

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra telekomunikační techniky



Bakalářská práce

Automatizace konfigurace síťových prvků pomocí Ansible

Miroslav Hudec

Studijní program: Komunikace, multimédia a elektronika

Studijní obor: Síťové a informační technologie

Vedoucí práce: Miloš Kozák

Praha, Březen 2016

Obsah

1	Úvod	6
1.1	Správa moderních sítí	6
1.1.1	Centralizovaná správa	7
1.1.2	Automatizace počítačových sítí	8
1.2	Současný stav	8
1.3	Zaměření a cíl práce	8
2	Analýza	9
2.1	Základní pojmy	9
2.1.1	Charakteristiky počítačové sítě	9
2.1.2	Základní síťové prvky	9
2.2	Základní požadavky na konfiguraci	11
2.3	Možnosti konfigurace síťových zařízení	11
2.4	Ansible	11
2.5	MikroTik	12
2.5.1	Možnosti konfigurace zařízení MikroTik	12
2.6	Shrnutí a vyhodnocení poznatků	14
3	Realizace	14
3.1	MikroTik API	14
3.2	Základní funkce	14
3.3	Moduly Ansible	14
3.4	Ansible Playbook	14
3.5	Příklady použití	14

Seznam použitých zkratk

BGP Border Gateway Protocol [11](#)

CLI Command Line Interface (*Příkazová řádka*) [6](#), [13](#)

DRP Disaster Recovery Plan [7](#)

EIGRP Enhanced Interior Gateway Routing Protocol [11](#)

IP Internet Protocol [10](#)

ISP Internet Service Provider (*Poskytovatel internetu*) [6](#)

ITIL Information Technology Infrastructure Library [7](#)

OSPF Open Shortest Path First [11](#)

RIP Router Information Protocol [11](#)

SNMP Simple Network Management Protocol [7](#), [8](#)

SPOF Single Point of Failure [9](#)

TCP Transmission Control Protocol [14](#)

VLAN Virtual Local Area Network [10](#)

YAML Yet Another Markup Language [12](#)

Seznam pojmů

RM ISO/OSI

Referenční model ISO/OSI je standardizovaný model abstrahující úkony telekomunikačního systému do 7 vrstev. Tato abstrakce umožňuje lépe vytvářet standardy a popisovat všeobecné principy síťové architektury. [8] [9](#)

Zadání projektu

Využijte nástroj Ansible pro automatickou správu síťových prvků. Zaměřte se na síťové prvky od společnosti Mikrotik a rozšiřte nástroj Ansible tak, aby bylo možné konfigurovat základní i pokročilé funkce těchto síťových prvků. Mezi základní funkce patří správa uživatelských účtů, logování a adresace síťových rozhraní. Mezi pokročilé funkce patří správa firewallu, překladu adres a síťových mostů. Navrhněte případy užití, na kterých ukážete funkci implementovaného rozšíření.

Motivace

V důsledku stálého rozvoje dnešních počítačových sítí a zvyšování nároků na ně kladených, je často potřeba zavádět do produkčního provozu větší množství nových a výkonnějších zařízení. V současnosti bývají tato zařízení konfigurována jedno po druhém, což prodlužuje dobu nutnou pro nastavení a tím samozřejmě zvyšuje náklady na implementaci. Cílem této práce je zefektivnění tohoto postupu navržením řešení, které umožní automatizovanou konfiguraci mnoha zařízení dle zadaných požadavků. Zařízení by tak bylo možné nakonfigurovat během velmi krátké doby a s minimálním úsilím. V rámci této práce se budu věnovat konfiguraci zařízení Mikrotik.

Abstrakt

1 Úvod

Rozvoj datových sítí v současné době probíhá velmi rychlým tempem. Poptávka po službách přenosu dat stále narůstá a datové sítě navíc přebírají role mnoha dalších sítí, jako jsou například hlasové či televizní sítě. Zvyšující se požadavky na propustnost a spolehlivost datových sítí spolu s rostoucím počtem internetových přípojek ústí v potřebu nasazovat velké počty síťových zařízení především do přístupové části sítě. Nasazování většího počtu síťových zařízení s sebou však přináší mnoho výzev. Nová zařízení je potřeba správně nastavit tak, aby odpovídala požadavkům dané sítě a umístění konkrétního prvku v topologii sítě. Dále je pak potřeba tato zařízení dohledovat, aby bylo možné odhalit případné poruchy či abnormality v síťovém provozu. V neposlední řadě je třeba umožnit vzdálenou správu zařízení tak, aby bylo možné provádět změny v konfiguraci každého zařízení v návaznosti na aktuální požadavky bez nutnosti fyzického přístupu k zařízení.

Ať už se jedná o nasazování aktivních prvků v síti [Internet Service Provider \(Poskytovatel internetu\) \(ISP\)](#) či v rozsáhlejší podnikové síti, vždy platí, že čas jsou peníze. A ekonomické hledisko bývá často až na prvním místě. Je proto velmi důležité celý proces implementace nových zařízení do provozu co nejvíce zefektivnit, a to jak po stránce potřebného času, tak nároků na lidské zdroje. V ideálním případě by tak měl celý proces implementace obstarat jeden technik v co nejkratším čase a samozřejmě bezchybně. K naplnění takových požadavků je však potřeba nalézt systém pro usnadnění a automatizaci celého procesu.

1.1 Správa moderních sítí

Korektně definovaná a aplikovaná správa počítačových sítí představuje fundamentální předpoklad pro jejich hladký a bezchybný provoz. Navzdory svojí důležitosti a nárokům kladeným na dnešní sítě však často administrátoři těchto sítí spoléhají na poměrně základní a přímočaré postupy: Změny v konfiguracích jednotlivých síťových prvků jsou převážně prováděny ručně pomocí zastaralých low-level nástrojů, jako je [Command Line Interface \(Příkazová řádka\) \(CLI\)](#). Takovýto přístup, který vyžaduje připojení se zvláště ke každému zařízení je však velmi časově náročný. Navíc ve světě, kde jsou sítě stále rozlehlejší, komplexnější a dynamičtější roste i riziko lidské chyby. Takováto chyba v konfiguraci byť jediného zařízení přitom může mít fatální dopady na provoz celé sítě.

Prvotní pokusy o lepší zvládnutí obsluhy síťových prvků vzešly právě od síťových administrátorů, kteří sepisovali a spouštěli nejrůznější skripty. Tyto skripty se však postupem času stávaly stále sofistikovanější, přičemž každý administrátor udržoval svůj vlastní "arsenál". Takovéto skripty mají však bohužel velmi specifické zaměření a vyžadují vysokou úroveň znalostí pro provedení

složitějších úkonů a zamezení nechtěným vedlejším účinkům. Sestavení takového skriptu navíc vyžaduje značnou odbornost a dlouhé testování metodou pokus-omyl.

Dalším nezbytným předpokladem pro provoz počítačové sítě je správa dokumentace, především udržování záznamů o stavu jednotlivých prvků a jejich konfigurací. Dojde-li například k poruše zařízení, je třeba zajistit, aby po opravě poruchy bylo možné uvést zařízení do původního stavu v co nejkratším čase. Za tímto účelem bylo vyvinuto mnoho softwarových řešení, které umožňují automatické zálohování konfigurací a jejich následnou obnovu v případě poruchy. Pro ochranu podnikové IT infrastruktury by také měl existovat [Disaster Recovery Plan \(DRP\)](#), který přesně definuje procesní postupy pro případ havárie.

1.1.1 Centralizovaná správa

Aby bylo možné naplnit veškeré požadavky na provoz sítě, je nutné celou její správu centralizovat. Jsou-li veškeré údaje o stavu sítě dostupné z jednoho kontrolního místa, je mnohem snazší dohlížet na provoz, odhalit případné poruchy a provádět změny v konfiguraci sítě s ohledem na všechny důsledky. Je velmi důležité, aby každá změna byla provedena s ohledem na všechny návaznosti a následně řádně zdokumentována. Provádění změn v síti tak říkájíc ad hoc, například na základě požadavků uživatelů bez jejich řádných zavedení do dokumentace sítě může vést k nestandardnímu chování celé sítě. Toto pak zvláště platí v případě, že o síť se stará více administrátorů. Aby se zamezilo takovýmto možným nedopatřením, měla by být v rámci správy sítě zavedena a striktně dodržována metodika procesního řízení. Jednou z takovýchto standardizovaných metodik je například [Information Technology Infrastructure Library \(ITIL\)](#). Jedná se o sadu praktik pro správu IT služeb, které se zaměřují na sladění IT s potřebami podniku [6]. Takováto metodika by poté měla být součástí širší podnikové politiky.

Monitoring počítačové sítě: Monitoringem počítačové sítě se rozumí nepřetržité sledování stavu zařízení v síti, které umožní odhalit selhávající komponenty sítě a upozornit na tuto skutečnost administrátora sítě. Lze tak jednak předejít možným výpadkům v provozu či určit kde k poruše došlo a značně tak zkrátit čas potřebný na její odstranění. Jedním z jednodušších přístupů k této problematice je centrální shromažďování logů z jednotlivých zařízení. Každé zařízení většinou ukládá informace o svém stavu a provedených úkonech do souboru označovaného jako log. Tyto logy lze pak shromažďovat na jednom centrálním místě, kterým bývá log server. Logy na tomto serveru jsou poté automaticky vyhodnoceny a na základě jejich závažnosti je provedena předdefinovaná akce, jako například upozornění administrátora prostřednictvím e-mailu či SMS.

Další možností pro získávání informací o stavu zařízení je implementace [Simple Network Management Protocol \(SNMP\)](#). Jedná se o protokol navržený pro shromažďování a organizaci

informací ze spravovaných zařízení. Většina dnešních zařízení, jako jsou routery, switche, servery, koncové stanice či tiskárny podporuje tento protokol, [SNMP](#) je proto hojně využíván v systémech síťové správy a jedná se de facto o standard. Informace ze zařízení jsou zde prezentována pomocí proměnných, které mohou být vyhodnocovány prostřednictvím softwaru, který je spuštěn na [SNMP](#) serveru. Hodnoty jako například aktuální vytížení procesoru či šířka využitého pásma lze následně vykreslit do grafů, což poskytuje administrátorovi dobrý přehled o stavu jednotlivých zařízení. Stejně jako v předchozím případě je i zde žádoucí předdefinovat akce pro klíčové hodnoty, při jejichž dosažení dojde k informování administrátora. Použití SNMP tak administrátorům poskytuje mnohem sofistikovanější a flexibilnější řešení než prosté shromažďování logů. Další výhodou tohoto řešení je také to, že některá zařízení umožňují i změnu svojí konfigurace prostřednictvím [SNMP](#).

Jednou z oblastí monitorování sítě je také sledování samotného síťového provozu. Informace o provozu směrem do a ze sítě administrátorům většinou zprostředkuje router či firewall a o stavu zařízení v síti jsou informováni díky [SNMP](#). Někdy je ovšem vhodné monitorovat i provoz uvnitř lokální sítě, což poskytuje administrátorům mnoho dalších informací za účelem optimalizace sítě či odhalení případných chyb. Monitorování vnitřního provozu navíc zvyšuje bezpečnost, protože díky němu lze odhalit nestandardní síťový provoz, který by mohl představovat potenciální rizika. Za tímto účelem se využívají takzvané síťové sondy, které shromažďují a analyzují veškerý síťový provoz. Získané statistiky pak poskytují informace například o tom, jaký druh provozu ve kterých částech sítě a v jakou dobu prochází. Takových údajů lze využít pro potřebné změny v konfiguraci jednotlivých aktivních prvků pro optimalizaci provozu. Síťové sondy jsou také schopny na základě behaviorální analýzy odhalit odchylky od standardního provozu a upozornit na ně administrátora sítě.

Centrální správa konfigurace:

Udržování dokumentace o konfiguraci na jednom místě, provádění záloh a obnov konfigurací

1.1.2 Automatizace počítačových sítí

Co je to automatizace, Současné trendy, SDN...

1.2 Současný stav

1.3 Zaměření a cíl práce

Co se cílem této práce a co už ne...

2 Analýza

2.1 Základní pojmy

2.1.1 Charakteristiky počítačové sítě

Mezi základní charakteristiky počítačové sítě patří: Topologie, rychlost, cena, bezpečnost, dostupnost, škálovatelnost a spolehlivost.

Pod pojmem topologie si lze představit určitý tvar či strukturu sítě. Zabývá se propojením síťových prvků v rámci sítě a zachycením reálné (fyzické) a logické podoby tohoto propojení. Fyzická topologie popisuje skutečnou konstrukci sítě s jednotlivými uzly a fyzickými zařízeními, včetně popisu kabeláže, která je propojuje. Logická topologie se pak vztahuje k tomu, jak jsou v síti přenášena data mezi jednotlivými uzly. Ačkoliv logická topologie může kopírovat topologii fyzickou (například v domácích sítích), ve větších sítích se tyto topologie liší. Důvodem je zejména zvýšení dostupnosti, spolehlivosti a tím celkové robustnosti sítě. Dojde-li například k poruše zařízení či závadě na kabelu, je možné pozměnit logickou topologii při zachování té fyzické. Provoz bude veden jinou cestou a síť bude stále dostupná a provozuschopná. Změnu logické topologie v případě poruchy jsou pak zařízení schopna provést sama bez vnějšího zásahu, s využitím zvláštních protokolů a ve velmi krátkém čase. S možností poruchy je třeba počítat již při návrhu fyzické topologie a pokud možno zamezit vzniku [Single Point of Failure \(SPOF\)](#), tedy bodu, jehož selhání povede k výpadku provozuschopnosti sítě.

Rychlost sítě udává, jaké množství informace je síť schopna přenést za určitý čas, základní jednotkou rychlosti je bit za sekundu (b/s, bps). ...

Cena...

Bezpečnost...

Dostupnost...

škálovatelnost...

Spolehlivost...

2.1.2 Základní síťové prvky

textbfPřepínač (Switch) Přepínač je síťově zařízení pracující na L2, tedy spojové vrstvě [RM ISO/OSI](#). Jedná se většinou o zařízení s více porty, kdy je ke každému portu připojeno koncové zařízení. Úkolem switche je pak předávat příchozí ethernetové rámce na příslušný port podle cílové MAC adresy. Switch si vede tabulku, ve které přiřazuje adresy zařízení připojených ke

konkrétním portům.

Přepínače nahradili v sítích dnes již zastaralé rozbočovače neboli Huby. Ty se od switchů lišily především v tom, že si nevedly tabulku adres připojených zařízení, ale pouze rozesílaly příchozí rámce na všechny porty s výjimkou příchozího. To velmi zvyšovalo provoz v síti. Představme si, že máme 10 osobních počítačů připojených prostřednictvím rozbočovače. Bude-li chtít počítač č. 1 komunikovat s počítačem č. 2, veškerá jejich komunikace v obou směrech bude zároveň odesílána na počítače č. 3-č. 10. Ty pak tato data v lepším případě zahodí, protože jim nejsou určena, v horším případě to umožňuje potenciálnímu útočníkovi odposlouchávat tuto komunikaci.

Toto „zbytečné“ kopírování dat navíc velmi negativně ovlivňuje propustnost sítě. Všechny koncové stanice propojené pomocí hubu tvoří tzv. kolizní doménu. Pokud chce daná stanice vysílat na sdíleném médiu, musí nejdříve naslouchat, je-li médium volné – tedy zda na něm právě ne-vysílá jiná stanice. V důsledku konečné rychlosti šíření elektromagnetické vlny v médiu však může nastat situace, že stanice začne vysílat krátký okamžik po tom, co začala vysílat jiná stanice a dojde tak ke kolizi. O vzniku této kolize jsou stanice informovány speciálním signálem, po kterém počkají náhodnou dobu, než začnou znovu vysílat. Oproti tomu v sítích, které používají switche, tvoří kolizní doménu vždy jeden port switche a zařízení k němu připojené. Tato metoda se označuje jako mikrosegmentace a zabraňuje vzniku kolizí. Moderní switche navíc podporují řadu pokročilých funkcí, jako je tvorba virtuálních lokálních sítí [Virtual Local Area Network \(VLAN\)](#), prioritizaci daného typu provozu a v neposlední řadě umožňují zabránit neoprávněným uživatelům v připojení do sítě.

Směrovač (Router) Směrovač je síťové zařízení, jehož úkolem je směrovat provoz mezi vícero sítěmi. Oproti přepínači pracuje pouze s [Internet Protocol \(IP\)](#) adresami, jedná se tedy o L3 zařízení. Router se většinou nachází na okraji lokální sítě a zpracovává data, která jsou určena k odeslání mimo lokální síť, nebo naopak do ní. Router bývá na odchozí straně připojen k jednomu nebo více dalším routerům. Zde je důležité zmínit, že každý port routeru má vlastní IP adresu a porty sousedních routerů, kterými jsou propojeny, se musí nacházet ve stejné podsíti neboli subnetu.

IP adresu můžeme rozdělit na část síťovou a hostovskou. Všechny stanice (angl. hosts) v rámci jedné sítě mají společnou síťovou část IP adresy. Aby bylo možné určit, kde je hranice mezi těmito dvěma částmi adresy, používá se síťová maska (angl. network mask). Ta je stejně jako IP adresa tvořena čtyřmi oktety, tedy 32 bity. Provedením bitové operace AND mezi IP adresou a síťovou maskou dostaneme adresu sítě:

IP adresa 192.168.1.100:	11000000.10101000.00000001.01100100
Síťová maska 255.255.255.0:	11111111.11111111.11111111.00000000
Adresa sítě 192.168.1.0:	11000000.10101000.00000001.00000000

Právě na základě síťové adresy router rozhodne, kam daný paket vyslat. Spadá-li cílová adresa paketu do definovaného adresního rozsahu, je paket odeslán na odpovídající další zařízení. V případě, že router nemá konkrétně definovaný rozsah, do kterého daná adresa spadá, využije výchozího záznamu, který se označuje jako výchozí brána. Tento proces se nazývá směrování a údaje podle kterých se router rozhoduje jsou obsaženy ve směrovací tabulce. Ta může být nastavená staticky administrátorem sítě, nebo dynamicky pomocí směrovacích protokolů jako je [RIP](#), [OSPF](#), [EIGRP](#), [BGP](#) a dalších.

2.2 Základní požadavky na konfiguraci

Jaká jsou základní nastavení Mikrotiku, kterých se snažím dosáhnout

2.3 Možnosti konfigurace síťových zařízení

Pomocí čeho a jak lze konfigurovat zařízení různých výrobců.

Vendor specific, opensource

Popis jednotlivých možností, výhody a nevýhody jednotlivých řešení

Někteří výrobci mají vlastní konfigurační nástroje...

Většina komerčních nástrojů se zaměřuje zejména na velké výrobce (Cisco, Huawei, HP, Juniper)

Opensourceové nástroje pro automatizaci (Chef, Puppet, Ansible)

2.4 Ansible

Pozn.: Tuto sekci rozšířím a přepíšu

Ansible je opensource softwarová platforma pro konfiguraci a management (především) serverů. Na rozdíl od ostatních takovýchto nástrojů je Ansible agent-less a nevyžaduje tedy žádnou instalaci agenta na koncových zařízeních. V terminologii Ansiblu se tato zařízení označují jako uzly (angl. *node*) [3]. Informace o jednotlivých uzlech jsou uvedeny v inventáři (angl. *Inventory*), kde je možné definovat skupiny uzlů a proměnné (angl. *Variables*). Proměnné lze použít například pro uživatelská jména a hesla, což značně usnadňuje další práci.

Ansible je napsaný v jazyce Python a primárně používá pro připojení k jednotlivým uzlům protokol SSH. Provádění samotných konfiguračních příkazů mají na starost skripty, tzv. moduly (angl. *modules*). Ty mají definovaný seznam argumentů, na jejichž základě provedou požadovanou operaci. Argumenty určují požadovaný stav nastavovaných hodnot, změny se tak provedou pouze v

případě, že aktuální hodnota neodpovídá té požadované.

Většina základních modulů, které jsou součástí instalace Ansiblu je však zaměřena na konfiguraci serverů a vyžaduje podporu jazyka Python na každém uzlu. Ansible pak na základě zvolených argumentů vytvoří skript a spustí jej na zvoleném uzlu. Z tohoto důvodu zatím Ansible neumožňuje konfiguraci většiny síťových prvků. Pro jejich konfiguraci je tak třeba vytvoření modulů, které se spouštějí pouze na straně Ansible serveru a uzlům pak předávají už pouze low-level příkazy. V terminologii Ansiblu se typ tohoto připojení označuje jako lokální (angl. *local*). Potom je pak možné zvolit libovolný typ připojení podporovaný síťovým prvkem (Telnet, SSH, API,...) a přizpůsobit mu daný modul.

Ansible umožňuje spouštění modulů ve dvou základních režimech. V režimu ad-hoc [2] se vždy spustí pouze jeden zvolený modul s danými parametry `??`. Ve druhém režimu se postupně spustí libovolný počet modulů ze seznamu. Tyto seznamy se označují jako playbooks [1]. Playbooks jsou psány v jazyce [Yet Another Markup Language \(YAML\)](#), díky čemuž jsou velmi přehledné a jednoduché na pochopení. Každý playbook obsahuje název skupiny uzlů, proti kterým budou příkazy použity. Příklad playbooku: `??`. Playbooky také umožňují práci s proměnnými, což značně usnadňuje použití modulů proti velkému počtu koncových uzlů.

2.5 MikroTik

Pozn.: Tuto sekci rozšířím a přepíšu

Společnost MikroTik byla založena v roce 1995 a zabývá se vývojem a prodejem síťových prvků používaných především poskytovateli internetu (dále ISP). Společnost je známá zejména díky svému operačnímu systému RouterOS, který je oblíbený pro svou robustnost, konfigurovatelnost a především výkon `??`.

Síťové prvky společnosti MikroTik nabízejí široké možnosti konfigurace, velké množství podporovaných funkcí a vysoký výkon za cenu někdy mnohonásobně nižší než u zařízení konkurenčních značek. Z těchto důvodů jsou velmi oblíbené u ISP pro realizaci přístupové části sítě, především pak pro bezdrátová řešení. Vzhledem k vysokému počtu nasazovaných zařízení je potřeba co nejvíce usnadnit a urychlit jejich prvotní konfiguraci, což je cílem tohoto projektu.

2.5.1 Možnosti konfigurace zařízení MikroTik

V této části se budu věnovat popisu jednotlivých možností konfigurace zařízení MikroTik.

Pozn.: Tuto sekci rozšířím a přepíšu

Síťové prvky s operačním systémem RouterOS nabízejí tyto možnosti konfigurace:

- Telnet
- SSH
- API
- Web GUI
- WinBox

Telnet

Zkratka z *Telecommunication Network* - představuje základní způsob pro nezabezpečené vzdálené připojení ke [CLI](#). Typicky je provozován na portu TCP 23.

SSH

Secure Shell - zabezpečený komunikační protokol, náhrada za Telnet. Stejně jako Telnet umožňuje vzdálené připojení ke CLI. Typicky je provozován na portu TCP 22.

API

Application Programming Interface - rozhraní pro programování aplikací. Jedná se o sbírku procedur, funkcí, tříd či protokolů nějaké knihovny, které může programátor využívat. API určuje, jakým způsobem jsou funkce knihovny volány ze zdrojového kódu programu. [7]. Rozhraní API tak umožňuje tvorbu vlastních softwarových řešení pro komunikaci s RouterOS zařízeními za účelem získání informací a úpravy jejich konfigurace. Ve výchozím stavu je tato služba provozována na portu TCP 8728.

Komunikace se zařízením RouterOS probíhá vysíláním vět, přičemž v odpovědi na jednu větu je přijata jedna nebo více vět. Věty se skládají ze slov a jsou ukončeny slovem nulové délky. Prvním slovem věty je příkaz (command word) následovaný atributy (attribute word) v libovolném pořadí [4].

Web GUI

Web Graphical User Interface - představuje ideální možnost konfigurace pro méně zkušené uživatele, jedná se o grafické prostředí běžící ve webovém prohlížeči.

WinBox

Jednoduchá utilita pro systémy Windows. Zobrazuje grafické prostředí umožňující konfiguraci zařízení. Jednotlivé sekce nastavení jsou co možná nejlíže konzolovým příkazům. Jedná se

o proprietární řešení společnosti MikroTik a ve výchozím stavu je pro připojení použit port [Transmission Control Protocol \(TCP\) 8921 \[5\]](#).

2.6 Shrnutí a vyhodnocení poznatků

Proč jsem se rozhodl pro API?

3 Realizace

3.1 MikroTik API

Popis komunikace API, struktura slov, jak to funguje

3.2 Základní funkce

API, RosAPI, RosRaw - jak fungují, co dělají

3.3 Moduly Ansible

Jaké jsou, co jsou zač, jak fungují, co musí každý modul splňovat

3.4 Ansible Playbook

3.5 Příklady použití

Reference

- [1] Ansible Docs. Intro to playbooks. http://docs.ansible.com/ansible/playbooks_intro.html, 2016. [Online].
- [2] Ansible Docs. Introduction to ad-hoc commands. http://docs.ansible.com/ansible/intro_adhoc.html, 2016. [Online].
- [3] Lorin Hochstein. *Ansible: Up and Running*. O'Reilly Media, Inc., May 2015.
- [4] MikroTik Wiki. Manual:api - mikrotik wiki. <http://wiki.mikrotik.com/wiki/Manual:API>. [Online].
- [5] MikroTik Wiki. Manual:winbox - mikrotik wiki. <http://wiki.mikrotik.com/wiki/Manual:Winbox>. [Online].
- [6] Wikipedia. Information technology infrastructure library — wikipedia, the free encyclopedia. <https://en.wikipedia.org/wiki/ITIL>, 2014. [Online].
- [7] Wikipedia. Api — wikipedia, the free encyclopedia. <https://cs.wikipedia.org/wiki/API>, 2016. [Online].
- [8] Wikipedia. Osi model — wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/OSI_model, 2016. [Online].