

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Skružný** Jméno: **Petr** Osobní číslo: **457860**
Fakulta/ústav: **Fakulta informačních technologií**
Zadávající katedra/ústav: **Katedra počítačových systémů**
Studijní program: **Informatika**
Studijní obor: **Bezpečnost a informační technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Automatická konfigurace serverů a instalace OS

Název bakalářské práce anglicky:

Automatic server configuration and OS installation

Pokyny pro vypracování:

Prozkoumejte volně dostupné systémy pro automatickou instalaci serverů, např. Foreman, Ironic, Crowbar, MaaS. Vyberte z nich jeden nejvhodnější na základě funkcionality, jednoduchosti instalace, použití a velikosti uživatelské základny a vytvořte konfiguraci pro instalaci Linuxu (ideálně s podporou více distribucí) a Windows na dvou druzích serverů (HPE, SuperMicro).
Systém by měl umět mimo instalace operačního systému na servery také nastavit IP adresy rozhraní pro správu serveru a rozhraní do datové sítě a konfigurovat RAID řadiče serverů.

Seznam doporučené literatury:

Dodá vedoucí práce.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Tomáš Vondra, Ph.D., katedra kybernetiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **31.12.2018** Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: _____

Ing. Tomáš Vondra, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Automatická konfigurace serverů a instalace OS

Petr Skružný

Katedra počítačových systémů
Vedoucí práce: Ing. Tomáš Vondra, Ph.D

13. května 2019

Poděkování

Chtěl bych poděkovat panu Ing. Tomáši Vondrovi, Ph.D, za trpělivost, ochotu a pomoc s tvorbou této práce.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 13. května 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Petr Skružný. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Skružný, Petr. *Automatická konfigurace serverů a instalace OS*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Tato práce se zabývá volbou vhodného systému pro konfiguraci serverů a instalaci operačního systému na servery v prostředí datového centra v Jihlavě. Po systémech je požadováno, aby byly schopny konfigurovat síťová rozhraní serverů a jejich napájecí rozhraní. Cílem práce je vybrat z volně dostupných systémů umožňujících vzdálené a automatizované konfigurování serverů pro zjednodušení poskytování služeb datového centra pro zákazníky.

Výsledkem práce bude kromě porovnání systémů také nasazení jednoho z nich v prostředí datového centra, kde pak bude dále používán v běžném provozu.

Klíčová slova konfigurace serveru, instalace OS, provision systém, open-source, HPE, SuperMicro, Foreman, Ironic, Cobbler, MaaS, Spacewalk

Abstract

This work focuses on choosing appropriate system for configuring servers and installing operation system on servers in data center located in Jihlava. The systems are required to be able to configure network devices of server and their power management interfaces. Goal of this work is to choose system capable of remote and automatic server configuration to allow for easier providing of customer service by the data center.

The result of this work aside from comparison of said systems will also be deployment of one of the system directly in the data center where it will be used in day-to-day operations.

Keywords server configuration, OS installation, provision system, open-source, HPE, SuperMicro, Foreman, Ironic, Cobbler, MaaS, Spacewalk

Obsah

Literatura	1
Úvod	3
1 Cíle práce	5
2 Základní pojmy	7
2.1 Bare metal provisioning	7
2.2 Průběh bare metal provisioning	8
3 Analýza a porovnání	11
3.1 Dostupné projekty	11
3.2 Foreman	11
3.3 Ironic	14
3.4 Cobbler	16
3.5 MaaS	17
3.6 Spacewalk	19
3.7 Závěrečný výběr	21
4 Realizace	23
4.1 Popis projektu Foreman	23
4.2 Instalace Foreman serveru	23
4.3 Konfigurace pro automatickou instalaci	26
4.4 Automatická instalace	28
Závěr	31
A Seznam použitých zkratk	33
B Obsah příloženého CD	35

Seznam obrázků

2.1	Průběh bare metal provision	9
3.1	Architektura Foreman	12
3.2	Architektura Ironic	14
3.3	Architektura MaaS	18
3.4	Architektura Spacewalk	20
4.1	Schéma nasazení projektu Foreman	24

Seznam tabulek

4.1	Tabulka portů	25
-----	-------------------------	----

Literatura

- [1] Sekera, J.: Automatická instalace linuxového serveru a základní konfigurace služeb. 2017, [PDF]. Dostupné z: <http://hdl.handle.net/10467/69647>
- [2] Sokol, J.: Infrastruktura pro centralizovanou automatickou instalaci serverů. 2017, [PDF]. Dostupné z: <http://hdl.handle.net/10467/69662>
- [3] *Preboot Execution Environment (PXE) Specification*. Intel Corporation, [online], [cit. 2019-05-02], září 1992. Dostupné z: <https://web.archive.org/web/20131102003141/http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf>
- [4] Droms, R.: *Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types*. Network Working Group, [online], [cit. 2019-05-02], září 2000. Dostupné z: <https://tools.ietf.org/html/rfc2939>
- [5] Sollins, K.: *The TFTP Protocol (Revision 2)*. Network Working Group, [online], [cit. 2019-05-02], červenec 1992. Dostupné z: <https://tools.ietf.org/html/rfc1350>
- [6] *Supermicro IPMI - What is it and what can it do for you?* Boston IT Solutions, [online], [cit. 2019-05-02], 16. listopad 2016. Dostupné z: <https://www.bostonindia.in/technical/2015/10/supermicro-ipmi-what-is-it-and-what-can-it-do-for-you.aspx>
- [7] Patterson, D. A.; Gibson, G. A.; Katz, R. H.: *A case for redundant arrays of inexpensive disks (RAID)*. Computer Science Division, University of California, [online], [cit. 2019-05-01], 1987. Dostupné z: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/1987/CSD-87-391.pdf>
- [8] Chapter 31. Advanced Networking. [online], [cit. 2019-05-05]. Dostupné z: <https://www.freebsd.org/doc/handbook/network-diskless.html>

- [9] *Introduction*. Foreman, [online], [cit. 2019-05-02], duben 2019. Dostupné z: <https://www.theforeman.org/introduction.html>
- [10] Quickstart Guide. [online], [cit. 2019-05-12], duben 2019. Dostupné z: https://www.theforeman.org/manuals/1.20/quickstart_guide.html
- [11] Welcome to Ironic's documentation! [online], [cit. 2019-05-05], září 2017. Dostupné z: <https://docs.openstack.org/ironic/latest/>
- [12] Install and configure for Ubuntu. [online], [cit. 2019-05-12], 17. ledna 2018. Dostupné z: <https://docs.openstack.org/ironic/stein/install/install-ubuntu.html>
- [13] Introduction to Cobbler. [online], [cit. 2019-05-12], 2019. Dostupné z: https://cobbler.github.io/manuals/2.8.0/1_-_About_Cobbler.html
- [14] Cobbler Quickstart Guide. [online], [cit. 2019-05-12], 2019. Dostupné z: <https://cobbler.github.io/manuals/quickstart/>
- [15] MAAS Documentation. [online], [cit. 2019-05-12], 2018. Dostupné z: <https://docs.maas.io/2.5/en/>
- [16] Install MAAS. [online], [cit. 2019-05-12], 2019. Dostupné z: <https://maas.io/install>
- [17] Spacewalk - Introduction. [online], [cit. 2019-05-12], 2019. Dostupné z: <https://spacewalkproject.github.io/>
- [18] Mraka, M.: HowToInstall. [online], [cit. 2019-05-12], 27. listopadu 2018. Dostupné z: <https://github.com/spacewalkproject/spacewalk/wiki/HowToInstall>
- [19] 1. Foreman 1.21 Manual. [online], [cit. 2019-05-12], 2019. Dostupné z: <https://www.theforeman.org/manuals/1.21/index.html#3.InstallingForeman>

Úvod

Práce se zabývá tématem volby vhodného systému pro správu neustále narůstajícího množství serverů. Se zvyšující se poptávkou po výpočetním výkonu jako odpověď vznikají a rozšiřují se datová centra. Aby byla schopna uspokojit poptávku, musí neustále rozšiřovat svou výpočetní kapacitu.

To znamená nákup nových strojů, jejich údržbu a správu. Pro snížení výdajů je pak také třeba znovu využívat i starší stroje, a ne je likvidovat. Aby byla taková operace vůbec možná, je zapotřebí systému, který umožňuje centralizovanou správu, poskytuje možnosti konfigurace a správy sítě a rozhraní k ní připojených, instalace operačních systémů na nové i staré stroje a monitorování celé infrastruktury.

Takový systém ovšem může být velmi drahý a pro menší datové centrum tím pádem nevhodný. Z toho důvodu se v této práci soustředím pouze na projekty licencované pod některou z open source licencí a na ty, které jsou zdarma k použití.

Výsledkem této práce bude nasazený systém pro konfiguraci serverů a instalaci operačních systémů přímo v prostředí datového centra v Jihlavě. Tento systém bude připraven k okamžitému používání zaměstnanci datového centra a měl by zjednodušit a zefektivnit jejich práci.

Důvodem, proč jsem si zvolil toto téma bakalářské práce, je můj zájem o automatizaci procesů. Proto chci nahlédnout za oponu datového centra, kde je automatizace nutností, abych pochopil, jak jsou používané procesy automatizovány. A protože se zajímám i o počítačovou bezpečnost, zajímají mě i možnosti zabezpečování těchto procesů a systémů.

Práce se dotýká obdobné problematiky, kterou se zabývali kolegové Jakub Sekera a Jan Sokol v pracích *Automatická instalace linuxového serveru a základní konfigurace služeb* a *Infrastruktura pro centralizovanou automatickou instalaci serverů* [1] [2]. Avšak na rozdíl od nich se zaměřím na jiné parametry u zkoumaných systémů.

V kapitole 2 zavedu a vysvětlím základní pojmy používané v této práci.

Kapitola 3 se zabývá nabídkou volně dostupných projektů umožňujících konfiguraci serverů a instalaci operačních systémů. Zabývám se jak splněním základních požadavků ze zadání práce, tak i přednostmi a nevýhodami jednotlivých projektů. Závěrem této kapitoly je výběr projektu, u kterého v kapitole 4 provedu instalaci a konfiguraci na serveru v datovém centru v Jihlavě. Následně udělám lehký průvod přidáním nového klienta a jeho instalací. Závěr shrnuje úkony provedené v práci, hodnotí dosažení cílů a možnosti dalšího rozšíření zvoleného řešení.

Cíle práce

S rostoucími nároky na výpočetní výkon stoupá obliba pronájmu virtuálních privátních serverů, virtuálních strojů přizpůsobených na míru uživateli. Pronájemem uživateli odpadají náklady na pořízení a údržbu hardware. Tyto stroje bývají umístěny v datových centrech a pracují se skutečným hardwarem, přičemž na jednom fyzickém serveru může fungovat více těchto virtuálních serverů.

Aby datová centra mohla uspokojit zvyšující se poptávku, musí se neustále rozšiřovat. Zároveň si ale musí udržet přehled a kontrolu nad velkým množstvím strojů, tudíž je důležité mít centralizovaný systém správy.

Konfigurace fyzických serverů a instalace operačních systémů je časově náročná činnost vyžadující během těchto operací přítomnost administrátora. Tudíž je záhodno mít systém umožňující jak správu i existujících serverů a virtuálních strojů na nich běžících, tak konfiguraci nových serverů a následnou instalaci operačního systému.

Tyto systémy bývají označovány jako tzv. „provision“ systémy. Na trhu existuje množství uzavřených i open source projektů, které takový systém nabízejí.

Cílem práce je prozkoumat nabídku open source projektů nabízejících automatickou konfiguraci a instalaci serverů a nevhodnější z nich vybrat.

Požadavky jsou nastavení IP adres do datové a management sítě, instalace jedné z několika distribucí systému Linux nebo Windows Server a nakonfigurování RAID řadiče. Tyto operace musí projekt umět vykonat nad servery značek HPE a SuperMicro.

Dalším kritériem pro výběr je množství dalších nabízených funkcionalit a složitostí, potažmo jednoduchost, instalace projektu, dále náročnost obsluhy a velikost uživatelské základny.

Základní pojmy

Tato kapitola je věnována vysvětlení pojmů a zkratky dále používané v této práci. Požadavky práce lze shrnout jedním termínem, a tím je bare metal provisioning. V dalších sekcích vysvětlím tento termín a k němu se vztahující další pojmy.

2.1 Bare metal provisioning

Bare metal provisioning je proces instalace operačního systému na počítačový hardware. Je složen z několika na sebe navazujících kroků a spojuje funkce několika služeb, aby mohlo být dosaženo žádaného výsledku.

2.1.1 PXE

Preboot Execution Environment(PXE). PXE má tři primární funkce:

1. Poskytuje metody pro inicializaci firmware klientského stroje pro spuštění PXE protokolu;
2. poskytuje sadu API ve firmware stroje, poskytující sadu služeb využitelných NBP nebo BIOS;
3. je protokolem umožňujícím klientovi vyžádat si síťovou adresu a stáhnout NBP soubor z příslušného serveru.

Schopnost zavádění přes síť zjednodušuje nasazení a správu serverů pro administrátora [3].

2.1.2 DHCP

Dynamic Host Configuration Protocol. Protokol poskytující rámec pro předávání konfiguračních parametrů, jako jsou IP adresy pro zařízení a služby na TCP/IP

síti [4]. Pomocí PXE a DHCP získá BIOS IP adresu pro síťové rozhraní a adresu serveru, na kterém je uložen NBP.

2.1.3 NBP

Network Bootstrap Program. Zavaděč jádra operačního systému obdobně jako GRUB či LILO, které jsou běžně používány pro lokální zavádění. Stejně jako zavaděč na pevném disku NBP je zodpovědný za načtení operačního systému do operační paměti, aby mohl být operační systém zaveden přes síť.

2.1.4 TFTP

Trivial File Transfer Protocol. Jednoduchý protokol pro přenos souborů běžně používaný pro automatický přenos konfiguračních souborů nebo zaváděcích souborů v místním prostředí. V prostředí PXE je TFTP používáno pro stahování NBP přes síť s pomocí informací získaných od DHCP serveru [5].

2.1.5 IPMI

Intelligent Platform Management Interface. IPMI je standardizované systémové rozhraní pro systémové administrátory sloužící pro out-of-band správu a monitorování počítačových systémů. Díky IPMI je možné spravovat systém, který nemusí odpovídat nebo může být vypnutý s pomocí pouze síťového připojení přímo k hardware namísto operačního systému [6].

2.1.6 RAID

Redundant Array of Inexpensive Disks. Technologie vyvinutá jako odpověď na problém spolehlivosti samostatných pevných disků. Jedná se o uspořádané pole pevných disků. Cílem tohoto uspořádání je primární navýšení spolehlivosti a získání redundance. V závislosti na uspořádání lze pak docílit též navýšení celkové kapacity či zvýšení rychlosti čtení.

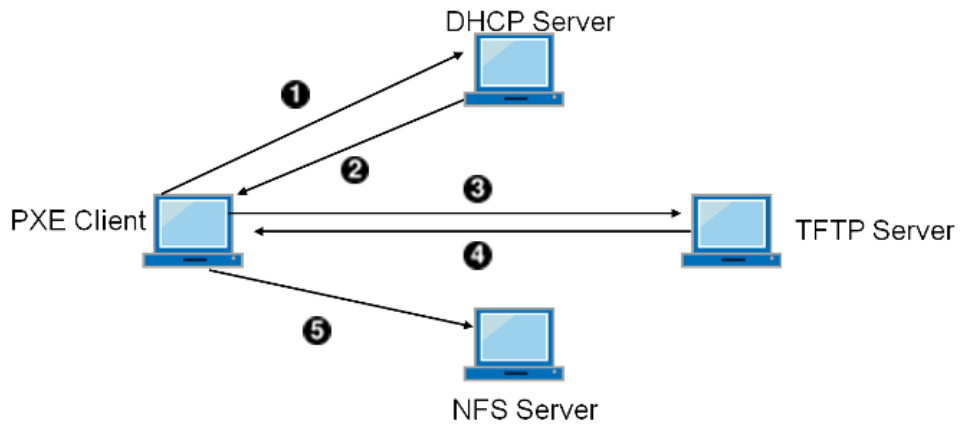
RAID řadič je zařízení spravující obsluhu a chod diskového pole. Díky tomu dochází ke snížení zatížení procesoru [7].

2.2 Průběh bare metal provisioning

Bare metal provisioning probíhá v několika na sobě závislých krocích. Tyto kroky využívají dříve zmíněné technologie.

1. Host vyšle DHCPDISCOVER zprávu.
2. DHCP server odpoví IP adresou, adresou next-server, jménem souboru a cestou ke kořenové složce.

Obrázek 2.1: Průběh bare metal provision



3. Host vyšle TFTP požadavek na next-server, ze kterého chce získat soubor, jehož jméno předtím obdržel.
4. TFTP server odpoví a pošle soubor hostovi.
5. Host soubor spustí, soubor načte jádro. Po spuštění jádra je přes NFS připojena kořenová složka, k níž host obdržel cestu od DHCP serveru [8].

Analýza a porovnání

Tato kapitola je věnována porovnání vybraných projektů a zda splňují požadavky, vyplývající ze zadání práce, které jsou na ně kladeny. V sekci 3.1 popíšu jednotlivé projekty. Poté v sekci 3.7 provedu porovnání daných projektů a zvolím výsledný projekt pro nasazení v datovém centru.

3.1 Dostupné projekty

V této kapitole popíši jednotlivé projekty, zda splňují požadavky práce a jejich největší výhody, popřípadě nevýhody.

Všechny zmiňované projekty jsou licencovány pod některou z open source licencí a jsou dostupné k použití bez poplatku.

Mezi takové projekty, kterým se budu dále věnovat patří:

- Foreman;
- Ironic;
- Cobbler;
- MaaS;
- Spacewalk.

V dalších sekcích uvedu český překlad představení jednotlivých projektů. Dále se pak budu věnovat jejich instalaci, způsobu používání, na to se váže uživatelská přívětivost, a zda a které z požadavků práce projekt splňuje.

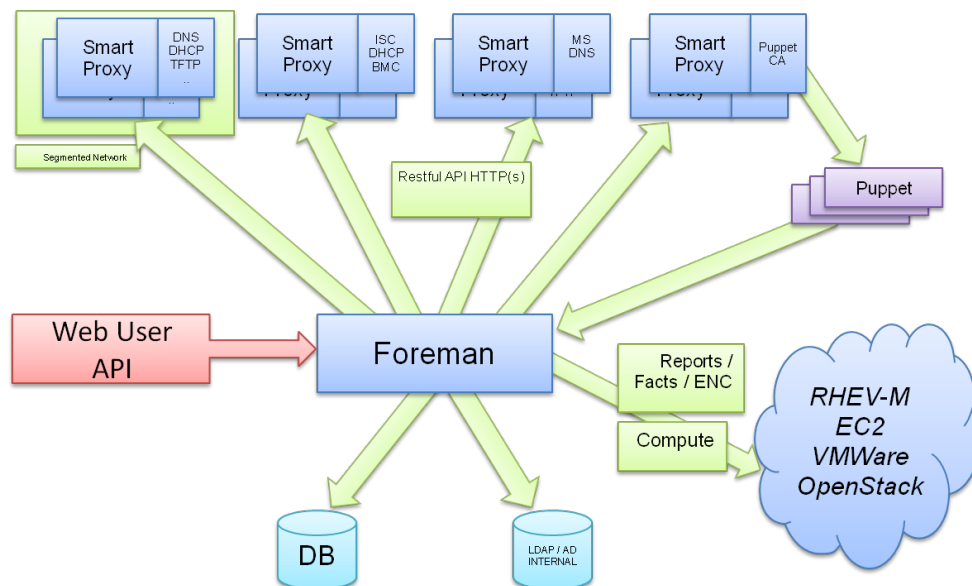
3.2 Foreman

Foreman je open source projekt pomáhající systémovým administrátorům spravovat servery během jejich životního cyklu od instalace a konfigurace po orchestraci a monitorování. Díky podpoře provision je snadné nastavovat nové

3. ANALÝZA A POROVNÁNÍ

servery a pomocí správců nastavení (Puppet, Ansible, Chef a Salt jsou podporovány) lze snadno zautomatizovat opakující se úlohy [9].

Foreman poskytuje webové rozhraní, CLI a RESTful API, umožňující vytvoření business logiky na vyšší úrovni.



Obrázek 3.1: Architektura Foreman

https://www.theforeman.org/static/images/foreman_architecture.png

3.2.1 Instalace

Instalace Foreman je vcelku jednoduchá. Na webových stránkách v sekci *Quick-start* je k dispozici návod pro stažení a instalaci instalátoru včetně systémových požadavků [10].

Jeden problém, který není v návodu popsán, je nutnost mít vytvořenou doménu. Foreman vyžaduje existenci domény pro spuštění instalátoru. Ovšem tato doména nemusí být nikde zaregistrována.

Následná instalace pomocí instalátoru je pak přímočará a jednoznačná. Instalátor používá přehledné dělení pro jednotlivé funkce Foreman. Díky tomu je celý proces instalace i konfigurace nenáročný.

3.2.2 Použití

Foreman je možné používat buď přes webové rozhraní, či CLI. Změny některých nastavení se dají dělat přímo v obou rozhráních. Většina změn se ovšem provádí přes *foreman-installer*. V něm je možné přidávat a odebírat služby či měnit jejich nastavení. Instalátor je dostupný pouze přes terminál.

Používání webového rozhraní hodnotím velmi kladně. Je přehledné a dobře strukturované. Lze přímo z něj zjistit stav služeb běžících na všech *foreman-proxy*, včetně jejich verzí, a u některých služeb, například DHCP nebo DNS i sítě a domény jimi spravované. Dále je možné zjistit stav klientů přidáných pod Foreman. U klientů je pak možné nahlédnout jejich konfiguraci, aktuální stav a různé další metriky. Tvorba a editace klientů je jednoduchá, po troše přípravy i z velké části automatická. Při použití skupin *Host Groups* se velká část údajů při tvorbě klienta vyplní automaticky.

Práce s CLI je obdobná práci s webovým rozhráním. K dispozici jsou dobře napsané manuálové stránky. Samotné příkazy jsou udělané tak, že uživatel již z názvu příkazu ví, co přesně dělá.

Používání CLI je rychlejší než používání webového rozhraní, ale vyžaduje již znalost potřebných příkazů. Avšak uživatele nerozptyluje grafika a soustředí se pouze na to, co píše. Výhodou webového rozhraní je naopak dostupnost. Pro používání CLI musí být uživatel připojen přímo k serveru, kdežto k webovému rozhraní může přistupovat i z venkovní sítě. Navíc zobrazení statistik do grafů je přehlednější než pouze jejich výpis.

Celkově jsou obě metody používání Foreman velice dobré. Osobně preferuji webové rozhraní, protože si nemusím pamatovat potřebné příkazy. Ještě jednou možností používání, kterou jsem zde nezmiňoval, je integrace Foreman do jiné služby. Toho lze dosáhnout díky poskytovanému API, které jednotlivé komponenty Foreman mají a využívají.

3.2.3 Splnění požadavků

Foreman podporuje velké množství linuxových distribucí. Instalaci Windows podporuje pouze z image. V oficiální dokumentaci je k dispozici podrobný návod, jak připravit konfiguraci operačního systému jak s pomocí síťového instalátoru, tak s pomocí image.

Kromě nastavení síťových rozhraní podporuje Foreman i možnost spravovat podsítě. Lze tak nakonfigurovat podsít' přímo v CLI nebo webovém rozhraní a rovnou jí přiřadit DHCP, DNS a TFTP server. Pomocí doplňku *dhcp infoblox* pak lze nahlížet i na to, kterým MAC adresám je přidělena jaká IP adresa.

Foreman nepodporuje ovládání RAID řadiče.

Repozitář na serveru github.com má 81 členů. Samotný Foreman má 286 přispěvatelů, Smart Proxy má 89 přispěvatelů, instalátor 28, webové stránky 166 a komunitní šablony jich mají 88.

Celkově je komunita kolem projektu Foreman velmi aktivní. V repozitářích na github dochází v průměru k 10 změnám za den a nahlášené problémy jsou rychle řešeny.

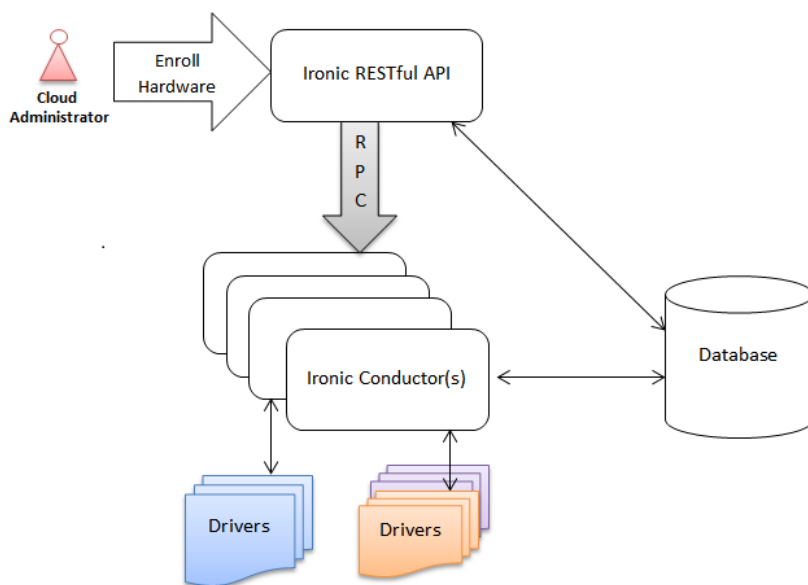
Foreman je sponzorován společnostmi a komunitami Rackspace, Red Hat, oVirt, Scaleway, Netways, JetBrains, ATIX a dm.

3.3 Ironic

Ironic je projekt ze skupiny OpenStack umožňující provision fyzických strojů. Lze ho použít samostatně, nebo jako součást OpenStack Cloud. Je možné ho integrovat s OpenStack Identity, Compute, Network, Image a Object službami.

Při patřičném nastavení Bare Metal služby s Compute a Network službami je možné poskytovat jak fyzické, tak virtuální stroje pomocí API služby Compute. Možnosti nabízených akcí u jednotlivých instancí jsou omezené kvůli různým charakteristikám fyzických serverů a přepínacího hardware.

Ironic využívá open source technologie, především PXE a IPMI, pro pokrytí velkého množství hardware [11].



Obrázek 3.2: Architektura Ironic

https://docs.openstack.org/ironic/pike/_images/deployment_architecture_2.png

3.3.1 Instalace

Instalace Ironic je náročná na instalující uživatele v tom smyslu, že uživatel instaluje jednotlivé služby, z nichž Ironic sestává. To je v celkem ostrém kon-

trastu s ostatními projekty, které toto dělají za uživatele. Na jednu stranu nejde o špatnou věc. Uživatel má díky tomu absolutní kontrolu nad tím, co, jak a kam se instaluje. Tomu velmi napomáhá podrobný instalační návod v dokumentaci OpenStack. Na druhou stranu by se nižší míra automatizace instalace dala považovat za zbytečnou zátěž uživatele, který chce projekt co nejrychleji nainstalovat a nakonfigurovat, ideálně s co nejmenším zásahem od něj, aby jej mohl co nejdříve používat ke svojí činnosti.

Další záležitostí, která by se dala označit jako výhoda i nevýhoda, je konfigurace služby. Výhodou je umístění celé konfigurace do jednoho souboru, tudíž veškeré editace stačí provádět nad tímto souborem a není třeba přeskakovat mezi soubory při rozsáhlejších úpravách. Nevýhodou toho je, že soubor má více než 4000 řádků, což činí jakékoliv vyhledávání složitým a zdlouhavým. Z toho zároveň vyplývá i určitá nepřehlednost, protože konfigurace všech komponent je na jednom místě [12].

Z mého osobního pohledu by bylo lepší mít konfigurační soubor logicky rozdělen na více souborů. Sice by pak následné úpravy konfigurace mohly vyžadovat editace více souborů, ale zvýšila by se tím celková přehlednost.

3.3.2 Použití

Samotný Ironic nabízí pouze CLI. Příkazy jsou srozumitelné a dobře strukturované. Výpisy s příkazy jsou taktéž dobře strukturované a tím pádem velmi dobře čitelné. Nevýhodou příkazů je ovšem jejich délka. To sice umožňuje dobrou míru srozumitelnosti a jednoznačnosti, ale na druhou stranu vnáší větší možnost učinit chybu, kterou je potom nutné odhalit a příkaz přepsat.

Tuto nevýhodu lze eliminovat použitím *openstack baremetal* klienta. Ten umožňuje tvorbu šablon, které lze pak dále používat. Při tvorbě šablony musí být specifikován parametr „-steps“, který má jednu z následujících tří hodnot:

- řetězec ve formátu JSON,
- cesta k souboru ve formátu JSON nebo
- „-“, pro čtení ze standardního vstupu.

Při integraci s dalšími službami OpenStack lze pak rozšířit celkovou funkcionalitu a navíc zpřístupnit webové grafické rozhraní pro přehlednější správu.

3.3.3 Splnění požadavků

V systému Ironic je použití image preferovanou metodou instalace. Díky tomu lze instalovat téměř libovolný systém, ať už Linux či Windows, ovšem za předpokladu, že image obsahuje i potřebné ovladače pro hardware, na který bude operační systém instalován.

Díky integraci služeb DNS a DHCP lze spravovat jednotlivá síťová rozhraní

klienta.

Konfiguraci RAID řadiče je možná za pomoci Ironic CLI nebo REST API. Pro reprezentaci aktuální konfigurace je používán formát JSON. Díky tomu je konfigurace snadná a přehledná.

Dle repozitáře na github má projekt Ironic samotný 366 přispěvatelů, a to je pouze jeden z projektů pod záštitou OpenStack.

Co se týče nevyřešených problémů, zde se počet pohybuje v řádech stovek, ale ze záznamů lze vyčíst, že na problémech se intenzivně pracuje. V době psaní tohoto odstavce bylo otevřeno 335 závad a 1570 bylo vyřešeno.

OpenStack je spozorován AT&T, Ericsson, Huawei, Intel, Rackspace, Red Hat, SUSE, Tencent Cloud, 99Cloud, Canonical, China Mobile, Cisco a mnoha dalšími.

3.4 Cobbler

Cobbler je linuxový instalační server, umožňující rychlé nastavení síťového instalačního prostředí. Spojuje a automatizuje mnoho navazujících úloh pro eliminaci potřeby přepínání několika různých příkazů a aplikací při nasazování nových systémů nebo změně již existujících. Cobbler může pomoci s provision, správou DNS a DHCP, aktualizací balíčků, správou napájení a dalšími [13].

3.4.1 Instalace

Instalace Cobbler je snadná a je dobře popsána v návodu v dokumentaci. Chtěl bych poukázat na velmi zajímavou a užitečnou funkcionalitu Cobbler. Jedná se o příkaz *cobbler check*. Tento příkaz provede kontrolu aktuální konfigurace Cobbler a poukáže na případné chyby. Navíc také představí návrhy a doporučení, co změnit v konfiguračním souboru a na co. Ve většině případů se jedná o doporučení, která nejsou nutná pro základní funkci Cobbler [14].

3.4.2 Použití

Obsluha Cobbler probíhá výhradně přes CLI. To by se dalo označit za lehkou nevýhodu. Uživatel si musí pamatovat příkazy nutné pro ovládání. Naštěstí jsou příkazy intuitivní a jejich výpisy jsou dobře strukturované a přehledné.

Navíc jsou v instalaci obsaženy i rozsáhlé a přehledné manuálové stránky, kde není problém najít strukturu příkazu, včetně vysvětlení jednotlivých částí.

Tato lehká nevýhoda se tedy může velmi brzy změnit na výhodu. Jakmile se uživatel naučí část příkazů, může velmi rychle pracovat. To vede k zefektivnění práce, protože velmi často může uživatele grafické rozhraní rozptylovat nebo nemusí být dobře navrženo.

3.4.3 Splnění požadavků

Cobbler je schopen spravovat služby jako DHCP, DNS a TFTP. Díky nim je možné vytvořit profily operačních systémů, které pak mohou být automaticky nainstalovány. Možné je tedy vytvořit profily pro mnoho distribucí Linux a zároveň lze vytvořit i profil pro Windows, který bude instalován z image.

Správu síťových rozhraní může zajišťovat služba DHCP. Editací konfiguračního souboru je pak možné alokovat nastavení pro specifická rozhraní ručně.

Cobbler neumožňuje konfiguraci RAID řadiče.

Cobbler má dle github 190 aktivních přispívajících uživatelů, ale aktivita repositáře je oproti ostatním projektům velmi malá.

Co se týče počtu otevřených a vyřešených závad, zde Cobbler také není příliš úspěšný. V době psaní bylo otevřeno 220 a uzavřeno 629 závad.

Projekt Cobbler je nejvíce podporován skupinami Eucalyptus, The Fedora Project, Spil Games, Puzzle ITC a Quancast. Kromě nich projekt sponzoruje i řada fyzických osob a společností.

3.5 MaaS

MaaS umožňuje administrátorovi chovat se k fyzickým serverům jako k virtuálním strojům. MaaS přetváří skupinu serverů v přizpůsobivý cloud.

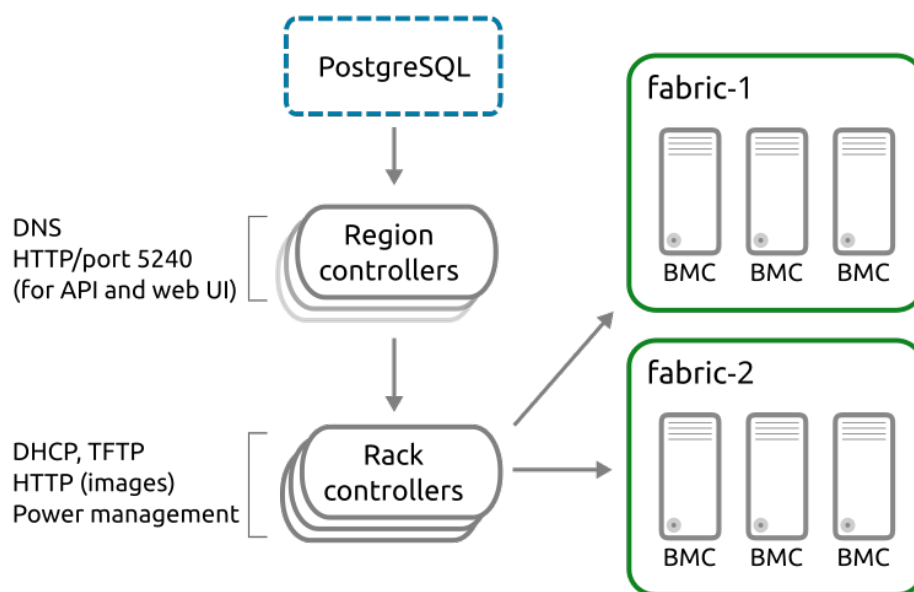
Stroje mohou rychle projít provision procesem a opět být rychle zničeny podobně jako u veřejných cloudů jako Amazon AWS, Google GCE a Microsoft Azure.

MaaS může sloužit jako samostatná služba využívající PXE a IPMI nebo může být integrován s jinými technologiemi. MaaS je navržen, aby velice dobře spolupracoval se službou Juju. Juju zajišťuje běžící služby na strojích a MaaS spravuje stroje samotné [15].

3.5.1 Instalace

Instalace MaaS je extrémně jednoduchá, stačí pomocí správce balíčku operačního systému MaaS nainstalovat, poté spustit inicializaci, během níž se vytvoří první uživatel. Poté už je MaaS k dispozici přes webové rozhraní na adrese serveru. Celý průběh instalace je popsán v dobře strukturovaném návodu. V rámci instalace jsou připraveny služby DNS, DHCP a BMC, které lze dále konfigurovat ve webovém rozhraní.

MaaS je hned po spuštění v *discovery* módu, kdy při spuštění serveru, který je třeba instalovat a má povolený PXE boot, nahraje na server dočasný image, který server zaregistruje v MaaS a následně se restartuje. Další konfigurace



Obrázek 3.3: Architektura MaaS

<https://docs.maas.io/media/intro-arch-overview.png>

pak probíhá ve webovém rozhraní. Díky tomu a jednoduchosti a přehlednosti rozhraní je práce s MaaS jednoduchá a uživatelsky přívětivá.

Konfigurace DHCP u sítí přímo připojených k serveru, je velmi jednoduchá. MaaS během instalace načte konfiguraci připojených sítí a automaticky je přidá do své správy. Díky tomu jsou sítě vidět ve webovém rozhraní a lze u nich povolit protokol DHCP. Následně lze přes toto webové rozhraní přidávat statické IP adresy, rezervovat rozsah adres, které nemají být přidělovány i sledovat aktuálně přidělené adresy [16].

3.5.2 Použití

MaaS nabízí čisté, přehledné a dobře organizované webové grafické rozhraní. Hledání a pohyb v něm je tedy nevyžaduje zbytečné hledání a je rychlé.

Instalace nových strojů je značně minimalistická. Požaduje pouze vyplnění jména, výběr domény, architektury, minimálního jádra, zónu, skupinu zařízení, do níž bude patřit, MAC adresy svých síťových rozhraní a typ správy napájení. U správy napájení pak vyžaduje další údaje jako IP adresu, MAC adresu a uživatelské jméno a heslo.

Díky vysoké míře automatizace a velkému množství podporovaného hardware je správa pomocí tohoto systému snadná a uživatelsky přívětivá.

3.5.3 Splnění požadavků

MaaS nabízí ve výchozí instalaci vlastní servery jako instalační zdroje pro operační systém Ubuntu. Kromě Ubuntu také nabízí možnost stáhnout image systému CentOS. Pro instalaci a použití jiných operačních systémů než Ubuntu a CentOS je nutné platit alespoň balíček Standard, který je naceněn na 5 amerických dolarů měsíčně za každý stroj registrovaný v MaaS.

Další možností instalace je použití vlastních zdrojů. Zde je potřeba specifikovat URL k instalačnímu image. Pro zajištění vyšší bezpečnosti lze použít GPG klíč uložený na serveru pro ověření stáhnutého image.

Ve výchozí konfiguraci používá MaaS DHCP pro všechna zařízení, která již nemají ručně přidělenou IP adresu. Přidělování IP adres zařízením není příliš přehledné a může být obtížné na pochopení.

MaaS neumožňuje nastavení hardware RAID řadiče.

Verzovací systém git a server github ukazují pro MaaS 46 aktivních uživatelů. To je poněkud málo v porovnání s dalšími porovnávanými projekty. Přesto je repositář celkem aktivní.

Stav nahlášených závad a jejich počet se mi nepodařilo najít.

Kromě toho, že MaaS je produktem společnosti Canonical, má také velké množství sponzorů a partnerů. Mezi nejznámější bych uvedl Quanta Cloud Technology, Amazon Web Services, IBM, Cisco, Emulex, Dell a Microsoft.

3.6 Spacewalk

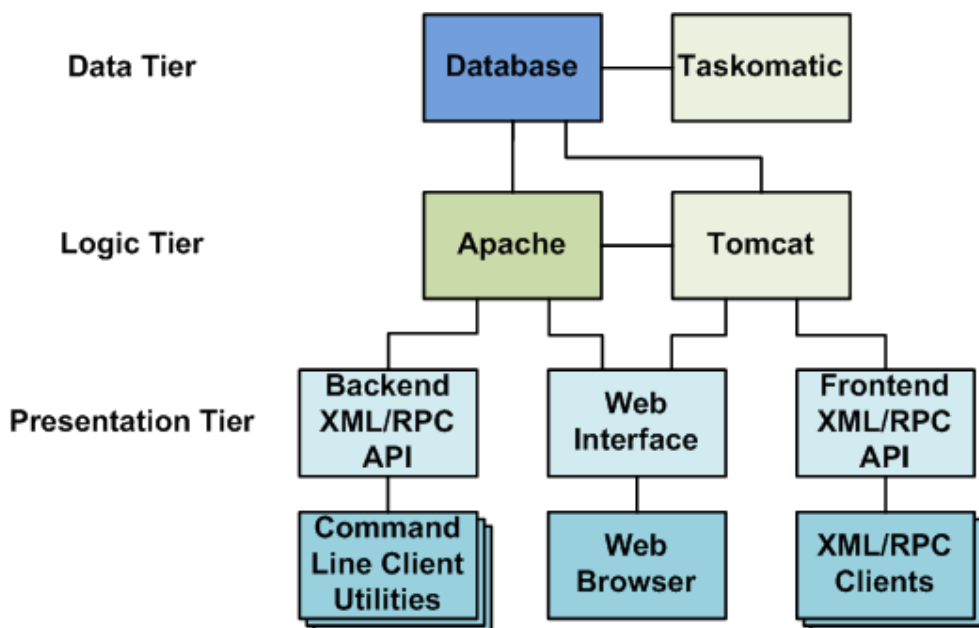
Spacewalk je open source řídicí řešení pro linuxové systémy. Umožňuje správu serverů během jejich životního cyklu, provision a nasazení konfiguračních souborů. Také obsahuje podporu virtualizace, podporující KVM a Xen.

Spacewalk server lze použít jako správce software obsahu systémů vycházejících ze systému Red Hat, jako jsou Fedora, CentOS a Scientific Linux. Díky tomu lze řídit nasazení aktualizací či nového softwaru skrze řízené systémy.

Součástí projektu Spacewalk je Spacewalk Proxy. Několik těchto proxy se může připojit k centrálnímu Spacewalk serveru a distribuovat obsah ze serveru do různých geografických oblastí snižujících zátěž na centrální server [17].

3.6.1 Instalace

Instalace Spacewalk je relativně jednoduchá, je ovšem složitější, potažmo obsahuje více kroků, než například instalace MaaS nebo Foreman. I tak je vcelku přímočará a jednoduchá. V průběhu instalace je vytvořen první uživatel a lokální certifikační autorita, včetně certifikátu. Opět je po instalaci k dispozici webové rozhraní [18].



Obrázek 3.4: Architektura Spacewalk

[https:](https://docs.oracle.com/cd/E92593_01/E85211/html/images/swk_arch.png)

[//docs.oracle.com/cd/E92593_01/E85211/html/images/swk_arch.png](https://docs.oracle.com/cd/E92593_01/E85211/html/images/swk_arch.png)

3.6.2 Použití

Rozhraní Spacewalk je velmi členité, ale neintuitivní pro nového uživatele. Navíc po instalaci je prostředí úplně „prázdné“. Veškerou konfiguraci je třeba ručně udělat a vzhledem k velkému množství položek, které je třeba vyplnit, se jedná o poměrně zdlouhavý proces. Výhodou tohoto přístupu je ovšem absolutní kontrola nad všemi aspekty správy.

Pro konfiguraci a instalaci serverů je pak třeba použít externí nástroj. Spacewalk samotný toto neumožňuje. Pro nové systémy je třeba použít systém Cobbler. Pro virtuální stroje či pro přeinstalaci existujícího stroje je nutné společně s Cobbler použít nástroj Koan. Jak Cobbler, tak Koan je třeba ovládat přes CLI, a to bez možnosti integrace do webového rozhraní Spacewalk.

Silnou stránkou Spacewalk je pak poměrně velmi dobrá a přehledná dokumentace.

3.6.3 Splnění požadavků

Systém Spacewalk je schopen splnit požadavky této práce, pouze pokud je zkombinován se systémem Cobbler.

Tím pádem se dá považovat za jakousi nadstavbu Cobbler. Lze tady in-

stalovat mnoho distribucí Linux i Windows pomocí předpřipraveného image.

Spacewalk je tedy schopen spravovat síťová rozhraní přes systém Cobbler. Navíc přes něj může spravovat služby jako DHCP, DNS či TFTP.

Tím, že Cobbler neumožňuje správu RAID řadiče, ani Spacewalk jí není schopen.

Spacewalk má na github uvedeno 149 přispěvatelů a ze záznamů lze vidět i jejich celkem slušnou aktivitu.

Seznam závad a stav řešení jsou k dispozici pouze po registraci.

3.7 Závěrečný výběr

Ze všech zkoumaných systémů vyšel jako nejhorší Spacewalk. Vyšel tak proto, že po něm požadované funkce je schopen poskytovat pouze při souběžném použití se systémem Cobbler.

Samotný Cobbler je tak díky nabízeným funkcím zcela dostačující, ale bohužel nepodporuje správu RAID řadiče. Navíc je poněkud limitován kvůli tomu, že nenabízí grafické rozhraní pro snadné monitorování.

Systém Ironic pak na rozdíl od Cobbler nabízí i možnost konfigurace RAID řadiče a při integraci s dalšími nástroji OpenStack pak i grafické rozhraní s přehledným monitorováním. Jeho velkou nevýhodou pak je náročná instalace. Sice je k dispozici velmi dobrá dokumentace, ale i přesto se jedná o velmi náročný proces.

MaaS nenabízí konfiguraci RAID řadiče, ovšem je zase podstatně jednodušší a rychlejší na instalaci a i samotné použití je velmi jednoduché. Zde je největší nevýhodou obtížná obsluha nastavování síťových rozhraní.

Nakonec je zde systém Foreman. Ten opět neumožňuje obsluhu RAID řadiče. Co se týče složitosti instalace, je u něj tento proces o něco složitější než u MaaS a Cobbler, ale zároveň je daleko snadnější než u Ironic a Spacewalk.

Ve výsledku jsem pak na základě tohoto srovnání zvolil systém Foreman, i přestože nesplňuje požadavek na ovládání RAID řadiče. Mé rozhodnutí bylo po konzultaci schváleno vedoucím práce. Pro ovládání RAID řadiče je možné použít sadu skriptů od společnosti HP, výrobce uvažovaného RAID řadiče.

Důvody pro tuto volbu jsou následující: Poměrně snadná instalace a velmi snadný způsob provádění změn v konfiguraci. Jednoduchá správa služeb jako DHCP či BMC. Snadné přidávání nových i existujících strojů. Velmi snadná správa síťových rozhraní těchto strojů. Základní možnosti monitorování již ve výchozí instalaci. Přehledné webové grafické rozhraní i CLI. Snadná škálovatelnost i přes několik různých sítí. A na závěr velké množství předpřipravených šablon pro rychlé nasazení a rozsáhlé možnosti modifikací.

3. ANALÝZA A POROVNÁNÍ

V další kapitole se budu věnovat instalaci a konfiguraci Foreman v prostředí datového centra.

Realizace

Tato kapitola se bude zabývat instalací zvoleného projektu Foreman a konfigurací jednotlivých služeb vyžadovaných pro splnění kladených požadavků. Dále bude v této kapitole probrána příprava nastavení operačního systému, k němu potřebných instalačních šablon a instalačních zdrojů.

V sekci 4.4 ukáží standardní postup přidání dvou nových klientů do systému. Na nich pak provedu instalaci jedné distribuce Linux a Windows.

4.1 Popis projektu Foreman

Instalace Foremana vždy obsahuje centrální instanci, která je zodpovědná za poskytování webového grafického rozhraní, konfigurace uzlů, prvotní konfigurační soubory klientů atd. . .

Pokud má ovšem Foreman zajišťovat automatickou instalaci, je třeba provést další operace pro úplné zautomatizování procesu. K těmto účelům slouží Smart Proxy, která spravuje vzdálené služby a běžně je instalována s Foremanem pro správu TFTP, DHCP, DNS, Puppet, Puppet CA, Ansible, Salt, Chef a dalších.

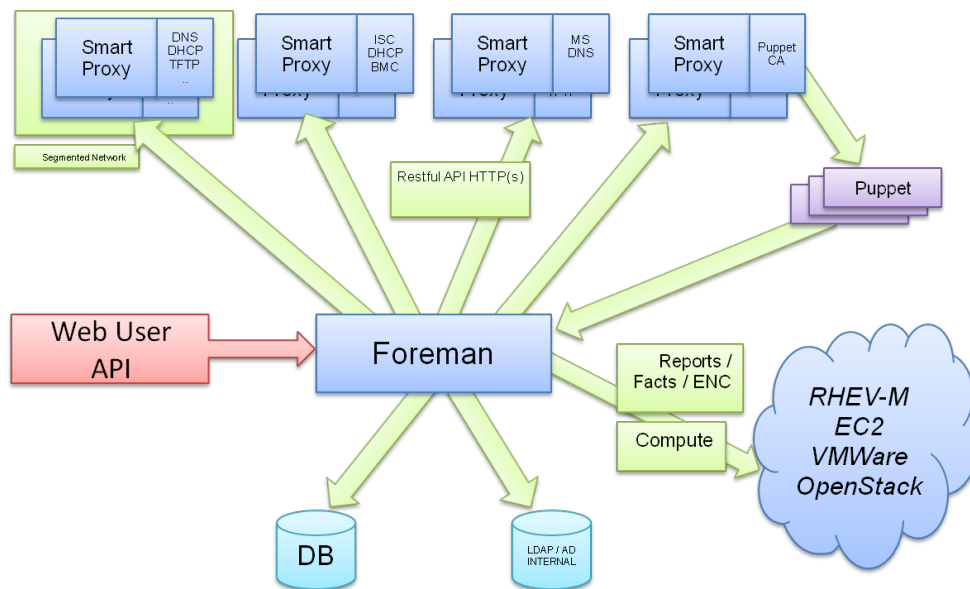
Schéma nasazení projektu Foreman je vyobrazeno na obrázku 4.1.

4.2 Instalace Foreman serveru

Tato sekce bude vcelku věrně kopírovat návod k instalaci dostupný v dokumentaci projektu Foreman. Hlavním rozdílem bude použití interaktivního instalátoru, narozdíl od použití výchozí instalace uváděné v návodu. Začnu popisem požadavků pro instalaci v sekci 4.2.1, v sekci 4.2.2 provedu základní nastavení používaných služeb jako DHCP, DNS a TFTP přímo v instalátoru. Následovat bude konfigurace podsítí a domén.

V sekci 4.3 se budu zabývat přípravou instalačních médií pro operační systémy, šablonami a tabulkami pro rozdělení disků.

Obrázek 4.1: Schéma nasazení projektu Foreman



Na závěr kapitoly ukáží postup přidání nového klienta a průběh jeho instalace v sekci 4.4

4.2.1 Požadavky na instalaci

Foreman podporuje následující distribuce Linux pro svou instalaci:

- Red Hat Enterprise Linux 7;
- CentOS, Scientific Linux nebo Oracle Linux 7;
- Ubuntu 16.04 (Xenial);
- Ubuntu 18.04 (Bionic);
- Debian 9 (Stretch).

Foreman je možné nainstalovat i na jiných operačních systémech, než které jsou uvedeny, ale je třeba pak použít jiné metody instalace.

Dále je potřeba Puppet verze 4 nebo vyšší. Puppet je možno nainstalovat z repozitářů operačního systému nebo repozitáře PuppetLabs.

Výchozí instalace Foreman včetně Puppet Server vyžaduje alespoň 4 GB operační paměti a minimálně 2 GB prostoru na disku.

Webové rozhraní Foreman podporuje všechny hlavní prohlížeče. Konkrétně se jedná o verze:

- Google Chrome 54 nebo vyšší,

- Microsoft Edge,
- Microsoft Explorer 10 nebo vyšší,
- Mozilla Firefox 49 nebo vyšší.

K dispozici je virtuální stroj VMware se 4 GB operační paměti, disk o kapacitě 7 GB a nainstalovaný operační systém Ubuntu 16.04. Kromě nenainstalovaného Puppetu jsou splněny všechny požadavky pro instalaci.

Instalaci Foreman zahájím instalací Puppet a všech nutných závislostí pro Foreman installer. Před spuštěním installeru je ale třeba udělat několik úprav na stroji. Je potřeba upravit záznam v `/etc/hosts` pro tento server aby při použití příkazu `ping hostname` vrátil IP adresu serveru, nikoliv 127.0.0.1. Poté je nutné zkontrolovat nastavení firewallu. V tomto případě je potřeba povolit porty zmíněné v tabulce 4.1.

Tabulka 4.1: Tabulka portů používaných Foremanem

Port	Protokol	Vyžadován pro
53	TCP a UDP	DNS server
67, 68	UDP	DHCP server
69	UDP	TFTP server
80, 443	TCP	HTTP a HTTPS
8140	TCP	Puppet Master
8443	TCP	Smart Proxy

4.2.2 Konfigurace služeb

Server je připojen do sítí 10.1.82.0/24 a 10.1.104.0/24. Síť 10.1.82.0 slouží jako management síť a adresy na ní jsou přidělovány manuálně. Primární funkcí této sítě je komunikace pomocí IPMI se základovými deskami serverů. V síti 10.1.104.0 jsou připojené síťové karty serverů. Ty slouží k přístupu na internet. Tato síť je označena jako datová a správu adres bude zajišťovat služba DHCP.

Ve Foreman installeru proto vyberu `foreman-proxy`. Dále vyberu bod `dhcp`, který nastavím na `true` pro povolení služby DHCP. DHCP bude naslouchat žádostem na zařízení `ens192`, které je připojeno do datové sítě. Proto nastavím na toto zařízení možnost `dhcp-listen-on`. Adresu DHCP serveru ponechám na místním serveru, jelikož služba DHCP bude fungovat na stejném serveru jako Foreman.

Další službou, kterou budu konfigurovat, je služba DNS. Opět tato služba není nutná v management síti, tudíž znovu stačí nastavit DNS pouze na jedno zařízení, na kterém bude naslouchat. Stále jsem v sekci `foreman-proxy` Foreman installeru. Zde nastavím hodnotu `dns` na `true` a hodnotu `dns-listen-on` na `ens192`. Adresu DNS serveru opět ponechávám na místním serveru ze stejného důvodu jako u DHCP. Installer si načel název domény z `/etc/hosts`,

tudíž není třeba vyplňovat položku *dns-lookup-domain*. Musí se ale vyplnit položka *dns-reverse-zones*. Zde nastavím hodnotu *104.1.10.in-addr.arpa*. Ta zajistí zpětné vyhledávání jmen serverů pomocí jejich IP adresy.

Nyní nastavím službu TFTP. Opět ji zapnu přepsáním hodnoty *tftp* na *true*. Velmi důležité je nyní nastavení *tftp-listens-on*, protože velká část automatických instalačních nástrojů nepoužívá protokol https, ale pouze http. Proto tuto možnost nastavím na *both*, aby všechny instalátory mohly využívat TFTP. Poslední hodnotou, kterou musím nastavit, je *tftp-servername*. Tu nastavím na *10.1.104.207* neboli adresu Foreman serveru v datové síti.

Poslední důležitou službou, kterou budu konfigurovat v installeru, je služba BMC, která pomocí IPMI bude obsluhovat základové desky serverů v management síti. Tuto službu stačí pouze aktivovat nastavením *bmc* na *true*.

Nyní již stačí celou tuto konfiguraci uložit a spustit [19].

4.3 Konfigurace pro automatickou instalaci

V této sekci popíši a ukážu postup přípravy podkladů pro automatickou instalaci klienta. Začnu přípravou instalačního média, následovaného tvorbou operačního systému ve webovém rozhraní Foreman. Následně se budu věnovat partition tabulkám pro vytvářený operační systém a tomu, jak je upravovat. Sekci završím krátkým popisem provisioning šablon.

4.3.1 Instalační médium

Prvním krokem pro přípravu operačního systému ve Foremanu je příprava média, z něhož bude instalován. Tudíž v záložce *Hosts – Installation Media* kliknu na tlačítko *Create Medium*.

Otevřená karta obsahuje 2 povinná pole. První je pole *Name*, jméno. Zde je dobré zadat jméno, ze kterého bude zřejmé o jaký zdroj instalačních dat se jedná. Například zda se jedná o data uložená na serveru v lokální síti, image operačního systému nebo soubory na serveru tvůrce operačního systému.

Druhou položkou je *Path*, cesta. Touto položkou nastavíme cestu k datům. Je zde možné zadat URL nebo funkční NFS server. Navíc mohou být v cestě použiti takzvané placeholders, proměnné hodnoty, které se nahradí hodnotami obsaženými u operačního systému. Takto lze snadno použít jedno instalační médium k poskytování více verzí téhož operačního systému.

Poslední a jedinou nepovinnou položkou tvorby média je výběr *Operating System Family*, rodiny operačního systému. Díky tomu je možné v pozdějších krocích filtrovat položky a usnadnit tak nastavení operačního systému. Kliknutím na tlačítko *Submit* dojde k vytvoření instalačního média.

4.3.2 Operační systém

Tvorba operačního systému je rozdělena na 2 kroky. Prvním krokem je pojmenování operačního systému a zadání major verze. V tomto kroku je též možné nastavit minor verzi operačního systému, jeho popis, přiřazení do rodiny, volbu používané funkce pro hash hesla uživatele root a přiřazení architektury. Ještě je možné v tomto kroku přiřadit partition tabulky a instalační média. Po dokončení tohoto kroku je třeba vše uložit kliknutím na tlačítko *Submit* a následuje krok druhý.

V druhém kroku je třeba přiřadit provisioning šablony k operačnímu systému. To provedu klepnutím na jméno provisioning šablony v *Hosts – Provisioning Templates*. Po otevření šablony vyberu záložku *Association*. Zde můžu šablonu přiřadit k nově vytvořeným či již existujícím operačním systémům. Pro uložení klepnu na tlačítko *Submit*.

Nyní pokračuji zpět v *Host – Operating Systems* v nastavování operačního systému z kroku 1. V záložce *Templates* se nyní objevilo několik polí. Ani jedno není povinné, ale bez dodání šablon neproběhne následná instalace, jak je potřeba. V těchto polích nastavím šablony pro jednotlivé části procesu automatické instalace. Po zvolení šablon opět uložím kliknutím na tlačítko *Submit*.

Tímto je dokončena tvorba operačního systému pro automatickou instalaci ve webovém rozhraní Foreman.

4.3.3 Partition tabulky

Další konfigurovatelnou položkou jsou partition tabulky. Ty určují, jak bude rozdělen disk při instalaci operačního systému na server. Pro potřeby práce stačí výchozí šablony poskytnuté při instalaci Foreman, proto se jimi nebudu dále zabývat.

4.3.4 Provisioning šablony

Provisioning šablony jsou šablony v jazyce EBR, ze kterých je při přechodu hosta do build režimu vygenerována výsledná šablona obsahující vyplněné proměnné podle parametrů vyplněných v konfiguraci hosta.

Šablony lze libovolně upravovat, a pokud je šablona přiřazena k některému z hostů, lze zobrazit její náhled. Díky tomu je možné najít případné chyby v konfiguraci ještě před spuštěním celého provision procesu.

4.3.5 Host groups

Host groups slouží k seskupování hostů do skupin na základě stejných parametrů. V Host group lze nastavit prostředí, v jakém budou hosti nasazeni, který server bude sloužit jako Puppet Master a Puppet CA. Dále lze nastavit,

jaké Puppet třídy budou hosti využívat, v jaké síti a doméně budou připojeni a jaký operační systém budou využívat.

Díky rozdělení do těchto skupin je pak snadnější příprava nových hostů i jejich správa, protože velkou část parametrů lze spravovat centrálně z jednoho místa.

Pro účely testování jsem vytvořil dvě skupiny. První skupinou je *provision-test-centos*. Tu používám pro testování instalace operačního systému CentOS 7 na servery. Druhou skupinou pak je *provision-test-windows*. Jak z názvu vyplývá, tato skupina je určena pro testování instalace Windows.

Obě skupiny mají kromě operačního systému stejná nastavení. Využívají Puppet třídu NTP pro synchronizaci času, jsou zapojené do stejné sítě a domény a využívají server *ubuntu.provision.local* jako Puppet Master a Puppet CA.

4.4 Automatická instalace

V sekci projdu postup přípravy nového klienta pro automatickou instalaci. Začnu pojmenováním klienta a pravidly řídicími pojmenovávání, volbu Host Group, Enviroment, Puppet Master a Puppet CA.

Poté se přesunu k výběru architektury, operačního systému, zdroje instalace, partition tabulce, nastavení hesla pro uživatele root a přiřazení provisioning šablon.

Dále pak popíši a projdu postup přidávání a nastavení síťových rozhraní pro nového klienta. Nakonec vše završím popisem přidávání Puppet tříd.

Přidání nového klienta provedu stiskem odkazu *Create Host* v záložce *Hosts*.

Prvním parametrem k vyplnění je *Name*, jméno. Jméno může obsahovat pouze malá písmena, čísla, pomlčky a tečky.

Další je výběr skupiny do jaké má klient patřit, pole *Host Group*. Zde je možné vybrat již existující a nastavenou klientskou skupinu pro zjednodušení procesu. Ostatní položky na této kartě jsem popsal v sekci 4.3.

Díky použití *Host Group* zbývá v druhé kartě, kartě *Operating System*, zaškrtnout položku *Build Mode*. Ta určuje zda, klient po vytvoření má rovnou přejít k instalaci, či vyčkat na manuální pokyn.

Dále je nevyplněné textové pole *Custom Partition Table*. Toto pole se vyplňuje pouze tehdy, pokud je potřeba použít rozdílnou partition tabulku, než jaká je nastavena v *Partition Table*. Dále je potřeba vyplnit heslo pro uživatele root. Heslo musí být alespoň 8 znaků dlouhé. Poslední zbývá tlačítko *Resolve* u *Provisioning Templates*. Po stisknutí tlačítka Foreman dodá provisioning šablony podle zvoleného operačního systému.

Poslední kartou vyžadující zvýšenou pozornost je karta *Interfaces*. Zde se nastavují jednotlivá síťová rozhraní klienta a opět díky použití *Host Group* je

velká část údajů předvyplněna. Konkrétně se jedná o položky *Domain*, *Subnet*, a pokud má síť nakonfigurované DHCP, pak i adresu.

Nastavování BMC rozhraní je velmi podobné nastavování klasického síťového rozhraní. Jediné rozdíly představují jiný typ rozhraní, položka *Provider*, která je povinná a určuje, jakým způsobem bude Foreman server komunikovat s BMC zařízením, a uživatelské jméno a heslo pro zařízení.

Kartu *Puppet Classes* není třeba znovu popisovat, byla již popsána v sekci 4.3.

V kartě *Parameters* je možné doplnit specifické parametry, jak pro Puppet třídy, tak i pro klienta, nebo nahradit hodnotu nějakého globálního parametru. Jelikož v práci není třeba tuto záložku používat nebudu se jí více věnovat.

Poslední kartou je karta *Additional Information*. Tato karta slouží, jak z názvu plyne, k doplnění informací o klientovy. Lze zde přiřadit vlastníka klienta. To je užitečné při propojení Foreman se službami LDAP nebo Windows Active Directory.

Dále je zde možnost vyřadit či přidat klienta do reportingu. Tato možnost je ve výchozím stavu zaškrtnuta a umožňuje tak sbírat údaje ze stroje po dokončení instalace.

Položka *Hardware Model* slouží k identifikaci stroje.

Poslední položka *Comment* jak z názvy vyplývá je prostor pro komentáře ke klientovy.

Tímto jsem popsal, jak vypadá proces přidávání klienta a jeho konfigurace do správy Foreman. Pro dokončení procesu stačí stisknout tlačítko *Submit*.

Závěr

V práci jsem se věnoval problematice bare metal provisioning a službám používaným při tomto procesu a jeho průběhu.

Dále jsem analyzoval volně dostupné open source projekty poskytující právě službu bare metal provisioning. Po projektech bylo požadováno, aby dokázaly nakonfigurovat síťová rozhraní serveru HPE a SuperMicro, k serveru přidělený RAID řadič a nainstalovat alespoň jednu distribuci Linux a jednu distribuci Windows.

Popisoval jsem, jak snadno se projekty instalují, konfigurují a používají, jaké mají možnosti obsluhy a zda splňují, či nesplňují požadavky na ně kladené.

Po popisu všech vybraných projektů jsem v závěru kapitoly 3 v sekci 3.7 zvolil výsledný projekt, tím se stal projekt Foreman, a zdůvodnil jsem jeho volbu.

Poté jsem v kapitole 4 popsal projekt Foreman a v dalších sekcích kapitoly jsem provedl jeho instalaci a konfiguraci v datovém centru v Jihlavě. Po dokončení instalace a konfigurace služeb jsem popsal proces konfigurace instalačního média a proces automatické instalace za použití webového grafické rozhraní Foreman.

Na práci by šlo navázat rozšířením funkcionality Foreman přidáním podpory konfigurace RAID řadiče.

Seznam použitých zkratek

OS	Operační systém
RAID	Redundant array of independent disks
HPE	Hewlett Packard Enterprise
PXE	Pre execution environment
DHCP	Dynamic host configuration protocol
BIOS	Basic input-output system
IP	Internet protocol
NBP	Network bootstrap program
TFTP	Trivial file transfer protocol
IPMI	Intelligent platform management interface
NFS	Network file system
CLI	Command line interface
REST	Representational state transfer
API	Application programming interface
MAC	Media access control
JSON	JavaScript object notation
BMC	Baseboard management controller
URL	Uniform resource locator
KVM	Kernel-based virtual machine

A. SEZNAM POUŽITÝCH ZKRATEK

HP Hewlett Packard

TCP Transmission control protocol

UDP User datagram protocol

HTTP Hypertext transfer protocol

HTTPS Hypertext transfer protocol secure

EBR Embedded Ruby

NTP Network time protocol

CA Certification authority

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
src	adresář se zdroji práce
_ images	adresář s obrázky
_ BP-Skruzny-Petr-B182.tex	zdrojová forma práce ve formátu \LaTeX
_ bibliography.bib	zdrojová forma citací
_ FITthesis.cls	zdroj šablon pro soubor thesis.tex
_ csn690.bst	zdroj formátování citací
text	text práce
_ BP-Skruzny-Petr-B182.pdf	text práce ve formátu PDF