-ansible \
--enable-foreman-cli-ssh \
--enable-foreman-compute-vmware \
--enable-foreman-proxy-plugin-remote-execution-script \
--puppet-runmode=none
```

Obrázek 8: Konfigurace managment serveru

- Organization – určení názvu organizace. Je důležitá, protože se vyskytuje v mnoha příkazech. Například se využívá v případě, kdy uživatel chce stáhnout z katello určitý balíček. V tomto případě se používá celá cesta od úrovně organizace. Je doporučeno, aby byla co nejjednodušší. V tomto případě byla zvolena zkratka organizace. V samotném rozhraní lze tento název libovolně měnit. Lable je však stálý. Jedná se o jeden z hodně těžko přepsatelných parametrů.
- Location – Prvotně při instalaci byl Foreman server vložen do lokality DCA. Následně byly zjištěny chyby, které způsobovalo spojení ostatních severů v DCA a Foreman serveru, proto byla vytvořena později pro tento server speciální lokalita FOREMAN.
- Admin-usermane – Funguje jako lokální administrátorský účet, který se však v běžné praxi nevyužívá. Je používán primárně pro prvotní přihlášení, dokud nejsou vytvořené jiné administrátorské účty (lokální, LDAP).
- Admin-password – heslo k prvotnímu administrátorskému účtu. Lze využít i jiné druhy zabezpečení než pouze heslem.
- Enable-foreman-cli-ansible – Slouží pro povolení ovládání Ansible pomocí Hammer.
- Enable-foreman-plugin-ansible – Povoluje vůbec samotné fungování Ansible.
- Enable-proxy-plugin-ansible – Povoluje správu severů pomocí Ansible za využití proxy serverů. U managment serveru je tento parametr využíván kvůli správě proxy serverů. (v případě tohoto řešení čtyř)

- Enable-foreman-cli-ssh – Díky tomu je povolena možnost pouštět příkazy i přes ssh bez využití Ansible.
- Enable-compute-vmware – Vzhledem k tomu, že od samého začátku bylo počítáno s připojením Vmware k Foreman, tak byl přidán i tento parametr. Díky tomu se přidalo potřebné nastavení, které je k propojení nutné.
- Enable-foreman-proxy-plugin-remote-execution-script – Pro instalaci nejen pomocí Ansible.
- Puppet-runmode=none – Pro vypnutí puppet agenta.

Po spuštění tohoto skriptu je Foreman nakonfigurován a je možné se k němu přihlásit přes webové rozhraní. Zároveň jsou přepsány hodnoty ve foreman-installer --scenario katello. Díky tomu je v případě problému možné tento skript pustit znovu a získat stejnou konfiguraci. Případně je možné konfiguraci, jakkoliv měnit. V tuto chvíli je zároveň možné dělat dílčí konfigurace i bez vytvořeného skriptu, a to zadáním příkazu foreman-installer --scenario katello a administrátorem vybraná konfigurace.

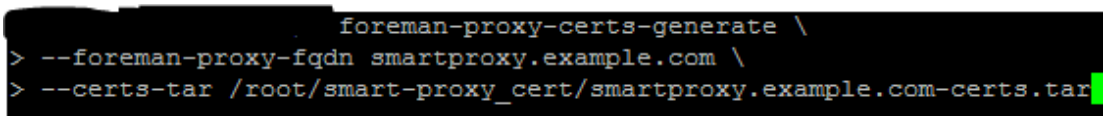
4.4.2 Smart proxy

Před instalací je potřeba udělat čtyři důležité predispozice. První z nich je přidání jedné jakékoliv licence obsahující Red Hat Enterprise Linux do rozhraní Foreman. To z důvodu, že veškeré Red Hat Enterprise Linux repozitáře je možné stahovat jednoduše přes rozhraní. Tento krok se dělá v záložce content – subscriptions – Manage manifest, kde je zvolen vygenerovaný soubor z Red Hat portálu. Je využit Simple Content Access, což znamená, že Foreman nehlídá, kolik serverů máme připojených a kolik licencí má organizace zakoupených. Je pouze důležité, aby věděl, že má alespoň jednu licenci a tím pádem existuje nárok na využívání oficiálních Red Hat repozitářů. Druhou činností, která je potřeba udělat před přidáváním proxy serverů, je přidání foreman, katello, katello-candlepin, pulpcore a foreman-plugins, puppet7 repozitářů do produktů. Díky tomu bude možné po přidání proxy serverů synchronizovat balíčky. V okamžiku, kdy se proxy server připojí k managment serveru, tak přestává čerpat data z internetu a svých lokálních repozitářů a začíná balíčky čerpat z managment serveru podle definovaného content view (bude vysvětleno později) (Dále jen „CV“). Třetí predispozicí je vytvoření aktivačního klíče pro proxy servery. Tento aktivační klíč slouží k tomu, aby server věděl, jaké repozitáře má povolené, tudíž z nich může server čerpat obsah. Čtvrtou a poslední predispozicí je vytvoření všech lokalit

ve kterých proxy servery budou. Jedná se tedy o DCA/TEST, DCA/PROD, DCB/TEST a DCB/PROD.

V tuto chvíli bylo možné vytvořit registraci hosta. U registrace se vyplňuje organizace, lokace, operační systém a aktivační klíč. Následně je vygenerován registrační příkaz, který je poté využit pro registraci serveru. Tento registrační příkaz je univerzální. Tudíž pokud jsou dva a více serverů, které mají mít stejné parametry, může být na jejich registraci využit stejný registrační příkaz. U proxy serverů se musí generovat, pro každou proxy zvlášť, protože každá se nachází v jiném prostředí. U běžných serverů jsou k dispozici další možnosti (host group a smart proxy). Před zadáním příkazu na smart proxy je potřeba server odregistrovat od subskripce pomocí příkazu subscription-manager unregister. Následně lze zadat vygenerovaný registrační příkaz. V tomto bodě nastala prvotně chyba, kdy registrace nemohla proběhnout kvůli tomu, že se smart proxy nedokázala ověřit vůči certifikátu managment serveru. Tento problém byl vyřešen přidáním parametru k do příkazu za znaménko - : curl -ksS … Parametr říká, aby registrace přeskočila krok kontroly certifikátu a celá registrace proběhla i tak.

Po registraci je nutné vygenerovat potřebný certifikát na managment serveru pro komunikaci se smart proxy. Používá se příkaz foreman-proxy-certs-generate a další parametry. Pro každý proxy server je příkaz upraven podle názvu serveru a složky kam se budou vygenerované certifikáty ukládat. Následně je nutné zkopírovat vygenerované certifikáty do domovské složky uživatele root na proxy serveru.



Obrázek 9: Generování certifikátů pro smart proxy

Poté už je potřeba nainstalovat potřebné balíčky a stejně jako u instalace managment serveru je nutné povolit moduly. Následně se spouští konfigurační skript. Tento konfigurační skript slouží primárně pro upřesnění komunikace mezi smart proxy a managment serverem. Zároveň je potřeba povolit Ansible, který bude následně využíván pro ovládání spravovaných serverů.

```
dnf install foreman-proxy-content
dnf install foreman-installer-proxy-content
dnf  module enable katello:el8 pulpcore:el8
dnf install foreman-proxy-content
sh foreman-proxy-install.sh
```

Obrázek 10: Instalace balíčků na smart proxy

```
foreman-installer \
--scenario foreman-proxy-content \
--certs-tar-file                    "/root/                cz-certs.tar" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-foreman-base-url    "
--foreman-proxy-trusted-hosts       "
--foreman-proxy-trusted-hosts       "
--foreman-proxy-oauth-consumer-key  "
--foreman-proxy-oauth-consumer-secret '
--puppet-runmode=none \
--enable-foreman-proxy-plugin-ansible \
--enable-foreman-proxy-plugin-remote-execution-script
```

Obrázek 11: Konfigurace smart proxy

4.5 LDAP

LDAP server zajišťuje v tomto řešení autentizaci účtů. Organizace má veškeré přihlašování do všech aplikací spravováno pomocí active directory, kde existuje více skupin (např. admin, users). V tomto řešení byla zvolena skupina admin. Je tomu tak, protože v současné době jsou všichni zaměstnanci, kteří mají svůj administrátorský účet zároveň pověřeni ke správě serverů. Později je samozřejmě možnost skupinu či skupiny měnit podle potřeby. Zároveň lze různým skupinám přidělovat různá oprávnění uvnitř systému Foreman. Lokální účty díky tomuto řešení nemají žádná oprávnění.

Pro samotnou instalaci bylo nutné získat certifikáty využívané pro službu active directory. Konkrétně se jednalo o tři certifikáty, které byly ve vzájemné struktuře od nejvyšší po nejnižší úroveň. S každým z nich bylo nutné udělat následující kroky. Všechny operace probíhaly na managment serveru.

1. Nedříve bylo nutné nahrát všechny certifikáty na managment server
2. Každý certifikát postupně nainstalovat.

```
install                 /etc/pki/tls/certs/
install                          /etc/pki/tls/certs/
install                               /etc/pki/tls/certs/
```

Obrázek 12: Instalace důvěryhodných certifikátů

3. Následně byly vytvořeny symbolické odkazy na tyto certifikáty a zároveň se generovaly hash hodnoty certifikátů. Díky tomu lze jednodušeji identifikovat certifikáty pomocí jejich hash hodnot než pomocí názvu souborů.

```
ln -s xx1.crt  /etc/pki/tls/certs/$(openssl x509 -noout -hash -in /etc/pki/tls/certs/xx1).0
ln -s xx2.crt  /etc/pki/tls/certs/$(openssl x509 -noout -hash -in /etc/pki/tls/certs/xx2).0
ln -s xx3.crt  /etc/pki/tls/certs/$(openssl x509 -noout -hash -in /etc/pki/tls/certs/xx3).0
```

Obrázek 13: Vytvoření symbolických odkazů

4. Následoval restart služeb httpd.service httpd a Foreman, pro znovunačtení certifikátů.

```
systemctl restart httpd.service
systemctl restart foreman
systemctl status foreman
```

Obrázek 14: Restart služeb pro znovunačtení certifikátů

5. V tento moment byly ve Foreman nastaveny údaje o AD serveru, oproti kterému se pak ověřují přihlašovací údaje. Bohužel nastala chyba, která byla zapříčiněna chybějícím kořenovým certifikátem. Proto byl vytvořen balíček, který obsahoval všechny tři certifikáty od nejvyšší úrovně po nejnižší. Ten byl následně naimportován do autorizovaných certifikátů. Následně došlo k aktualizaci autorizovaných certifikátů a znovu proběhl restart služeb.
6. Bohužel ani předchozí krok nepomohl. Po následné analýze bylo zjištěno, že problém se ukrýval v zabezpečení jedno z certifikátu využívaných na AD serveru. Je tomu tak historicky z důvodu jiných systémů, které by mohly mít problém se silnějším druhem zabezpečení. Na druhou stranu na základě tohoto zjištění byly zahájeny kroky k nápravě. Samotný problém byl nakonec vyřešen ponížením požadavků, které managment server na certifikáty má. Bylo tak docíleno pomocí změny politiky z původní default na legacy. Toto řešení není zcela ideální. V budoucnu, až bude vyřešen problém s certifikátem, bude nastavení politiky přepnuto zpátky, v ideálním případě povýšeno.

```
update-crypto-policies --set LEGACY
```

Obrázek 15: Aktualizace politik

7. Znovu proběhl restart veškerých služeb. Proběhla kontrola funkčnosti, která byla úspěšná.
8. Pro možnost přístupu byla nastavena skupina admin, tak aby se běžný uživatel nemohl do systému Foreman přihlásit.

Došlo k sebrání veškerých pravomocí, které lokální účty měly. Jedinou výjimkou je záložní účet, který by byl použit pouze v případě nefunkčnosti active directory.

4.6 Správa repozitářů

Jednou z nejdůležitějších částí, která byla v rámci celé práce implementována je celková správa repozitářů a jejich balíčků. Prvotními kroky jsou příprava veškerých repozitářů a k nim určení pravidel synchronizací. Následně jsou tyto produkty spojovány do takzvaných CV. Správa obsahu je velmi důležitá, protože se jedná o základ celkové automatizace aktualizačního procesu všech serverů připojených k systému Foreman.

4.6.1 Produkt

Produkty jsou úplným základem celého aktualizačního procesu. Jedná se o veškeré repozitáře, které jsou na připojených serverech. Z tohoto důvodu bylo důležité nejdříve zjistit, které repozitáře a jaké linuxové distribuce se vyskytují napříč celou organizací. Každý produkt má v sobě obvykle mnoho repozitářů. Příkladem může být databázový systém Postgresql. Jméno produktu je Postgresql a pod tímto produktem je schovaných mnoho verzí (Postgresql 10 až 15 a zároveň pro každou verzi linuxové distribuce, na kterých bude daný repozitář zapotřebí). Každý repozitář má svůj unikátní lable, pomocí kterého je možné s produktem pracovat na konzoli. Důležitou volbou je pak Published At – tato volba určuje, jakým způsobem je obsah ukládán. V případě volby Immediate jsou veškerá data repozitáře uložena přímo na serveru v plné velikosti. U volby On Demant jsou na server stahovány pouze metadata. V případě, že je daný repozitář potřeba na některém ze systémů využít, pak se teprve stahují potřebné soubory na základě předem stažených metadat. V předchozích verzích Foreman se u velkých repozitářů občas stávalo, že při vyvolání kompletních dat na základě metadat docházelo k chybě. V aktuální verzi (3.7), která je současně instalována chyba objevena nebyla. Zároveň je možnost volbu kdykoliv změnit. Vždy po změně možnosti je u repozitáře nutné zvolit volbu Reclaim space, která smaže původní obsah a podle nového nastavení stáhne příslušná data.

Většina repozitářů je nastavena na volbu On Demand z důvodu ušetření místa na serveru. Při prvotním přidávání produktů došlo k zahlcení disku při pokusu o stažení dvou verzí linuxové distribuce Ubuntu. Jediné repozitáře, které jsou v současné době udržovány ve stavu Immediate jsou repozitáře Red Hat Enterprise Linux. Zde je doporučeno, aby

balíčky byly přímo staženy. Zároveň má organizace nejvíce serverů právě na linuxové distribuci od společnosti Red Hat.

4.6.2 Content View

Jedná se o balíčky repozitářů, které jsou využívány na jednom nebo více stejných serverech. Základní myšlenkou je možnost spravování verzí repozitářů na jednotlivých serverech. To bylo i jedním z hlavních zadání a cílů celé první fáze automatizace v organizaci. Každý server může mít přidělené pouze jedno CV. Na základě toho, do kterého CV patří, pak na serveru je možné provádět aktualizace a instalace potřebných balíčků. Pokud některý z repozitářů je ve stavu On Demand, pak i CV je automaticky v tomto stavu. Každé CV má vlastní verzi, kterou spravuje vždy administrátor, tento krok lze případně automatizovat. Automatizace však není zcela doporučována, protože by veškeré verze balíčků měly být pro servery zpřístupňovány až po expertním rozhodnutí. V rámci CV lze využívat lifecycle environment, ve kterém jsou v tomto řešení vytvořeny dvě fáze. Jedná se o Fázi 1 a Fázi 2. Fáze 1 představuje testovací prostředí. Fáze 2 prostředí produkční. Správce si vždy může vybrat, které verze zpřístupní pro danou fázi. Nemusí přidávat verzi do žádné fáze, ale zároveň může do obou. Myšlenkou tohoto řešení je možnost předejít chybám v produkčním prostředí. V současné době jsou aktualizace dělány v testovacím prostředí a pokud se nevyskytnou chyby přibližně po týdnu, jsou zpřístupněny a instalovány i na produkční servery. Velkou výhodou CV je tedy možnost zamrazení jednotlivých verzí a následná práce s nimi. Veškerá činnost je historizována, tudíž lze zjistit, kdo a kdy vytvořil jakou verzi CV a které komponenty obsahovala.

Existují dva typy CV. Klasické a kompozitní.

- Klasické – funguje tak, že jsou vybrány repozitáře na základě požadavku. Většinou se jedná o obsah všech repozitářů, které byly na určitém serveru, pro který je CV tvořeno.
- Kompozitní – jedná se o CV, které slučuje dvě a více klasických CV. Velkou výhodou je, že jednotlivé CV lze aktualizovat podle potřeby a následně v kompozitním CV využít verzi, kterou administrátor chce. V tomto řešení je využito například pro aktualizaci Zabbix serveru. Zde je nutné, aby se aktualizace systému dělaly jednou týdně, ale zároveň balíčky Zabbix server zůstaly stále na stejné verzi. Proto je využito kompozitní CV, díky kterému lze neustále aktualizovat CV

obsahující repozitáře pro Red Hat Enterprise Linux 8 a zároveň držet stále stejnou verzi Zabbix server repozitáře, dokud není rozhodnuto o nutnosti aktualizace.

Je více přístupů, jakým CV zpracovávat. Vždy záleží na aktuálních potřebách a podmínkách, které v dané infrastruktuře jsou. V rámci této práce byly vyzkoušeny dva přístupy. Prvním přístupem bylo vytvoření jednoho CV pro celou jednu verzi linuxové distribuce. Do toho CV se přidaly veškeré repozitáře, které byly využívány napříč všemi servery dané linuxové distribuce a verze. To se však neukázalo jako zcela vhodné, protože veškeré servery, které do tohoto CV patřily, měly přístup ke všem repozitářům, a to i těm, které na nich nebyly využívány. Díky tomu byla menší přehlednost, co na kterých serverech je a není instalováno. Druhým přístupem, který je zároveň i využíván, je vytvoření CV pro servery, které využívají stejné repozitáře. Takových serverů sice není napříč organizací úplně mnoho a tím pádem vzniká i větší množství CV, ale na druhou stranu je větší přehled o tom, co a kde je instalováno. V tomto druhém přístupu jsou také využívány kompozitní CV.

Lze tvořit také filtry pro jednotlivá CV. Tyto filtry se aplikují na jednotlivé repozitáře, které jsou přidány. Jsou vybírány balíčky, které buď jsou do daného CV použity nebo naopak, které administrátor chce vynechat. Filtry byly původně využity u repozitářů Zabbix. Cílem bylo, aby v každém základním contet view byly balíčky obsahující Zabbix agent a pouze na samotném Zabbix serveru byly všechny balíčky. Bohužel po aplikování filtrů bylo zjištěno, že tyto filtry způsobují problémy při vytváření nových verzí a následné synchronizaci na proxy serverech. Proto byly filtry odstraněny. V tuto chvíli je problém vyřešen pomocí kompozitního CV, kde pro Zabbix server bylo vytvořeno CV, ve kterém je pouze Zabbix produkt, a tím se zvlášť tvoří verze balíčku a zároveň se zvlášť tvoří verze systémových balíčků. U normálních serverů bylo zjištěno, že Zabbix agent je již obsažen v systémových repozitářích.

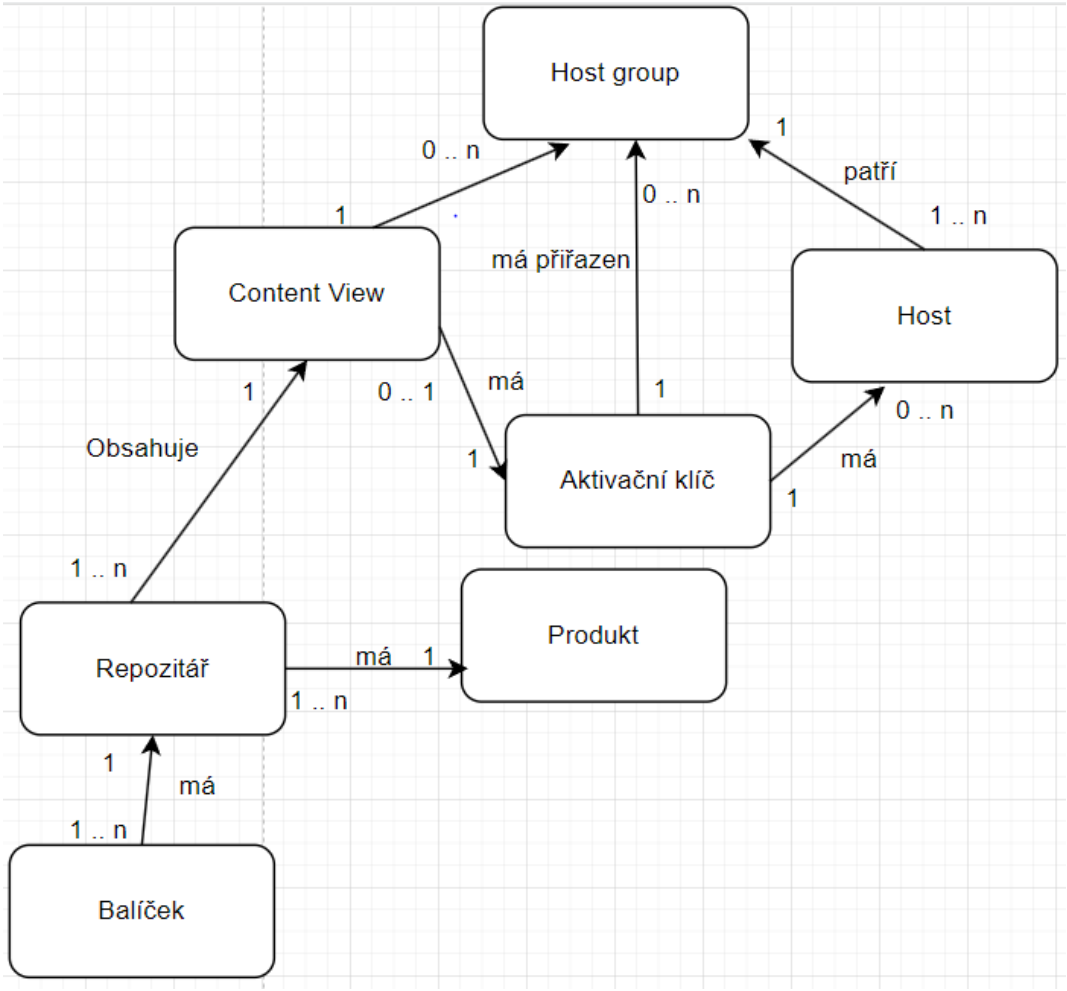
4.6.3 Aktivační klíč

Každý server neboli host má přiřazený právě jeden aktivační klíč. Tento aktivační klíč se využívá pro registraci serverů. V rámci tvorby musí administrátor zadat jméno, fázi do které bude spadat server registrovaný tímto klíčem, a CV, které server má mít přiděleno. Po vytvoření následuje povolení všech potřebných repozitářů. Zpravidla se povolují pouze repozitáře, které spadají do vybraného CV. Je tomu tak z důvodu, že server, který je

registrován daným aktivačním klíčem má zpřístupněné pouze repozitáře, které jsou povolené. Samozřejmě tento stav není konečný. Jedná se pouze o stav, který bude po samotné registraci. Následně lze pomocí správy samotného serveru a CV libovolně přidávat a odebírat obsah podle potřeb administrátora

4.7 Správa serverů

Jak již vyšlo z analýzy procesů, které v rámci organizace má na starosti IT oddělení je jedním z hlavních bodů této práce správa serverů. Hlavní snahou je celkový proces, co možná nejvíce zjednodušit a zároveň od samého začátku důsledně dbát na to, aby celkový systém byl připraven pro kompletní automatizační řešení. V rámci prostředí Foreman se serverem rozumí host. Na schématu níže jsou vidět kompletní závislosti v rámci Foreman.



Obrázek 16: Schéma závislostí v rámci Foreman

4.7.1 Skupiny hostů

Jedná se o další velmi důležitou predispozici pro kvalitní zpracování konfiguračního managmentu. Funkcionalita Host Groups je využita k vytvoření primární stromové struktury pro organizaci spravovaných serverů za účelem parametrizování jejich konfigurace (přiřazování aktivačního klíče). Následně je tento strom využit pro konfigurační management pomocí Ansible. Je velmi důležité, aby na nejvíce podrobné úrovni byly pouze servery, které jsou svou konfigurací úplně totožné. V případě, kdy dojde k jakékoliv odchylce je nutné rozdělit danou úroveň na dvě a více, či vytvořit celou další úroveň stromu. Každá skupina hostů má vlastní jméno. Celkové rozdělení má velký vliv na samotnou automatizaci a logiku celého řešení. Je proto dobré při zakládání nových serverů myslet na určitou logickou strukturu.

Tato logická struktura je následně přiřazena k danému proxy serveru podle lokality a prostředí, do které skupina spadá. Fáze, je vybrána automaticky na základě smart proxy, případně CV, pokud je rozhodnuto, že bude přiřazeno na úrovni skupiny hostů. Tento způsob v této práci není využit. Druhým způsobem přiřazování obsahu skupinám hostů je pomocí přiřazených aktivačních klíčů. Kdy při automatické registraci na základě parametrů serveru se přiřazuje patřičný aktivační klíč, se kterým je příslušené CV následně spojeno. Tento postup bude později podrobněji popsán z důvodu využití při implementaci řešení. K jednotlivým skupinám lze také přidávat různé Ansible role.

Struktura je navržena, tak aby co nejvíce reflektovala situaci v rozložení systémů.

1. CORE – Tvoří kořen stromu. Obsahuje veškeré výchozí nastavení pro celou infrastrukturu.
2. LOKACE – Definuje fyzickou lokalitu. Vychází ze zkratky lokalit. Zatím DCA a DCB.
3. PROSTŘEDÍ – Definuje prostředí, ve kterém se servery nachází. Testovací, Produkční.
4. TYP – Rozřazení podle typu serveru. Management server, aplikační server, databázový server.
5. APLIKACE – Skupina konkrétních systémů, které spolu souvisí a mají stejnou konfiguraci. (např. jsou provozovány pro konkrétní aplikaci, nebo tvoří jeden cluster)
6. DRUH – Jedná se o upřesnění funkce aplikace.

Samotné zakládání skupin serverů je možno dělat dvěma způsoby. První způsob je vytváření skupin po jedné. Tento postup je velmi časově náročný a vyplatí se v případě, kdy je zapotřebí vytvořit jen malý počet skupin. Druhý způsob, který byl i použit, využívá pro přidání příkaz hammer. Hammer je specifický nástroj pro ovládání Formana, který se spouští na managment serveru. To, co lze ručně zadat ve webovém rozhraní, je možné pomocí hammer upravit v příkazovém řádku na managment serveru. Nejprve byl vytvořen excel soubor, ve kterém jednotlivé sloupce představovaly úrovně ve stromové struktuře. Tento soubor vznikl na základě analýzy prostředí a dokumentace LAN tvořené externím dodavatelem. Následně byl tento soubor vložen na managment server. Pomocí příkazu jsou poté všechny záznamy upraveny do požadované formy. Výstup je uložen jako skript, který je následně spuštěn. Tento krok je potřeba opakovat pro všechny úrovně. V případě této implementace se jednalo pouze o úroveň 6 a 7. Všechny nižší úrovně byly vzhledem k jejich rozsahu vytvořeny ručně. Systém si zároveň sám hlídá, že není některý ze záznamů duplicitní. $ v příkazu níže značí o kolikátý sloupec se jedná. V případě, kdy byla tvořena vrstva 7 bylo místo $6 napsáno $7 a za $5 bylo napsáno "/"$6.

Příkazy: cat testovaci_prostredi.csv | awk -F ',' '{ print "hammer hostgroup create --name "$6" --parent-title core/"$3"/"$4"/"$5 }' | uniq > hg.sh

sh hg.sh

4.7.2 Zakládání hosta

Tím, že je založen host, Foreman ví, že nějaký takový server existuje. Neznamená to však, že je server registrován a může být skrze tento systém spravován. Tento postup je dělán z důvodu, že následná registrace může být automatizována. V případě, kdy jsou servery správně založeny, je možné na základě skupin hostů provést jejich hromadnou registraci. Samozřejmě v případě, že mají veškeré predispozice, které jsou k tomu potřeba. O automatické registraci bude psáno v následující kapitole Registrace hosta.

K vytváření hosta jsou v rámci organizace tři základní přístupy. Tyto přístupy se liší v závislosti na počtu serverů, které je potřeba založit. V případě, kdy administrátor potřebuje přidat jeden či dva servery je možné hosta přidat přes webové rozhraní. Přiřadí se mu jméno, jméno organizace, lokace, skupina hostů, zbytek předvoleb si systém vyplní sám na základě závislostí, které host group má (včetně Ansible rolí a upravených parametrů). V případě, že je zapotřebí přidat větší počet serverů je vhodné stejně jako u vytváření skupin hostů

využít nástroj hammer. Tento postup byl využit i při tvorbě této práce, kdy pro prvotní nahrání byl hammer využit. Ke zpracování údajů bylo využito stejného souboru, jako při vytváření skupin hostů. Jediným rozdílem bylo zvolení jiných parametrů, sloupců a přidání určitých parametrů, které jsou specifické pro tvorbu nových záznamů hosta. U možnosti – location-title je parametr toupper to z důvodu, že v excel souboru bylo určení lokalit malými písmeny. Naopak ihned po instalaci byly nastaveny lokace velkými písmeny. Tento parametr převádí malá písmena na velká. Všechny další konfigurace si stejně jako u přidávání jednotlivých serverů přebírá podle přiřazení do skupiny hostů.

cat testovaci_prostredi.csv | awk -F ',' '{ printf "hammer host create --name "$1" --build false --managed false --ip "$2" --organization-title xxx --location-title "toupper($3)"/"toupper($4)" --hostgroup-title core/"$3"/"$4"/"$5"/"$6; print ($7 != "") ? "/"$7 : "" }'

Třetím a posledním přístupem využívaným v této práci bylo vytvoření serveru přes VMware ve kterém organizace má veškeré virtuální servery. Napojení bude popsáno později.

4.7.3 Zakládání serverů ve VMware

Zajímavou funkcionalitou je propojení VMware s Foreman. Pomocí, kterého lze vytvářet pomocí image nové servery a případně je rovnou přiřazovat do skupin hostů a registrovat. Hlavním cílem této práce byla prostá instalace nového serveru a přiřazení ho do seznamu hostů. Samotná registrace se pak dělá zvlášť. VMware se do Foreman přidává jako Compute Resources. Díky tomu ho dokáže zobrazovat při vytváření nových serverů. Do Vcentra přistupuje pouze managment server. Smart proxy zde nemají žádnou úlohu. Jedno Compute Reseources značí jedno datové centrum. Po zadání údajů, jako jméno, provider (VMware), Vcenter, username, password a datacenter je nutné vybrat lokalitu ve které se VCentrum nachází. Organizace je v případě této implementace vyplněna automaticky (existuje pouze jedna).

Po připojení je nutné přidat požadovanou image, do záložky Images. Image je z pohledu Foreman, ve VMware se jedná o template. Administrátor musí udat, jaký template odpovídá na straně VMware. Jméno by ideálně mělo odpovídat jménu template. Username je vždy root. Dále se určuje operační systém, architektura, uživatel a image, které odpovídá té na straně VMware. Skrze Foreman nelze nastavovat serverům IP adresy. Proto se využívá balíček cloud-init, který se spustí ihned po prvním startu nového serveru. Díky němu je

možné si z Foreman stáhnout patřičné počáteční nastavení. Druhou variantou je využití vm-tools, které nabízí obdobné možnosti jako cloud-init. Aby server po načtení mohl využívat vm-tools či cloud-init je nutné zakliknout volbu User Data. Ve finálním řešení jsou využity vm-tools. Bylo též nutné při prvotní instalaci nastavit subnets, který se využívá při vytváření hosta.

Samotná tvorba nového hosta probíhá ve webovém rozhraní v záložce create host. Vytváření funguje stejně, jako pro vytváření jednoho či pár serverů. Rozdílem však je, že u možnosti Deploy On musí být zakliknuto dané vcentrum. Zároveň se v záložce Virtual Machine musí nastavit potřebné parametry, které odpovídají konfiguraci image (Disky nemusí být nastaveny. Nastavují se z přidělené template). V kolonce Boot order se nechává pouze Harddisk, protože síť je řešena pomocí vm-tools. V záložce Operating System zvolit v Provisioning Method Image Based, vybrat příslušná operační systém, image a nastavit root password. A zmáčknout tlačítko resolve. Poslední nastavení je v záložce Intefaces, kde se upravují údaje o síťové komunikaci serveru. Celkově lze proces zjednodušit na úrovni skupiny hostů, kde může být rovnou předepsáno, že se má pro servery této skupiny využívat VMware. Z tohoto důvodu je možné připravit Compute profile ve kterém lze nastavit konkrétní parametry pro, které se jinak nastavují v záložce Virtual Machine. Samozřejmě Compute profile lze využít i pro zakládání serverů bez použití host groups. Zároveň také vznikl parametr ssh_authorized_keys, který je upravován vždy na úrovni prostředí. Do hodnoty je zadán klíč smart proxy, která k prostředí přísluší. Tento klíč pak používá vm-tools po zapnutí, aby se použila nastavená konfigurace.

Pro instalace vznikla template minimalistické instalace, která bude následně upravována dle potřeb. V minimalistické instalaci je pouze uživatel Foreman se čtyřmi klíči (klíč každé smart proxy) a s právy administrátora. Po instalaci je využita role pro registraci hosta. V tu chvíli lze skrze Foreman provádět, jakékoliv další konfigurace, včetně přidávání repozitářů a dalších uživatelů. V budoucnu se počítá s tím, že veškerá další konfigurace bude dělána pomocí Ansible rolí.

4.7.4 Registrace hosta

Nejednoduší cestou, jak registrovat server je ručně ve webovém rozhraní Foreman zvolit záložku host – register host. Tam je nutné vyplnit kolonku organizace, lokalita, skupina hostů, smart proxy a případně operační systém, ten však není nutný, protože po registraci si Foreman sám dokáže zjistit, jaký operační systém se na serveru vyskytuje.

Poslední možností je aktivační klíč. Automaticky jsou nabízeny klíče, které splňují podmínku správné fáze a zařazení v závislosti na skupině hostů. Po dokončení tohoto procesu je nutné vygenerovat registrační příkaz. Tento registrační příkaz je podobný tomu, který byl využíván při registraci proxy serverů, akorát s jinými parametry. Zároveň bylo zjištěno, že je dobré vždy zvolit možnost insecure. V tu chvíli totiž registrovaný systém nepožaduje administrátorské heslo. Na závěr je potřeba registrační příkaz spustit na serveru.

Předchozí proces byl sice relativně jednoduchý, ale vzhledem k tomu, že bylo potřeba postupně přidat přes 200 serverů, tak bylo rozhodnuto, že bude tento proces, pokud možno, co nejvíce automatizován pomocí Ansible. Zároveň díky němu je v současné době možné registrovat mnoho serverů najednou. I zde však nastaly problémy. Největším problémem je různorodost nastavení systémů. Pro správné fungování Ansible přes Foreman je nutné, aby na serverech byl uživatel, který „vyvolává“ potřebné kroky registrace a v budoucnu jakékoliv spravování pomocí Ansible. Dalším problémem byla nefunkčnost samotného Ansible na serverech. I přesto se ukázalo, že krok celkové automatizace registrace serverů byl dobrý, protože tyto chyby, které nastaly by se stejně v budoucnu musely řešit, aby mohly být automatizovány další procesy.

Po studii analýzy prostředí bylo vedením rozhodnuto, že do tohoto systému budou vkládány pouze servery obsahující linuxovou distribuci Red Hat Enterprise Linux a Ubuntu. Nejdříve byl vytvořen playbook, který dokázal automaticky registrovat pouze Red Hat Enterprise Linux servery. Foreman je celkově primární nástroj pro správu Red Hat Enterprise Linux serverů, a proto je nutné pro registraci Ubuntu serverů provádět více kroků, tak aby vše fungovalo, jak má. I přesto jsou však některé funkce nedostupné. Například kontrola počtu balíčků připravených k instalaci. Z tohoto důvodu vznikla role, díky které je možné automatizovaně registrovat servery obou distribucí.

Před samotným použitím role je nutné na každou ze smart proxy nainstalovat kolekci community.general. Při prvotní instalaci byla výstupem chyba. Poté autor přišel na to, že Foreman bere jako zdroj kolekcí pro Ansible složku /etc/ansible/collections. Po vytvoření této složky a opětovné instalaci už role fungovala, tak jak má.

Role je spouštěna pomocí playbooku. V tomto případě playbook obsahuje tři proměnné. První proměnná udává, jaký aktivační klíč pro registraci bude použit. Důležitou úlohu zde mají skupiny hostů. Právě na nich je upravován parametr podle, kterého role ví, jakým aktivačním klíčem má server registrovat. Parametr je nastavován zvlášť pro každou skupinu hostů v záložce Activation Keys. Pokud je tato možnost prázdná, bere

automaticky klíč, který se jmenuje podle názvu distribuce, verze a – test (např. rhel9-test). Pokud je možnost vyplněna využívá vyplněný klíč. Úplně tento příkaz říká: Pokud parametr nebude vyplněný, pak vypiš název distribuce, verzi a -test, pokud však vyplněný je, použij vyplněnou hodnotu. Druhým parametrem je parametr smart proxy. Tento parametr udává, ke které smart proxy bude server připojen a ze kterého bude brát obsah. Údaj se bere znovu podle skupiny hostů. Posledním parametrem v playbooku je parametr organizace. Organizace je pro všechny servery stejná. Je však doplňována na základě přiřazené smart proxy. V některých případech může být k jednomu systému Foreman připojeno více organizací.

Playbook na registraci host:

```
- hosts: all
  gather_facts: yes
  become: true
  vars:
    foreman_client_activationkey: <%= @host.params['kt_activation_keys'].blank? ? "rhel{{ ansible_distribution_major_version }}-test" : @host.params['kt_activation_keys'] %>
    foreman_client_smart_proxy: "<%= @host.content_source %>"
    foreman_client_organization_id: "<%= @host.rhsm_organization_label %>"
  roles:
    - foreman_client
```

Ve složce defaults a následně v souboru main.yaml jsou uchovány veškeré proměnné, které celá role využívá. Zároveň jsou zde popsány, některé příklady, které neexistují, a proto musí být vždy zadány. Jsou jimi aktivační klíč, smart proxy a organizace. Zadávají se skrze Foreman v rámci spouštění playbooku.

Create_host/defaults/main.yml

```
---
foreman_client_redhat_backup_yum_repolist_destination: "/tmp/repolist-old.txt"
foreman_client_redhat_backup_repo_files_destination: "/etc/yum.repos.d/backup"
foreman_client_activationkey: "SOME-ACTIVATION-KEY"
foreman_client_smart_proxy: "foreman.example.com"
foreman_client_organization_id: "ORG"
foreman_client_deb_backup_repo_files_destination: "/etc/apt/sources.list.d/backup"
```

foreman_client_deb_rhsm_url: "http://{{ foreman_client_smart_proxy }}/pulp/content/{{ foreman_client_organization_id }}/Library/custom/{{ ansible_distribution|lower }}/subscription-manager-{{ ansible_distribution_version | ansible. builtin.regex_replace('\\.','-') }}/ stable main"

foreman_client_deb_rhsm_repo_name: "{{ foreman_client_organization_id }}_{{ ansible_distribution|lower }}_subscription-manager-{{ ansible_distribution_version | ansible.builtin.regex_replace('\\.','-') }}"

Jako první akci, kterou role udělá po incializaci playbookem je, že zkontroluje o jakou linuxovou distribuci se jedná. Zdali ubuntu nebo red hat enterprise linux. Pokud na zvoleném serveru je jedna z těchto dvou distribucí, pak proces pokračuje. Pokud ne, je výsledkem například centos is NOT SUPPORTED by this role (centos není podporován pro tuto roli). A vypíše jaké distribuce jsou podporovány. Pokud je podporovaná, tak se spustí další úloha, která se jmenuje buď ubuntu.yml nebo rhel.yml. Vždy tomu tak je v závislosti na linuxu, který je na registrovaném systému. Dvě specifické úlohy vznikly z důvodu velké rozlišnosti kroků na obou druzích systému.

Create_host/task/main.yml:

```
---
- name: Check if OS distribution is supported by this role
  ansible.builtin.assert:
    that: ansible_distribution|lower in main_foreman_client_supported_distros
    success_msg: "{{ ansible_distribution }} is supported."
    fail_msg: "{{ ansible_distribution }} is NOT SUPPORTED by this role. Currently supported distributions: {{ main_foreman_client_supported_distros|join(', ') }}."
  tags: always

- name: Include distro specific tasks for RHSM preparation
  include_tasks: "{{ ansible_distribution|lower }}.yml"
  tags: always
```

Prvním krokem u Red Hat Enterprise Linux serverů je uložení aktuálního seznamu repozitářů do složky, která je definována ve složce defaults a souboru main.yml. Následně jsou vymazána veškerá metadata. Pak v případě, že je systém registrován, proběhne jeho

odregistrování. Poté je vytvořena složka. Jméno a místo je definováno stejně, jako u ukládání seznamu repozitáře v defaults/main.yml. Následně je vytvořena záloha všech repozitářů, a to pro případ, kdy by registrace neproběhla v pořádku a administrátor by se potřeboval vrátit k předchozímu stavu. Záloha je uložena do předem vytvořené složky. Poté je nainstalován certifikát pro smart proxy, ke které je registrovaný server přiřazen. Na závěr proběhne samotná registrace pomocí aktivačního klíče a organizace definovaných v playbooku.

```
---
- name: Save current repolist
  ansible.builtin.shell: "/usr/bin/yum repolist > {{ foreman_client_redhat_backup_yum_repolist_destination }}"

- name: Clean yum metadata
  ansible.builtin.shell: "/usr/bin/yum clean all"

- name: Unregister if registered
  community.general.redhat_subscription:
    state: absent

- name: Create backup folder for old repo files
  ansible.builtin.file:
    path: "{{ foreman_client_redhat_backup_repo_files_destination }}"
    state: directory
    owner: root
    group: root
    mode: '0644'

- name: Backup all repo files
  ansible.builtin.shell: mv /etc/yum.repos.d/*.repo {{ foreman_client_redhat_backup_repo_files_destination }}
  ignore_errors: true

- name: Install ca-consumer rpm
  ansible.builtin.yum:
    name: "http://{{ foreman_client_smart_proxy }}/pub/katello-ca-consumer-latest.noarch.rpm"
    disable_gpg_check: yes
    state: present

- name: RHSM Register
  community.general.redhat_subscription:
    state: present
    activationkey: "{{ foreman_client_activationkey }}"
    org_id: "{{ foreman_client_organization_id }}"
```

Obrázek 17: Create_host/task/redhat.yml

U ubuntu serverů je registrace poněkud složitější, protože systém Foreman pro něj není primárně tvořen. Jako první krok se na přidávaný server musí nahrát dočasný soubor, který obsahuje red hat subscritpion manager, který slouží pro registraci systému do Foreman. Následně je subscription manager nainstalován. Poté je ze smart proxy stažen script, který je zároveň rovnou spuštěn. Pomocí tohoto skriptu je nastaven nainstalovaný subscription manager. Následně stejně jako u systému Red Hat Enterprise Linux, proběhne registrace systému. Proběhne znovu první krok, z důvodu předchozí registrace systému, kdy tato registrace zakáže využívání subscription manageru. Následně je vytvořena složka pro zálohu a je udělána samotná záloha veškerých repozitářů. Poté bylo zjištěno, že u verze Ubuntu 20.04 musí být u repozitáře subscription manager nastavena architektura amd64. Pokud se jedná o jinou verzi je krok přeskočen. Následuje instalace katello-host-tools, což jsou nástroje pro správu serveru přes Foreman. Na závěr je zapnuta služba rhsmcertd.service, díky které začne server komunikovat s managment serverem pomocí smart proxy.

```
---
- name: Temporary repo source file for RHSM
  ansible.builtin.template:
    src: 'etc/apt/sources.list.d/rhsm.list.j2'
    dest: '/etc/apt/sources.list.d/rhsm.list'
    owner: root
    group: root
    mode: '0644'
  tags:
    - ubuntu_rhsm_install

- name: RHSM Package
  ansible.builtin.apt:
    name: subscription-manager
    update_cache: true
    state: present
  tags:
    - ubuntu_rhsm_install

- name: Bootstrap katello-rhsm-consumer script
  ansible.builtin.shell: "/usr/bin/curl -sL http://{{ foreman_client_smart_proxy }}/pub/katello-rhsm-consumer | bash -"
  tags:
    - ubuntu_rhsm_register

- name: RHSM Register
  community.general.redhat_subscription:
    state: present
    activationkey: "{{ foreman_client_activationkey }}"
    org_id: "{{ foreman_client_organization_id }}"
  tags:
    - ubuntu_rhsm_register

- name: Empty sources.list file
  ansible.builtin.copy:
    dest: '/etc/apt/sources.list'
    content: ''
    backup: true
  tags:
    - ubuntu_rhsm_register

- name: Temporary repo source file for RHSM
  ansible.builtin.template:
    src: 'etc/apt/sources.list.d/rhsm.list.j2'
    dest: '/etc/apt/sources.list.d/rhsm.list'
    owner: root
    group: root
    mode: '0644'
  tags:
    - ubuntu_rhsm_install

- name: Create backup folder for old sources.list files
  ansible.builtin.file:
    path: "{{ foreman_client_deb_backup_repo_files_destination }}"
    state: directory
    owner: root
    group: root
    mode: '0644'

- name: Backup all repo files
  ansible.builtin.shell: mv /etc/apt/sources.list.d/*.list {{ foreman_client_deb_backup_repo_files_destination }}
  ignore_errors: true
```

Obrázek 18: Create_host/task/ubuntu.yml

```
- name: Add repo-override for Architectures
  shell: "/usr/bin/subscription-manager repo-override --repo {{ foreman_client_deb_rhsm_repo_name }} --add Architectures:amd64"
  when: ansible_distribution_version == "20.04"
  tags:
    - ubuntu_rhsm_register

- name: Install katello-host-tools
  ansible.builtin.apt:
    name: ['katello-host-tools','katello-host-tools-tracer']
    update_cache: true
    state: present
  tags:
    - ubuntu_post_register

- name: Service rhsmcertd started
  ansible.builtin.service:
    name: rhsmcertd.service
    state: started
    enabled: true
  tags:
    - ubuntu_post_register
```

Obrázek 19: Create_host/task/ubuntu.yml

Ve složce vars a souboru main.yml je pouze seznam linuxových distribucí, které jsou rolí podporovány.

```
---

main_foreman_client_supported_distros: ['redhat','ubuntu']
```

Obrázek 20: Create_host/vars/main.yml

4.8 Ansible

Ansible v prostředí Foreman má v zásadě dvě úlohy. První, která se nazývá remote execution slouží pro spouštění vzdálených akcí. Jedná se tedy o klasické využití Ansible. Druhou úlohou, kde se Ansible v rámci Foreman využívá je správa a práce se skupinami hostů. Těmto skupinám lze přidávat role a parametry rolí, pomocí, kterých lze na této úrovni spravovat dané skupiny.

V době implantace je prostředí rozděleno do čtyřech základních lokalit. Každá lokalita má vlastní smart proxy, přes které se následně aplikuje na servery obsah. Proto, aby bylo možné z každé smart proxy ovládat pomocí Ansible spravované servery bylo nejdříve nutné udělat několik prerekvizit.

1. Připravit uživatele, který bude využíván pro spouštění Ansible požadavků. V každém prostředí má stejný uživatel jiný ssh klíč. Tento ssh klíč je totožný s ssh klíčem smart proxy, dle lokality. Zároveň tento uživatel musí mít právo přepnutí na administrátora (sudo). Tento uživatel se nazývá „foreman“. Zároveň od chvíle implementace je tento uživatel vytvářen hned při zakládání nových serverů.
2. Pro vytváření tohoto uživatele byl využit ansible playbook a role, které byly pouštěny, pod již existujícími účty ze serveru, kterým se pomocí Ansible dalo na potřebné servery připojit.
    a. Playbook obsahuje název serveru/skupiny serverů, na který se má připojovat, dále je uvedeno, aby se přihlásil pod administrátorský účet (root). Následně se využije ansible role, kde je proměnná users vyměněna za jméno uživatele. V tomto případě Foreman.

```
- hosts:
  become: yes
  become_user: root
  roles:
     - { role: users }

  vars:
    users:
      - foreman
```

Obrázek 21: Playbook pro vytvoření uživatele

    b. Samotná role má tři základní složky defaults, files a task. Ve složce files se nachází veškeré certifikáty, které tato role umí využít. Složka task obsahuje soubor main.yml. Tento soubor obsahuje činnosti, které má role po inicializaci udělat. Nejdříve tedy založí uživatele se jménem specifikovaným

v playbooku a následně ho přidá do skupiny wheel, která je v linuxové distribuci Red Hat Enterprise Linux využívána pro administrátorské účty. V druhém kroku se přidá autorizační certifikát, který se bere ze složky files a je pojmenován názvem uživatele plus -key.pub. Tento klíč nastaví, jako aktivní. Stejná role byla využita i pro ubuntu servery akorát s rozdílem, že skupina byla změněna z wheel na sudo.

```
---
- name: Add a new user named XXX
  user:
    name: "{{ item }}"
    groups: "wheel"
  with_items: "{{ users }}"

- name: Add authorized keys
  authorized_key:
    user: "{{ item }}"
    key: "{{ lookup('file', 'files/'+ item + '-key.pub') }}"
    state: "present"
  with_items: "{{ users }}"
...
```

Obrázek 22: Jednoduchá role pro založení uživatele

3. Správná funkčnost se ověřovala vzdáleným přihlášením z proxy serveru na budoucí spravovaný server, kde byl uživatel vytvořen. Tento krok se dělal pouze u prvního serveru, aby bylo ověřeno, že vše funguje, jak má. Zároveň to však ukázalo problém s DNS záznamy, které nespadají do obsahu práce. Tento problém byl vyřešen. Díky tomu Ansible využívá DNS záznamy nikoliv IP adresy serverů.
4. Nastavení uživatele pro Ansible na managment serveru. Jsou dvě možnosti, jak postupovat
    a. Globálně v nastavení. Nastavení ssh user. Využívá se v případě, kdy je pro všechny smart proxy používá stejný uživatel.
    b. Druhou a zároveň zvolenou možností je přidání parametru remote_execution_user do host group. Tento postup se využívá ve chvíli, kdy je rozhodnuto o tom, že každá smart proxy bude mít svého vlastního uživatele pro vzdálenou správu. A to jak pomocí Ansible, tak remote execution. I přesto, že v konečném řešení vznikl pouze jeden uživatel je možné v případě potřeby kdykoliv tento parametr změnit. Parametr byl zadán zvlášť na třetí úroveň skupiny hostů.
5. Zkouška testovacího příkazu v rozhraní Ansible na připojený server.

4.8.1 Role

Veškeré role by měly být, co možná nejvíce univerzální. Neměly by tedy v sobě mít jakékoliv konkrétní údaje. Vše by mělo být na bázi proměnných. Neměly by obsahovat žádná data konkrétního použití. Samotné použití je pak specifikováno v playbooku nebo na úrovni skupiny hostů. Všechny role jsou uloženy ve složce roles na managment serveru. Nejdůležitější je složka task a soubor main.yml. Ten se spustí vždy, když je role inicializována. Odtud se dále volají veškeré další komponenty. Jedná se tedy o takový rozcestník. Každá role je vždy nahrána, jak na managment server, tak na všechny smart proxy, které roli mohou využívat. Aktuálně veškeré proměnné, které jsou v rolích jdou upravovat na jednotlivých skupinách hostů. Foreman při nahrání role sám dokáže tyto proměnné identifikovat.

Do Foreman rozhraní se role může nahrát buď z managment serveru nebo z jakékoliv smart proxy. Role musí být ve složce /etc/ansible/roles. Následně je nutné podle potřeby upravit parametry na každé skupině hostů. Bohužel v současné verzi 3.7 se musí parametry upravovat přes záložku parameters nikoli ansible roles. Lze očekávat, že v budoucnu budou parametry ansible rolí přesunuty do záložky ansible roles.

4.9 Zakládání uživatelů

Dalším procesem, který byl zvolen k automatizaci je zakládání administrátorských uživatelů operačního systému Linux. Pro komplexní tvorbu byla vytvořena Ansible role. Tato role spravuje lokální uživatelské účty na kompatibilních OS distribucích. Databáze administrátorských účtů včetně jejich parametrů je definována proměnnou localusers_userdb ve složce vars. Proměnnou localusers_realize upravovanou často na úrovni skupiny hostů nebo samotném serveru. Případně na úrovni playbooku. Je možné definovat nastavení jednotlivých účtů na systémech a popřípadě přepisovat některé jejich parametry. Před spuštěním role je nutné nainstalovat na všechny proxy servery kolekci ansible.posix (ansible-galaxy collection install ansible-posix -p /etc/ansible/collection)

Soubor Task/main.yml udává samotné založení uživatele na serveru. Vzhledem k tomu, že Ansible nedokáže v rámci jednoho souboru udělat cyklus, který by vytvářel víc uživatelů najednou musel vzniknout druhý task, který je na tento napojen. V něm jsou popsány veškeré konfigurační činnosti, které role může novému uživateli přiřazovat. Velmi důležitá je složka vars, ve které je soubor main.yml. Tento soubor v sobě nese veškeré

informace o všech administrátorských účtech. Odtud jsou načítána data do tasku user_task.yml, jako proměnné. Každý uživatel může mít zadané úplně jiné údaje. Zároveň při znovu spuštění se nově zadané údaje přepisují. Na konci souboru jsou definovány administrátorské skupiny pro obě podporované linuxové distribuce. Druhou možností, jak držet informace o účtech bylo udržování účtů přímo v systému na úrovni skupiny hostů. To by mělo výhodu, že by se všechny údaje udržovali pouze ve Foremanovi a nemusel by se používat gitlab. Na druhou stranu vzhledem k myšlence robustního konfiguračního managmentu nebylo toto řešení vybráno. V současném vybraném řešení jsou ukládány údaje o administrátorských účtech mimo Foreman, ale informace o tom, jaké administrátorské účty na daných skupinách dají upravovat v rámci parametrů na úrovni skupin hostů. Pro skupinu core, je zadán parametr local_realized: []. Na každé skupině hostů je před využitím role nutné upravit tento parametr podle aktuální potřeby. Může to být klasicky jen napsáním účtu ze seznamu (- foremantest) nebo je možné některého z uživatelů upravit i na této úrovni. Například, když se bude vytvářet uživatel foremantest a administrátor bude chtít k jeho konfiguraci přidat navíc komentář, tak může pomocí parametru napsaném ve formě yaml. Takto lze přepsat jakýkoliv parametr. Co se týče ssh klíčů, tak zde jsou dvě možnosti zápisu. Buď jako odkaz na soubor ve složce file nebo rovnou zadáním celého klíče do konfigurace účtu. Řešení, které je primárně využíváno je řešení s klíči ve složce file. Lze si také zvolit, pro které prostředí bude klíč využit. Je proto nutné zadat parametr env. Všichni administrátoři mají v rámci organizace dva klíče, pro produkční a testovací prostředí.

Tabulka 3: Možné parametry pro administrátorský účet

Parametr	Popis
Name	Posix uživalské jméno (default=**userid**).
State	present/absent (default: present)
Uid	Posix UID uživatele (pokud se neuvede, přiřadí se volné v pořadí).
Gid	Posix GID uživatele (pokud se neuvede, přiřadí se volné v pořadí).
Groups	Seznam skupin, kterých je uživatel členem. Souvisí s parametry: append a sudo. Pokud není definováno, členství ve skupinách není řízeno (zůstává bez změny). Prázdný list [] spolu s append: false a sudo: false ruší členství uživatele ve

	všech skupinách. S parametrem apend: true provede přidání uživatele do skupin (se zachováním aktuálního členství).
Append	Ovlivňuje chování parametru groups. V případě true nedochází k odebírání členství ve skupinách, které nejsou uvedené. V případě false je členství ve skupinách striktně upraveno dle parametru groups a sudo.
Sudo	Přidává do seznamu skupin v parametru groups skupinu pro sudo dle ansible_os_family. Zdrojem tohoto seznamu je proměnná localusers_sudo_group_by_os_family definovaná jako role variables (./vars/main.yml). Např. pro os family RedHat je seznam skupin obohacen o wheel, v případě Debian sudo. true/false (default)
Shell	Nastavuje posix parametr shell.
Comment	Nastavuje posix parametr comment neboli gecos.
Home	Nastavuje posix parametr home (cesta k domovskému adresáři).
Create_home	true (default) / false – zakládá/nezakládá uživatelský home adresář při vytvoření uživatele.
Password	Hodnotou může být heslo ve formě kryptovaného řetězce**.**
Update_password	always – aktualizace hesla probíhá pokaždé, pokud se liší. on_create – heslo je nastaveno pouze při vytvoření uživatele.
Password_lock	true – nastavuje blokaci možnosti použít heslo k autentizaci. false – nastavuje odblokování uživatelského hesla.
Ssh_keys	Spravuje public ssh klíče v souboru ~/.ssh/authorized_keys. Identifikátorem klíče je samotný klíč specifikovaný v parametru key jako string, nebo v parametru file jako název souboru umístěného v adresáři ./files/. Pomocí parametru env lze specifikovat použití klíče pouze tam, kde env odpovídá proměnné localusers_environment (např. test/prod/). List of dictionary s klíči key/file, env (volitelně) a state (default: present)

```
---

- name: Manage users
  vars:
    user_nameid: "{{ item_user if item_user is string else item_user.name | mandatory }}"
  include_tasks: "user_tasks.yml"
  loop: "{{ localusers_realized }}"
  loop_control:
    loop_var: item_user
```

Obrázek 23: Create_user/tasks/main.yml

- name: Realize user '{{ user_nameid }}'

  vars:

    groups_defined: "{{ item_user.groups is defined or localusers_userdb[user_nameid].groups is defined or (item_user.sudo|default(localusers_userdb[user_nameid].sudo)|default(false)|bool) }}"

    groups_sudo: "{{ [localusers_sudo_group_by_os_family[ansible_os_family|lower]] if item_user.sudo|default(localusers_userdb[user_nameid].sudo)|default(false)|bool else []}}"

  ansible.builtin.user:

    name: "{{ localusers_userdb[user_nameid].name | default(user_nameid) }}"

    state: "{{ item_user.state | default(localusers_userdb[user_nameid].state) | default('present') }}"

    append: "{{ item_user.append | default(localusers_userdb[user_nameid].append) | default(omit) }}"

    comment: "{{ item_user.comment | default(localusers_userdb[user_nameid].comment) | default(omit) }}"

    create_home: "{{ item_user.create_home | default(localusers_userdb[user_nameid].create_home) | default(omit) }}"

    expires: "{{ item_user.expires | default(localusers_userdb[user_nameid].expires) | default(omit) }}"

    groups: "{{ item_user.groups | default(localusers_userdb[user_nameid].groups) | default([]) + groups_sudo if groups_defined else omit }}"

    home: "{{ item_user.home | default(localusers_userdb[user_nameid].home) | default(omit) }}"

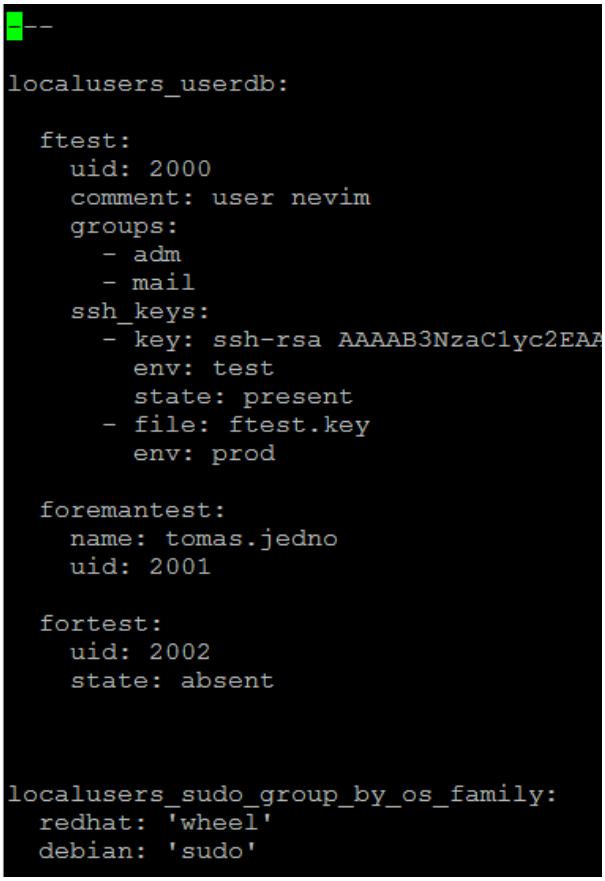
    password: "{{ item_user.password | default(localusers_userdb[user_nameid].password) | default(omit) }}"

```
        password_lock: "{{ item_user.password_lock |
default(localusers_userdb[user_nameid].password_lock) | default(omit) }}"
        shell: "{{ item_user.shell | default(localusers_userdb[user_nameid].shell) |
default(omit) }}"
        uid: "{{ item_user.uid | default(localusers_userdb[user_nameid].uid) |
default(omit) }}"
        update_password: "{{ item_user.update_password |
default(localusers_userdb[user_nameid].update_password) | default(omit) }}"
    - name: Manage Authorized Keys for user '{{ user_nameid }}'
      ansible.posix.authorized_key:
        user: "{{ localusers_userdb[user_nameid].name | default(user_nameid) }}"
        key: "{{ item.key|default(lookup('file','files/'+item.file) if item.file is defined else
'') }}"
        state: "{{ item.state|default('present') }}"
      loop: "{{
item_user.ssh_keys|default(localusers_userdb[user_nameid].ssh_keys)|default([])|rejectattr
('env','defined')|list +
item_user.ssh_keys|default(localusers_userdb[user_nameid].ssh_keys)|default([])|selectattr
('env','defined')|selectattr('env','equalto',localusers_environment)|list }}"
    when: not item_user.state | default(localusers_userdb[user_nameid].state) |
default('present') == 'absent'
```

Create_user/tasks/create_user.yml

```
---
localusers_environment: 'test'
localusers_realized: []
```

Obrázek 24: Create_user/defaults/main.yml



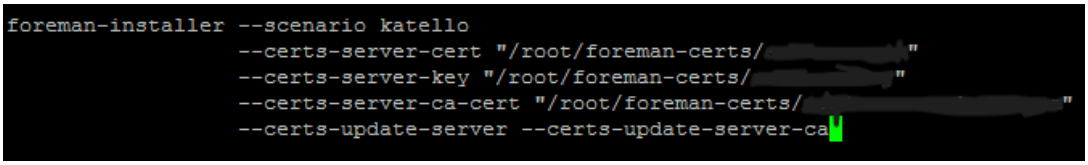
Obrázek 25: Create_user/vars/main.yml

4.10 Přidávání certifikátu

Organizace využívá dva typy certifikátů. Pro webové rozhraní WildCard SSL certifikát (*.doména.cz) vydaný certifikační autoritou. Druhý typ vydává vlastní certifikační autorita. Organizace chtěla, aby byl prvotně přidán certifikát pro webové rozhraní. Případně později, by byly vytvořeny certifikáty vlastní autoritou pro komunikaci mezi proxy servery a spravovanými servery. Nejdříve byl implementován hvězdičkový certifikát s tím, že by proxy servery měly být schopné tento certifikát přijmout a nemusely by se prozatím přidávat i na ně. Tato myšlenka byla správná. Bohužel pouze do chvíle, než byla provedena aktualizace systému Foreman a proxy serverů. V tu chvíli došlo k nekonzistenci. Na základě jiných certifikátů se nedokázal synchronizovat obsah na smart proxy. Z tohoto důvodu musel být nakonec vrácen původní stav, tak aby vše fungovalo. Aktuálně se debatuje nad dalším postupem, protože tato chyba ukázala na problém, kdy managment server a zároveň smart proxy nedokáží mít certifikáty od jiných certifikačních autorit, což celkově rozbilo koncept, kterým řešení mělo být uděláno. V případě, že by se certifikáty měnili, muselo by tak být na všech pěti serverech najednou. Vzhledem k časové náročnosti a byrokracii, která kolem

kroků s vytvářením certifikátů je, není konečné řešení v práci zaznamenáno. Je velmi důležité do budoucna hlídat platnost certifikátů. Ve chvíli, kdy vyprší jakýkoliv certifikát se rozbije celá struktura a nemusí se jít do Foreman vůbec přihlásit, Přidání certifikátů však případně bude probíhat stejně, jako při prvotním pokusu, který je popsán níže.

1. Prvotně je nutné mít všechny certifikáty struktury plus klíč k nim (v našem případě dva). Klíč bylo potřeba nejdříve extrahovat pomocí příkazu openssl pkcs12 -in xxx.pfx -out xxx.key -nocerts
2. Následně bylo nutné rozšifrovat vytvořený klíč openssl rsa -in xxx.key -out xxx.key-decrypt.
3. Spojení všech certifikátů, které jsou ve struktuře. K nim je dobré přidat ještě aktuální certifikát, pro zachování funkčnosti (/var/www/html/pub/katello-server-ca.crt)
4. Ověření, zda jsou certifikáty v pořádku. katello-certs-check -t foreman -c /root/foreman-certs/xxx.pem -k /root/foreman-certs/xxx.cz.key -b /root/foreman-certs/xxx.cz.ca_chain.pem
5. V případě, že ověření proběhlo v pořádku vygenerují se automaticky příkazy, který mi se certifikát buď prvotně instaluje nebo aktualizuje. Tyto příkazy se následně pustí.



Obrázek 26: Instalace a aktivace certifikátů

6. Po instalaci je nutné na všechny smart proxy nahrát nový chain balíček, který je na managment serveru. Jinak spolu nedokážou servery komunikovat. rpm -Uvh http://název_managment_serveru/pub/katello-ca-consumer-latest.noarch.rpm

4.11 Připojení Gitlab a správa rolí

Foreman dokáže komunikovat s Gitlabem. Pomocí něho je spravován obsah Ansible rolí. Výhodou je možnost verzování a zároveň kontroly, kdo jaké úpravy na roli udělal. Role je vždy upravována na jedné ze smart proxy pod vlastním účtem a následně je pomocí příkazu push aktualizována na gitlab. Odtamtud pomocí playbooku je synchronizována na managment server a zbylé proxy servery. Propojení je důležité z pohledu budoucího využívání a celkového přístupu ke konfiguračnímu managmentu.

Playbook má tři proměnné, které jsou upravovány v rámci tamplate imputs v rozhraní Foreman. Jedná se o proměnné –

1. location – charakterizuje složku do které může zapisovat a přistupovat uživatel foreman-proxy.
2. git_repository_port – charakterizuje port přes, který gitlab komunikuje
3. git_repository – který charakterizuje ssh cestu na gitlab

Dlouho byl problém se spouštěním playbooku. Nakonec bylo zjištěno, že byl problém v ssh klíči. Proto musel být do autorizovaných klíčů na všech smart proxy přidán klíč managment serveru (/home/foreman/.ssh/authorized_keys).

Po spuštění ansible playbook se nejdříve na vybrané proxy vytvoří složka definovaná v imputs (/opt/foreman-ansible). Následně se do této složky zkopírují veškerá data z Gitlabu. Dá výpis, že vše proběhlo v pořádku a následně přesune veškerý obsah ve vytvořené složce do /etc/ansible/roles. V tu chvíli jsou aktualizovány role, pro vybranou smart proxy.

Playbook pro distribuci aktuální verze rolí na smart proxy:

```
---
- hosts: all
  gather_facts: false
  become: true
  tasks:
    - name: Manage source directory
      ansible.builtin.file:
        path: "<%= input('location') %>"
        state: directory
        owner: foreman-proxy
        group: foreman-proxy
        mode: "0755"
    - name: Sync git
      vars:
        ansible_become_user: foreman-proxy
      ansible.builtin.git:
        repo: "<%= input('git_repository') %>"
        dest: "<%= input('location') %>"
```

```
            ssh_opts: "-i /var/lib/foreman-proxy/ssh/id_rsa_foreman_proxy -o
IdentitiesOnly=yes -o StrictHostKeyChecking=accept-new -F /dev/null -p <%=
input('git_repository_port') %>"
        #       key_file: "/var/lib/foreman-proxy/ssh/id_rsa_foreman_proxy"
              force: true
            register: out
          - name: Show git output
            ansible.builtin.debug: var=out
          - name: Manage symlink to ansible
            ansible.builtin.file:
              path: /etc/ansible/roles
              src: "<%= input('location') %>/roles"
              state: link
              force: true
```

5 Výsledky a diskuse

Jako první bylo nutné provést analýzu prostředí, zdrojů dat a nástrojů, které lze využít společně s Ansible.

Z analýzy prostředí se potvrdilo, že má organizace velmi různorodé prostředí. Nejednotné je nejen z pohledu operačních systémů na serverech, verzí těchto operačních systémů, tak i v různorodosti repozitářů na serverech, které by měly být totožné. Z tohoto důvodu bylo nejdříve nutné servery postupně upravovat, tak aby byl nejen přehled o tom, co kde je, ale zároveň připravit vše pro následný jednotný pohled na hromadnou správu pomocí automatizace. Tyto kroky byly dělány mimo záběr diplomové práce. Jednalo se hlavně o kroky týkající se skenování serverů, pro zjištění unikátních repozitářů napříč všemi servery, zjišťování IP adres, upravování nastavení v případě nalezených problémů a tvorba jednotného uživatele pro automatizaci. Z této analýzy též byly zjištěny zdroje dat (konfigurace linuxových serverů). Organizace má celkově okolo 450 serverů. Z nichž bylo pro tuto práci využito přibližně 250, které splňovaly vstupní podmínky pro operační systém v daných verzích. Podmínkami byly operační systémy Red Hat Enterprise Linux 7 a novější a Ubuntu 20.04 a novější. Zbylé historické servery a Windows servery nebyly do projektu zatím zapojeny.

Na základě vstupní analýzy byly vybrány tři procesy. Těmi jsou správa repozitářů, serverů a administrátorských uživatelů. Tyto procesy byly vybrány na základě dohody s vedením a zároveň po poradě oddělení IT. Bylo konstatováno, že vybrané procesy jsou časově náročné vzhledem k množství serverů, repozitářů a administrátorských uživatelů, které je nutné spravovat a současně neexistuje jednotný nástroj pomocí, kterého by tyto procesy šlo dělat jednoduše a centralizovaně.

Následovala analýza dostupných nástrojů, které lze využít společně s nástrojem Ansible pro automatizaci právě vybraných procesů. Na výběr byly čtyři nástroje. Ansible Automation Platform, Oracle Automation, Microsoft System Center a Foreman. Na základě debaty, srovnání výhod a nevýhod bylo rozhodnuto, že bude využit open-source nástroj Foreman, neboť v sobě integruje Ansible. Tento nástroj byl zvolen nejen vzhledem k jeho cenové výhodnosti, ale také kvůli dřívějším zkušenostem, které organizace měla. V průběhu testovaní se ukázalo, že Foreman disponuje většinou funkcí, jako ostatní placené nástroje. Samozřejmě každý nástroj má nějaké malé funkce navíc, ale dosud nebylo zjištěno, že by některá z potřebných funkcí v rámci Foreman chyběla. Zdánlivou nevýhodou, která

u nástroje byla zatím zjištěna je pouze komunitní podpora. Vzhledem k tomu, že se jedná o hodně využívaný nástroj, tak i toto není zas takový problém, jak se někdy uvádí. V případě problémů byla většinou komunikace a zároveň samotné řešení problémů velmi pohodlné a rychlé.

Před samotnou instalací bylo prostředí rozděleno na čtyři části. Vzhledem k tomu, že má organizace dvě hlavní datová centra a v každém z nich testovací a produkční prostředí. Rozdělení tedy bylo uděláno podle datového centra a příslušného prostředí (DCA/TEST, DCB/TEST, DCA/PROD, DCB/PROD). Pro každé jedno ze čtyř prostředí byl instalován proxy server s“mart proxy“, která slouží pro distribuci obsahu na spravované servery. Nad těmito smart proxy je Foreman server neboli také managment server. Odtud se distribuuje veškerý obsah na smart proxy a zároveň se odtud spouští veškeré operace. Také jsou zde uloženy veškeré údaje o serverech. Slouží i jako hlavní server pro správu konfiguračního managmentu.

Jako první po instalaci a nastavení Foreman serveru a smart proxy se autor zaměřil na správu repozitářů. Neboť jakmile bude server zaregistrován, tak server nadále nebude mít přístup k repozitářům a do internetu. Tudíž ve chvíli, kdy by nebyly připraveny repozitáře a k nim další komponenty, tak by servery nemohly aktualizovat operační systém a aplikace. U tohoto bodu autor vycházel z analýzy prostředí. Nejdříve bylo nutné nahrát veškeré unikátní repozitáře, které byly rozděleny do produktů (Postresql. Docker, Node atd.). V každém z produktů je několik jednotlivých repozitářů podle typu operačního systému a jeho verze. Následovala tvorba CV znovu na základě analýzy prostředí, tak aby bylo zajištěno, že všechny servery měly po registraci potřebné zdroje. CV byly využity dvojího typu. Kompozitní a klasické. U většiny serverů je využíváno kompozitního. Je tomu, tak z důvodu, že byl požadavek na možnost zvlášť aktualizace operačního sytému a dalších speciálních repozitářů. Následovala tvorba aktivačních klíčů, které jsou určeny pro každý server při registraci. Pro skupinu serverů se stejnými parametry je vytvořen vždy pouze jeden aktivační klíč. Na základě něho probíhá prvotní přiřazení, povolení repozitářů a CV. Je zároveň využíván v Ansible roli pro registraci nových systémů.

Pro lepší správu Ansible rolí byl k Foreman připojen Gitlab, který organizace využívá. Zároveň byl vytvořen Ansible playbook, který zajišťuje synchronizaci veškerých rolí na Foreman server a všechny smart proxy. Veškeré Ansible role jsou tedy uloženy na jednom místě a do Foreman se pomocí playbook synchronizují.

Následně se práce zabývala samotnou správou serverů. Na základě analýzy prostředí bylo nutné jako nejdůležitější predispozici vytvořit skupiny hostů, ty slouží, jako základ pro konfigurační managment. Zároveň na nejnižší vrstvě skupiny hostů by měly být dány servery, které mají úplně stejnou konfiguraci (stejné uživatele, stejné repozitáře a další nastavení). Tento krok byl hodně zdlouhavý z důvodu velké rozmanitosti infrastruktury organizace. Jen málo serverů má úplně stejnou konfiguraci. Po vytvoření skupin hostů bylo možné začít přidávat servery. Nejdříve bylo ručně přidáno několik testovacích serverů. Následně byly hromadně vytvořeny záznamy o serverech, které postupně byly následně registrovány do Foreman. Proto, aby celý proces registrace byl co nejvíce pohodlný, vznikla Ansible role, která automaticky dělá jednotlivé kroky registrace. Rozlišuje také o jaký operační systém se jedná, zda-li Ubuntu nebo Red Hat Enterprise Linux. Z důvodu, že oba operační systémy mají jiné kroky registrace. Třetí možností zakládání serverů je vytvoření serveru ve VMware. Pro tento účel vznikla minimální instalace systému Red Hat Enterprise Linux 9.3 a z této image se následně skrze konfiguraci ve Foreman instaluje a konfiguruje celý server. Po registraci serveru je možné libovolně skrze Foreman spouštět jakékoliv příkazy, ansible playbook, či ansible role. Zároveň lze veškeré činnosti dělat na více serverech najednou.

Posledním procesem, který autor práce implementoval je správa administrátorských uživatelů operačního systému Linux. Ta v organizaci dříve byla řešena vždy pouze v rámci daného serveru. Nebyla zároveň žádná dokumentace, kdo na kterém serveru má účet. K vytváření uživatelů existovala jednoduchá role a playbook, ve kterém administrátor vždy musel zadat jednoho uživatele, kterého chce založit. Pokud tedy vznikal nový server, tak přidávání všech potřebných uživatelů mohlo být zdlouhavé. Proto vznikla ansible role, díky které nyní lze přidávat a zároveň evidovat uživatele na všech serverech. Pro každého uživatele lze upravovat 15 různých parametrů. Vše je upravováno na úrovni skupin hostů, kde existuje parametr, pomocí kterého lze definovat, kteří uživatelé na dané skupině budou. Na každé skupině může mít jeden uživatel jiné nastavení podle potřeby a zadaných parametrů. Následně po aplikaci role na skupinu hostů, proběhne synchronizace, která na základě konfigurace v roli uživatele na serverech založí, zakáže nebo aktualizuje. Proces synchronizace probíhá automaticky každou noc.

V tuto chvíli v rámci organizace skončila první část projektu automatizace. Autor práce se v částech diplomové práce zaměřil na postup jednotlivých kroků v rámci celé implementace řešení. Zároveň považuje vybraný nástroj za vhodný. Na druhou stranu

v rámci celé práce pozoroval, že nástroj Foreman je velmi komplexní a má mnoho funkcionalit. Může proto být v některých ohledech náročný i pro samotného správce. Zároveň však nabízí velké množství možností, kam lze dále v rámci organizace tento projekt posouvat. Autor během své práce musel řešit několik problémů. Jedním z nich byl problém s certifikátem ve webovém rozhraní. Bylo zjištěno, že správa certifikátů v rámci celé architektury Foreman je celkem obtížná. Vzhledem k tomu, že je nutné pravidelně měnit certifikáty, jak na Foreman serveru, tak i všech čtyřech smart proxy. Další problémy byly zjištěny většinou po aktualizaci celého systému. Zároveň však většinou dané problémy byly vyřešeny následnou další aktualizací. Proto je dle autora důležité pravidelně aktualizovat celý systém. Mezi dalšími kroky v tomto projektu autor očekává, postupnou implementaci dalších serverů, a to hlavně z produkčního prostředí. To mělo být sice náplní již první části projektu, ale vzhledem k velké rozmanitosti infrastruktury a zároveň většímu počtu dodavatelů se kterými organizace spolupracuje nebylo zatím možné produkční prostředí plně implementovat.

V rámci diplomové práce se autor zaměřoval hlavně na správu linuxových serverů. Ať už z hlediska správy administrátorských účtů, správy repozitářů, tak správy samotných serverů. V tomto ohledu autor očekává, že v rámci druhé části projektu automatizace se organizace zaměří na rozšiřování v rámci konfiguračního managmentu. Ten je sice již využíván v rámci této diplomové práce, ale má potenciál i v dalších oblastech správy serverů. V tuto chvíli, jak již bylo psáno výše, je možné pomocí Foreman založit server ve VMware. Při současném testování bylo zjištěno, že čtyři servery pro Postgresql, které mají stejnou konfiguraci, lze nainstalovat v rámci několika minut. Jde však samozřejmě o to, jakým způsobem jsou připraveny veškeré predispozice, které jsou u vytváření serveru ve Foreman potřeba udělat. Občas v tuto chvíli ještě může být rychlejší server založit ručně ve VMware. Tento postup by však v budoucnu neměl být vhodný vzhledem k automatizačním krokům v rámci Foreman. Další konfigurace však zatím probíhá ručně. V další části projektu lze tedy očekávat, že se organizace zaměří na plnou konfiguraci v rámci Foreman, tedy na všechny kroky od vytvoření serveru až po kompletní nakonfigurování a založení všech záznamů v dalších přidružených systémech (Vytvoření požadavku, Rezervace volné IP adresy, kontrola DNS záznamu). Autor také očekává, že v budoucnu v rámci dalších fází projektu bude automatizováno více IT procesů, a to i z jiných oblastí než jen správy serverů. Nástroj například umožňuje správu infrastruktury. Kdy organizace v rámci dalšího projektu implementuje nástroj NetBox, který slouží pro kompletní dokumentaci infrastruktury.

Foreman má možnost modulu, který dokáže s NetBox spolupracovat. Autor se tedy domnívá, že by v budoucnu mohlo být užitečné, kdyby tyto dva důležité nástroje společně dokázaly komunikovat. Zároveň lze také očekávat, že bude nástroj rozšířen i o Windows servery, kterých má organizace též hodně.

.

6 Závěr

Cílem práce byla automatizace zvolených procesů ve vybrané instituci na základě analýzy prostředí s využitím nástroje Ansible v souladu s ITIL. Spolu s analýzou stávajícího prostředí a procesů a identifikace oblasti, které bylo možné automatizovat. Na jejím základě pak navrhnout řešení pro automatizaci správy operačních systémů pomocí nástroje Ansible v souladu s procesy ITIL a implementovat navržené řešení.

Cílů bylo dosaženo v rámci první etapy projektu automatizace ve vybrané organizaci. Nejdříve proběhla analýza prostředí a procesů. Z analýzy prostředí bylo potvrzeno, že má organizace velmi různorodé prostředí. Pro samotnou automatizaci bylo ze 450 serverů vybráno 250, které splňovaly podmínky zvolené podmínky. Dále bylo zjištěno, že organizace v rámci IT má 23 hlavních procesů a subprocesů. Z nich byly tři procesy vybrány pro následnou automatizaci. Jednalo se o správu repozitářů, správu serverů a správu administrátorských uživatelů operačního systému Linux. Na základě analýzy vhodných nástrojů, které využívají Ansible. Následovala volba vhodného nástroje. Z této analýzy vyšel nejlépe nástroj Foreman, který se stal důležitou součástí v rámci implementace vlastní části práce. Po instalaci Foreman serveru a smart proxy následovala samotná automatizace tří vybraných IT procesů, kterými se IT oddělení v organizaci zabývá. V rámci správy repozitářů autor vycházel z analýzy prostředí a zdrojů dat, kde byly do Foreman nahrané veškeré potřebné repozitáře, které se nacházely na serverech vybraných pro projekt. V rámci správy serverů byla vytvořena stromová struktura host group, na jejímž základě lze efektivně využívat tento nástroj. Následně bylo možné přidávat a registrovat jednotlivé servery. Registrace probíhala pomocí vytvořené Ansible role. Posledním automatizovaným procesem byla správa administrátorských uživatelů operačního systému Linux. Zde vznikla komplexní Ansible role, ve které lze nastavit 15 různých parametrů, které administrátor může mít. Výběr uživatelů na jednotlivých systémech je určen na úrovni host group. Dále byla ověřena funkčnost celého naimplementovaného řešení v praxi. Tím byla potvrzena funkčnost celkového návrhu řešení. Nyní lze automatizovaně spravovat repozitáře pro veškeré vybrané servery infrastruktury. Zároveň lze automatizovaně spravovat veškeré přidané servery (například aktualizace systémů, spouštění jakýchkoliv operací). A také lze automatizovaně spravovat administrátorské uživatele operačního systému Linux. Všechny činnosti, které byly v rámci vlastní práce realizovány se řídily podle metodiky ITIL, která

byla popsána v teoretické části práce. Veškerá teoretická východiska byla ve vlastní části využita.

Ve vybrané instituci je již naplánována druhá etapa projektu, která navazuje na dokončenou první etapu a bude se ještě více věnovat samotnému konfiguračnímu managmentu. V rámci této etapy bude též řešení plně implementováno do produkčního prostředí. Implementace do celého produkčního prostředí nemohla být zatím provedena s ohledem na velkou rozmanitost infrastruktury a zároveň většímu počtu dodavatelů, se kterými organizace spolupracuje. Také lze do budoucna počítat s propojováním Foreman s dalšími systémy organizace.

7 Bibliografie

Alvao. 2023. Alvao. *Alvao.* [Online] 2023. [Citace: 20. Červenec 2023.] https://www.alvao.com/.

Balátě, Jaroslav. 2003. *Automatické řízení.* Praha : BEN - technická literatura, 2003. ISBN 80-7300-020-2.

Comidor. 2023. Comidor. *Comidor.* [Online] 1. Únor 2023. [Citace: 10. Srpen 2023.] https://www.comidor.com/news/industry-news/it-automation-trends/.

Danel, Roman. 2011. *Metodiky řízení IS/ICT.* Ostrava, Česká republika : Vysoká škola báňská - Technická univerzita Ostrava, 2011.

Heap, Michael. 2016. *Ansible - From Beginner to Pro.* Reading : Apress, 2016. ISBN 978-1-4842-1660-6.

ILX Marketing Team. 2018. ITIL Training. *ITIL Training.* [Online] 6. Listopad 2018. [Citace: 15. Červenec 2023.] https://www.itiltraining.com/eur/blog/itil-history.

ITIL® Foundation. 2019. *ITIL 4 edition.* Norwich : TSO, 2019. ISBN 9780113316076.

Kissflow. 2023. Kissflow. *Kissflow.* [Online] 31. Leden 2023. [Citace: 25. Červenec 2023.] https://kissflow.com/workflow/bpm/business-process-automation/it-process-automation/.

Knowledgehut. 2023. Knowledgehut. *Knowledgehut.* [Online] 2023. [Citace: 14. Červenec 2023.] https://www.knowledgehut.com/tutorials/itil4-tutorial/itil-four-dimensions-it-service-management.

Kolektiv autorů. 2012. *Automatizace a automatizační technika 1.* Brno : Computer Press, 2012. ISBN 978-80-251-3628-7.

Manageengine. 2023. Manageengine. *Manageengine.* [Online] 2023. [Citace: 20. Červenec 2023.] https://www.manageengine.com/products/service-desk/itil-release-management/.

McKendrick, Russ. 2018. *Learn Ansible.* Birmingham : Packt Publishing, 2018. ISBN 978-1-78899-875-8.

Procházka, Jaroslav a Klimeš, Cyril. 2011. *Provozujte IT jinak: agilní a štíhlý provoz, podpora a údržba informačních systémů a IT služeb.* Praha : Grada, 2011. ISBN 978-80-247-4137-6.

RVX. 2023. ISO-certifikace. *ISO-certifikace.* [Online] 22. Únor 2023. [Citace: 25. Červenec 2023.] https://iso-certifikace.cz/.

Servicenow. 2023. Servicenow. *Servicenow.* [Online] 2023. [Citace: 20. Červenec 2023.] https://www.servicenow.com/.

Sesto, Vincent. 2021. *Practical Ansible: Configuration Management from Start to Finish.* Auckland : Apress, 2021. ISBN 978-1-4842-6484-3.

Shah, Gourav. 2015. *Ansible Playbook Essentials.* Birmingham : Packt Publishing, 2015. ISBN 978-1-78439-829-3.

Şimşek, Hazal. 2023. AI Multiple. *AI Multiple.* [Online] 8. Srpen 2023. [Citace: 10. Srpen 2023.] https://research.aimultiple.com/it-automation-trends/.

Svobodová, Kristýna. 2023. Safetica. *Safetica.* [Online] 14. Únor 2023. [Citace: 25. Červenec 2023.] https://www.safetica.com/cs/blog/iso/iec-27001.

8 Seznam obrázků, tabulek, grafů a zkratek

8.1 Seznam obrázků

8.2 Seznam tabulek

8.3 Seznam použitých zkratek

CV – Content View

IT – Informační technologie

ITIL - Information Technology Infrastructure Library