



## ZADÁNÍ BAKALÁ SKÉ PRÁCE

<b>Název:</b>	Automatická instalace linuxového serveru a základní konfigurace služeb
<b>Student:</b>	Jakub Sekera
<b>Vedoucí:</b>	Ing. Zden k Muziká , CSc.
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Informa ní technologie
<b>Katedra:</b>	Katedra po íta ových systém
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

Navrhn te a vytvo te nástroj, který umožní vytvo it virtuální linuxový server ve form OVA/OVF souboru tak, aby byl p ípravený k okamžitému nasazení do produk ního prost edí. Automatická instalace musí nastavit vybrané parametry serveru (HW nastavení, IP adresa, hostname atd.) a nainstalovat požadované služby a jejich konfiguraci. Vstupní rozhraní ve formátu XML musí umožnit nastavovat jednotlivé parametry bez hlubších znalostí uživatele. V testovacím prost edí konfiguraci ov te.

### Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Róbert Lórencz, CSc.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
d kan

V Praze dne 8. listopadu 2016



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA POČÍTAČOVÝCH SYSTÉMŮ



Bakalárska práca

## **Automatická inštalácia linuxového servera a základná konfigurácia služieb**

*Jakub Sekera*

Vedúci práce: Ing. Zdeněk Muzikář, CSc.

7. mája 2017



---

## Pod'akovanie

Rád by som sa poďakoval vedúcemu práce Ing. Zdeňkovi Muzikářovi, CSc. za cenné rady, pripomienky a zaujímavé postrehy, ktoré mi pomohli pri vypracovaní mojej bakalárskej práce. Ďalej by som chcel poďakovať Bc. Vladimírovi Rajchartovi, ktorý mi pomohol s vytvorením témy a Ing. Tomášovi Kadlecovi, ktorý mi pomohol s technológiami, pomocou ktorých sa mi podarilo vytvoriť túto prácu.



---

# Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov. V súlade s ustanovením § 46 odst. 6 tohoto zákona týmto udeľujem bezvýhradné oprávnenie (licenciu) k užívaniu tejto mojej práce, a to vrátane všetkých počítačových programov ktoré sú jej súčasťou alebo prílohou a tiež všetkej ich dokumentácie (ďalej len „Dielo“), a to všetkým osobám, ktoré si prajú Dielo užívať.

Tieto osoby sú oprávnené Dielo používať akýmkoľvek spôsobom, ktorý nezníži hodnotu Diela, a za akýmkoľvek účelom (vrátane komerčného využitia). Toto oprávnenie je časovo, územne a množstevne neobmedzené. Každá osoba, ktorá využije vyššie uvedenú licenciu, sa však zaväzuje priradiť každému dielu, ktoré vznikne (čo i len čiastočne) na základe Diela, úpravou Diela, spojením Diela s iným dielom, zaradením Diela do diela súborného či zpracovaním Diela (vrátane prekladu), licenciu aspoň vo vyššie uvedenom rozsahu a zároveň sa zaväzuje sprístupniť zdrojový kód takého diela aspoň zrovnateľným spôsobom a v zrovnateľnom rozsahu ako je zprístupnený zdrojový kód Diela.

V Prahe 7. mája 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Jakub Sekera. Všetky práva vyhradené.

*Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.*

### **Odkaz na túto prácu**

Sekera, Jakub. *Automatická inštalácia linuxového servera a základná konfigurácia služieb*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Práca je zameraná na to ako zjednodušiť a zefektívniť inštalácie a konfigurácie virtuálnych linuxových serverov postavených na distribúciách Ubuntu a Debian. Cieľom je to, aby celkový čas strávený pri inštalácii a následnej konfigurácii servera bol čo najkratší a umožnil človeku nastaviť jednotlivé parametre servera (už pred samotnou inštaláciou) bez hlbších znalostí daného systému. Jeden z vedľajších cieľov je aj ten, aby inštalované servery mali rovnakú štruktúru prostredia (konfiguračné súbory, skripty, užívateľov atď.) a človek, ktorý bude tento systém ďalej spravovať mal uľahčenú správu systému. Cieľom práce je vytvoriť nástroj, pomocou ktorého je možná automatická inštalácia virtuálneho linuxového servera a základná konfigurácia vybraných služieb.

**Kľúčová slova** automatická inštalácia, linuxový server, konfigurácia služieb, Debian, Ubuntu, správa systému

---

# Abstract

The Bachelor thesis focused on how to simplify and make more efficient installations and configurations of virtual Linux servers, which are based on distributions like Ubuntu and Debian. The main goal is that overall time, which is spend by installation and following by configuration of server, will be as short as possible and allows user to set parameters of server (before installation) without deeper knowledge of the system. One of the goals is, that the installed servers will have same structure of environment (configuration files, scripts, users etc.) and user, who will administrated this system will have easier management system. The aim of this thesis is to create a tool that allows automatic installation of virtual Linux server and fundamental configuration of selected services.

**Keywords** automatic installation, linux server, service configuration, Debian, Ubuntu, system management

---

# Obsah

Úvod	1
<b>1 Cieľ práce</b>	<b>3</b>
<b>2 Teoretický popis virtualizácie</b>	<b>5</b>
2.1 Virtualizácia a jej typy . . . . .	5
2.2 Výhody serverovej virtualizácie . . . . .	8
2.3 Nevýhody serverovej virtualizácie . . . . .	9
2.4 Virtuálny stroj . . . . .	10
<b>3 Hypervízor, virtualizačný softvér a konfiguračné nástroje</b>	<b>11</b>
3.1 Hypervízor . . . . .	11
3.2 Virtualizačný softvér . . . . .	15
3.3 Konfiguračné nástroje . . . . .	18
<b>4 Prehľad a analýza existujúcich riešení</b>	<b>19</b>
4.1 Spacewalk . . . . .	19
4.2 FAI . . . . .	21
4.3 Cobbler . . . . .	22
4.4 Foreman . . . . .	23
<b>5 Návrh riešenia</b>	<b>27</b>
5.1 Požiadavky systému . . . . .	27
5.2 Možnosti užívateľských rozhraní . . . . .	28
5.3 Architektúra aplikácie . . . . .	29
5.4 Nastavovanie parametrov a inštalácia služieb . . . . .	31
5.5 Použité virtualizačné prostredia a inštalované operačné systémy	32
<b>6 Popis a implementácia aplikácie</b>	<b>35</b>
6.1 Použité nástroje a prostredia . . . . .	35

6.2	Použité programovacie jazyky . . . . .	35
6.3	Implementácia aplikácie . . . . .	36
6.4	Návrh a štruktúra GUI aplikácie . . . . .	37
6.5	Vstupný súbor vo formáte XML . . . . .	38
6.6	Packer . . . . .	39
6.7	Ansible . . . . .	40
6.8	Obsah a tvorba balíčka vo formáte deb . . . . .	45
<b>7</b>	<b>Testovanie programu a inštalovaných služieb</b>	<b>47</b>
7.1	Priebeh testovania . . . . .	47
7.2	Vyhodnotenie testovania . . . . .	48
	<b>Záver</b>	<b>49</b>
	<b>Literatúra</b>	<b>51</b>
	<b>A Zoznam použitých skratiek</b>	<b>55</b>
	<b>B Uživateľská príručka</b>	<b>57</b>
B.1	Popis aplikácie . . . . .	57
B.2	Požiadavky . . . . .	57
B.3	Inštalácia . . . . .	58
B.4	Spustenie a používanie aplikácie . . . . .	58
B.5	Popis a funkcie grafických objektov aplikácie . . . . .	60
B.6	Popis a prístup k nainštalovaným službám . . . . .	60
B.7	Odinštalovanie aplikácie . . . . .	62
	<b>C Grafické rozhranie aplikácie</b>	<b>63</b>
	<b>D Obsah priloženého DVD</b>	<b>71</b>

---

## Zoznam obrázkov

2.1	Tradičná verzus virtuálna architektúra [1] . . . . .	6
2.2	Diagram virtualizácie [2] . . . . .	6
3.1	VMware ESX Server architektúra [3] . . . . .	12
3.2	Xen Project architektúra [4] . . . . .	15
4.1	Časť webového rozhrania Spacewalk [5] . . . . .	20
4.2	Úvodná obrazovka nástroja FAI [6] . . . . .	21
4.3	Webové rozhranie nástroja Cobbler [7] . . . . .	23
4.4	Webové rozhranie nástroja Foreman [8] . . . . .	24
5.1	Štruktúra aplikácie . . . . .	30
C.1	Úvodná obrazovka aplikácie . . . . .	64
C.2	XML dialóg . . . . .	65
C.3	V poradí 2. obrazovka aplikácie . . . . .	66
C.4	Dialóg služby Zabbix . . . . .	67
C.5	V poradí 3. obrazovka aplikácie . . . . .	68
C.6	V poradí 4. obrazovka aplikácie . . . . .	69



---

# Zoznam tabuliek

5.1	Porovnanie definovania parametrov virtuálneho stroja . . . . .	33
-----	--	----





---

# Úvod

Každá väčšia firma (nielen IT) má zvyčajne svoje vlastné dátové centrum, v ktorom má postavenú svoju vlastnú IT infraštruktúru alebo využíva produkty firiem, ktoré poskytujú služby v oblasti dátových centier. Medzi najviac využívané služby patrí:

- *Serverhosting*<sup>1</sup>
- *Outsourcing*<sup>2</sup> IT infraštruktúry

Servery (fyzické, ale aj virtuálne) sú veľmi významnou súčasťou dátových centier. Na týchto serveroch sú prevádzkované vyššie uvedené služby, a preto je veľmi dôležité zaoberať sa ich inštaláciou, konfiguráciou a následným monitorovaním ich stavu. Inštalácia a konfigurácia servera na ktorom budú prevádzkované jednotlivé služby (napr. monitorovanie rôznych zariadení, služba dátového úložiska, *webhosting*<sup>3</sup>) je veľmi dlhý proces.

Pri ručnej inštalácii serverov, ktorá sa veľmi často opakuje len s malými zmenami môžu vznikáť rôzne chyby. Pri následnej konfigurácii služieb, ktorá môže trvať aj niekoľko hodín môžu vznikáť nekonzistentné konfigurácie naprieč servermi, ktoré sú určené na rovnaké použitie. Inštalácia nového servera ručne, bez automatickej inštalácie a s malými konfiguračnými zmenami je strata času. V dnešnom uponáhľanom a rýchlo sa rozvíjajúcom svete je veľmi dôležité to, aby sa veci robili, čo najrýchlejšie, najefektívnejšie a čo najjednoduchšie. Z vyššie uvedených dôvodov som sa rozhodol pre výber témy, ktorá sa zaoberá automatickou inštaláciou a konfiguráciou linuxových virtuálnych serverov.

---

<sup>1</sup>prenájom alebo umiestnenie počítačového servera v priestoroch dátového centra

<sup>2</sup>uskutočňovanie určitých činností pomocou externých zdrojov

<sup>3</sup>služba prenájmu „vlastného miesta na Internete“ určená na prevádzku www stránok, e-mail a pod.



---

## Cieľ práce

Cieľom tejto práce je analýza, návrh a implementácia systému, ktorý umožní automaticky nainštalovať linuxový virtuálny server a vybrané služby, ktoré tiež automaticky nakonfiguruje. Nástroj bude primárne určený pre užívateľov, ktorí nemajú hlboké znalosti v danej problematike. Títo užívatelia avšak potrebujú mať minimálne prehľad o základných činnostiach jednotlivých komponentov daného systému. V neposlednom rade bude určený aj pre skúsených správcov systémov. Pomocou nástroja bude možná jednoduchšia a časovo menej náročná inštalácia a následná konfigurácia virtuálnych serverov. Tvorba programu bude inšpirovaná systémami, ktoré sa podobnou problematikou zaoberajú. Definované budú aj služby a parametre, ktoré sa budú na cieľový virtuálny stroj inštalovať alebo nastavovať.

Ďalším z cieľov bude implementovať jednoduché vstupné rozhranie programu vo formáte XML súbora, pomocou ktorého budú užívatelia nastavovať jednotlivé parametre virtuálneho stroja, ktorý sa bude následne automatickou inštaláciou vytvárať. Grafické rozhranie programu následne umožní tieto údaje zo súbora vo formáte XML načítať a požiada užívateľa o vyplnenie ďalších doplňujúcich parametrov, ktoré sú potrebné k inštalácii virtuálneho stroja.

Program umožní vytvoriť virtuálny stroj založený na operačnom systéme linuxovej distribúcie Debian alebo Ubuntu pre *hypervízor*<sup>4</sup> VirtualBox alebo KVM. Cieľom tejto práce bude otestovať tento vytvorený virtuálny stroj v testovacom prostredí a overiť konfigurácia jednotlivých parametrov a služieb.

---

<sup>4</sup>počítačový softvér, firmvér alebo hardvér, ktorý vytvára a spúšťa virtuálne stroje



## Teoretický popis virtualizácie

Táto kapitola je venovaná virtualizácií, typom virtualizácie a virtualizačným technikám. Ďalšia časť tejto kapitoly je venovaná výhodám a nevýhodám serverovej virtualizácie a popisom virtuálneho stroja.

### 2.1 Virtualizácia a jej typy

Zdroj [9] uvádza, že virtualizácia systému alebo častí systému ako je procesor, pamäť, alebo I/O zariadenia na určitej abstraktnej úrovni mapuje rozhranie a viditeľné zdroje systému do rozhrania a zdrojov základného, prípadne iného reálneho systému. V dôsledku toho sa reálny systém javí ako odlišný virtuálny systém alebo dokonca ako odlišné viacnásobné virtuálne systémy. Systém virtuálneho stroja poskytuje kompletne prostredie, v ktorom dokáže operačný systém navzájom spolupracovať s procesmi, ktoré patria rôznym užívateľom.

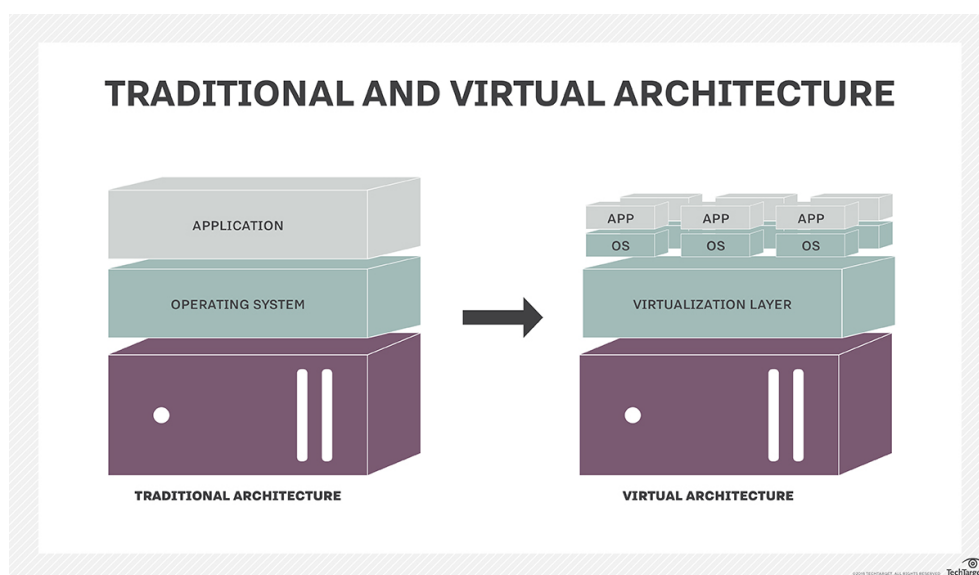
Koncept virtualizácie nemusí byť aplikovaný len na subsystemy ako sú disky, ale môže byť aplikovaný aj na celý stroj. K implementácií virtuálneho stroja vývojári pridávajú fyzickému stroju softvérovú vrstvu, ktorá pomáha virtuálnemu stroju využiť hardvér fyzického stroja. Týmto spôsobom dokáže virtuálny stroj zanedbať obmedzenia spôsobené hardvérovou kompatibilitou a nedostatkom hardvérových zdrojov. Porovnanie tradičnej a virtuálnej architektúry je možné vidieť na obrázku 2.1 a diagram virtualizácie je na obrázku 2.2.

#### 2.1.1 Typy virtualizácie

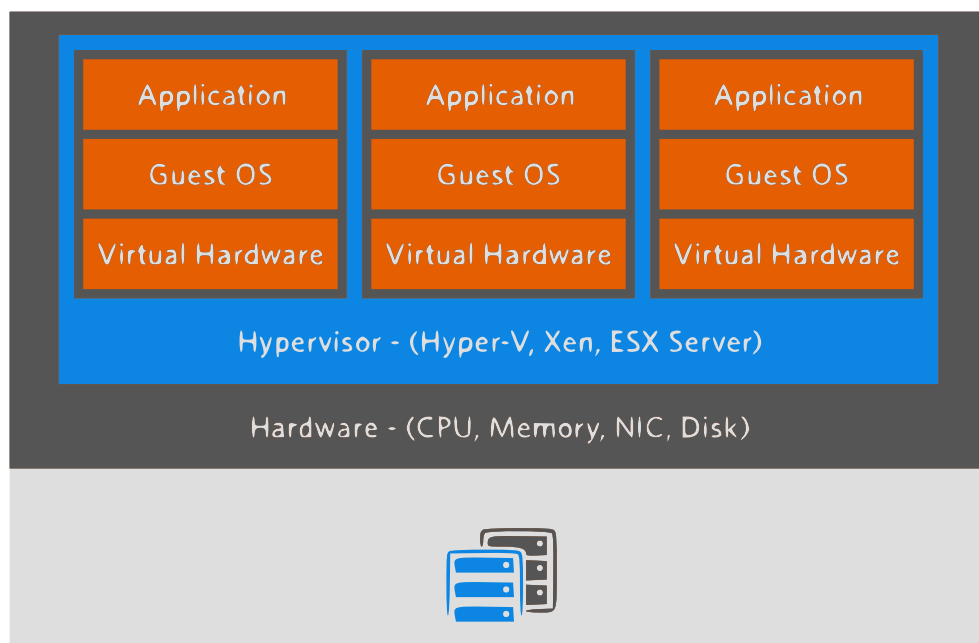
Podľa zdroja [10] existuje šesť oblastí v IT, v ktorých virtualizácia robí pokroky:

- **Sieťová virtualizácia** – spôsob kombinácie dostupných zdrojov v sieti rozdelením dostupného pásma do kanálov, z ktorých každý je nezávislý na ostatných a môže byť pridelený k určitému serveru v reálnom čase

## 2. TEORETICKÝ POPIS VIRTUALIZÁCIE



Obr. 2.1: Tradičná verzus virtuálna architektúra [1]



Obr. 2.2: Diagram virtualizácie [2]

- **Disková virtualizácia** – spôsob spojenia fyzických diskov (z viacerých rôznych sieťových úložných zariadení) do jedného diskového zariadenia
- **Serverová virtualizácia** - spôsob maskovania a delenia zdrojov servera, ktorého zámerom je rozdeliť fyzický server do menších virtuálnych serverov, čo pomáha lepšie využiť a maximalizovať zdroje fyzického servera
- **Dátová virtualizácia** – spôsob prístupu k správe dát, ktorý umožňuje aplikácií získať a manipulovať s dátami bez toho, aby aplikácia vedela informácie o vnútornej štruktúre dát (napr. ako sú dáta formátované alebo to, kde sú fyzický uložené)
- **Aplikačná virtualizácia** – spôsob abstrakcie aplikačnej vrstvy od operačného systému, ktorý umožňuje beh aplikácie v zapuzdrenej forme bez toho, aby to záviselo na operačnom systéme, na ktorom aplikácia beží (napr. aplikácia určená pre beh na operačnom systéme Windows beží na operačnom systéme Linux)
- **Virtualizácia stolných počítačov** – spôsob, ktorý umožňuje virtualizovať pracovnú stanicu a umožňuje užívateľovi pristúpiť k ploche vzdialene, typicky použitím tenkého klienta, ktorý sa pripája na server (kde beží pracovná stanica)

### 2.1.2 Virtualizačné techniky

Zdroj [11] definuje tri virtualizačné techniky:

1. **Paravirtualizácia** – typ virtualizačnej techniky, počas ktorej hostujúci operačný systém vie o tom, že je hosťom a namiesto zadávania príkazov hardvéru jednoducho vydáva príkazy hostiteľskému počítaču (patrí sem aj pamäťový a vláknový manažment, ktorý zvyčajne vyžaduje nedostupné privilegované inštrukcie procesora)
2. **Plná virtualizácia** – typ virtualizačnej techniky, počas ktorej hostujúci operačný systém nevie o tom, že je vo virtualizovanom prostredí, a preto je hardvér virtualizovaný hostiteľským počítačom (hosť si myslí, že zadáva príkazy skutočnému hardvéru, no v skutočnosti sa jedná len o simulované hardvérové zariadenie vytvorené hostiteľským počítačom)
3. **Čiastočná virtualizácia** – typ virtualizačnej techniky, ktorej základnou vlastnosťou je virtualizácia adresného priestoru (virtuálny stroj simuluje viacero inštancií základného hardvérového prostredia, najmä adresného priestoru), v ktorom každý virtuálny stroj pozostáva z nezávislého adresného priestoru

### 2.2 Výhody serverovej virtualizácie

Virtualizáciou dátových centier podľa zdroja [12] získame desať benefitov, ktoré nám serverová virtualizácia prináša a získame tak možnosti a schopnosti, ktoré nie sú vo svete fyzických strojov realizovateľné. Medzi výhody serverovej virtualizácie patrí:

1. **Úspora energie** – migrácia fyzických serverov do virtuálnych strojov a ich konsolidácia na oveľa menší počet fyzických serverov znamená zníženie mesačných nákladov na energiu a chladenie v dátových centrách
2. **Úspora priestoru v dátovom centre** – konsolidácia spojená s virtualizáciou ušetrí miesto v dátovom centre (menej serverov, menej sieťových zariadení a menej stojanov, v ktorých sú umiestnené fyzické servery)
3. **Testovacie virtuálne prostredie** – virtualizácia umožňuje jednoducho vytvárať sebestačné testovacie prostredie, ktoré beží vo vlastnej izolovanej sieti
4. **Rýchlejšie vytvorenie serverov** – serverová virtualizácia poskytuje možnosť vytvoriť alebo prípadne naklonovať už existujúci server za čo najkratšiu dobu
5. **Možnosť výberu dodávateľov hardvéru** – obmedzenie sa na jedného dodávateľa hardvéru môže byť často frustrujúce, ale pretože serverová virtualizácia abstrahuje od hardvéru a nahradzuje ho virtuálnym hardvérom existuje väčšia flexibilita vo výbere hardvéru
6. **Zvýšenie doby prevádzky systému** – funkcie, akými sú živá migrácia, migrácia dátového úložiska, odolnosť voči chybám, vysoká dostupnosť a plánovanie distribuovania dostupných zdrojov ponúkajú možnosť rýchlo sa zotaviť z neplánovaných odstávok
7. **Zlepšenie zotavenia systému po havárii** – zahŕňa tri dôležité komponenty, ktorými sú schopnosť abstrakcie hardvéru, jednoduchšia replikácia servera a systém, ktorý pomáha zautomatizovať prevzatie služby pri zlyhaní
8. **Izolovanie aplikácií** – ponúka izoláciu aplikácií a odstraňuje problémy s kompatibilitou aplikácií konsolidáciou mnohých virtuálnych strojov naprieč oveľa menším počtom fyzických serverov
9. **Predĺženie životnosti starších aplikácií** – virtualizáciou a zapuzdrením aplikácie a jej prostredia môžeme predĺžiť životnosť aplikácie a stále tak zachovať jej prevádzkyschopnosť



10. **Pomáha presúvať veci do cloudu**<sup>5</sup> – jedným z prvých krokov môže byť presun od jednoduchého virtualizovaného dátového centra do privátneho cloudu

## 2.3 Nevýhody serverovej virtualizácie

Na druhej strane existujú aj nevýhody, ktoré so sebou serverová virtualizácia prináša. Podľa zdroja [13] existuje sedem nevýhod, ktoré nám môže serverová virtualizácia priniesť:

1. **Zväčšené fyzické výpadky** – v prípade behu väčšieho počtu virtuálnych serveroch na jednom fyzickom stroji je potrebné zabezpečiť, aby tento stroj bol lepšie odolný proti výpadkom hardvéru, a preto existujú spôsoby ako tomu predísť, no znamená to zainvestovať do redundantného hardvéru, čo menej alebo viac eliminuje výhodu serverovej virtualizácie
2. **Zníženie výkonu** – virtualizácia vyžaduje ďalšie hardvérové nároky a problémom je to, že je takmer nemožné vždy dopredu odhadnúť koľko prostriedkov bude virtualizácia vyžadovať, dokonca existujú aplikácie, ktoré vyžadujú viac prostriedkov, keď bežia na virtuálnom stroji, ako keby bežali na stroji fyzickom
3. **Nové zručnosti** – je potrebné zmeniť pracovné návyky v prípade, že prechádzame od fyzických strojoch k virtuálnym, kde pridelovanie prostriedkov funguje trochu iným spôsobom a teda aj skutočnosť, že viacero správcov zdieľa jeden fyzický server spôsobuje problémy, ktoré predtým neexistovali
4. **Komplexnejšia analýza príčin problémov** – virtualizácia servera znamená veľké zmeny v celom systéme a pridaná nová vrstva môže spôsobiť nové problémy
5. **Nové nástroje pre správu systému** – často sa stáva, že nástroje, ktoré s virtualizáciou prichádzajú nie sú dostačujúce a podporujú len základné funkcie správy, to znamená, že sú potrebné ďalšie nástroje, ktoré stoja čas a peniaze
6. **Expanzia virtuálnych strojov** – problém je ten, že počet serverov rastie rýchlejšie ako počet správcov, ktorí sú schopní sa o ne starať (niektoré z nich existujú len preto, že ich je ľahké vytvoriť vo virtuálnom prostredí, a preto je potrebné si dať pozor na plytvanie zdrojmi nepotrebnými virtuálnymi strojmi)

---

<sup>5</sup>sieť serverov, z ktorých každý má inú funkciu

7. **Virtuálne návyky** – je veľký trend v zavádzaní virtualizácie, no niekedy je lepšie namiesto virtualizačnej technológie použiť klasický fyzický server

### 2.4 Virtuálny stroj

Virtuálny stroj je svojím spôsobom emulácia počítačového systému. Je založený na počítačovej architektúre a poskytuje rovnakú funkčnosť ako fyzický stroj. Jeho implementácia môže obsahovať špecializovaný hardvér, softvér, prípadne ich kombináciu. Jednotlivé virtuálne stroje sú od seba oddelené (bežia vo vlastnom prostredí), to znamená, že jeden stroj nemôže narušiť beh druhého stroja. Požiadavky virtuálnych strojov spracováva vrstva, ktorú zabezpečuje virtualizácia a posieľa ich reálnemu hardvéru, o ktorý sa virtuálne stroje delia (RAM, procesor, diskový priestor).

*Host*<sup>6</sup> VM a *hostiteľ*<sup>7</sup> VM sú dva komponenty, ktoré spolu tvoria virtuálny stroj. Host je softvérová komponenta virtuálneho stroja, ktorá funguje ako keby sa jednalo o fyzický stroj. Hostiteľ je serverová komponenta, ktorá poskytuje príslušným hostom výpočtové zdroje (výpočtový výkon, pamäť, disk, sieťové vstupy/výstupy). Užívateľ, ktorý virtuálny stroj alebo stroje používa má rovnaký pocit ako keby používal fyzický stroj a operačné systémy, ktoré na strojoch bežia väčšinou nemajú informácie o tom, že bežia vo virtualizovanom prostredí. [14]

Virtuálne stroje efektívnejším spôsobom využívajú hardvér, čo znižuje množstvo hardvéru a nákladov, ktoré sú s tým spojené. Virtualizácia a virtuálne stroje vyžadujú väčšiu šírku pásma, väčšiu diskovú kapacitu a viac výpočtového výkonu ako tradičné fyzické servery. Táto situácia nastane v prípade, že fyzický hardvér bude hostiť viac virtuálnych serverov naraz. Tieto virtuálne stroje je možné ľahko kopírovať a presúvať medzi jednotlivými hostiteľskými servermi s cieľom optimalizovať využitie hardvérových prostriedkov. [15]

---

<sup>6</sup>z anglického slova Guest

<sup>7</sup>z anglického slova Host

---

# Hypervízor, virtualizačný softvér a konfiguračné nástroje

## 3.1 Hypervízor

Hypervízor vytvára viac virtuálnych serverov v rámci jedného fyzického servera. Inštaláciou hypervízora na hostiteľský server je možné doceliť spustenie viacerých operačných systémov súčasne, pomocou virtualizačnej technológie. Hypervízor spravuje jeden alebo viac virtuálnych strojov, kde každému virtuálnemu stroju poskytuje prostriedky na beh operačného systému a na beh aplikácií, tak ako keby bol spustený v normálnom prostredí a priamo na skutočnom hardvéri. Zdroj [16] uvádza, že existujú dva typy hypervízorov:

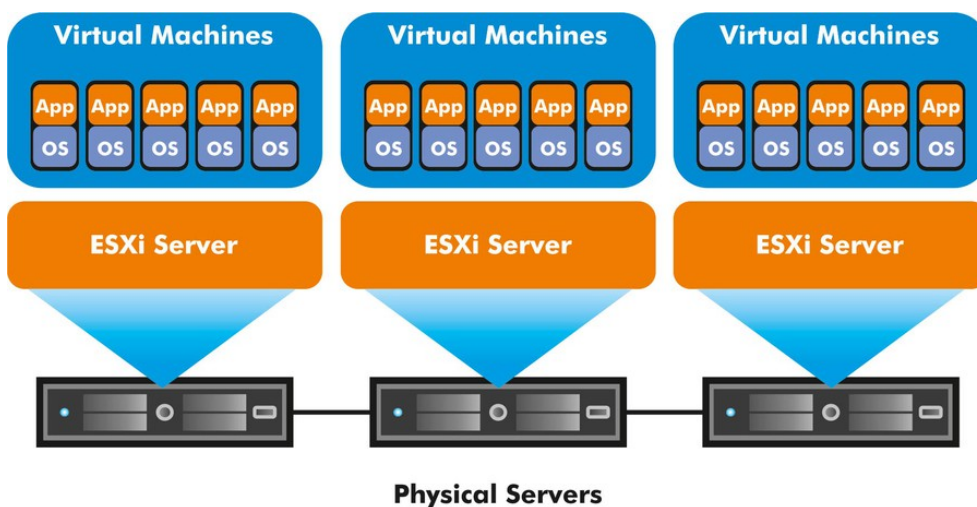
1. **Typ 1 hypervízor** – beží priamo na hardvéri systému, tiež nazývaný ako *bare-metal*<sup>8</sup> hypervízor
2. **Typ 2 hypervízor** – beží na operačnom systéme hostiteľa (vedľa existujúcich aplikácií, ktoré bežia na tomto hostiteľovi), ktorý poskytuje virtualizačné služby ako I/O a správu pamäte, tiež známy ako hostovaný hypervízor

### 3.1.1 VMware ESX Server

VMWare ESX Server je základom pre dynamickú a automatickú optimalizáciu IT infraštruktúry. Je robustná, produkčne-otvorená platforma, ktorá abstrahuje procesor, pamäť, diskové úložisko a sieťové zdroje do niekoľkých virtuálnych strojov. Poskytuje vlastnosti akými sú výkonnosť, škálovateľnosť a robustnosť, ktoré sú požadované v podnikovom IT prostredí. Typickú architektúru VMware ESX servera je možné vidieť na obrázku 3.1.

---

<sup>8</sup>zvyčajne sa pod tým pojmom rozumie pevný disk, médium na ktoré sa inštaluje operačný systém



Obr. 3.1: VMware ESX Server architektúra [3]

Medzi benefity VMware ESX servera patrí:

- zvyšovanie využitia hardvéru a dramatické znížovanie prevádzkových nákladov zdieľaním hardvérových prostriedkov, ktoré využíva veľký počet virtuálnych strojov
- zlepšovanie úrovne poskytovaných služieb pokročilou správou zdrojov, vysokou dostupnosťou a bezpečnostnými funkciami

ESX Server sa inštaluje priamo na hardvér servera (typ 1 hypervízor) a vkladá robustnú virtualizačnú vrstvu medzi hardvér a operačný systém. Rozdeľuje fyzický server do viacerých bezpečných a prenosných virtuálnych strojov, ktoré dokážu bežať vedľa seba na rovnakom fyzickom serveri. Architektúra typu 1 hypervízor dáva ESX serveru úplnú kontrolu nad alokáciou serverových zdrojov každému virtuálnemu stroju a umožňuje tak dosiahnuť takmer natívny výkon virtuálneho stroja. [17]

#### 3.1.2 VirtualBox

VirtualBox je virtualizačná aplikácia (typ 2 hypervízor), ktorá môže bežať pod rôznymi platformami. Znamená to, že sa môže inštalovať na existujúce počítače alebo servery, ktoré obsahujú procesory Intel alebo AMD a na ktorých bežia operačné systémy ako Windows, MAC, Linux alebo Solaris. Takisto rozširuje možnosti počítača na ktorom beží tak, že v rovnakom čase je možné prevádzkovať viacero virtuálnych strojov (viac operačných systémov). Medzi funkcie, ktoré VirtualBox ponúka patrí:

- **prenositelnosť** – dokáže bežať na veľkom počte (32 a 64-bitových) operačných systémov a virtuálne stroje môžu byť ľahko importované a exportované pomocou formátu OVA alebo OVF (priemyselný štandard, ktorý bol vytvorený pre tento účel)
- **beh bez hardvérovej virtualizácie** – nepotrebuje procesorové technológie ako Intel VT-x alebo AMD-V
- **lepší výkon a komunikácia s hostiteľským strojom** – umožňuje inštalovať rozšírenia ako je zdieľanie priečinkov s hostiteľským strojom, automatické nastavenie rozlíšenia obrazovky, ale aj podpora 3D virtualizácie (rozšírenia sa inštalujú ako softvérový balík a zlepšujú výkonnosť celého virtuálneho stroja)
- **veľká hardvérová podpora** – SMP, podpora USB zariadení, hardvérová kompatibilita, plná podpora ACPI, PXE, viacnásobné rozlíšenie obrazovky cez veľké množstvo obrazoviek pripojených k hostiteľskému stroju
- **možnosť vytvorenia snímok (z angl. snapshot)** – možnosť uloženia ľubovoľného stavu virtuálneho stroja (vytváranie a mazanie snímok počas toho ako virtuálny stroj beží)
- **možnosť delenia virtuálnych strojov do skupín** – funkcia, ktorá umožňuje organizovať a kontrolovať virtuálne stroje kolektívne (jednotlivé operácie je možné vykonávať nad celou skupinou virtuálnych strojov)
- **jednoduché ovládanie** – virtuálny stroj je možné ovládať cez GUI, príkazový riadok a dokonca aj cez vzdialený prístup (VRDE)

VirtualBox sa môže javiť ako jednoduchá aplikácia, no v skutočnosti je to veľmi výkonná aplikácia. Môže bežať v rôznych prostrediach, od malých vstaných systémov cez klasické desktopové počítače až po nasadenie v dátových centrách a dokonca aj v prostredí cloudu. [18]

### 3.1.3 KVM

KVM je virtualizačné riešenie (typ 1 hypervízor) pre operačný systém Linux a hardvér platformy x86, ktorý obsahuje rozšírenie virtualizácie (Intel VT alebo AMD-V). Skladá sa z jadrového modulu (kvm.ko), ktorý poskytuje základnú virtualizačnú infraštruktúru a špecifického modulu procesora (kvm-intel.ko alebo kvm-amd.ko). KVM je *open-source*<sup>9</sup> softvér a kernelová komponenta KVM je súčasťou operačného systému Linux od verzie 2.6.20. [19]

<sup>9</sup>vo všeobecnosti akákoľvek informácia dostupná verejnosti, za podmienky, že možnosť jej slobodného šírenia zostane zachovaná

Na uloženie obrazov virtuálnych strojov je možné použiť akýkoľvek typ úložiska, ktorý je podporovaný linuxovým kernelom vrátane lokálnych diskov, SCSI alebo sieťovo-pripojených diskových úložisk (NFS alebo SAN). Podporuje tiež obrazy virtuálnych strojov, ktoré sú uložené na zdieľaných súborových systémoch ako je GFS2 a ktoré umožňujú ich zdieľanie naprieč viacerými fyzickými hosťiteľmi. Možno je takisto vykonávať živé migrácie a presuny bežiacich virtuálnych strojov medzi fyzickými hosťiteľmi bez prerušenia ich behu.

Pretože KVM architektúra hostuje jednotlivé obrazy virtuálnych strojov ako regulárne linuxové procesy je možné použiť štandardné linuxové bezpečnostné opatrenia k izolovaniu a kontrole zdrojov. K izolovaniu strojov sa používa SELinux (bezpečnostný modul linuxového kernelu) spolu s sVIRT, ktorý integruje SELinux a virtualizáciu. Ku kontrole, obmedzeniu a monitorovaniu zdrojov sa používa *cgroups*<sup>10</sup>.

Virtuálne stroje môžu pomocou KSM zdieľať jednotlivé pamäťové stránky. Ak majú jednotlivé virtuálne stroje identické stránky pamäti, tieto stránky môžu byť zlúčené do jednej stránky, ktorá je potom zdieľaná.

KVM hypervízor podporuje širokú škálu operačných systémov (rôzne linuxové distribúcie, Microsoft Windows a ďalšie platformy vrátane OpenBSD, FreeBSD, OpenSolaris, Solaris x86 a MS DOS). [20]

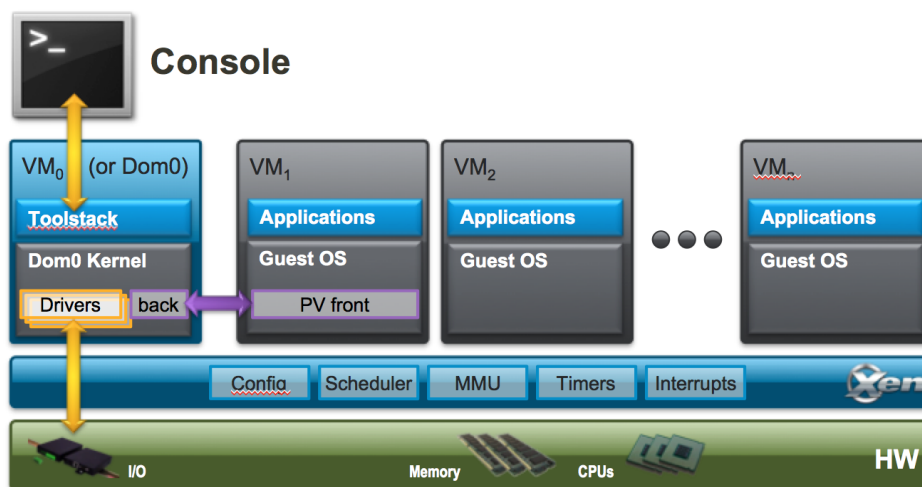
#### 3.1.4 Xen Project

Xen projekt hypervízor je open-source (typ 1 hypervízor), ktorý umožňuje spustiť mnoho inštancií operačného systému alebo rôzne operačné systémy súčasne, na jednom počítači. Používa sa ako základ pre rôzne komerčné a open-source aplikácie ako napr. serverová virtualizácia, IaaS, virtualizácia desktopov, bezpečnostné aplikácie a vstavané hardvérové zariadenia. Na obrázku 3.2 je možné vidieť typickú architektúru Xen projektu. Podľa zdroja [21] medzi kľúčové vlastnosti hypervízora patrí:

1. **malé rozmery a rozhranie** – používa microkernel dizajn s malými nárokmi na pamäť a s obmedzeným rozhraním pre hosťa, a preto je robustnejší a bezpečnejší ako ostatné hypervízory
2. **použitie výhradne 1 operačného systému** – väčšina inštalácií beží pod linuxovým jadrom ako hlavným ovládacím prvkom („doména 0“), ale môžu byť použité aj ďalšie operačné systémy (napr. NetBSD a OpenSolaris)
3. **izolácia ovládačov** – umožňuje, aby hlavný ovládač zariadenia pre systém bežal vnútri virtuálneho stroja, a ak tento ovládač zlyhá alebo je ohrozený môže byť virtuálny stroj spolu s ovládačom reštartovaný bez toho, aby to ovplyvnilo zvyšok systému

---

<sup>10</sup>systém kontroly skupín



Obr. 3.2: Xen Project architektúra [4]

4. **paravirtualizácia** – plne paravirtualizovaný hosť bol optimalizovaný tak, aby bežal ako virtuálny stroj a to mu umožňuje bežať oveľa rýchlejšie ako s rozšírením hardvéru (HVM) a navyše umožňuje, aby hypervízor bežal na hardvéri, ktorý nepodporuje rozšírenie virtualizácie

Medzi kľúčové aspekty Xen Project architektúry, ktoré musí užívateľ pochopiť na to, aby si vybral najlepšiu voľbu patrí:

1. **typ hosťa** – hypervízor môže bežať ako plne virtualizovaný (HVM), alebo ako paravirtualizovaný (PV) hosť
2. **doména 0** – architektúra využíva špeciálnu doménu, nazývanú ako „doména 0“, ktorá obsahuje ovládače pre hardvér, ako aj nástroj pre kontrolu virtuálnych strojov
3. **nástroje** – obsahom sú rôzne nástroje, ktoré poskytujú front-end a je potrebné si uvedomiť dôsledky používania každého z nich

## 3.2 Virtualizačný softvér

S rýchlym rozvojom v oblasti virtualizácie a *cloud computing*<sup>11</sup> postupne narastá aj priemerný počet virtuálnych strojov, ktoré je potrebné postupne vytvárať a o ktoré sa musia správcovia systémov starať. Manuálne vytvoriť

<sup>11</sup>model vývoja a používania počítačových technológií

celý virtuálny stroj so základnou konfiguráciou je pomerne dosť časovo náročný proces v dnešných virtualizačných prostrediach (hypervízoroch) akými sú napr. VMware, KVM, VirtualBox alebo Xen.

Samozrejme všetky moderné hypervízory podporujú funkcie ako je napr. klonovanie virtuálneho stroja, prípadne jeho exportovanie do rôznych formátov. Avšak väčšinou je potrebné vo virtuálnom stroji, ktorý je vytvorený touto cestou upraviť konfiguráciu jednotlivých služieb, nastaviť alebo zmeniť vybrané hardvérové (napr. veľkosť pamäte RAM, veľkosť disku a počet CPU) a systémové (napr. nastavenie IP adresy a nastavenie hostname) parametre, prípadne vytvoriť nových užívateľov.

Nástroje akými sú napr. Vagrant alebo Packer (nástroje sú bližšie popísané v podsekciiach 3.2.1 a 3.2.2) nám dokážu čiastočne alebo úplne vyššie uvedené požiadavky splniť. Pomocou nich je možné jednoducho, ale hlavne rýchlo generovať a vytvárať nové virtuálne stroje napr. pre rôzne hypervízory, prípadne pre iné platformy. Touto formou je možné virtuálny stroj jednoducho zrušiť a zmazať, a v priebehu niekoľkých sekúnd alebo minút vytvoriť nový virtuálny stroj. Tento novovytvorený stroj môže byť napr. identický ako ten predtým, môže obsahovať malé zmeny, no môže byť aj úplne odlišný.

Existuje ešte podobný softvér Docker, ktorý používa trochu odlišný prístup ako vyššie spomínané nástroje. Docker je open-source platforma, ktorá je založená na kontajneroch v ktorých bežia jednotlivé aplikácie, ktoré sú od seba navzájom izolované. Nevytvára nový virtuálny stroj s operačným systémom, ale využíva už existujúce jadro operačného systému. Hlavnou úlohou je automatizácia nasadenia jednotlivých aplikácií vnútri softvérových kontajnerov.

#### 3.2.1 Vagrant

Vagrant je nástroj na tvorbu kompletných vývojových prostredí. S ľahko použiteľným *workflow*<sup>12</sup> a sústredením sa na automatizáciu znižuje čas, ktorý je potrebný na vytvorenie vývojového prostredia (napr. virtuálneho servera). Poskytuje jednoducho konfigurovateľné, reprodukovateľné a prenosné pracovné prostredie postavané na najnovších technologických štandardoch, ktoré pomáhajú zvyšovať produktivitu a flexibilitu.

Virtuálne stroje (prostredia) sú pomocou nástroja Vagrant vytvárané pre hypervízory akými je napr. VirtualBox, VMware alebo AWS. O automatickú inštaláciu a konfiguráciu softvéru a služieb na virtuálnom stroji sa môžu starať shell skripty, prípadne orchestračné nástroje ako Chef, Puppet alebo Ansible (bližšie sú tieto nástroje popísané v sekcii 3.3). Medzi výhody, ktoré môže tento nástroj priniesť patrí:

- **pre vývojára** – izoluje závislosti a ich konfiguráciu v rámci jedného jednorázového a konzistentného prostredia, zabezpečuje, že sa kód spustí

---

<sup>12</sup>pracovný alebo technologický postup



stále v rovnakom prostredí, ktoré bude stále nakonfigurované rovnako (na základe rovnakých závislostí)

- **pre priemyselného inžiniera** – dáva možnosť vytvorenia jednorázového testovacieho prostredia pre vývoj a testovanie aplikácií, prípadne testovanie skriptov pre správu infraštruktúry, umožňuje jednoducho testovať skripty napísané v jazyku shell, Chef recepty, Puppet moduly použitím lokálnej virtualizácie ako VirtualBox alebo VMware
- **pre dizajnéra** – automaticky nastaví všetko, čo je požadované pre webovú aplikáciu, aby sa dizajnér mohol zamerať na to najdôležitejšie, čo je obsahom jeho práce (dizajn) a nemusel sa tak vždy starať o vytvorenie prostredia

Postupom času sa Vagrant stal dôveryhodným a používaným nástrojom celou radou jednotlivcov, ale aj používaným nástrojom vo vývojových tímoch veľkých spoločností. [22]

### 3.2.2 Packer

Zdroj [23] uvádza, že Packer je open-source nástroj na tvorbu identických obrazov virtuálnych strojov pre rôzne platformy pomocou jedného konfiguračného súboru. Jednoduchý nástroj, ktorý beží skoro na každom operačnom systéme a dokáže vytvárať obrazy virtuálnych strojov pre viacero platforiem paralelne. Jeho úlohou nie je nahradiť konfiguračné nástroje ako Puppet, Chef alebo Ansible, ale je schopný ich využiť pri tvorbe obrazu virtuálneho stroja. Obraz virtuálneho stroja, ktorého základná kostra je dopredu pripravená má svoje výhody, no väčšinou nie je možné tieto výhody využiť naplno pre ich príliš časovo náročné vytváranie a spravovanie (tento prístup používa nástroj Vagrant). Packer to trochu mení tým, že otvára nové príležitosti a automatizuje vytváranie akéhokoľvek typu obrazu virtuálneho stroja. Packer prináša nasledujúce výhody:

- **rýchle nasadenie infraštruktúry** – umožňuje spustiť kompletne nainštalovaný a nakonfigurovaný stroj v priebehu niekoľkých sekúnd
- **prenositelnosť medzi platformami** – umožňuje vytvoriť identický obraz virtuálneho stroja pre viacero platforiem (VMware, VirtualBox a pod.)
- **vylepšená stabilita** – inštaluje a konfiguruje softvér už počas vytvárania obrazu virtuálneho stroja, čím je možné oveľa skôr predísť chybám, ktoré môžu vzniknúť v inštaláčnych skriptoch
- **väčšie možnosti testovania** – potom ako sa obraz vytvorí je možné jednoducho otestovať funkčnosť systému a tým prípadne overiť, že všetky ďalšie stroje vytvorené z toho obrazu budú fungovať správne

### 3.3 Konfiguračné nástroje

Konfiguračné nástroje dokážu správcovi systémov pomôcť s automatickou inštaláciou a konfiguráciou služieb veľkého množstva serverov. Ich doménou je automatizácia opakujúcich sa úloh, paralelne nasadenie aplikácií a služieb skupine serverov, ale aj konfigurácia nových serverov od úplného začiatku. Zdroj [24] porovnáva dva nástroje (Chef a Ansible), ktoré spĺňujú tento popis, no existuje ešte jeden nástroj (Puppet), ktorý stojí za zmienku. [25]

- **Chef** – nástroj, ktorý je napísaný v programovacom jazyku Ruby a založený na architektúre hlavný kontrolný server – agent (na pracovnej stanici), konfigurácie jednotlivých služieb sú ukladané do JSON súborov, ktoré sa nazývajú recepty
- **Ansible** – nástroj, ktorý je napísaný v programovacom jazyku Python, ktorý využíva SSH prístup a balí všetky príkazy do YAML modulov, ktoré spoločne vytvárajú výsledný *playbook*<sup>13</sup>
- **Puppet** – nástroj, ktorý je napísaný v programovacom jazyku Ruby a je založený na architektúre klient-server, konfigurácie jednotlivých služieb sú uložené v súboroch, ktoré sa nazývajú *manifesty*<sup>14</sup>

---

<sup>13</sup>súbor, ktorého obsahom je postupnosť príkazov (vo formáte YAML), ktoré sa majú vykonať na cieľovom stroji

<sup>14</sup>konfiguračný súbor pre nástroj Puppet

---

## Prehľad a analýza existujúcich riešení

V nasledujúcej kapitole je rozobraných niekoľko nástrojov, ktoré slúžia na automatickú inštaláciu linuxových serverov. Pred samotnou tvorbou riešenia je dôležitý prehľad a analýza existujúcich riešení.

Cieľom je analýza toho, aké existujú spôsoby inštalácie linuxových strojov (hlavne tých virtuálnych) a aké sú konfiguračné možnosti jednotlivých nástrojov. Ďalším cieľom analýzy je zistenie výhod a nevýhod nástrojov, ale aj zistenie spôsobu ich ovládania užívateľom.

### 4.1 Spacewalk

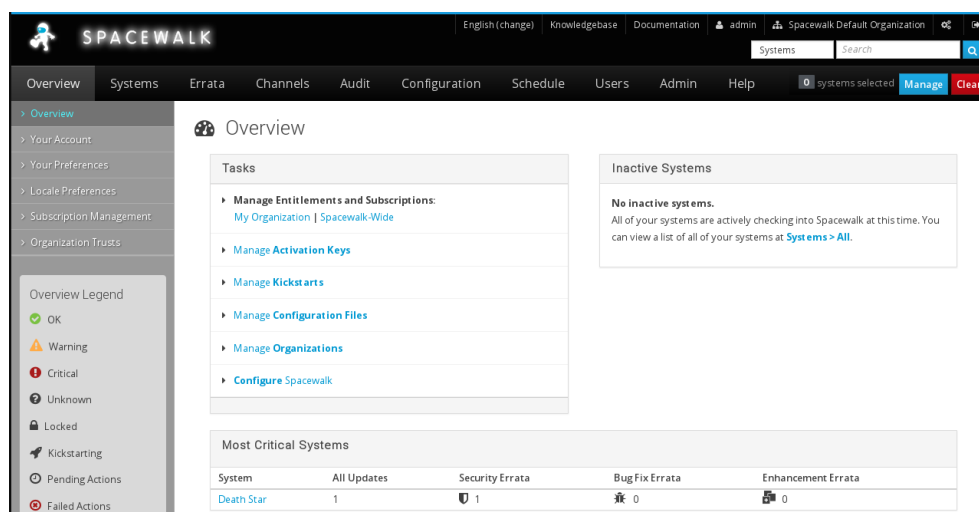
Spacewalk [26] je open-source nástroj na manažment alebo správu systémov. Umožňuje spravovať softvérové aktualizácie pre distribúcie odvodené od Red Hat (napr. Fedora, CentOS). Centrálné webové rozhranie umožňuje zobrazovať stav systému a stav aktualizácie jednotlivých softvérov. Taktiež umožňuje spravovať systémy počas celého ich životného cyklu. Spacewalk pomocou *kickstart*<sup>15</sup> inštaluje systém a kopíruje konfiguračné súbory. Medzi jeho schopnosti napr. patrí:

- inventarizácia systému (hardvérové a softvérové informácie)
- inštalácia a aktualizácia softvéru
- inštalácia systému pomocou kickstart
- ovládanie a konfigurácia virtuálnych serverov

---

<sup>15</sup>inštaláčna metóda určená na automatickú inštaláciu a konfiguráciu operačného systému

## 4. PREHLAD A ANALÝZA EXISTUJÚCICH RIEŠENÍ



Obr. 4.1: Časť webového rozhrania Spacewalk [5]

### 4.1.1 Analýza nástroja

Spacewalk ako nástroj patrí pod skupinu Red Hat a je dostupný len vo forme RPM balíčkov, čo môže byť niekedy problém kvôli nekompatibilitě správcu balíčkov systému, na ktorý sa bude Spacewalk inštalovať. Nástroj Spacewalk inštaluje celý systém pomocou kickstart metódy. Táto metóda sa používa hlavne pri inštalácii Red Hat distribúcií, čo obmedzuje možnosť inštalácie iných distribúcií. Výhodou nástroja je ovládanie pomocou webového rozhrania, čo umožňuje jeho jednoduchšie ovládanie. Podporuje hypervízory ako KVM a XEN, no chýba podpora VirtualBox, ale aj VMware.

- odvodený od projektu Red Hat Satellite
- dostupný len vo forme RPM balíčkov, čo obmedzuje inštaláciu na niektoré iné linuxové distribúcie
- k svojmu behu potrebuje Java Virtual Machine a databázový server (PostgreSQL alebo Oracle RDBMS)
- potrebuje, aby boli prístupné rôzne prichádzajúce porty napr. 80 a 443
- nástroj Spacewalk je možné konfigurovať aj cez súbor, ktorého obsahom sú odpovede na hodnoty, ktoré je potrebné zadať pri inštalácii napr. admin-email = root@localhost
- pozostáva z rôznych služieb a každá z nich má vlastný init skript
- prístup cez webové rozhranie
- podporuje inštalácie pre hypervízory KVM a Xen

```

GNU GRUB  version 2.02~beta2-22

+-----+
| Fully Automatic Installation |
| FAI 5.0   build 2015 Nov 23 - 17:07 |
| (c) Thomas Lange  lange@debian.org |
+-----+
Username/password to start the installation:  fai/install

*FAI client standalone installation - select installation type from menu
FAI server installation - using external DHCP
FAI server installation - using a fixed IP
FAI rescue system, no installation
Boot OS from first partition on first disk
Memory test (memtest86+)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.

```

Obr. 4.2: Úvodná obrazovka nástroja FAI [6]

## 4.2 FAI

FAI [27] je neinteraktívny systém na inštaláciu a manažment systémov založených na operačnom systéme Linux. Tento nástroj sa zaoberá softvérovou konfiguráciou fyzických a virtuálnych strojov. Je to nástroj na hromadné nasadenie alebo inštaláciu linuxových strojov, ktoré sú spolu so systémom a s kompletnou konfiguráciou nainštalované bez interakcie užívateľa. Na obrázku 4.2 je možné vidieť úvodnú obrazovku nástroja FAI. Medzi hlavné výhody nástroja patrí:

- inštalácia a aktualizácia operačných systémov napr. Debian, Ubuntu, CentOS, RHEL alebo SUSE
- centralizovaná správa systému a konfiguračný manažment
- jednoduché nastavenie softvérového RAID a LVM
- plná kontrola systému cez SSH počas inštalácie

### 4.2.1 Analýza nástroja

FAI je neinteraktívny systém, čo znamená, že ho nie je možné konfigurovať pomocou nejakého grafického rozhrania. Tento nástroj je možné ovládať len pomocou príkazového riadku daného systému a spustiť pomocou jednoduchého menu, ktoré je vytvorené v príkazovom riadku. S tým je aj spojené to, že užívateľ na to, aby mohol naplno využívať tento systém musí vedieť presnú

syntax a konfiguračné možnosti alebo voľby nástroja. Medzi jednu z najväčších výhod je možné považovať inštaláciu rôznych druhov distribúcií a podporu inštalácie virtuálnych strojov na rôzne hypervízory. Pred inštaláciou nástroja je potrebné na systéme pripraviť prostredie, ktoré bude tento nástroj pri inštalácii strojov využívať.

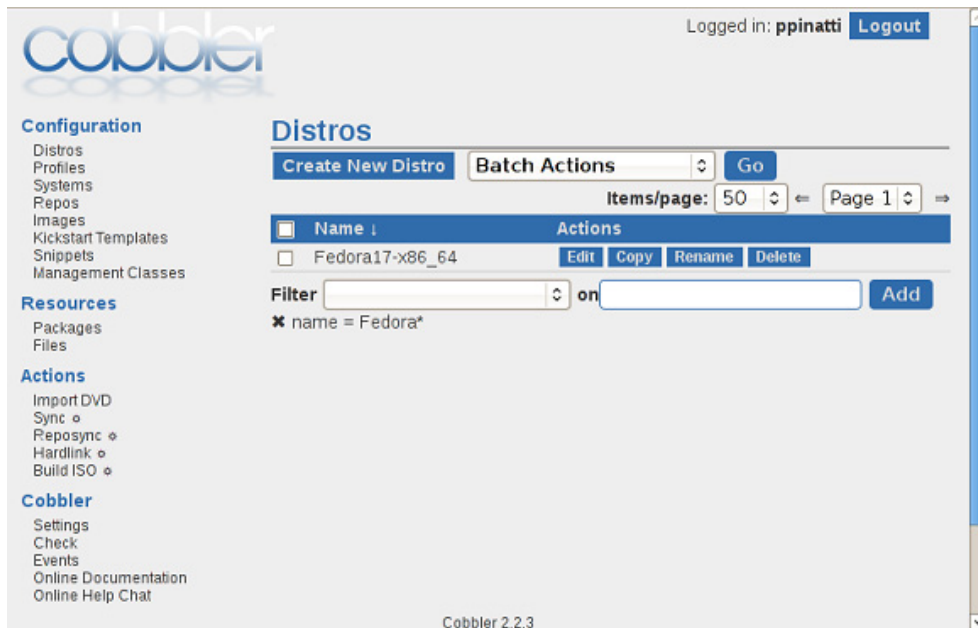
- ovládanie pomocou príkazového riadku (neinteraktívny systém)
- klient, ktorý bude inštalovaný pomocou FAI je zavedený cez sieťovú kartu alebo z CD/DVD, USB média
- podporuje inštaláciu virtuálnych strojov použitím hypervízorov akými je KVM, XEN alebo VirtualBox
- systém, na ktorý sa inštaluje a pomocou neho ovláda FAI poskytuje DHCP, TFTP a NFS služby a konfiguračné dáta pre všetkých inštalovaných klientov pomocou FAI
- je potrebné nakonfigurovať alebo nastaviť systém a vytvoriť na tomto systéme prostredie pre FAI
- pri inštalácii jednotlivých klientov je potrebné vedieť štruktúru konfiguračných súborov a modulov, ktoré FAI ponúka

### 4.3 Cobbler

Cobbler [28] je linuxový inštalačný server, ktorý dokáže pomôcť napr. so správou TFTP, DHCP a DNS za účelom vykonávania inštalácií operačných systémov prostredníctvom siete. Dokáže zautomatizovať proces inštalácie serverov od úplného začiatku, ale aj pomôcť s vytváraním virtuálnych strojov pre rôzne hypervízory. Cobbler je možné používať cez príkazový riadok, webové rozhranie a existuje niekoľko možností prístupu cez API. Na obrázku 4.3 je možné vidieť webové rozhranie nástroja Cobbler.

#### 4.3.1 Analýza nástroja

Cobbler je možné ovládať pomocou príkazového riadku, ale aj pomocou webového rozhrania a API, čo je možné považovať za veľkú výhodu nástroja. Pred použitím nástroja je vhodné mať znalosti v oblasti inštalácie strojov pomocou PXE a poznať spôsoby automatických inštalácií jednotlivých distribúcií, čo môže byť pre menej skúseného užívateľa problém. Pri inštalácii nástroja je potrebné urobiť malé konfiguračné zmeny do systému, na ktorý sa tento nástroj inštaluje. Podporuje inštalácie virtuálnych strojov na rôzne hypervízory a taktiež podporuje inštalácie rôznych linuxových distribúcií.



Obr. 4.3: Webové rozhranie nástroja Cobbler [7]

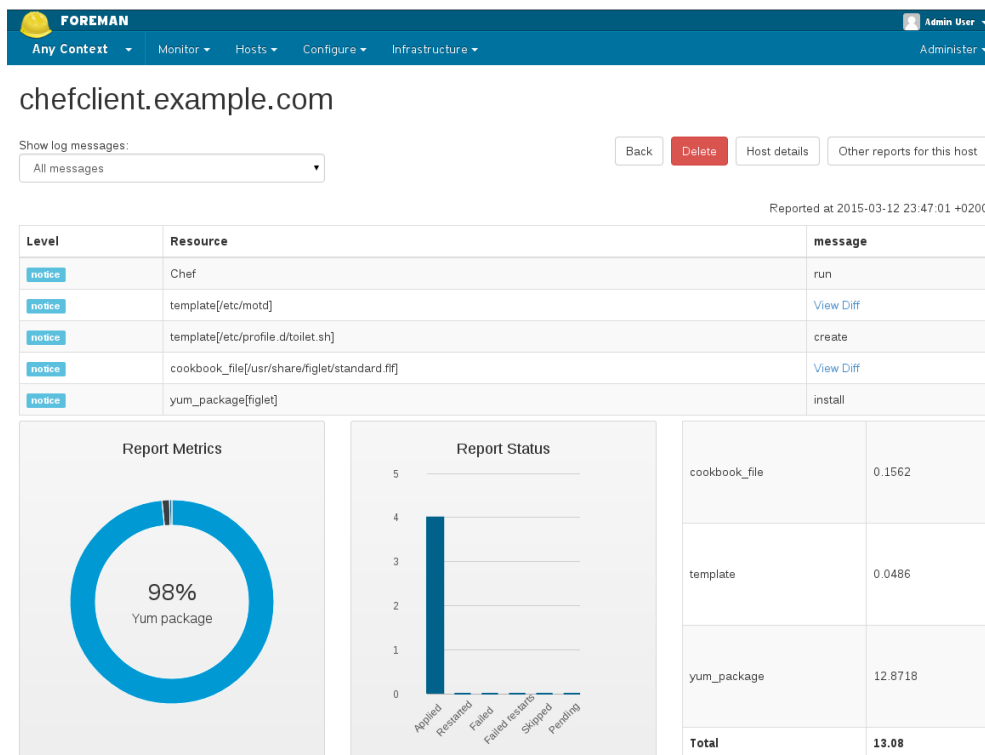
- možnosť ovládania nástroja cez príkazový riadok, ale aj cez webové rozhranie
- pred používaním nástroja je potrebné mať dobrú znalosť toho, ako funguje technológia PXE a automatizovaný proces inštalácie vybranej linuxovej distribúcie
- pri používaní nástroja je vhodné zakázať *SELinux*<sup>16</sup> na systéme
- podporuje virtualizované inštalácie (Xen, QEMU, KVM a niektoré varianty VMware)
- pomáha s aktualizáciou balíčkov a správou konfigurácie pomocou konfiguračných nástrojov (napr. pomocou Puppet)
- používa sa hlavne na inštaláciu Red Hat distribúcií, ale je ho možné použiť aj na inštaláciu Debian a Ubuntu

## 4.4 Foreman

Foreman [29] je open-source projekt, ktorý pomáha systémovým administrátorom spravovať servery počas celej doby ich existencie, od vytvorenia a konfigurácie cez konfiguračné nástroje až po monitorovanie. Použitím nástrojov ako

<sup>16</sup>bezpečnostný modul kernelu, ktorý podporuje bezpečnostnú politiku riadenia prístupu

## 4. PREHLAD A ANALÝZA EXISTUJÚCICH RIEŠENÍ



Obr. 4.4: Webové rozhranie nástroja Foreman [8]

je Puppet, Chef, Salt a Ansible je možné jednoducho zautomatizovať opakujúce sa úlohy, nainštalovať aplikácie a aktívne reagovať na zmeny, a to všetko v prostredí virtuálnych, ale aj fyzických serveroch a dokonca aj v cloude. Interakcia s Foreman je možná cez webové rozhranie, príkazový riadok ale aj cez Restful API. S týmto nástrojom je napr. možné:

- aktualizácia celej fyzickej infraštruktúry (serverov)
- rozdeľovanie zariadení do skupín a ich správa vo veľkom
- automatické vytváranie systémových obrazov s cieľom optimalizovať ich nasadenie

Na obrázku 4.4 je možné vidieť webové rozhranie nástroja Foreman.

### 4.4.1 Analýza nástroja

Foreman sa dá považovať za najlepší nástroj zo všetkých uvedených nástrojov, ale aj tak má svoje nevýhody. Umožňuje inštalovať, konfigurovať, ale aj monitorovať jednotlivé stroje. Obsahuje viacero rozšíriteľných modulov a podporuje



rôzne konfiguračné nástroje pomocou ktorých je možné konfigurovať jednotlivé služby. Cez príkazový riadok, webové rozhranie, ale aj API je možné tento nástroj ovládať. Je určený pre inštalovanie operačných systémov založených na linuxových distribúciach Red Hat a CentOS, čo ho trochu obmedzuje. Je predovšetkým určený na inštalácie do cloudu a nepodporuje inštalácie do hypervízorov ako XEN, VirtualBox alebo KVM.

- základná súčasť komerčného produktu Red Hat Satellite 6 (Red Hat projekt)
- má integrovaný konfiguračný nástroj Puppet
- umožňuje inštalovať linuxové distribúcie Red Hat a CentOS
- pomocou neho je možné nasadiť hardvérové systémy (systém s MAC adresou bootovateľný cez DHCP) a virtuálne systémy (napr. Libvirt a OpenStack)
- pomocou pluginov je možné rozšíriť jeho základnú funkcionality



---

## Návrh riešenia

Nasledujúca kapitola sa venuje návrhu a architektúre celého systému, pomocou ktorého bude možné automaticky vytvárať linuxové virtuálne stroje, špecifikácií inštalovaných služieb a parametrom, ktoré je možné meniť pri inštalácii strojov.

### 5.1 Požiadavky systému

Na začiatku (pri návrhu systému) je potrebné sa zaoberať funkcionalitou a požiadavkami, ktoré sa od nástroja očakávajú. Vzhľadom k tomu, že nástroj budú používať aj menej skúsení používatelia je vhodná prítomnosť užívateľského rozhrania, ktoré by malo byť prehľadné a jednoduché na používanie a ovládanie.

Jednou z požiadaviek je vstupné rozhranie vo formáte XML, ktoré umožní nastavovať jednotlivé parametre virtuálneho stroja bez hlbších znalostí užívateľa. Nevyhnutné je teda definovať obsah a štruktúru súboru vo formáte XML, ktorý bude súčasťou nástroja. Užívateľ bude samozrejme môcť upravovať jednotlivé parametre alebo atribúty XML súboru ľubovoľným textovým editorom, no od nástroja sa očakáva to, aby umožnil súbor vo formáte XML vytvoriť (aj s možnosťou uloženia) a pomohol tak menej skúseným užívateľom. Úlohou nástroja bude načítať parametre z vytvoreného alebo upraveného XML súboru a prípadne umožniť užívateľovi ešte tieto parametre zmeniť cez užívateľské rozhranie. Vedľajšou funkciou nástroja bude analyzovať vybrané parametre vzdialeného servera pomocou SSH a vytvoriť XML súbor, ktorého obsahom budú takto získané parametre (vytvorenie čiastočného klonu systému).

Ďalšou požiadavkou je nastaviť vybrané parametre servera a nainštalovať požadované služby spolu s ich automatickou konfiguráciou. Nástroj bude umožňovať nastaviť parametre ako je napr. IP adresa, hardvérové nastavenia atď. (bližšie to je popísané v podsekcii 5.4.1) a inštalovať a konfigurovať služby ako napr. rancid, snmp, zabbix atď. (bližšie to je popísané v podsekcii 5.4.2)

Poslednou požiadavkou je inštalovať virtuálne servery založené na operačných systémoch linuxových distribúcií Ubuntu a Debian. Nástroj musí umožniť

inštalovať virtuálne stroje určené pre hypervízor VirtualBox spolu s exportom do formátu OVA alebo OVF. Ďalší hypervízor pre ktorý budú virtuálne stroje inštalované bude KVM spolu s exportom do formátu qcow2 a raw.

Pre zhrnutie môžeme požiadavky a to čo nám nástroj poskytne zhrnúť do nasledujúcich bodov:

- **užívateľské rozhranie**
- **vstupné rozhranie vo formáte XML**
- **nastavovanie vybraných parametrov**
- **inštalácia a konfigurácia jednotlivých služieb**
- **inštalácia operačných systémov Debian a Ubuntu**
- **virtuálne stroje určené pre VirtualBox a KVM**

### 5.2 Možnosti užívateľských rozhraní

Pri návrhu užívateľského rozhrania prichádzajú do úvahy 3 nasledujúce možnosti:

1. príkazový riadok
2. webové rozhranie
3. grafické rozhranie

#### 5.2.1 Príkazový riadok

Medzi základné vlastnosti príkazového riadku patrí to, že je rýchly, nenáročný a užívateľ k jeho používaniu nepotrebuje grafické užívateľské rozhranie (všetko sa zaobíde bez klikania, nie je potrebná myš). Je to nástroj pomocou, ktorého je možné zadávať rôzne systémové príkazy, využívať nástroje bez použitia grafickej nadstavby, ale aj meniť a upravovať rôzne konfiguračné súbory.

Správcovia systémov väčšinou využívajú k správe serverov príkazový riadok, pretože je to nástroj pomocou ktorého je možné rýchlejšie, ale hlavne efektívnejšie spravovať jednotlivé systémy.

Na druhú stranu pre začínajúceho alebo menej skúseného užívateľa, ktorý využíva len grafické rozhranie systému, môže byť zo začiatku problém používať a zapamätať si jednotlivé príkazy a ich prepínače. Aj keď existujú manuálové stránky, ktoré vo väčšine prípadov dokážu pomôcť je to skôr pre užívateľa zo začiatku spomalenie práce ako zrýchlenie.

Na základe toho, že nástroj má byť prehľadný a jednoduchý na ovládanie a používanie je táto voľba aj pre menej skúseného používateľa nevhodná a nepoužiteľná.

### 5.2.2 Webové rozhranie

Webové rozhranie sa dá považovať za najkomfortnejšie riešenie pre užívateľa, pretože hlavnou výhodou je to, že pre prístup k aplikácii je potrebný len webový prehliadač. Webový prehliadač je v dnešnej dobe prítomný skoro vo všetkých operačných systémoch. Táto skutočnosť nám umožňuje pristupovať k aplikácii pomocou webového prehliadača z ľubovoľného operačného systému.

Miernou nevýhodou tohto riešenia by mohla byť nutnosť implementácie webového servera, no v dnešnej dobe balíčkovacích systémov to nie je veľký problém a môže sa to realizovať napr. pomocou webového servera Apache HTTP Server.

Zmeny, ktoré sa v aplikácii vykonajú nemusia byť určitým spôsobom distribuované smerom k užívateľovi, ale automaticky sa zobrazia alebo zmenia pri načítaní aplikácie po aktualizácii, avšak aj napriek spomínaným výhodám je pre niektoré aplikácie vhodnejšie použiť grafické rozhranie.

### 5.2.3 Grafické rozhranie

Grafické rozhranie nám pomáha vyriešiť problém, ktorý sa týka užívateľského komfortu (prehľadnosť a jednoduché ovládanie) a eliminuje nedostatky príkazového riadku. Je určené aj pre menej skúsených užívateľov, ktorí sú zvyknutí na grafické rozhranie.

Výhodou riešenia je to, že nepotrebuje napr. webový server pre prístup k aplikácii, no nevýhodou môže byť to, že pri aktualizácii aplikácie je nutná distribúcia zmien napr. pomocou balíčkovacieho systému. Do určitej miery je riešenie viazané na operačný systém pre ktorý je aplikácia určená, no v tomto prípade sa bude aplikácia zaoberať inštaláciou linuxových virtuálnych serverov, a teda bude primárne určená len pre linuxové operačné systémy.

Aj napriek niektorým nevýhodám je pre návrh riešenia a následnú implementáciu použité grafické rozhranie.

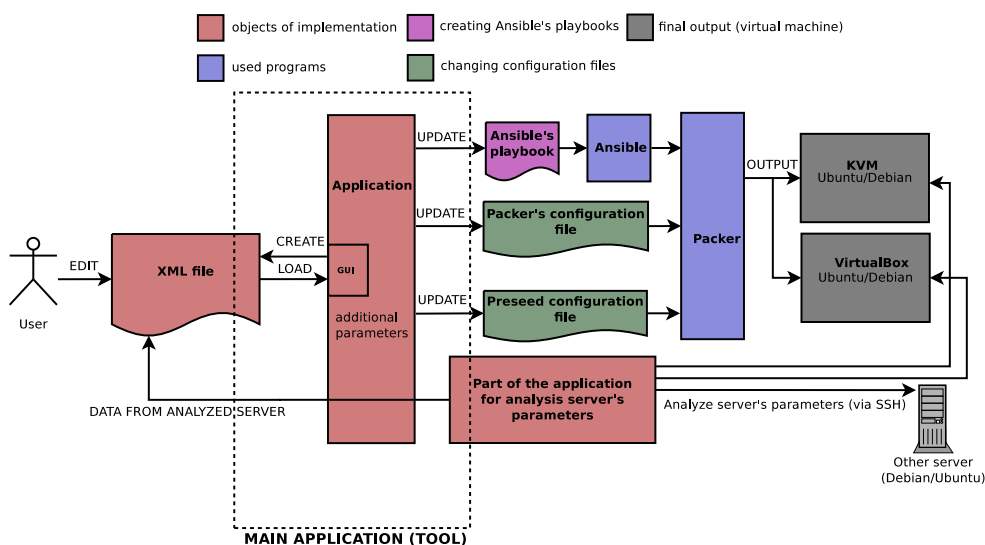
## 5.3 Architektúra aplikácie

Na obrázku 5.1 je možné vidieť architektúru celej aplikácie, ktorá sa skladá z niekoľkých na seba nadväzujúcich častí.

Hlavnou časťou je desktopová aplikácia, ktorá poskytuje rozhranie medzi súborom vo formáte XML, z ktorého sa načítavajú jednotlivé parametre inštalovaného servera, grafickým rozhraním pomocou ktorého je možné doplniť jednotlivé doplňujúce parametre alebo inštalované služby, konfiguračným nástrojom Ansible [30] a virtualizačným softvérom Packer. [31]

Súbor vo formáte XML vytvorí užívateľ pomocou GUI aplikácie alebo len upraví hodnoty už existujúceho XML súboru pomocou ľubovoľného textového editora, no musí dodržať príslušné elementy s ktorými aplikácia počíta. Ďalšou úlohou aplikácie je buď načítať vytvorený alebo upravený XML súbor

## 5. NÁVRH RIEŠENIA



Obr. 5.1: Štruktúra aplikácie

s možnosťou ďalšej zmeny parametrov. Aplikácia ponúka možnosť čiastočne analyzovať vzdialený server pomocou SSH (IP adresa servera, meno a heslo) a z tejto analýzy získať a následne vytvoriť niektoré parametre pre inštalovaný server.

Ak aplikácia získa všetky potrebné informácie upraví všetky konfiguračné súbory, ktoré jednotlivé nástroje potrebujú a spustí aplikáciu Packer, ktorá v spolupráci s Ansible vytvorí virtuálny stroj.

### 5.3.1 Architektúra MVC

Pri návrhu vnútornej štruktúry aplikácie nebude použitá typická MVC architektúra, aj v závislosti na tom, že použitá technológia používa Model-View koncept (Controller je súčasťou View). Podľa zdroja [32] je MVC návrhový vzor rozdeľujúci aplikáciu do troch vrstiev (model, view a controller). Každá z týchto troch vrstiev je od ostatných vrstiev oddelená abstraktnými hranicami a komunikácia medzi vrstvami prebieha naprieč týmito hranicami.

#### 5.3.1.1 Dátová vrstva

Model je obecné označenie pre triedu, ktorá je súčasťou dátovej vrstvy aplikácie. Úlohou tejto triedy ako aj celej vrstvy je manipulácia a práca s dátami. V aplikácii bude úlohou dátovej vrstvy vykonávať príkazy na základe toho, čo bude predané z logickej vrstvy.

Medzi hlavné úlohy tejto vrstvy bude patriť upravovanie konfiguračných súborov, analýza systému a kontrola vstupov od užívateľa.

### 5.3.1.2 Prezentačná vrstva

Prezentačná vrstva je ďalšou vrstvou aplikácie a jej hlavnou úlohou je vykresľovanie dát užívateľovi a interakcia s užívateľom. View je obecné označenie pre triedu, ktorá je súčasťou prezentačnej vrstvy aplikácie.

Pomocou nástroja Qt Designer [33] bude vytvorený návrh a dizajn GUI, a vygenerovaná hlavná časť prezentačnej vrstvy aplikácie.

### 5.3.1.3 Logická vrstva

Poslednou vrstvou je logická vrstva, ktorá tvorí rozhranie medzi prezentačnou a dátovou vrstvou. Jej hlavnou úlohou je riadenie toku dát medzi týmito dvoma vrstvami.

V okamihu keď prezentačná vrstva obdrží od užívateľa nejaký požiadavok, aplikácia vyberie z logickej vrstvy triedu, ktorá dokáže tento požiadavok obslužiť. Táto trieda môže požiadať dátovú vrstvu o to, aby vykonala potrebné operácie so súborovým systémom a týmto spôsobom získala alebo upravila niektoré dáta. Takto získané dáta prevedie do vhodného formátu a poskytne opäť prezentačnej vrstve, ktorá ich užívateľovi vykreslí.

## 5.4 Nastavovanie parametrov a inštalácia služieb

Pred samotnou inštaláciou virtuálneho servera musí užívateľ vyplniť jednotlivé parametre servera a zvoliť si inštalované služby. Vyplnenie parametrov a služieb sa delí na 2 časti:

1. vyplnenie niektorých vybraných parametrov pomocou súboru vo formáte XML
2. vyplnenie ostatných parametrov a inštalovaných služieb pomocou grafického rozhrania aplikácie (formulár)

### 5.4.1 Vyplnenie údajov pomocou súboru vo formáte XML

Pomocou súboru bude možné vyplniť nasledujúce parametre servera:

- hardvérové nastavenia virtuálneho stroja (meno virtuálneho stroja, veľkosť pamäte RAM, veľkosť videopamäte, veľkosť disku a počet CPU)
- sieťové nastavenia (typ sieťovej konfigurácie (staticky alebo dynamicky), IP adresu, sieťovú masku, predvolenú bránu a IP adresu DNS servera)
- serverové nastavenia (hostname a heslo užívateľa root)

### 5.4.2 Vyplnenie údajov pomocou grafického rozhrania aplikácie

Pomocou grafického rozhranie bude možné nasledujúce:

- vytváranie užívateľov (meno, heslo, SSH kľúč)
- nastavenie hesla pre servisný účet vagrant
- úprava konfiguračného súboru `/etc/hosts` (IP adresa a hostname)
- možnosť výberu typu virtuálneho stroja (stroj určený pre monitorovanie alebo podľa výberu inštalovaných služieb užívateľom)
- možnosť voľby inštalovaných služieb (rancid, snmp, postfix, ftp, webmin, zabbix, apache2, syslog-ng)
- možnosť voľby typu hypervízora (VirtualBox alebo KVM)
- možnosť voľby operačného systému (Debian alebo Ubuntu)
- vyplnenie popisu virtuálneho stroja
- možnosť výberu formátu exportu virtuálneho stroja (VirtualBox - OVA/OVF, KVM - qcow2/raw)
- možnosť vytvorenia boxu pre nástroj Vagrant

Pri niektorých službách bude možné ešte pred inštaláciou doplniť niektoré údaje, prípadne vytvoriť zoznamy zariadení, užívateľov a pod.

Pri službe Rancid bude možné zmeniť heslo pre užívateľa rancid, nastaviť meno a heslo pre prihlásenie do zariadení a vytvoriť zoznam sledovaných zariadení (IP adresa a hostname). Pri službe Postfix bude možné vyplniť hodnotu premennej relay host (IP adresa alebo hostname) a pri službe Zabbix bude možné vytvoriť zoznam monitorovaných zariadení (IP adresa a hostname) a vytvoriť užívateľov (meno, priezvisko, alias, skupina a heslo).

V tabuľke 5.1 je možné nájsť porovnanie spôsobov definovania parametrov inštalovaného virtuálneho servera. Časť parametrov je možné naplniť vytvorením alebo upravením súboru vo formáte XML, analýzou servera, ktorá vytvorí čiastočne definovaný súbor vo formáte XML a druhú časť parametrov je možné vyplniť pomocou grafického rozhrania aplikácie.

## 5.5 Použité virtualizačné prostredia a inštalované operačné systémy

Ako vyplýva zo zadania, že cieľom je vytvoriť virtuálny linuxový server vo formáte OVA/OVF je potrebné zvoliť technológiu pomocou ktorej sa tento cieľ



Tabuľka 5.1: Porovnanie definovania parametrov virtuálneho stroja

Parameter	XML	Analýza servera	GUI
meno virtuálneho stroja	✓	×	×
hostname	✓	✓	×
počet CPU	✓	✓	×
veľkosť pamäte RAM	✓	✓	×
veľkosť videopamäte	✓	×	×
veľkosť disku	✓	✓	×
typ konfigurácie IP adresy	✓	×	×
IP adresa	✓	✓	×
predvolená brána	✓	✓	×
maska podsiete	✓	✓	×
IP adresa DNS servera	✓	✓	×
heslo užívateľa root	✓	×	×
vytváranie užívateľov v systéme	×	×	✓
heslo užívateľa vagrant	×	×	✓
úprava /etc/hosts	×	×	✓
typ virtuálneho stroja	×	×	✓
voľba inštalovaných služieb	×	×	✓
typ hypervízora	×	×	✓
typ operačného systému	×	×	✓
popis virtuálneho stroja	×	×	✓
formát exportu VM	×	×	✓
box pre Vagrant	×	×	✓

dá dosiahnuť. Z množstva virtualizačných technológií a hypervízorov umožňuje vytvárať virtuálne stroje vo formáte OVA/OVF nástroj VirtualBox (detailne popísaný v podsekcii 3.1.2). Okrem nástroja VirtualBox bude pre vytváranie virtuálnych strojov použitý aj hypervízor KVM (detailne popísaný v podsekcii 3.1.3), ktorý podporuje formáty ako je qcow2 a raw.

Aplikácia sa bude vyvíjať pod operačným systémom linuxovej distribúcie Debian GNU/Linux verzie Jessie, no predpokladom je, že aplikácia bude fungovať aj napr. na Ubuntu. Nutnou podmienkou je, že distribúcia (na ktorej bude aplikácia používaná) bude musieť podporovať všetky potrebné nástroje a programy, ktoré aplikácia potrebuje k svojej činnosti.

Okrem toho, že sa Debian použije ako distribúcia pod ktorou sa bude aplikácia vyvíjať bude taktiež použitý ako operačný systém, ktorý sa bude inštalovať na vytvárané virtuálne stroje. Konkrétne bude použitý pre inštaláciu virtuálnych strojov operačný systém Debian (verzie 8.7.1 Jessie) bez GUI. Debian je bezplatný operačný systém podporovaný komunitou obsahujúci balíčkovací systém apt, pomocou ktorého je inštalácia nových aplikácií a služieb veľmi jednoduchá. Okrem operačného systému Debian sa bude na virtuálne

## 5. NÁVRH RIEŠENIA

---

stroje inštalovať aj linuxová distribúcia Ubuntu (serverová verzia 14.04.5 LTS - Trusty Tahr). Ubuntu je dnes najrozšírenejším zástupcom linuxových distribúcií obsahujúci taktiež balíčkovací systém apt.

---

## Popis a implementácia aplikácie

Obsahom úvodnej časti tejto kapitoly je zoznam použitých nástrojov, prostredí a programovacích jazykov. V ďalšej časti kapitoly je popísaná implementácia aplikácie a návrh jej GUI. Kapitola obsahuje aj popis štruktúry vstupného súboru vo formáte XML a použitie nástrojov Packer a Ansible. Na konci tejto kapitoly sa nachádza popis tvorby inštalačného balíčka aplikácie pre operačný systém Debian.

### 6.1 Použité nástroje a prostredia

Pri vývoji programu bola použitá linuxová distribúcia operačného systému Debian, ktorý poskytol prostredie aj pre ďalšie spolupracujúce programy akými sú Ansible [30] a Packer [31]. Pomocou nástroja Ansible sú na cieľový stroj inštalované služby ako Apache2, FTP server (vsftpd), Postfix, Rancid, Snmp, Syslog-ng, Webmin a Zabbix. Pomocou nástroja Packer sa vytvára virtuálny stroj, inštaluje sa základný operačný systém, menia sa vybrané parametre a exportuje sa virtuálny stroj.

Qt Designer [33] bol použitý pri návrhu a tvorbe GUI aplikácie a pomocou vývojového prostredia PyCharm [34] bol napísaný zdrojový kód aplikácie.

### 6.2 Použité programovacie jazyky

Na implementáciu samotnej aplikácie bol použitý programovací a skriptovací jazyk Python verzie 3.4 [35] spolu s PyQt verzie 5 [36] (multiplatformový GUI toolkit Qt pre programovací jazyk Python). Pomocou GUI toolkit PyQt5 bola naprogramovaná GUI časť aplikácie. Okrem jazyka Python bol použitý aj skriptovací jazyk Bash, v ktorom sú napísané skripty, ktoré využíva nástroj Ansible pri svojej činnosti. Pomocou YAML (formát pre serializáciu dát textových súborov) boli vytvorené tzv. playbook pre nástroj Ansible a pomocou dátového formátu YAML vytvorené konfiguračné súbory pre nástroj Packer.

### 6.2.1 Licencia

Z dôvodu licenčnej kompatibility s PyQt verzie 5 [36] musí byť kód aplikácie vydaný pod licenciou GNU GPL verzie 3 [37]. Úplne znenie licencie je súčasťou zdrojových kódov aplikácie, ktoré je možné nájsť na priloženom médiu.

## 6.3 Implementácia aplikácie

Implementácia riešenia aplikácie obsahuje niekoľko tried a funkcií:

- triedu `MainWindow` (hlavná časť aplikácie)
- dialógové triedy ako `XMLDialog`, `RancidDialog`, `CreateUserDialog`, `ErrorDialog`, `PostfixDialog`, `ZabbixDialog` a `UpdateHostsDialog`, `AnalyzeDialog`
- triedu `XML` (načítanie a tvorba súbora vo formáte XML)
- funkcie, ktoré sa starajú o zmeny v konfiguračných súboroch jednotlivých nástrojov
- funkcie, ktorých úlohou je kontrola vstupných dát zadaných užívateľom

### 6.3.1 Trieda `MainWindow`

Obsahom tejto triedy je hlavné okno aplikácie, ktoré obsluhuje jednotlivé požiadavky užívateľa. Úlohou tejto triedy je reagovať na tlačidlá, ktoré užívateľ stláča a na základe toho vytvárať ďalšie inštancie tried (dialógov), analyzovať systém a pomocou nástroja `Packer` vytvoriť virtuálny stroj. Ďalšou úlohou je prijímať od užívateľa vstupné dáta z klávesnice a analyzovať ich pomocou príslušných funkcií.

### 6.3.2 Dialógové triedy

Aplikácia obsahuje niekoľko dialógových tried, ktoré sa starajú o prijímanie vstupov od užívateľa (definovanie jednotlivých parametrov a inštalovaných služieb), o výpis chybových hlášok a o analýzu parametrov vzdialeného servera. Medzi dialógové triedy patria triedy ako:

- `XMLDialog` (spolupracuje s triedou `XML`, jej úlohou je načítanie spojené s editáciou a vytvorenie XML súbora)
- `RancidDialog` (úlohou je vytvoriť databázu zariadení, definovať heslo pre užívateľa `rancid` a nastaviť login prístup k zariadeniam)
- `CreateUserDialog` (úlohou je vytvoriť užívateľov v systéme)
- `UpdateHostsDialog` (úlohou je upraviť konfiguračný súbor `/etc/hosts`)

- PostfixDialog (úlohou je nastaviť službu Postfix)
- ZabbixDialog (úlohou je vytvoriť databázu monitorovaných zariadení a užívateľov v službe Zabbix)
- AnalyzeDialog (úlohou je analyzovať vzdialený server pomocou SSH a získať z neho parametre pre ďalšiu inštaláciu virtuálneho stroja)
- ErrorDialog (úlohou je výpis chýb, ktoré môže vzniknúť napr. pri nesprávnych vstupoch od užívateľa)

### 6.3.3 Trieda XML

Trieda XML využíva Python knižnicu ElementTree pomocou ktorej vytvára alebo načítava súbor vo formáte XML. Súčasťou tejto triedy je aj syntaktická analýza súboru XML.

### 6.3.4 Funkcie

Aplikácia obsahuje 2 rozdielne skupiny funkcií. Prvá skupina funkcií sa stará o zmeny v konfiguračných súborov nástrojov Ansible, Packer a v preseeding konfiguračnom súbore. Druhá skupina funkcií obsahuje funkcie, akými sú:

- kontrola správneho formátu IP adresy
- kontrola výskytu konkrétneho znaku v reťazci
- kontrola komplexnosti hesla (8 znakov, minimálne 1 malé, veľké písmeno a číslo)
- kontrola správneho formátu hostname
- kontrola, či je zadaný reťazec číslo nachádzajúce sa v určitých intervaloch
- kontrola, či nie je niektorý z parametrov prázdny

## 6.4 Návrh a štruktúra GUI aplikácie

Na obrázku C.1 je možné vidieť úvodnú obrazovku aplikácie. Ako je možné vidieť na obrázku, aplikácia umožňuje vytvoriť alebo načítať súbor vo formáte XML. Ďalšou možnosťou je analýza vzdialeného servera pomocou SSH a na základe tejto analýzy vytvoriť XML dialóg a definovať niektoré vybrané parametre inštalovaného servera (hostname, IP adresu, sieťová maska, predvolenú brána, IP adresu DNS servera, veľkosť pamäte RAM, počet CPU a veľkosť disku).

Obrázok C.2 zobrazuje XML dialóg, ktorý vznikne po stlačení niektorého z tlačidiel umiestnených na úvodnej obrazovke aplikácie. XML dialóg teda môže vzniknúť:

- stlačením tlačidla **Load** na úvodnej obrazovke a výberom správneho súboru vo formáte XML (načítanie údajov z XML súboru)
- stlačením tlačidla **Create** a následným vyplnením údajov užívateľom (zadanie údajov manuálne)
- stlačením tlačidla **Analyze**, vyplnením údajov o vzdialenom serveri (IP adresa servera, meno a heslo užívateľa), následnou analýzou vzdialeného servera pomocou SSH a vyplnením niektorých chýbajúcich údajov užívateľom (čiastočná analýza a zadanie chýbajúcich údajov manuálne)

Na obrázku C.3 je možné vidieť v poradí druhú obrazovku aplikácie. Pomocou nej môže užívateľ definovať užívateľov v systéme, zmeniť heslo pre servisný účet vagrant, upraviť konfiguračný súbor `/etc/hosts`, vybrať typ *appliance*<sup>17</sup> a zvoliť si inštalované služby.

Obrázok C.4 zobrazuje Zabbix dialóg vytvorený po tom, ako si užívateľ pomocou druhej obrazovky aplikácie vybral to, že chce nainštalovať službu Zabbix. Pomocou dialógu je možné vytvoriť zoznam monitorovaných zariadení, ale aj užívateľov do systému Zabbix.

Pomocou tretej obrazovky aplikácie C.5 je možné zvoliť typ hypervízora, vybrať inštalovaný operačný systém, pridať popis virtuálneho stroja pre hypervízor, zvoliť formát exportu virtuálneho stroja alebo prípadne vytvoriť box pre nástroj Vagrant.

Posledná obrazovka aplikácie C.6 zobrazuje základné informácie o inštalovanom serveri a zoznam inštalovaných služieb.

### 6.5 Vstupný súbor vo formáte XML

Jedným z možných vstupov aplikácie môže byť súbor vo formáte XML. Tento súbor je možné vytvoriť pomocou grafického rozhrania aplikácie, vytvoriť alebo upraviť ručne pomocou ľubovoľného textového editora, no v tomto prípade je potrebné dodržať nasledujúce:

- hlavička musí byť definovaná vo formáte UTF-8
- súbor musí mať definovaný tzv. root element (názov elementu je **data**)
- v prípade, že nie je uvedený niektorý z definovaných elementov, ktorý aplikácia pozná, bude hodnota parametru príslušného elementu v aplikácii nedefinovaná (prázdna)
- v prípade, že niektorý z definovaných elementov neobsahuje hodnotu, hodnota parametru príslušného elementu bude v aplikácii nedefinovaná (prázdna)

---

<sup>17</sup>systém poskytujúci špecifické zdroje

- ak je uvedený nejaký iný element (aplikácia ho nepozná) alebo je nejaký element zle uzatvorený jedná sa o chybu syntaktickej analýzy XML dokumentu

V ukážke 1 je možné vidieť typický vstupný súbor vo formáte XML.

```
<?xml version='1.0' encoding='utf-8'?>
<data>
  <name_of_virtual_machine>Test</name_of_virtual_machine>
  <hostname>testujem1235</hostname>
  <cpus>1</cpus>
  <RAM>512</RAM>
  <video_memory>32</video_memory>
  <disk_size>10000</disk_size>
  <type_of_ip_config>static</type_of_ip_config>
  <ip_address>192.168.10.1</ip_address>
  <subnet_mask>255.255.255.0</subnet_mask>
  <default_gateway>192.168.10.254</default_gateway>
  <dns_server>192.168.10.254</dns_server>
  <root_password>test12345A</root_password>
</data>
```

Listing 1: XML súbor (test.xml)

## 6.6 Packer

Packer vytvorí pomocou *preseeding*<sup>18</sup> inštalácie virtuálny stroj a zmení mu niektoré parametre (hostname, meno a popis virtuálneho stroja, veľkosť pamäte RAM, počet CPU, veľkosť videopamäte a veľkosť disku). Po inštalácii vyexportuje virtuálny stroj v zadanom formáte, prípadne vytvorí z virtuálneho stroja box pre nástroj Vagrant.

Tento nástroj je riadený šablónovými súbormi vo formáte JSON. Dôležité je tieto konfiguračné súbory vytvoriť a správne definovať ich vnútornú štruktúru. Vytvorené sú súbory pre operačné systémy Debian a Ubuntu a pre hypervízor VirtualBox a KVM. Pred samotnou inštaláciou (pred spustením nástroja Packer) musí aplikácia správne vybrať konfiguračný súbor (nástroja Packer) určený pre daný operačný systém a hypervízor, a zmeniť jeho parametre podľa parametrov, ktoré sú zadané užívateľom pomocou grafického rozhrania aplikácie. Výstupom nástroja je virtuálny stroj uložený do príslušného adresára v definovanom formáte.

<sup>18</sup>spôsob, akým je možné ešte pred samotnou inštaláciou odpovedať na otázky, ktoré sa inštalátor pýta pri inštalácii, a tým pádom už nie je nutné manuálne odpovedať na otázky počas inštalácie

### 6.6.1 Preseeding konfiguračný súbor

Preseeding konfiguračný súbor používa nástroj Packer k tomu, aby nainštaloval virtuálny stroj v základnej konfigurácii. Jeho obsahom je nastavenie času a časovej zóny, nastavenie a výber balíčkov, rozdelenie disku, nastavenie účtov, nastavenie balíčkovacieho systému apt, spustenie príkazov po inštalácii a inštalácia *grub*<sup>19</sup>. Preseeding súbor sa načítava pri inštalácii z lokálneho webového servera vytvoreného dočasne nástrojom Packer.

## 6.7 Ansible

Po tom ako nástroj Packer pomocou preseeding inštalácie nainštaluje základný systém a nastaví niektoré parametre servera spustí playbook nástroja Ansible. Jeho hlavnou úlohou je zmeniť parametre, ktoré Packer nezmenil (heslo užívateľa root, typ IP konfigurácie a prípadne IP adresu, sieťovú masku, predvolenú bránu a IP adresu DNS servera), nainštalovať a nakonfigurovať jednotlivé služby definované užívateľom ešte pred samotnou inštaláciou. Po tom ako Ansible vykoná všetky svoje operácie, nástroj Packer vypne a následne vyexportuje virtuálny stroj.

Pomocou Ansible sa spustí tzv. hlavný playbook, ktoré úlohou je zmeniť niektoré parametre servera, ale hlavne aktivovať a spustiť jednotlivé roly, ktoré taktiež obsahujú svoje vlastné playbook. Roly sú rozdelené podľa jednotlivých služieb a ich hlavnou úlohou je inštalácia a konfigurácia týchto služieb. Ich popis spolu s popisom hlavného playbook sa nachádza v nasledujúcich dvoch podsekciiach.

### 6.7.1 Hlavný playbook

Hlavný playbook sa dá rozdeliť na 3 časti:

1. úvodná časť
2. jednotlivé roly
3. záverečná časť

V úvodnej časti je úlohou Ansible pomocou playbook nainštalovať na inštalovaný virtuálny stroj základné nástroje pre systémového administrátora (screen, autossh, fail2ban a rsync), upraviť konfiguráciu SSH, nakopírovať na cieľového hostiteľa skript **hosts** (spolu so súborom vo formáte CSV, ktorého obsahom sú záznamy vo formáte IP adresa;hostname) a skript **autousers** (spolu s adresárom, kde sa nachádzajú jednotlivé užívateľské účty). Následne

---

<sup>19</sup>linuxový zavádzač



Ansible spustí na cieľovom stroji skript **autouser**, ktorý vytvorí tieto užívateľské účty v systéme. Na konci tejto úvodnej časti je úlohou nainštalovať a nakonfigurovať službu ntp, a upraviť `/etc/hosts` súbor pomocou skriptu **hosts**.

V časti jednotlivé roly sú definované len tie roly (služby), ktoré sa budú na cieľového hostiteľa inštalovať a každá z nich má vytvorenú vlastnú adresárovú štruktúru pre nástroj Ansible.

Úlohou záverečnej časti je zmeniť sieťové nastavenia (IP adresu, sieťovú masku a predvolenú bránu), upraviť konfiguračný súbor `/etc/resolv.conf` (adrese DNS servera), zmeniť heslo užívateľovi root a zakázať mu vzdialený prístup na server pomocou SSH.

### 6.7.1.1 Skript hosts

Skript **hosts** je shell skript spúšťaný príkazovým shell interpreterom Bash, ktorého úlohou je upraviť konfiguračný súbor `/etc/hosts`. Nachádza sa na cieľovom hostiteľovi v adresári `/root/scripts/hosts`. Ako vstupný súbor tento skript očakáva súbor `hosts.csv`, ktorý má definovanú svoju vlastnú vnútornú štruktúru.

Súbor `hosts.csv` je vytváraný užívateľom pomocou GUI aplikácie (aplikácia zabezpečuje správnosť hlavičky a aj správnosť vyplnenia údajov) a kopíruje sa do domovského adresára užívateľa `vagrant`. Na prvom riadku súboru musí byť hlavička vo formáte **IP address;hostname** nasledovaná jednotlivými záznamami. Po hlavičke nasledujú jednotlivé záznamy, ktoré musia spĺňať jej formát.

### 6.7.1.2 Skript autousers

Skript **autousers** je shell skript spúšťaný príkazovým shell interpreterom Bash, ktorého úlohou je vytvoriť užívateľské účty v systéme. Nachádza sa na cieľovom hostiteľovi v adresári `/root/scripts/autousers`. Vstupom skriptu sú adresáre pomenované podľa názvov užívateľov.

Užívateľov skript vytvára podľa názvov adresárov. Obsahom adresára daného užívateľa je heslo a prípadne SSH kľúč. Heslo užívateľa (pri vytvorení je použitá kryptografická hašovacia funkcia SHA-512) musí byť definované, no ak nie je definovaný SSH kľúč tak to nevedí a táto možnosť sa ignoruje. Spolu s vytvorením domovského adresára sa užívateľia pridávajú do skupiny `users`, nastaví sa im heslo a prípadne pridá SSH kľúč.

Jednotlivé užívateľské adresáre a v nich nachádzajúce sa heslá, prípadne SSH kľúče definuje a vytvorí užívateľ pomocou GUI aplikácie.

## 6.7.2 Roly

V nasledujúcej podsekcii sú popísané jednotlivé roly nástroja Ansible starajúce sa o inštaláciu a konfiguráciu služieb.

Pri výbere monitoring appliance cez GUI aplikácie sa automaticky vyberú roly snmp, postfix, apache2 a zabbix. Pri výbere custom appliance je možné definovať ľubovoľné zloženie inštalovaných rolí.

### 6.7.2.1 Snmp

Úlohou tejto roly je stiahnuť niektoré neštandardné MIB súbory a nainštalovať služby snmp, snmpd a snmp-mibs-downloader.

Táto rola upraví konfiguračný súbor, aby bolo možné stiahnuť MIB súbory. Následne sa kopírujú vybrané MIB súbory z hostiteľského stroja priamo na cieľový stroj. Tieto súbory pomáhajú službe Zabbix pri monitorovaní určitých špecifických zariadení pomocou SNMP. Poslednou úlohou je stiahnutie MIB súborov z repozitárov a reštart služby snmpd.

### 6.7.2.2 Postfix

Úlohou tejto roly je nainštalovať a nakonfigurovať službu Postfix. Túto službu bude využívať služba Zabbix na to, aby mohla posilať notifikácie užívateľom.

Táto rola nainštaluje službu Postfix, upraví jej konfiguračný súbor a nakoniec službu Postfix reštartuje.

### 6.7.2.3 Apache2

Úlohou tejto roly je nainštalovať a nakonfigurovať webový server Apache2.

Táto rola nainštaluje hlavne službu Apache2, ale aj PHP verzie 5 a ostatné podporné moduly. Po inštalácii služieb zapne moduly alias, info, status, cgi, SSL a rewrite. Následne aktivuje SSL virtuálneho hosťa a upraví konfiguráciu predvoleného hosťa. Pomocou modulu status je možné získať informácie o výkone a aktivite webového servera. Modul info poskytuje prehľad serverovej konfigurácie a oba moduly sú prístupné len z *localhost*<sup>20</sup>. V poslednom kroku definuje nový logovací formát a reštartuje službu Apache2.

### 6.7.2.4 Zabbix

Hlavnou úlohou tejto roly je nainštalovať službu Zabbix verzie 3.2 a automaticky vytvoriť zoznam monitorovaných zariadení a užívateľov v službe Zabbix.

Táto rola pridá do repozitára službu Zabbix. Následne nainštaluje podporné moduly a služby (MySQL, PHP a Zabbix server). Ďalšou úlohou je nakopírovať skript **zabbix\_hosts** (spolu so súborom vo formáte CSV, ktorého obsahom je zoznam monitorovaných zariadení) a skript **zabbix\_users** (spolu so súborom vo formáte CSV, ktorého obsahom je zoznam užívateľov). Ďalším krokom je správne nastavenie a nakonfigurovanie databázy, a úprava

---

<sup>20</sup>v počítačovej terminológii to znamená odkaz na práve používaný počítač

konfiguračných súborov služieb akými sú Zabbix, Apache2 a PHP. V poslednom kroku reštartuje jednotlivé služby a automaticky spustí na cieľovom stroji oba skripty.

Skript **zabbix\_hosts** je shell skript spúšťaný príkazovým shell interpretom Bash, ktorého úlohou je vytvoriť zoznam monitorovaných zariadení. Nachádza sa na cieľovom hostiteľovi v adresári `/root/scripts/zabbix_hosts`. Vstupom skriptu je súbor vo formáte CSV, ktorý na prvom riadku obsahuje hlavičku vo formáte **IP;Host**. Po hlavičke nasledujú jednotlivé záznamy, ktoré musia spĺňať jej formát. Súbor je vytvorený užívateľom pomocou GUI aplikácie, ktorá zabezpečí správnosť vyplnenia údajov a kopíruje sa do domovského adresára servisného účtu vagrant. Hlavný skript obsahuje ďalšie 3 skripty spúšťané v nasledujúcom poradí:

1. skript, úlohou ktorého je získať token k API služby Zabbix
2. skript vytvorí zoznam monitorovaných zariadení
3. skript odstráni monitorované zariadenia, ktoré sa nachádzali v službe Zabbix, ale už neboli obsahom vstupného súboru

Skript **zabbix\_users** je shell skript spúšťaný príkazovým shell interpretom Bash, ktorého úlohou je vytvoriť užívateľské účty v službe Zabbix. Nachádza sa na cieľovom hostiteľovi v adresári `/root/scripts/zabbix_users`. Vstupom skriptu je súbor vo formáte CSV, ktorý na prvom riadku obsahuje hlavičku vo formáte **Name;Surname;Alias;Password;Group**. Po hlavičke nasledujú jednotlivé záznamy, ktoré musia spĺňať jej formát. Súbor je vytvorený užívateľom pomocou GUI aplikácie, ktorá zabezpečí správnosť vyplnenia údajov a kopíruje sa do domovského adresára servisného účtu vagrant. Hlavný skript obsahuje ďalšie 3 skripty spúšťané v nasledujúcom poradí:

1. skript, úlohou ktorého je získať token k API služby Zabbix
2. skript vytvorí užívateľov
3. skript odstráni užívateľov, ktorí sa síce nachádzali v službe Zabbix, ale už neboli obsahom vstupného súboru

#### 6.7.2.5 Webmin

Hlavnou úlohou tejto roly je nainštalovať službu Webmin, ktorá pomôže menej skúseným užívateľom pri upravovaní konfiguračných súborov a pri správe serverových služieb.

Táto rola pridá službu Webmin do repozitára a následne zabezpečí jej stiahnutie a nainštalovanie. Súčasťou tejto roly je len základná inštalácia bez ďalšej konfigurácie.

### 6.7.2.6 Rancid

Úlohou tejto roly je nainštalovať službu Rancid, ktorá sa postará o sťahovanie konfigurácií jednotlivých zariadení.

Najprv sa prekopíruje skript **rancid\_database** spolu so súborom vo formáte CSV, ktorého obsahom je zoznam zariadení, z ktorých sa budú sťahovať konfigurácie. Potom sa nainštalujú podporné služby a pomocou skriptu sa nainštaluje a následne nakonfiguruje služba Rancid a nastaví sa heslo pre užívateľa rancid. Následne sa na cieľovom stroji spustí skript, ktorý vytvorí databázu prvkov a aktualizuje sa *cron*<sup>21</sup>. Do cron sa pridajú úlohy ako je spúšťanie služby Rancid a mazanie logov v pravidelných časových intervaloch. Poslednou úlohou je reštartovanie služby Rancid.

Skript **rancid\_database** je shell skript spúšťaný príkazovým shell interpretom Bash, ktorého úlohou je vytvoriť databázu prvkov pre službu Rancid a aktualizovať súbor `/etc/hosts`. Nachádza sa na cieľovom hostiteľovi v adresári `/root/scripts/rancid_database`. Vstupom skriptu je súbor vo formáte CSV, ktorý na prvom riadku obsahuje hlavičku vo formáte **IP address;hostname**. Po hlavičke nasledujú jednotlivé záznamy, ktoré musia spĺňať jej formát. Súbor je vytvorený užívateľom pomocou GUI aplikácie, ktorá zabezpečí správnosť vyplnenia údajov a kopíruje sa do domovského adresára servisného účtu vagrant.

### 6.7.2.7 Syslog-ng

Úlohou tejto roly je nainštalovať službu Syslog-ng, teda pripraviť systém na prijímanie a ukladanie logov do definovaných adresárov. Logy zo zariadení sa budú ukladať do adresára `/var/log/remote-devices/zariadenie/log`. Názov adresára **zariadenie** je v tvare meno zariadenia-facility-dátum logu.

### 6.7.2.8 Ftp

Úlohou tejto roly je nainštalovať FTP server (vsftpd) spolu s grafickým webovým rozhraním net2ftp pomocou ktorého je možné sťahovať, nahrávať (pomocou FTP) a prehľadávať jednotlivé súbory a adresáre.

Táto rola nainštaluje a zmení konfiguráciu FTP servera vsftpd, vytvorí adresár pre službu FTP a symbolický link na tento adresár. Ďalej nainštaluje a zmení konfiguráciu net2ftp, pomocou ktorého bude možné pristupovať k FTP adresáru pomocou webového rozhrania.

---

<sup>21</sup>program, ktorý poskytuje možnosť automaticky spúšťať nejaký príkaz resp. proces (skript, program a pod.) v pravidelných časových intervaloch

## 6.8 Obsah a tvorba balíčka vo formáte deb

Aplikácia je distribuovaná pomocou Debian balíčka spolu s konfiguračnými súborami nástrojov Packer a Ansible. V nasledujúcej sekcii je popísaný obsah a proces tvorby balíčka.

Ešte pred tvorbou adresárovej štruktúry balíčka je potrebné vytvoriť zo zdrojových súborov spustiteľnú aplikáciu. Pomocou nástroja `pyinstaller` príkazom:

```
$ pyinstaller --onefile main.py --name virtual-creator
```

vytvoríme spustiteľný súbor aplikácie `virtual-creator`, pomocou ktorého bude možné aplikáciu spustiť.

Obsahom adresárovej štruktúry balíčka je adresár `usr` a `DEBIAN`. Adresár `usr` obsahuje ďalšie 2 adresáre, ktoré sa pri inštalácii balíčka kopírujú na cieľového hostiteľa. Prvý adresár je `/usr/bin` obsahujúci spustiteľný súbor nástroja Packer (`packer`) a druhý adresár je `/usr/share/virtual-machine-creator` obsahujúci adresáre `ansible`, `application` a `packer`. V adresároch `ansible` a `packer` sa nachádzajú konfiguračné súbory nástrojov Ansible a Packer. V adresári `application` sa nachádza spustiteľný súbor aplikácie, adresár `dialogues` a adresár `xml`, kde sú uložené ukážkové súbory vo formáte XML.

Adresár `DEBIAN` obsahuje súbory `control` a `md5sums`, predinštalovaný skript `preinst`, poínštalovaný skript `postinst` a skript `postrm`, ktorý sa spustí pri odinštalovaní aplikácie. Súbor `control` obsahuje informácie o balíčku ako je meno autora, verzia, popis, závislosti atď. Medzi závislosti aplikácie patria `dkms`, `build-essential`, `virtualbox`, `qemu-kvm`, `virt-manager`, `gksu` alebo `kdesu` a `ansible`. Okrem súboru `control` sa v adresári nachádza aj súbor `md5sums` obsahujúci kontrolné súčty umožňujúce overiť integritu. Kontrolný súčet zaistí to, že balíčkovací systém identifikuje prípadne poškodenie balíčka a nedovolí balíček nainštalovať.

Obsahom skriptu `preinst` je pridanie repozitára kvôli inštalácii najnovšej verzie nástroja Ansible (2.3.0), pretože systém pre ktorý je primárne aplikácia určená (Debian verzia 8.7) obsahuje vo svojich základných repozitároch Ansible verzie 1.7.2. Obsahom skriptu `postinst` je príkaz `modprobe vboxdrv` zabezpečujúci správnu funkciu nástroja VirtualBox. Obsahom skriptu `postinst` je aj príkaz, ktorý vytvorí symbolický link na spustiteľný súbor aplikácie do adresára `/usr/local/bin` tak, aby bolo možné aplikáciu spustiť príkazom `virtual-creator`. Obsahom skriptu `postrm` je odstránenie symbolického linku na spustiteľný súbor aplikácie. Inštalovaný balíček aplikácie sa vytvorí príkazom:

```
$ dpkg-deb -b package \  
> virtual-machine-creator-1.0_amd64.deb
```

## 6. POPIS A IMPLEMENTÁCIA APLIKÁCIE

---

Súčasťou aplikácie sú inštalačné súbory (bez úprav) operačných systémov Debian [38] a Ubuntu [39] vo formáte ISO, ktoré však nie sú súčasťou inštalačného balíčka kvôli ich veľkosti. Postup inštalácie balíčka spolu s návodom, kde tieto inštalačné súbory, ktoré sú súčasťou priloženého DVD nakopírovať sa nachádza v dodatku B.

---

# Testovanie programu a inštalovaných služieb

## 7.1 Priebeh testovania

Program bol priebežne testovaný pri vývoji na lokálnom počítači s operačným systémom Debian verzie 8.7 (8192 MB RAM) a vo virtualizovanom prostredí VMware, ktoré podporovalo vytváranie 64-bitových virtuálnych strojov (operačný systém Debian verzie 8 Jessie, 4096 MB RAM, 4 CPU). Po vytvorení hlavných častí aplikácie bol vytvorený inštalačný balíček v formáte deb a následne bolo otestované správne nainštalovanie alebo odinštalovanie balíčka a programová funkčnosť aplikácie. V nasledujúcich bodoch sú uvedené jednotlivé testy programovej funkčnosti aplikácie zamerané na vytváranie virtuálnych strojov s rôznymi parametrami a inštalovanými službami:

1. virtuálny stroj – načítanie parametrov už z vytvoreného XML súboru test.xml (bez zmeny parametrov), statická IP konfigurácia, vytvorenie troch užívateľov v systéme, nastavenie hesla pre užívateľa vagrant, pridanie jedného záznamu do konfiguračného súboru /etc/hosts, výber monitoring appliance (služby Postfix, Zabbix, Apache2, Snmp), nastavenie premennej relay host služby Postfix, vytvorenie nového zariadenia a užívateľa do služby Zabbix, hypervízor VirtualBox, operačný systém Debian, formát exportu virtuálneho stroja OVA bez vytvorenia box pre nástroj Vagrant (doba inštalácie vo virtualizovanom prostredí spojená s exportom  $\approx$  30 minút)
2. virtuálny stroj – načítanie parametrov už z vytvoreného XML súboru test\_dhcp.xml (úprava hesla), dynamická IP konfigurácia, vytvorenie jedného užívateľa v systéme s SSH kľúčom, nastavenie hesla pre užívateľa vagrant, bez zmeny konfiguračného súboru /etc/hosts, výber custom appliance (služby Webmin, Rancid, Ftp, Apache2), definovanie prístupu

a vytvorenie jedného zariadenia pre službu Rancid, hypervízor Virtual-Box, operačný systém Ubuntu, formát exportu virtuálneho stroja OVF spolu s vytvorením box pre nástroj Vagrant (doba inštalácie vo virtualizovanom prostredí spojená s exportom a vytvorením box pre nástroj Vagrant  $\approx$  45 minút)

- virtuálny stroj – definovanie parametrov analýzou servera pomocou aplikácie, zmenou a doplnením parametrov ručne (bez uloženia), dynamická IP konfigurácia, nastavenie hesla pre užívateľa vagrant, bez vytvorenia nového užívateľa a s pridaním troch záznamov do konfiguračného súboru `/etc/hosts`, custom appliance (inštalácia služieb Syslog-ng, Ftp a Apache2), hypervízor KVM, operačný systém Ubuntu, formát exportu qcow2, bez vytvorenia box pre nástroj Vagrant (doba inštalácie na fyzickom stroji spojená s exportom a bez vytvorenia box pre nástroj Vagrant  $\approx$  25 minút)
- virtuálny stroj – definovanie parametrov ručne užívateľom pomocou aplikácie (spojené s uložením do súboru vo formáte XML), statická IP konfigurácia, nastavenie hesla pre užívateľa vagrant, vytvorenie 3 nových užívateľov v systéme a bez zmeny konfiguračného súboru `/etc/hosts`, custom appliance (inštalácia všetkých služieb spojená s definovaním parametrov), hypervízor VirtualBox, operačný systém Debian, formát exportu OVA bez vytvorenia box pre nástroj Vagrant (doba inštalácie na fyzickom stroji spojená s exportom a bez vytvorenia box pre nástroj Vagrant  $\approx$  20 minút)

### 7.2 Vyhodnotenie testovania

Pri testovaní sa podarilo odhaliť a opraviť niekoľko chýb, ktoré vznikli pri inštalácii balíčka aplikácie, ale aj pri samotnom inštalovaní virtuálnych strojov pomocou aplikácie.



---

## Záver

Táto práca mala za cieľ realizovať nástroj pomocou ktorého bude možné jednoducho a efektívne vytvárať linuxové virtuálne stroje postavené na operačných systémoch ako je Ubuntu a Debian pre hypervízor KVM a VirtualBox.

Tvorba nástroja bola inšpirovaná do určitej miery podobnými nástrojmi a vychádzala z porovnaní už vytvorených nástrojov, ktoré sa podobnej problematike venujú. Existuje viacero možností alebo nástrojov, pomocou ktorých je možné finálnu aplikáciu vytvoriť. Implementácia a návrh celej aplikácie pomocou nástroja Vagrant nebola tá správna cesta, a preto sa neskôr pri návrhu a implementácii prešlo na nástroj Packer.

Vytvorená desktopová aplikácia umožňuje jednoducho a komfortne vytvárať virtuálne stroje a poskytuje užívateľovi grafické rozhranie pre ľahšiu konfiguráciu jednotlivých parametrov a služieb virtuálneho stroja.

Jednotlivé vlastnosti a funkcie podporných nástrojov Ansible a Packer sa dajú nastaviť pomocou konfiguračných súborov. Táto možnosť teda umožňuje administrátorovi alebo užívateľovi v určitej miere prispôsobiť aplikáciu jeho požiadavkám. Celá aplikácia bola integrovaná a testovaná pod operačným systémom linuxovej distribúcie Debian pre ktorý je aj primárne určená.

Najbližšie rozšírenie aplikácie by mohlo obsahovať rozšírenie inštalovaných operačných systémov, pridanie nových inštalovaných služieb, prípadne možnosť definovať aj iné parametre servera, ktoré je možné pomocou podporných nástrojov meniť. Vzhľadom ku štruktúre aplikácie by nemala byť implementácia týchto rozšírení problematická. Jednou z väčších zmien by mohlo byť prepísanie vnútornej štruktúry aplikácie tak, žeby sa všetky parametre nachádzali v jednom súbore vo formáte XML a nebolo potrebné nastavovať niektoré vybrané parametre servera ručne, pomocou GUI aplikácie.

Výsledná implementácia aplikácie umožňuje inštaláciu jedného virtuálneho servera pre hypervízor VirtualBox alebo KVM. Ďalšie vylepšenie aplikácie by mohlo spočívať práve v paralelnej inštalácii viacerých virtuálnych serverov.

Pre uľahčenie inštalácie vytvorenej aplikácie bol vytvorený inštalateľný balíček `virtual-machine-creator-1.0_amd64.deb`.

## ZÁVER

---

Všetky ciele bakalárske práce a požiadavky stanovené na aplikáciu sa podarilo splniť. Výsledná implementácia aplikácie slúžiaca na vytváranie linuxových virtuálnych strojov sa nachádza na priloženom médiu.

---

## Literatúra

- [1] Traditional and Virtual architecture [obrázok]. [cit. 2017-03-14]. Dostupné z: [http://cdn.ttgtmedia.com/rms/onlineImages/server\\_virtualization-traditional\\_virtual\\_architecture.jpg](http://cdn.ttgtmedia.com/rms/onlineImages/server_virtualization-traditional_virtual_architecture.jpg)
- [2] Virtualization in the trenches with VMware, Part 1: Basics and benefits [obrázok]. [cit. 2017-03-16]. Dostupné z: <https://cdn.arstechnica.net/wp-content/uploads/2011/02/part1-hardware-virtualization2.png>
- [3] VMware Basics [obrázok]. [cit. 2017-03-17]. Dostupné z: [https://www.starline.de/typo3temp/pics/VMware-Basics\\_ENG\\_1b4c3ab65f.jpg](https://www.starline.de/typo3temp/pics/VMware-Basics_ENG_1b4c3ab65f.jpg)
- [4] Xen arch diagram [obrázok]. [cit. 2017-03-17]. Dostupné z: [https://wiki.xenproject.org/images/6/63/Xen\\_Arch\\_Diagram.png](https://wiki.xenproject.org/images/6/63/Xen_Arch_Diagram.png)
- [5] Your Spacewalk [obrázok]. [cit. 2017-03-29]. Dostupné z: [http://spacewalk.redhat.com/img/screenshots/your\\_spacewalk.png](http://spacewalk.redhat.com/img/screenshots/your_spacewalk.png)
- [6] FAI project slideshow (prvý obrázok zo slideshow) [obrázok]. [cit. 2017-03-29]. Dostupné z: <http://fai-project.org/slideshow/>
- [7] Automate and manage systems installation with Cobbler [obrázok]. [cit. 2017-03-29]. Dostupné z: <https://www.ibm.com/developerworks/library/l-cobbler/fig-2.jpg>
- [8] Foreman reports applied [obrázok]. [cit. 2017-03-29]. Dostupné z: [https://theforeman.org/plugins/foreman\\_chef/0.1/reports\\_applied.png](https://theforeman.org/plugins/foreman_chef/0.1/reports_applied.png)
- [9] Smith, J.; Nair, R.: *Virtual Machines: Versatile Platforms for Systems and Processes*. San Francisco: Morgan Kaufmann, prvé vydání, jún 2005, ISBN 978-1-55860-910-5.

- [10] Rouse, M.; Kirsch, B.: Virtualization. *TechTarget*, október 2016, [cit. 2017-03-14]. Dostupné z: <http://searchservervirtualization.techtarget.com/definition/virtualization>
- [11] Shrivastava, A.: What is full virtualization, partial virtualization and paravirtualization? *Quora*, február 2016, [cit. 2017-03-18]. Dostupné z: <http://searchservervirtualization.techtarget.com/definition/virtual-machine>
- [12] Marshall, D.: Top 10 benefits of server virtualization. *InfoWorld*, november 2011, [cit. 2017-03-14]. Dostupné z: <http://www.infoworld.com/article/2621446/server-virtualization/server-virtualization-top-10-benefits-of-server-virtualization.html>
- [13] Pietroforte, M.: Seven disadvantages of server virtualization. *4synops*, júl 2008, [cit. 2017-03-14]. Dostupné z: <https://4sysops.com/archives/seven-disadvantages-of-server-virtualization/>
- [14] Rouse, M.: guest virtual machine (guest VM). *TechTarget*, jún 2011, [cit. 2017-03-16]. Dostupné z: <https://4sysops.com/archives/seven-disadvantages-of-server-virtualization/>
- [15] Rouse, M.; Kirsch, B.: virtual machine (VM). *TechTarget*, júl 2016, [cit. 2017-03-16]. Dostupné z: <http://searchservervirtualization.techtarget.com/definition/virtual-machine>
- [16] Sridharan, S.: *A Performance Comparison of Hypervisors for Cloud Computing*. Diplomová práca, University of North Florida School of Computing, Jacksonville, FL 32224, USA, 8 2012.
- [17] VMware, Inc.: *VMware ESX Server [online]*. [cit. 2011-07-07]. Dostupné z: [http://www.vmware.com/pdf/esx\\_datasheet.pdf](http://www.vmware.com/pdf/esx_datasheet.pdf)
- [18] Oracle VM VirtualBox: *Chapter 1. First steps [online]*. [cit. 2017-03-18]. Dostupné z: <https://www.virtualbox.org/manual/ch01.html>
- [19] KVM: *Kernel Virtual Machine [online]*. [cit. 2017-03-18]. Dostupné z: [https://www.linux-kvm.org/page/Main\\_Page](https://www.linux-kvm.org/page/Main_Page)
- [20] International Business Machines Corporation: *KVM overview [online]*. [cit. 2017-03-18]. Dostupné z: <https://www.ibm.com/support/knowledgecenter/linuxonibm/liaat/liaatkvmover.htm>
- [21] Xen Project: *Xen Project Software Overview [online]*. [cit. 2017-03-18]. Dostupné z: [https://wiki.xenproject.org/wiki/Xen\\_Project\\_Software\\_Overview](https://wiki.xenproject.org/wiki/Xen_Project_Software_Overview)

- 
- [22] HashiCorp, Inc.: *WHY VAGRANT?* [online]. [cit. 2017-03-19]. Dostupné z: <https://www.vagrantup.com/docs/why-vagrant/>
- [23] HashiCorp, Inc.: *Introduction to Packer* [online]. [cit. 2017-03-19]. Dostupné z: <https://www.packer.io/intro/>
- [24] UpGuard: [Infographic] Ansible vs Chef. *UpGuard*, 2016, [cit. 2016-12-06]. Dostupné z: <https://www.upguard.com/articles/ansible-vs-chef>
- [25] Rouse, M.: Puppet (Puppet Labs). *TechTarget*, január 2017, [cit. 2017-03-19]. Dostupné z: <http://searchenterprise-linux.techtarget.com/definition/Puppet>
- [26] Red Hat, Inc.: *Free & Open Source Systems Management* [online]. [cit. 2017-03-27]. Dostupné z: <http://spacewalk.redhat.com/>
- [27] fai-project: *FAI - Fully Automatic Installation* [online]. [cit. 2017-03-28]. Dostupné z: <http://fai-project.org/>
- [28] Cobbler: *Welcome to Cobbler!* [online]. [cit. 2017-03-28]. Dostupné z: <http://cobbler.github.io/>
- [29] Foreman: *What is Foreman?* [online]. [cit. 2017-03-28]. Dostupné z: <https://theforeman.org/introduction.html>
- [30] Ansible Community / Ansible, Inc. / Red Hat, Inc. Ansible 2.2.1.0 [software]. [prístup 1. marca 2017]. Dostupné z: <https://www.ansible.com/>
- [31] HashiCorp, Inc. Packer v1.0.0 [software]. [prístup 27. apríla 2017]. Dostupné z: <https://www.packer.io/>
- [32] Apple, Inc.: *Model-View-Controller* [online]. [cit. 2017-04-16]. Dostupné z: <https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>
- [33] Qt Group plc. Qt Designer 5.3.2 [software]. [prístup 16. marca 2017]. Dostupné z: <https://www.qt.io/>
- [34] JetBrains. PyCharm Community Edition 2017.1 [software]. [prístup 1. marca 2017]. Dostupné z: <https://www.jetbrains.com/pycharm/>
- [35] Python Software Foundation. Python 3.4.0 [software]. [prístup 1. marca 2017]. Dostupné z: <https://www.python.org/download/releases/3.4.0/>
- [36] Riverbank Computing. PyQt5 5.8.2 [software]. [prístup 1. marca 2017]. Dostupné z: <https://pypi.python.org/pypi/PyQt5>

- [37] FREE SOFTWARE FOUNDATION, Inc.: *GNU General Public License [online]*. [cit. 2017-04-25]. Dostupné z: <https://www.gnu.org/licenses/gpl-3.0.en.html>
- [38] Debian Project (Software in the Public Interest). Debian Jessie 8.7.5 [software]. [prístup 1. marca 2017]. Dostupné z: <http://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-8.7.1-amd64-netinst.iso>
- [39] Canonical Ltd., Ubuntu community. Ubuntu 14.04.5 [software]. [prístup 1. marca 2017]. Dostupné z: <http://releases.ubuntu.com/14.04/ubuntu-14.04.5-server-amd64.iso>

## Zoznam použitých skratiek

**ACPI** Advanced Configuration and Power Interface

**API** Application programming interface

**AWD** Amazon Web Services

**CPU** Central Processing Unit

**CSV** Comma-separated values

**DHCP** Dynamic Host Configuration Protocol

**FAI** Fully Automatic Installation

**GFS2** Global File System

**GNU GPL** GNU General Public License

**GUI** Graphical User Interface

**HVM** Hardware Virtual Machine

**IP** Internet Protocol

**IaaS** Infrastructure as a Service

**I/O** Input/Output

**JSON** JavaScript Object Notation

**KSM** Kernel Same-page Merging

**KVM** Kernel-based Virtual Machine

**LVM** Logical Volume Manager

**MIB** Management Information Base

## A. ZOZNAM POUŽITÝCH SKRATIEK

---

**MVC** Model-View-Controller

**NFS** Network File System

**OVA/OVF** Open Virtualization Format

**PXE** Preboot eXecution Enviroment

**RAID** Redundant Array of Independent Disks

**RAM** Random Access Memory

**RDBMS** Relational database management syste

**SAN** Storage Area Network

**SCSI** Small Computer System Interface

**SELinux** Security-Enhanced Linux

**SHA** Secure Hash Algorithm

**SSL** Secure Sockets Layer

**SMP** Symmetric multiprocessing

**SNMP** Simple Network Management Protocol

**SSH** Secure Shell

**TFTP** Trivial File Transfer Protocol

**USB** Universal Serial Bus

**UTF** Unicode Transformation Format

**VM** Virtual Machine

**VRDE** VirtualBox Remote Desktop Extension

**XML** Extensible Markup Language

**YAML** YAML Ain't Markup Language



---

# Užívateľská príručka

## B.1 Popis aplikácie

Aplikácia je určená na vytváranie virtuálnych strojov založených na operačných systémoch linuxových distribúcií Ubuntu a Debian. Jej hlavnou úlohou je nastavovať vybrané parametre inštalovaného servera, inštalovať vybrané služby a niektoré služby aj konfigurovať. Výstupom aplikácie je virtuálny stroj pre hypervízor VirtualBox alebo KVM, ktorý je uložený vo formáte, ktorý si užívateľ vybral pomocou GUI aplikácie.

## B.2 Požiadavky

Aplikáciu je možné inštalovať na systémy využívajúce balíčkovací systém apt podporujúci inštalovanie balíčkov formátu deb. Aplikácia bola inštalovaná a testovaná pod linuxovou distribúciou operačného systému Debian verzie 8 (Jessie). Aplikáciu nie je vhodné testovať napr. v nástroji VirtualBox, pretože vzniká problém vytvorenia virtuálneho stroja vo virtuálnom stroji (vnorená virtualizácia). Nie žeby to nebolo možné, ale aplikácia inštaluje 64-bitové systémy a VirtualBox podporuje pomocou vnorenej virtualizácie inštaláciu len 32-bitových virtuálnych strojov.

Aplikáciu je teda vhodné testovať na fyzickom stroji, prípadne v prostredí, ktoré dovoľuje pomocou vnorenej virtualizácie vytvárať aj 64-bitové virtuálne stroje. Na fyzickom stroji (v BIOS), prípadne vo virtualizovanom prostredí, v ktorom sa to testuje musí byť povolená virtualizačná technológia Intel VT-x/EPT alebo AMD-V/RV, musí byť dostupné pripojenie k internetu a dostatok miesta na disku.

V prípade testovania vo virtualizovanom prostredí je okrem vyššie spomínaných obmedzení potrebné mať ešte nainštalovaný napr. v prípade nástroja VMware softvérový balíček VMware tools.

### B.3 Inštalácia

Postup inštalácie aplikácie (pod užívateľom root) je rozdelený do niekoľkých bodov:

- spustiť príkaz na inštaláciu balíčka aplikácie:  

```
$ dpkg -i virtual-machine-creator-1.0_amd64.deb
```
- aktualizovať systém a nainštalovať jednotlivé závislosti aplikácie príkazom:  

```
$ apt-get update && apt-get install -f -y
```
- skopírovať obsah adresára ISO z priloženého DVD média (inštaláčne súbory operačných systémov Debian [38] a Ubuntu [39]) do adresára **/usr/share/virtual-machine-creator/packer/iso**

Po inštalácii aplikácie je vhodné nainštalovať balíček VirtualBox Guest Additions, ktorý optimalizuje operačné systémy virtuálnych strojov pre lepší výkon a použiteľnosť (inštaláciu balíčka je možné spustiť priamo pomocou GUI aplikácie nástroja VirtualBox).

### B.4 Spustenie a používanie aplikácie

Inštalčný balíček vytvorí symbolický link na spustiteľný súbor aplikácie do adresára /usr/local/bin. Aplikácia vyžaduje práva užívateľa root, a preto je aplikáciu potrebné spustiť s týmito právami. Keďže sa jedná o GUI aplikáciu je potrebné sa prepnúť na non-root užívateľa a spustiť aplikáciu a následne zadať heslo užívateľa root príkazom:

```
$ gksu virtual-creator
```

alebo

```
$ kdesu virtual-creator
```

Pre interakciu s užívateľom je aplikácia vytvorená ako desktopová aplikácia s interaktívnym GUI pomocou ktorého užívateľ aplikáciu ovláda. Aplikácia sa skladá z hlavného okna aplikácie a je navrhnutá formou inštalátora, to znamená, že až po správnom vyplnení všetkých povinných údajov v danom dialógovom okne aplikácia povolí prístup k nasledujúcemu dialógovému oknu.

Pomocou interaktívneho GUI aplikácie, ktorého časť je možné vidieť v dodatku C je možné jednoducho načítať alebo vytvoriť XML súbor obsahujúci niektoré parametre servera, nastavovať ďalšie vybrané parametre a vybrať inštalované služby. Pri vytváraní užívateľov, zariadení alebo pri zmenách konfigurácií niektorých služieb pomocou dialógových okien je potrebné dať pozor

na to, že pri každom ďalšom otvorení toho istého dialógového okna dochádza k vytvoreniu nového konfiguračného súboru, súboru s užívateľmi alebo so zariadeniami a v tomto prípade je potrebné celý zoznam (zariadení alebo užívateľov) definovať odznova.

Kliknutím na tlačidlo **Create virtual machine** (umiestneného v poslednom okne GUI aplikácie) sa spustí terminálové okno a v ňom séria príkazov, ktorá spustí nástroj Packer, ktorý virtuálny stroj s definovanými parametrami a službami nainštaluje a vyexportuje do cieľového formátu.

Virtuálny stroj je po inštalácii a následnom exporte uložený do adresára `/usr/share/virtual-machine-creator` podľa typu operačného systému a hypervízora v tvare `packer/(ubuntu|debian)/virtual_machines/(virtualbox-iso|qemu-kvm)/názov_stroja` s užívateľskými právami (600) pre užívateľa root. Preto je pri testovaní virtuálneho stroja potrebné použiť užívateľa root alebo zmeniť práva k exportovanému súboru.

Čas inštalácie virtuálneho stroja závisí hlavne od rýchlosti pripojenia k internetu a od hardvérových parametrov stroja, pod ktorým sa aplikácia používa.

Pri používaní aplikácií je potrebné dodržať nasledujúce veci:

- počas vytvárania virtuálneho stroja nie je vhodné manipulovať s inštalovaným virtuálnym strojom, pretože môže dôjsť k prerušeniu automatickej inštalácie (v prípade, že VirtualBox zobrazí pri inštalácii nejaké dialógové okno je potrebné toto dialógové okno zavrieť, aby sa správne nainštaloval virtuálny stroj)
- aplikácia potrebuje k svojej správnej funkčnosti užívateľské práva užívateľa root
- pri definovaní parametrov ako je veľkosť RAM, disku a počet CPU je potrebné dbať na hardvérové parametre stroja, pod ktorým sa inštaluje virtuálny stroj (v prípade chyby alebo prekročenia hardvérových limitov sa virtuálny stroj nenainštaluje)
- pri analýze vzdialeného servera, ktorý nebol vytvorený aplikáciou je potrebné dbať na správne nastavenie SSH služby na tomto serveri (možnosť prihlásenia sa cez SSH na non-root užívateľa pomocou hesla)
- pri analýze vzdialeného servera, v prípade, že nefunguje prístup na vzdialený server môže byť problém s kľúčom v súbore `known_hosts`, a preto je vhodné v tomto prípade použiť príkaz (pod užívateľom root) na jeho odstránenie:

```
$ ssh-keygen -R IP_adresa_servera
```

Pri analýze servera pomocou aplikácie sa v prípade parametra veľkosť disku analyzuje len veľkosť root adresára / a pomocou aplikácie nie je možné inštalovať paralelne viacero virtuálnych strojov.

## B.5 Popis a funkcie grafických objektov aplikácie

### B.5.1 Tlačidlá

- CREATE – zobrazenie formulára na vytvorenie súboru vo formáte XML pomocou aplikácie (definovanie parametrov)
- LOAD – načítanie parametrov zo súboru vo formáte XML
- ANALYZE SERVER – analýza vzdialeného servera a získanie parametrov
- Create users – vytvorenie užívateľov, ktorí budú vytvorený v inštalovanom systéme
- Update file – aktualizácia konfiguračného súboru /etc/hosts
- Create virtual machine – finálne vytvorenie virtuálneho stroja

### B.5.2 Iné objekty

- Password for user vagrant – zmena hesla pre užívateľa vagrant
- Type of appliance – možnosť výberu typu appliance
- Services – možnosť výberu inštalovaných služieb
- Hypervisor – možnosť výberu typu hypervízora
- OS type – možnosť výberu typu operačného systému
- Virtual machine description – popis virtuálneho stroja
- Export options – formát exportu virtuálneho stroja
- Basic informations – informácie o parametroch inštalovaného virtuálneho stroja
- Informations about services – informácie o inštalovaných službách

## B.6 Popis a prístup k nainštalovaným službám

V nasledujúcej sekcii je v bodoch popísané to, čo aplikácia mení alebo inštaluje pri inštalovaní jednotlivých služieb a aj to ako je možné pristupovať k webovým rozhraniám vybraných služieb.

## B.6.1 Popis zmien konfigurácií jednotlivých služieb

Pre detailnejšie informácie o tom, čo všetko sa mení pri inštalácií jednotlivých službách je vhodné preskúmať adresár ansible, ktorý je súčasťou aplikácie a nachádza sa aj na priloženom médiu.

### B.6.1.1 Apache2

- inštalácia apache2 a php5
- zmena LogLevel a LogFormat v `/etc/apache2/apache2.conf`
- povolenie modulov alias, info, status, cgid, ssl, rewrite
- povolenie info a status modulu zmenou konfigurácie `/etc/apache2/sites-enabled/000-default.conf`

### B.6.1.2 Ftp

- inštalácia vsftpd a net2ftp
- vytvorenie adresára `/home/ftproot` a symbolického linku `/ftproot`
- zmena konfigurácie v `/etc/vsftpd.conf` a `/etc/vsftpd.chroot_list`

### B.6.1.3 Postfix

- inštalácia postfix a zmena konfigurácie v `/etc/postfix/main.cf`

### B.6.1.4 Rancid

- inštalácia rancid, cvs a expect
- zmena v `/home/rancid/etc/rancid.conf` a `/home/rancid/.cloginrc`
- vytvorenie rancid databáze a pridanie záznamov do cron

### B.6.1.5 Snmp

- inštalácia snmp, snmpd a snmp-mibs-downloader
- kopírovanie MIB do `/usr/share/snmp/mibs` a inštalácia MIB

### B.6.1.6 Syslog-ng

- inštalácia syslog-ng-core a syslog-ng
- zmena v `/etc/syslog-ng/syslog-ng.conf`

### B.6.1.7 Webmin

- inštalácia webmin

### B.6.1.8 Zabbix

- inštalácia zabbix-server, php5 a mysql
- zmeny v konfiguračných súboroch napr. v `/etc/php5/apache2/php.ini` a `/etc/zabbix/zabbix_server.conf`
- vytvorenie zabbix databáze spojené s konfiguráciou

## B.6.2 Prístup k webovým rozhraniam vybraných služieb

K niektorým nainštalovaným službám je možné pristupovať aj pomocou webového rozhrania. Medzi tieto služby patrí:

- zabbix (nástroj na monitoring) – **https://ip\_adresa/zabbix**, prihlásenie pomocou užívateľov vytvorených pri inštalácii, prípadne pomocou užívateľa **root** s heslom **modra@KOCKA#**
- webmin (nástroj na konfiguráciu servera) – **https://ip\_adresa:10000**, prihlásenie pomocou užívateľa **root** s heslom definovaným pri inštalácii
- apache2 (webový server) – **https://127.0.0.1/server-(info|status)**, informácie o nainštalovanom webovom serveri (prístup možný len z localhost cez IP adresu 127.0.0.1)
- ftp (nástroj na prenos súborov) – **https://ip\_adresa/net2ftp**, prihlásenie na FTP server (zadáť localhost alebo IP adresu servera) pomocou užívateľských účtov v systéme, okrem užívateľa root

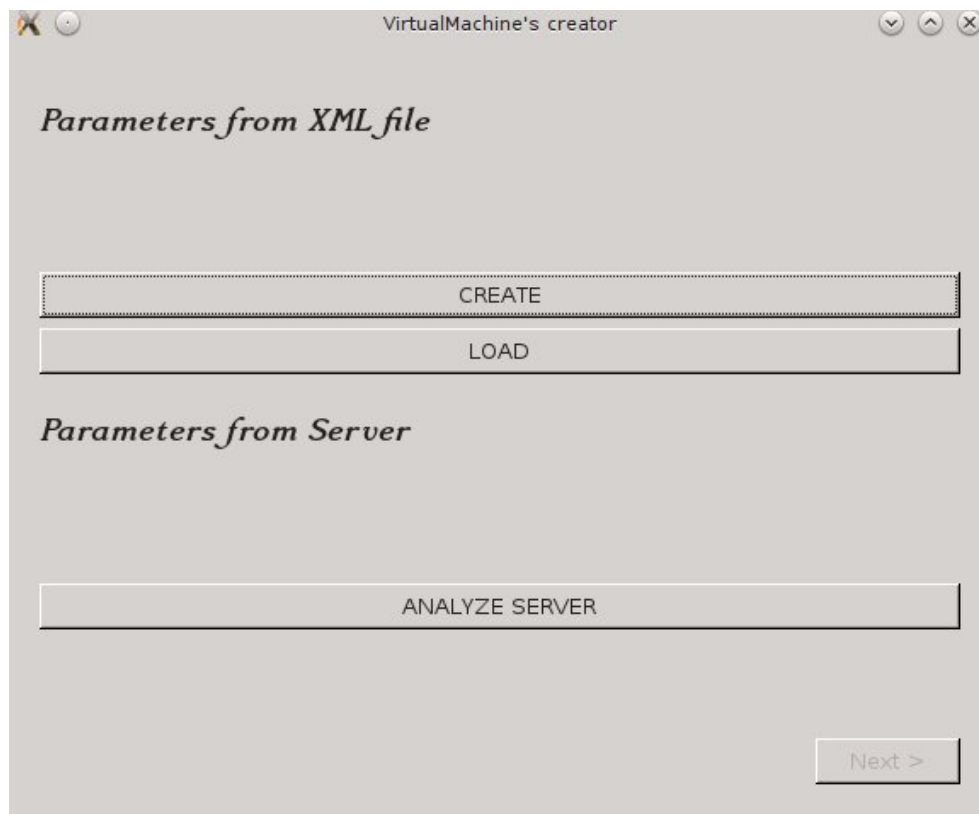
## B.7 Odinštalovanie aplikácie

Aplikáciu je možné odinštalovať pomocou príkazu:

```
$ dpkg -r virtual-machine-creator
```

Všetky inštalované súbory operačných systémov vo formáte ISO kopírované do adresára aplikácie, všetky virtuálne stroje vytvorené aplikáciou a všetky vytvorené alebo zmenené konfiguračné súbory sa pri odinštalovaní balíčka pomocou vyššie spomínaného príkazu neodstránia.

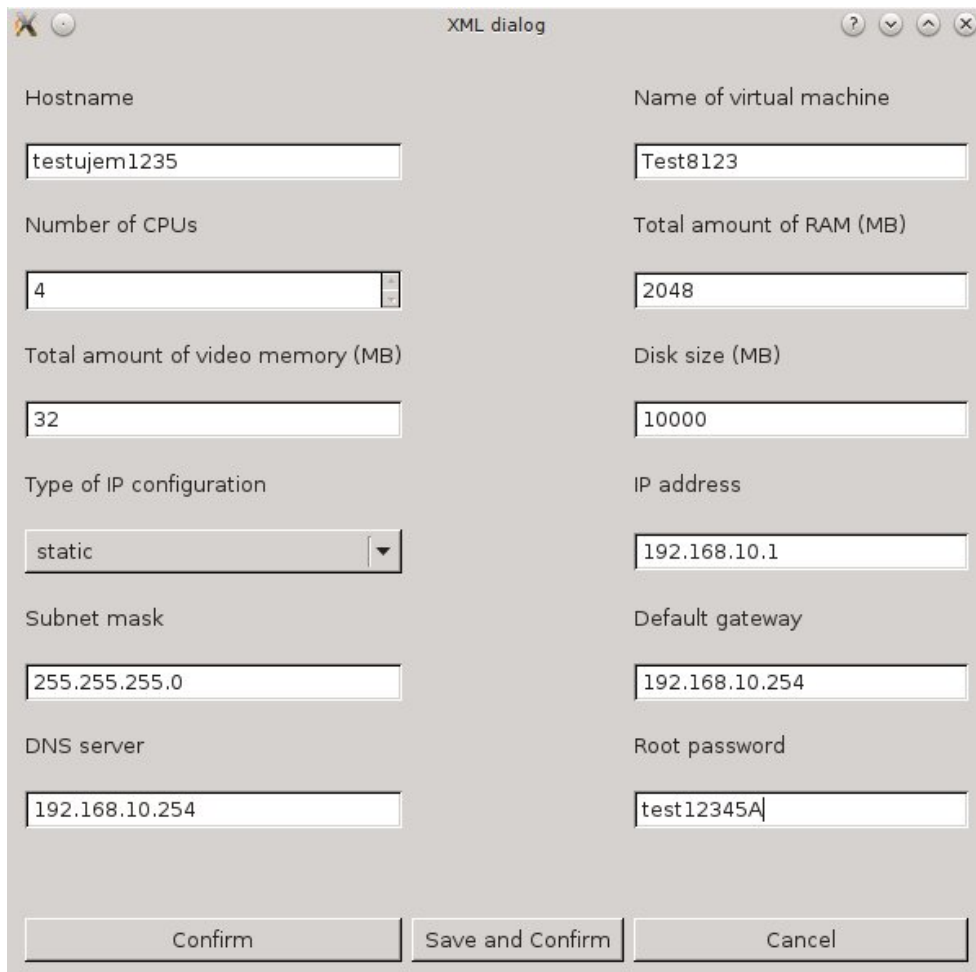
## **Grafické rozhranie aplikácie**



Obr. C.1: Úvodná obrazovka aplikácie



---

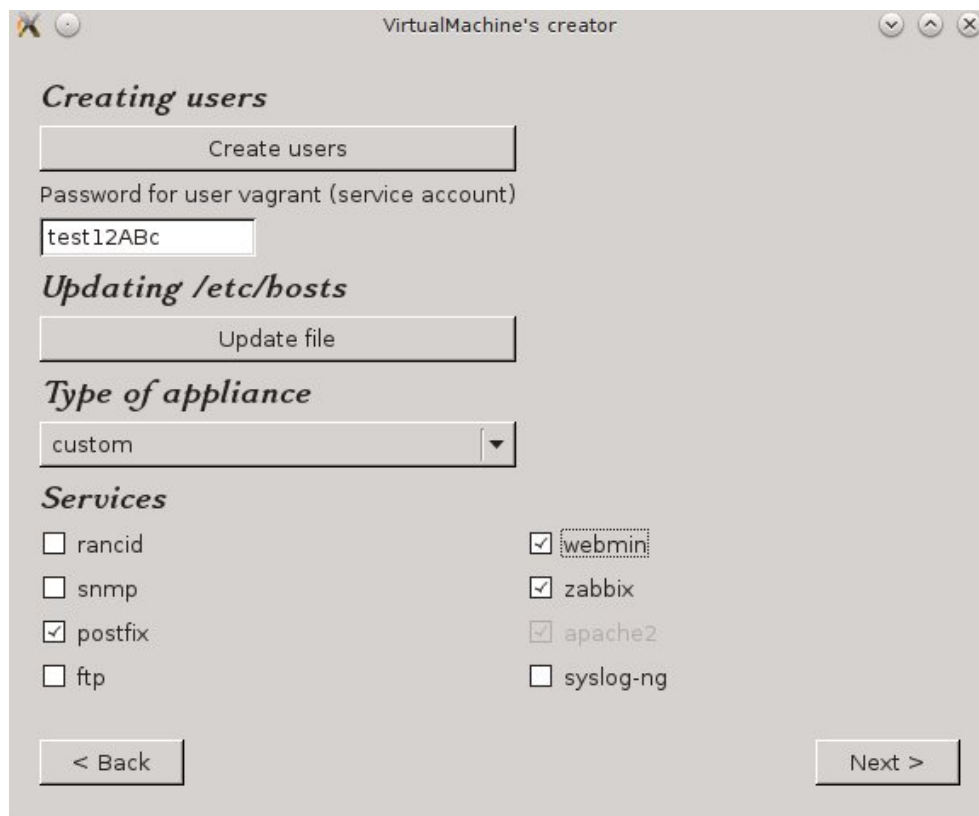
  


The image shows a dialog box titled "XML dialog" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog is organized into two columns of input fields. The left column contains: "Hostname" (text box with "testujem1235"), "Number of CPUs" (spin box with "4"), "Total amount of video memory (MB)" (text box with "32"), "Type of IP configuration" (dropdown menu with "static" selected), "Subnet mask" (text box with "255.255.255.0"), and "DNS server" (text box with "192.168.10.254"). The right column contains: "Name of virtual machine" (text box with "Test8123"), "Total amount of RAM (MB)" (text box with "2048"), "Disk size (MB)" (text box with "10000"), "IP address" (text box with "192.168.10.1"), "Default gateway" (text box with "192.168.10.254"), and "Root password" (text box with "test12345A"). At the bottom, there are three buttons: "Confirm", "Save and Confirm", and "Cancel".

Hostname	testujem1235	Name of virtual machine	Test8123
Number of CPUs	4	Total amount of RAM (MB)	2048
Total amount of video memory (MB)	32	Disk size (MB)	10000
Type of IP configuration	static	IP address	192.168.10.1
Subnet mask	255.255.255.0	Default gateway	192.168.10.254
DNS server	192.168.10.254	Root password	test12345A

Buttons: Confirm, Save and Confirm, Cancel

Obr. C.2: XML dialóg



Obr. C.3: V poradí 2. obrazovka aplikácie

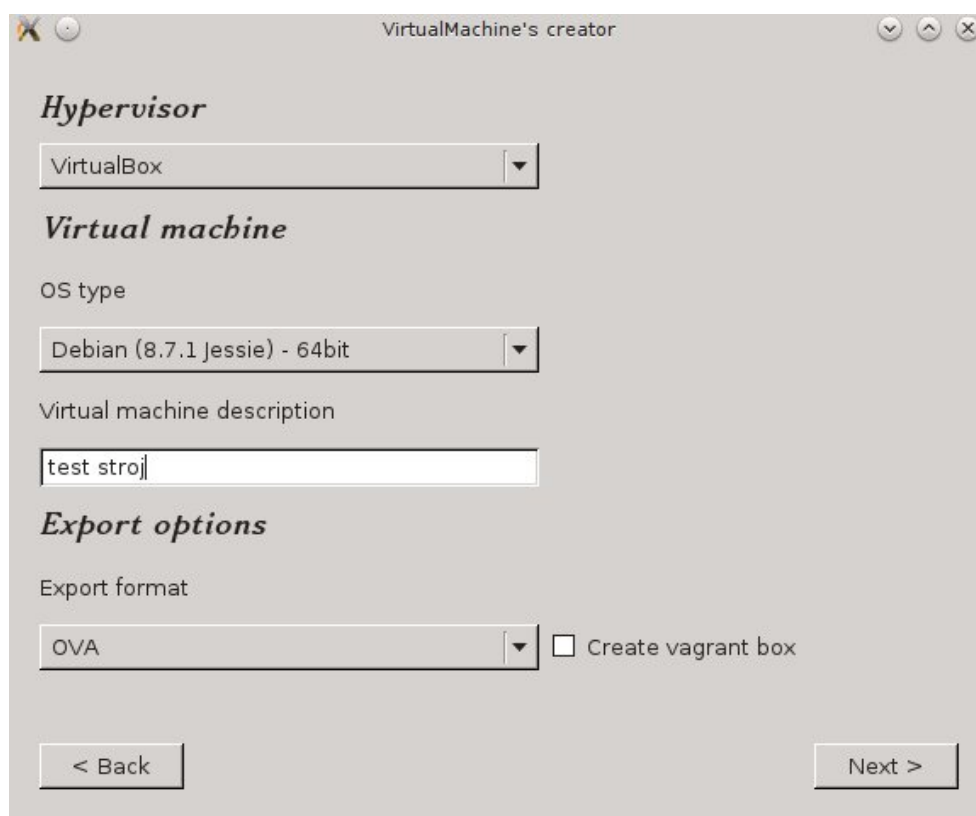
---

The image shows a window titled "Zabbix" with standard window controls. It contains two sections for configuration:

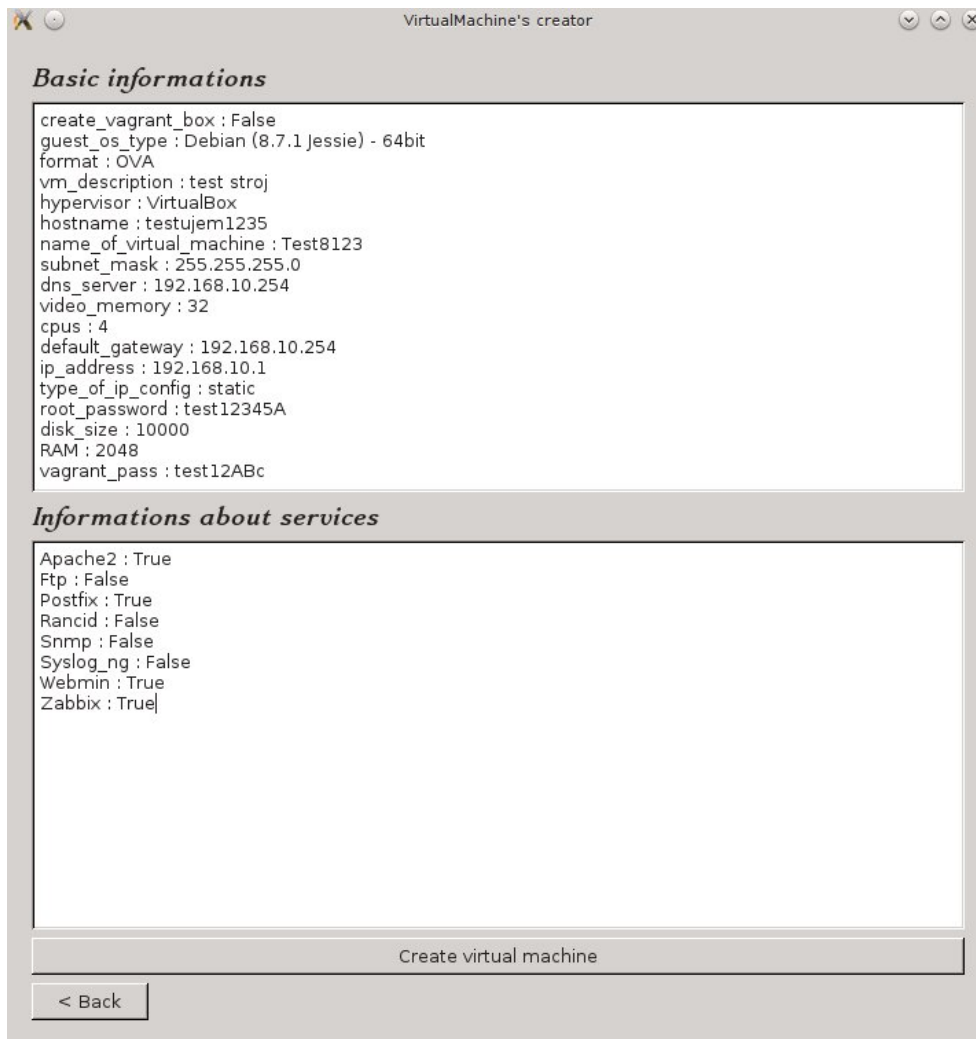
- Create zabbix hosts**
  - IP address: 192.168.10.1
  - Host name: testABC
  - Button: Create zabbix host
- Create zabbix users**
  - Name: Jakub
  - Surname: Sekera
  - Alias: sekerjak
  - Group: Zabbix user (dropdown menu)
  - Password: nb123A4e|
  - Button: Create zabbix user

An "OK" button is located at the bottom right of the dialog.

Obr. C.4: Dialóg služby Zabbix



Obr. C.5: V poradí 3. obrazovka aplikácie



Obr. C.6: V poradí 4. obrazovka aplikácie



---

## Obsah priloženého DVD

readme.txt	.....	stručný popis obsahu DVD
package	.....	adresár s inštalačným balíčkom a ISO adresárom
├── ISO	.....	adresár obsahujúci ISO obrazy operačných systémov
├── virtual-machine-creator-1.0_amd64.deb	.....	inštalačný balíček
src	.....	zdrojové kódy implementácie
├── application	.....	zdrojové kódy aplikácie
├── ansible	.....	konfiguračné súbory a playbooky nástroja Ansible
├── packer	.....	konfiguračné súbory nástroja Packer
text	.....	text práce
├── BP_Sekera_Jakub_2017.pdf	.....	text práce vo formáte PDF
├── src	.....	zdrojové súbory práce vo formáte L <sup>A</sup> T <sub>E</sub> X