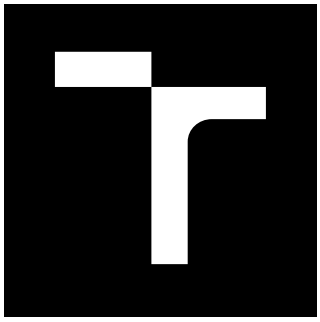



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS


# REALIZACE A OPTIMALIZACE DATOVÉHO ULOŽIŠTĚ
REALIZATION AND OPTIMALIZATION OF DATA STORAGE SERVER

**BAKALÁŘSKÁ PRÁCE**
BACHELOR'S THESIS


**AUTOR PRÁCE** — **Martin Holík**
AUTHOR

**VEDOUCÍ PRÁCE** — **prof. Ing. Miloslav Filka, CSc.**
SUPERVISOR



**BRNO 2016**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

# Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

***Student:*** Martin Holík ***ID:*** 164284
***Ročník:*** 3 ***Akademický rok:*** 2015/16

**NÁZEV TÉMATU:**

## Realizace a optimalizace datového uložiště

**POKYNY PRO VYPRACOVÁNÍ:**

Vyberte a implementujte vhodnou virtualizační platformu pro serverovou stanici umožňující následující služby - datové uložiště FTP/Samba, automatická záloha dat a automatická synchronizace složek (pomocí OwnCloud/Dropbox). Ověřte funkčnost na klientských stanicích a vytvořte návod pro instalaci serveru a klientských stanic.

**DOPORUČENÁ LITERATURA:**

[1] FILKA, M. Optoelektronika pro telekomunikace a informatiku. CENTA, Brno 2009.

[2] GOLFARELLI, M., RIZZII, S. Data Warehouse Design: Modern Principles and Methodologies (1 ed.). McGraw-Hill, Inc., New York, NY, USA, 2009.

[3] PORTNOVI, M. Virtualization Essentials. Indianapolis, IN: John Wiley & Sons, 2012.

***Termín zadání:*** 1.2.2016 ***Termín odevzdání:*** 1.6.2016

***Vedoucí práce:*** prof. Ing. Miloslav Filka, CSc.
***Konzultant bakalářské práce:*** Ing. Tomáš Horváth

**doc. Ing. Jiří Mišurec, CSc.**, *předseda oborové rady*

# Abstrakt

Tato bakalářská práce je zaměřena na realizaci a optimalizaci datového úložiště, zejména na problematiku návrhů serverů. V práci jsou teoreticky rozebrány dílčí problémy návrhu, především virtualizace, virtualizační platforma, sdílení dat nebo tvorba cloudového úložiště. Výsledný návrh je implementován na fyzické zařízení.

# Klíčová slova

Cloud, cloud computing, Dropbox, Debian, FTP, hypervizor, KVM, Linux, OwnCloud, paravirtualizace. RHEL, Samba, virtualizace, WMware, Xen

# Abstract

This bachelor thesis deals with realization and optimization of datacentres, specializing on the servers' design area. The assessment comprises of theoretical problem analysis, such as virtualization, virtualization platform, data sharing, as well as cloud storage production design. Final conception is implemented on the physical server.

# Keywords

Cloud, cloud computing, Dropbox, Debian, FTP, hypervisor, KVM, Linux, OwnCloud, paravirtualization. RHEL, Samba, virtualization, WMware, Xen

HOLÍK, M. *Realizace a optimalizace datového uložiště.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 76 s. Vedoucí bakalářské práce prof. Ing. Miloslav Filka, CSc..

## Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Realizace a optimalizace datového úložiště“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona c. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonu (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb“

V Brně dne ..... .....

podpis autora

# Poděkování

Chtěl bych poděkovat prof. Ing. Miloslavu Filkovi, CSc. za vedení bakalářské práce. Dále děkuji Ing. Petru Münsterovi, Ph.D a Ing. Tomášovi Horváthovi za odborné vedení, připomínky a cenné rady.


V Brně dne ……..  .....

podpis autora

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této bakalářské práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj pro inovace

# Obsah

# ÚVOD

Datové úložiště je běžným zařízením ve většině počítačových sítí, podnikových i domácích. Mezi nejjednodušší datová úložiště můžeme považovat v domácím prostředí obyčejné fyzické počítače se sdílenými daty prostřednictvím počítačových sítí LAN. Tento způsob není dostačující pro organizace a podniky, proto se setkáváme s datovými úložišti typu NAS a se servery s diskovými poli, které mimo jiné mohou poskytovat i jiné služby než jen přístup k datům.

Cílem této bakalářské práce je realizovat a optimalizovat datové úložiště na přiděleném dvouprocesorovém serveru s minimálními pořizovacími náklady a optimálně využít dostupné hardwarové prostředky. Dalším cílem této práce zvážit výhody a nevýhody virtualizačních platforem, vybrat nejvhodnější, implementovat ji a vytvořit vhodné prostředí pro služby, které bude server poskytovat. Serveru bude přidělena veřejná IP adresa z VUT rozsahu a proto musí být řádně zabezpečen proti útokům z internetu. Všechny poskytované služby by měli být dostupné pouze ze sítě VUT a zároveň dostupné pro omezenou komunitu z internetu. Samotná data by měla být zabezpečena proti ztrátě nebo havárii.

# 1 LINUX

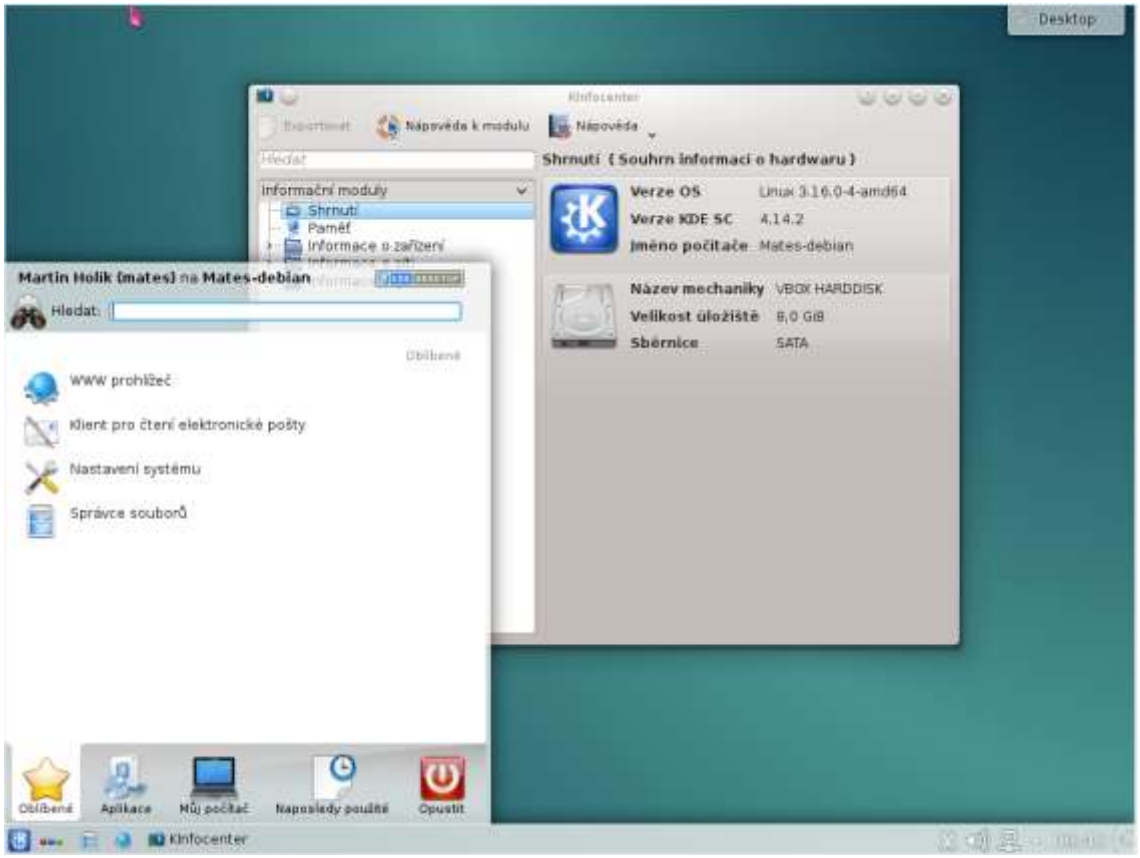
## 1.1 Přestavení

Linux je svobodný operační systém založený na jádře systému Unix. Počátky systému šahají až do 60. let minulého století a na jeho vývoji se podílel i otec programovacího jazyk C Dennis Ritchie. Už v tomto období si Unix zakládal na možnosti spouštět více aplikací zároveň a běžet na jakémkoliv dostupném hardwaru. V roce 1991 napsal vysokoškolský student Linus Torvalds dnes nejznámější unixovský založený operační systém Linux. Zkratka Linux znamená Linux Is Not Unix. Rozdíl mezi Linuxem a Unixem je značný. Linux je založen sice na bázi systému Unix, ale používá své vlastní jádro. Jádro systému disponuje otevřenými zdrojovými kódy, které může kdokoliv upravovat, šířit a volně používat. Tato myšlenka je také někdy označována jako GNU. Někdy tento operační systému označujeme jako GNU/Linux.

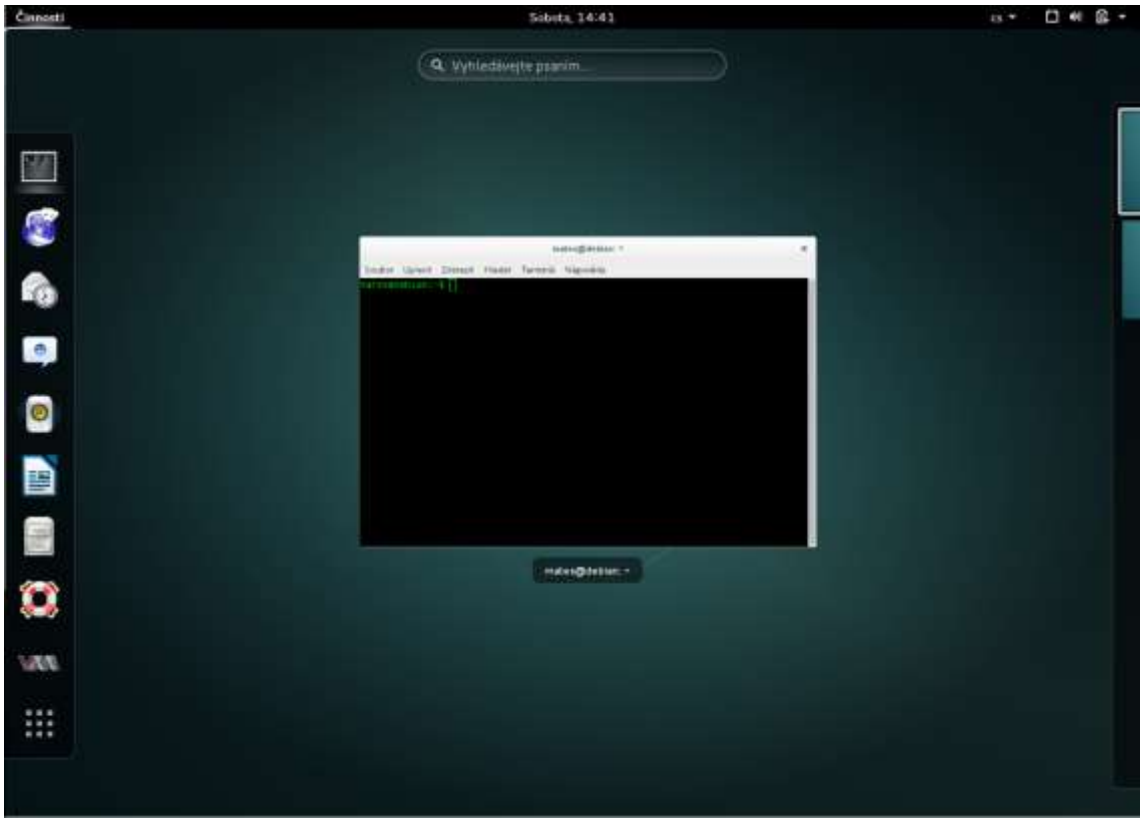
Ačkoliv systém Linux má menšinový podíl na trhu s desktopovými počítači tak na serverových platformách dominuje. Výhodou nasazení Linuxu na serverech je jeho otevřenost, stabilita a snadná správa. Při vhodné virtualizaci, která už je podporována i v samém jádře systému, lze na něm provozovat mnoho dalších virtuálních strojů. Tento způsob virtualizace je označován jako KVM (Kernel-based Virtual Machine). Mezi nejčastější konfigurace můžeme považovat LAMP u poskytovatelů webhostingu. Jde o Linux, na kterém běží web server Apache s podporou skriptů PHP a databází MySQL.

Výhodou systému Linux je jeho koncepce. Při jeho instalaci si uživatel může vybrat, jestli má zájem o systém s příkazovým nebo grafickým rozhraním. Příkazové rozhraní je vhodné pro servery, protože odlehčuje zatížení systému zbytečnostmi. Grafické nebo též desktopové rozhraní je v Linuxu označované jako nástavbový prvek. GUI je vhodné především pro běžné uživatele, kterým je tak usnadněna práce s operačním systémem, ale za cenu vyšších nároků na hardware. V současnosti existuje několik grafických prostředí, ale mezi nejznámější patří GNOME a KDE.

Cílem vývoje prostředí KDE bylo vytvořit jednoduché, avšak době odpovídající grafické rozhraní. Počátky vývoje jsou v roce 1996 a od té doby je zastoupeno v mnoha Linuxových distribucích, ať již vůbec nebo nepatrně pozměněné. Rozhraní je jednoduché a koncepčně podobné operačnímu systému Windows.




**Obrázek 1 – Desktopové prostředí KDE SC**




**Obrázek 2 – Grafické rozhraní GNOME**

Desktopové prostředí GNOME vychází z odlišné filosofie než KDE. Je založeno na myšlence vytvořit jednoduché prostředí pro aplikace běžící na Linuxu. Oproti KDE je méně náročný na hardware počítače, a tudíž je více prostředků dostupných pro běžící aplikace.

## 1.2 Linuxové distribuce

Jádro systému Linux je pouze program, který hardware pouze uvede do funkčního stavu. Další částí systému Linux je vytvořit nadstavbu, která bude komunikovat s uživatelem a bude obsahovat další programy a funkce. Týmy vývojářů pracující na uvedených nástavbových balících a tvoří takzvané distribuce. Jednotlivé distribuce mohou vyvíjet jednotlivci, společnosti nebo skupiny dobrovolníků. Ať se ale od sebe jednotlivé distribuce liší jakkoliv, vždy jde o systém Linux. Každá distribuce má své výhody a nevýhody, mohou být nainstalovány na počítači nebo spouštěny z přenosných médií.

Většina verzí má vyvinuté i tzv. Live distribuce. Live distribuce je verze, která lze spustit pouhým nabootováním systému bez nutnosti instalace. Výhoda těchto edicí spočívá v přenositelnosti. Uživatel tak může svůj systém spustit kdekoliv, pracovat na něm a ukládat nebo neukládat nový aktuální stav systému. Dalším typickým využitím těchto distribucí je při hackování.

Distribucí systému Linux je nepřeberné množství, nejznámějšími jsou Debian, Red Hat Linux, Slax, Kali, Fedora nebo nejvíce uživatelské přístupné Ubuntu.

### 1.2.1 Red Hat Enterprise Linux

RHEL je komerčně spravovaný produkt společností Red Hat. RHEL je úzce spojen s distribucí Fedora. Cílem tohoto operačního systému je nabídnout kvalitní rozhraní. Ačkoliv jde o placenou jejím zakoupením je podporována i zdarma dostupná Fedora. Všechny funkce Red Hat Enterprise Linuxu jsou v prvotním stádiu vyvinuty pro Fedoru, na které probíhá testování a ladění. RHEL je systém založený pro nasazení na specifickém hardwaru a svou koncepcí se od Fedory velmi liš, zejména možností univerzálního nasazení. První verze RHEL nepodporovala grafické rozhraní a instalace včetně budoucí správy tak byla možná pouze přes příkazové prostředí. Od verze RHEL 6.1 distribuce obsahuje i známá grafická rozhraní GNOME a KDE, která je možné zvolit při instalaci. Od verze 6.2 systém obsahuje i

českou lokalizaci. Nové verze jsou vydávány v intervalu několika let a disponují až 13 letou placenou podporou ze strany Red Hatu.

### 1.2.2 Ubuntu

Ubuntu je uživatelsky přívětivá distribuce odvozená z Debianu. Vlastní své desktopové prostředí zvané Unity, ale jsou známy i verze s jinými prostředími (Kubuntu, Xubuntu,…). Je určeno především pro nasazení na desktopové počítače, ale existují i verze pro servery. Každého půl roku vychází nová verze, která je vydávána bezplatně s devíti měsíční podporou. Každé dva roky je vydávána verze s označením LTS. Tato verze je stabilní a nabízí až pětiletou podporu. Distribuce Ubuntu se snaží nabídnout jednoduché prostředí s jednoduchým instalátorem a nástroji pro správu. Ubuntu i jeho odvozené verze podle desktopového prostředí mají nízké nároky na hardware, dokonce nižší než 64 bitové verze operačního systému Windows 7.

### 1.2.3 Debian

Debian je svobodná a nekomerční distribuce, na jejímž vývoji se podílí tisíce lidí po celém světě. Je velmi oblíbený mezi pokročilými uživateli Linuxu. Další programy a funkce do systému můžeme přidat pomocí balíčků, kterých je v současné distribuci přes 43 tisíc. Balíčky lze snadno spravovat a vyhledávat pomocí správce balíčků. Při instalaci systému volíme, jestli chceme nainstalovat nějaké desktopové prostředí nebo používat pouze příkazový režim. Aktuální verze 8 nese kódové označení Jessie. Celý systém je vyvíjen v duchu svobody a proto všechny balíčky i odvozené distribuce z Debianu jako je například Ubuntu by měli být svobodné. Všechny náklady jako vývoj, hardware nebo připojení k síti jsou hrazeny z příspěvků a darů.

#### 1.2.3.1 Vytvoření RAID-1

Nesmírně důležitou částí serveru je ochrana dat proti havárii pevného disku. Pevné disky, jsou namáhány a jejich životnost sledována technologií S.M.A.R.T. Tato technologie umožňuje sledovat stav, ale nedokáže přesně předpovědět délku života disku. I při včasné výměně disků však je nutno data přemístit ze starého na nový a eliminovat tak výpadek serveru na co nejmenší čas. Vhodnou ochranu proti havárii disku lze vyřešit využitím pole RAID. Při tvorbě polí RAID, můžeme použít RAID hardwarový, softwarový nebo SataRAID. Podle úrovně zabezpečení disků dělíme RAID na typy 0, 1, 5, 01, 10 atp.

Hardwarový RAID, je technologie, která pro svůj chod vyžaduje přídavný hardware. Tento hardware je umístěn v serveru ve slotech PCI a disponuje vlastním řadičem pro připojení disků, procesorem a zálohovací baterií. Při použití hardwarového RAID pole jsou sníženy nároky na procesor serveru, protože všechny redundance a operace pro spravování pole jsou realizovány vlastním procesorem. V případě havárie disku, lze provést opravu před i po zavedení operačního systému. Pořizovací náklady na hardwarový RAID jsou velmi vysoké a pohybují se okolo 10 000,- korun.

Pravým opakem je RAID softwarový. Výpočty parity a správu pole provádí operační systém, který však výrazně zatěžuje procesor serveru a tím snižuje i výkon virtuálním strojům. Hlavní nevýhodou je oprava disků, protože nelze provádět před zavedením systému. Pokud je poškozen operační systém spravující RAID a nelze ho nabootovat, data jsou nenávratně ztracena a neexistuje možnost jak data z pole obnovit.

SataRAID je technologie založená na předchozích dvou. Je implementována výrobcem v řadiči pevných disků základní desky ale neobsahuje vlastní procesor. Zátěž je tak rovnoměrně rozdělena mezi řadiče a procesor serveru. Procesor je také zatěžován jako u softwarového RAIDu, ale zátěž je výrazně nižší. Pole lze spravovat před spuštěním i po spuštění operačního systému, který musí mít doinstalovány specifické ovladače řadiče disku. Po spuštění serveru je inicializována utilita pro správu pole a následně až zaveden BIOS.

Tento server technologií SataRAID disponuje a umožňuje vytvářet pole typu RAID 1 (zrcadlení disku na disk), RAID 0 (spojení více disků v jedno pole) a RAID 10 (vytvoří 2 pole RAID 1 a spojí je v jedno pomocí RAIDu 0). Server má hotswap sloty pouze pro tři pevné disky. Tyto sloty jsou obsazeny disky o kapacitách 2x 320 GB a 2 TB. RAID 10 vytvořit nelze, protože jeden disk chybí. RAID 0 není bezpečný, protože při výpadku jednoho pevného disku přijdeme o všechna data v celém poli a proto použijeme RAID 1.

Při spuštění serveru vstoupíme do BIOSu, povolíme SataRAID a restartujeme počítač. Ihned po startu jsou inicializovány disky a stiskem CTRL + E otevřeme utilitu pro konfigurací polí. V menu vybereme *Configure – Easy Configuration* vybereme dostupné disky a potvrdíme klávesou ENTER. Dále přejdeme do konfigurace vlastností pole stiskem klávesy F10. Nastavíme vlastnosti pole na RAID 1 a nastavíme maximální velikost pole. Vše potvrdíme tlačítkem Accept. Ukončíme průvodce vytvářením polí a inicializujeme disky z menu výběrem nabídek *Object – Logical Drive – Logical Drive 0 – Initialize*. Tímto máme pole vytvořené a můžeme pokračovat v instalaci.

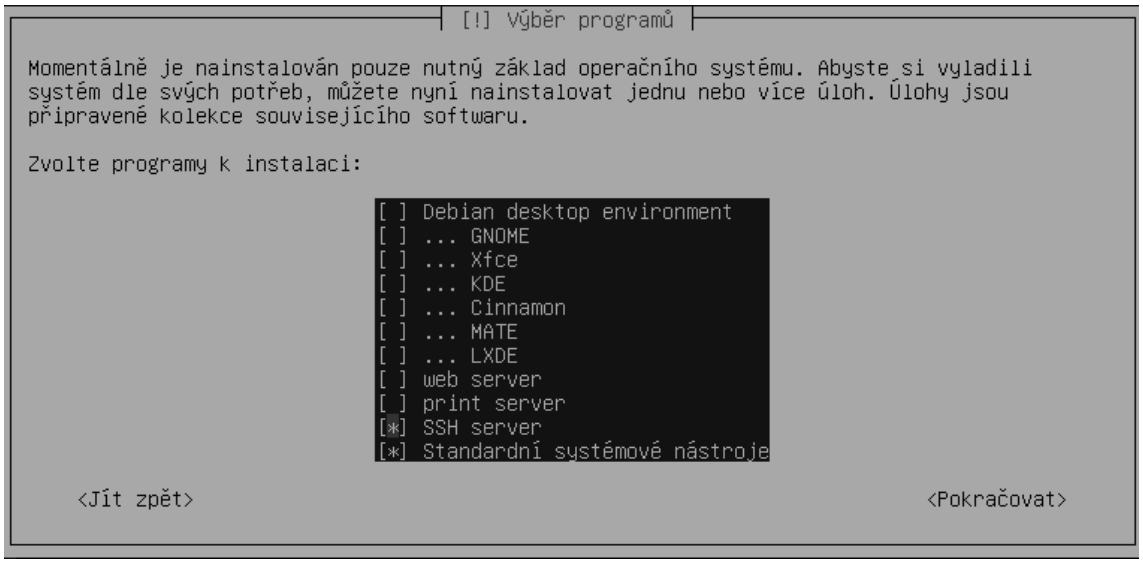
### 1.2.3.2 Instalace hlavního operačního systému

Hostitelský (hlavní) operační systém je nejdůležitějším objektem každého serveru. Jeho nestabilita, chyba nebo zaseknutí výrazně ovlivní celý server, včetně hostovaných strojů. Na stabilitě se výrazně podílí i ochrana proti havárii disku. Tato ochrana je zabezpečena polem typu RAID 1, na kterém je systém nainstalován. Debian, stejně jako další distribuce Linuxu, disponuje internetovými zrcadly se vždy aktuálním obrazem distribuce. Instalační CD lze pořídit, ale vzhledem k rychlosti vývoje operačního systému je výhodnější zavést systém z USB disku a celou instalaci provést ze síťového zrcadla. Po zavedení operačního systému z USB povolíme instalaci na SataRaid pole následujícím postupem. Posunem kurzor na možnost *Install* a stiskneme klávesu TAB. Do spouštěcího dialogu je nutno zavést další parametr *dmrait=true.*


**Obrázek 3 – Nastavení SataRAIDu při instalaci**

Instalaci spustíme stisknutím klávesy Enter. Instalace je intuitivní a funguje na principu dotazů. Pozastavíme se u nastavení hesel a účtů. V tomto bodě je důležité zvážit vytvoření superuživatele root. V případě serveru je výhodnější nechat heslo prázdné, účet root bude deaktivován a prvnímu uživateli bude umožněno provádět operace nad systémem s příkazem *sudo,* nebo se přepnout na superuživatele příkazem *sudo –i*. Tento krok je zásadní

v bezpečnosti serveru. Deaktivací účtu se případnému útočníkovi znesnadní práce s prolomením přístupu, protože musí uhádnout nejen heslo, ale i jméno uživatele. Po nastavení práv instalace může pokračovat. Po spuštění nástroje na rozdělení disku použijeme asistované rozdělení a vybereme diskové pole RAID k rozdělení. Pokud při instalaci je zobrazena pouze USB klíčenka se zavaděčem. Instalaci je nezbytné přerušit a spustit prostředí shell. Při instalaci se také spouští démon i pro softwarový RAID, který blokuje načtení SataRAID pole. Softwarový RAID zastavíme v terminálu zadáním příkazu *mdadm –S /dev/md127* a po deaktivaci softwarového RAIDu je následně potřeba aktivovat SataRAID příkazem *dmraid –ay*. Po tomto kroku opět spustíme nástroj pro rozdělení disku a pokračujeme v instalaci do doby, kdy se systém zeptá na výběr programů. Zde odznačíme možnost *Debian desktop enviroment*. Tím deaktivujeme grafické rozhraní a systém zůstane pouze v textovém režimu. SSH server označovat nemusíme, protože jej budeme instalovat později.




**Obrázek 4 – Okn výběru programů**

V instalaci pokračujeme až k zavaděči systému GRUB u kterého pravděpodobně nastanou problémy. Pokud instalace zavaděče selže, nainstalujeme systém bez zavaděče a restartujeme operační systém. Opět nabootujeme z instalačního USB, v menu vybereme *Advanced Options* a kurzor přesuneme na možnost *Rescue Syste*m. Po stisku klávesy TAB opět upravíme spouštěcí parametry připsáním parametru *dmraid=true*. Při dotazu na kořenový souborový systém vybereme první možnost */dev/md-x* (x je číslo). Nyní se spustí systém v záchranném módu. Do terminálu napíše příkaz *bash,* a pomocí editoru editujeme soubor se zavaděčem */etc/default/grub*. V souboru zavaděče odkomentujeme řádek

GRUB_DISABLE_LINUX_UUID=true a uložíme změny. V posledním kroku již zbývá aktualizovat zavaděč systému příkazem *grub-install /dev/mapper/* a doplníme jméno RAID pole (jména se vypíší po opětovném stisknutí klávesy tabulátoru). Bash ukončíme příkazem *exit* a restartujeme příkazem *reboot system*. Po restartování zařízení nabootuje nainstalovaný systém.

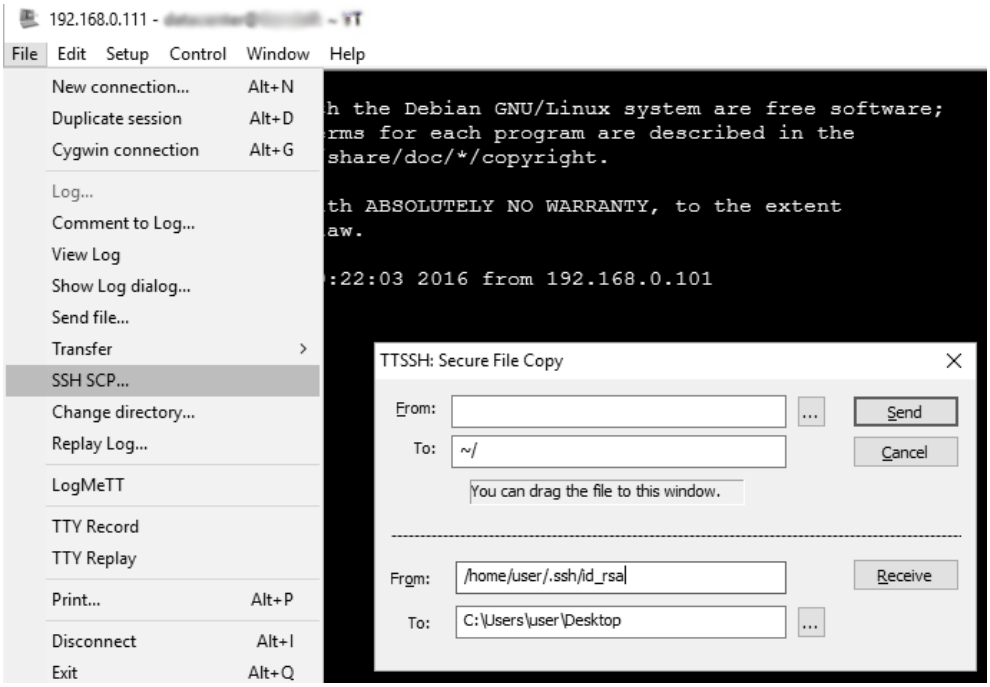
#### 1.2.3.3 Instalace SSH serveru

Systémy Linux disponují možností správy systému prostřednictvím terminálu. K terminálu se lze připojit prostřednictvím protokolů telnet a SSH. Telnet je textový protokol, který odesílá data na server v nezašifrované podobě, a proto není bezpečný pro správu serverů. Zabezpečenou komunikaci mezi serverem a klientem zabezpečuje protokol SSH, který všechny zprávy přenáší v zašifrované podobě. SSH protokol, ale umožňuje i vytvoření SSH tunelu, skrze které lze využívat i jiné protokoly nebo služby (např. RDP, SPICE).

SSH server opět nainstalujeme pomocí balíčkového systému aptitude příkazem *apt-get install openssh-server*.

#### 1.2.3.4 Zabezpečení SSH přístupu

K serveru lze přistupovat prostřednictvím protokolu SSH s ověřením uživatelského jména a hesla. Nevýhodou je pamatovat si obě kombinace a použití hesla zvyšuje potenciál prolomení hesla útočníkem. Vyšší zabezpečení získáme použitím privátního a veřejného RSA klíče pro ověření přístupu. Klíč vygenerujeme příkazem *ssh-keygen*. Při generování se generátor dotáže na heslo, které slouží jako dodatečná ochrana při použití RSA klíče. Toto heslo není důležité a nemusí být zadáno. Po ukončení generátoru máme ve skryté složce uživatele *~/.ssh/* k dispozici dva klíče *id_rsa* a *id_rsa.pub*. První klíč je privátní, tudíž nesmí být sdílen a používá ho uživatel pro přihlášení. Druhý klíč je veřejný a je uložen na serveru. Protože oba klíče jsou uloženy na straně serveru, je nutné privátní klíč bezpečně přenést k uživateli například prostřednictvím protokolu pro bezpečný přenos SCP. Pro přenos do Windows lze použít prostředí Tera Term.



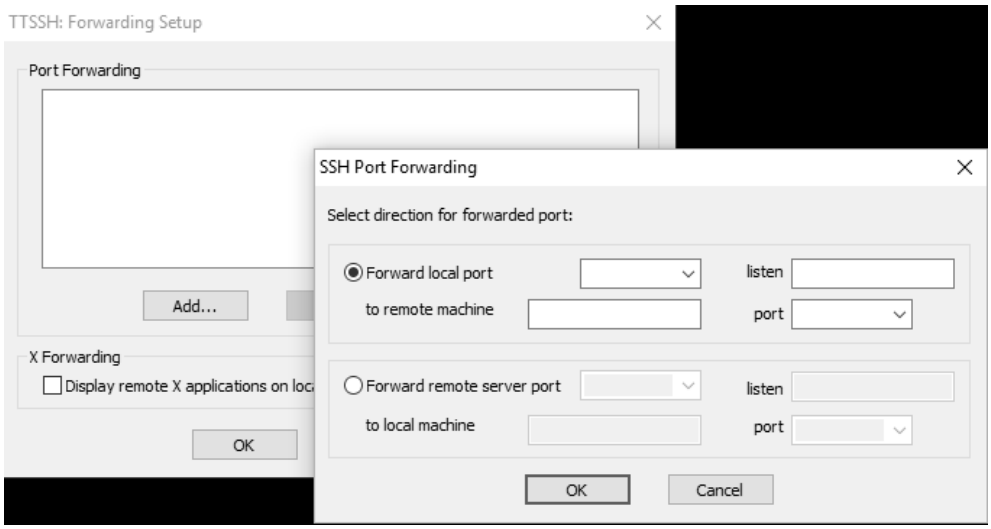

**Obrázek 5 – Tera Term přenos přes SCP**

Jak bylo výše zmíněno, použití RSA klíče je bezpečnější než použití hesla. Pokud chceme server více zabezpečit, můžeme zakázat přihlášení všem uživatelům bez svého privátního klíče. Tato volba se zapíná odkomentováním řádku *PasswordAuthentication no* v konfiguračním souboru */etc/ssh/sshd_config*.

Aktivace změn v nastavení SSH serveru dokončíme příkazem pro restartování služby *service ssh restart*.

### 1.2.3.5 Vytvoření SSH tunelu

Zabezpečený tunel můžeme v prostředí Tera Term připojit až po úspěšném připojení. Tunel vytvoříme v nabídce *Setup – SSH Forwarding*. V otevřeném okně TTSSH: Forwarding Setup klikneme na tlačítko ADD a nastavíme tunel.




**Obrázek 6 – Vytvoření SSH tunelu**

Do položky Forward Local port nastavíme libovolný nepoužitý port klientského PC a políčko Listen nevyplníme. V druhém řádku nastavíme vzdálenou IP adresu zařízení, které danou službu poskytuje. Můžeme zadat i IP adresu virtuálního stroje, který se skrývá za NATem. Do posledního políčka nastavíme číslo portu, na kterém služba na vzdáleném PC naslouchá. Po potvrzení kliknutím dvakrát na tlačítko OK je SSH tunel aktivní. Vzdálenou službu máme nyní dostupnou na adrese *localhost:port*, kde port je číslo, které jsme zadali do prvního políčka.

# 2 VIRTUALIZACE

Při současném vývoji internetu, přenosných zařízení a dostupnosti internetové konektivity prochází vývojem i datová úložiště. Mezi nejznámější datová úložiště patří například Google Drive, Dropbox nebo OneDrive od firmy Microsoft. Každé úložiště musí být stabilní. Stabilita takového systému je závislá na síťové konektivitě, ale i na výkonu samotného zařízení.

Virtualizace je softwarová technologie, která nám umožní lépe využít dostupné hardwarové kapacity a provozovat na jednom fyzickém počítači více na sobě nezávislých virtuálních strojů.

## 2.1 Typy virtualizace

### 2.1.1 Desktopová virtualizace

Každý uživatel dnes vlastní minimálně jeden osobní počítač s vlastními nainstalovanými aplikacemi. Tyto počítače je potřeba spravovat. U každého nového zařízení je potřeba provést opětovnou instalaci, která může být ve většině případů zdlouhavá.

Výše zmíněné problémy je možné vyřešit desktopovou virtualizací, která celý systém včetně aplikací převede a uloží na server datového centra. Takový uživatel má na svém počítači nainstalovaný pouze hypervizor a pomocí vzdálené plochy se přihlašuje na server ke svému virtuálnímu počítači.

Desktopová virtualizace disponuje mnoha výhodami. Administrátor při výměně počítače nemusí celý systém předinstalovávat, ale nainstaluje pouze základní komponenty pro přístup. Správně provedené desktopová virtualizace může ušetřit nemalé finanční prostředky při budování infrastruktury podniku, zejména při nákupu softwarových licencí nebo snížením nákladů na nové počítače. Organizace nemusí tak často nakupovat nový hardware, protože uživatelské aplikace nepotřebují prostředky fyzického hardwaru, ale spotřebovávají výkon datového centra. Datové centrum musí být pro tuto virtualizaci dostatečně naddimenzované, aby nedocházelo k výpadkům. Takto virtualizovaný systém je snadno dostupný prostřednictvím počítačové sítě nebo internetu a tím se zvyšuje i mobilita zaměstnanců. Některé virtuální počítače také mohou pracovat v tzv. offline režimu. Uživatel provádí běžné úkony na počítači a po připojení k serveru jsou změny zabezpečeně synchronizovány.

### 2.1.2 Serverová virtualizace

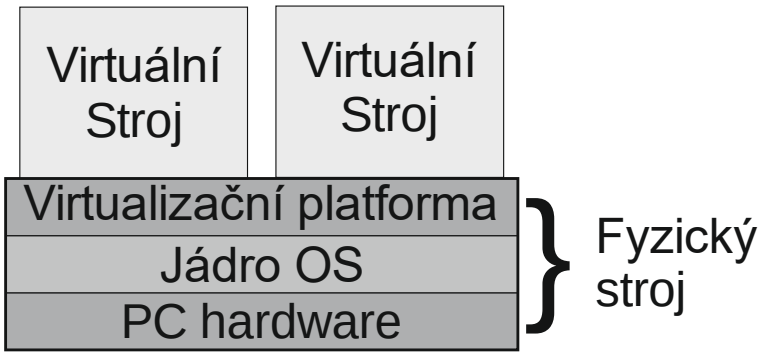
V dnešní době servery disponují obrovskými hardwarovými kapacitami, které nelze u většiny aplikací efektivně využít. Umístění více aplikací na jeden fyzický stroj není z hlediska bezpečnosti možné, a proto je výhodné použít serverovou virtualizaci.

Tyto virtuální servery lze přizpůsobit potřebám uživatelů. Díku tomu je lze lépe spravovat, vytvářet i nasazovat. Vytváření virtuálních serverů je ve většině případů rychlejší než instalace serverů fyzických. Takto oddělené servery lze spravovat samostatně. Havárie jednoho virtuálního stroje neovlivní jiné virtuální stroje a můžeme provozovat i různé operační systémy či aplikace na jednom fyzickém hardwaru současně. Díky virtualizaci můžeme také značně ušetřit při budování infrastruktury sítě, protože nemusíme pořizovat více fyzických serverů, ale zainvestovat pouze do jediného. Virtualizace také umožňuje snížit i provozní náklady, do kterých patří elektřina nebo klimatizace.

## 2.2 Druhy virtualizace

### 2.2.1 Plná virtualizace

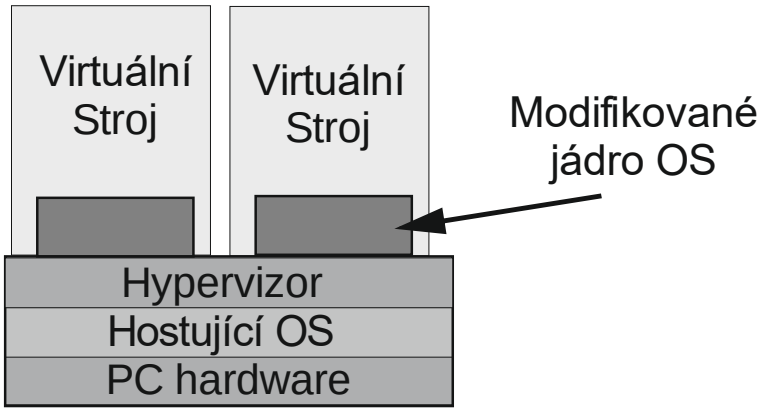
Plná virtualizace, nebo též nativní virtualizace je technologie, která nepředpokládá žádnou úpravu jádra systému a virtualizuje všechny dostatečné prostředky nezbytné pro nezávislý běh virtuálního systému. Fyzická vrstva systému je plně oddělena a virtuální stroj běží pouze na virtuálním hardwaru. Přístup k fyzickému hardwaru musí být zprostředkován. Plná virtualizace přináší mnoho výhod. Především snadnou a rychlou obsluhu virtualizovaného systému, rychlejší instalaci a reinstalaci, snadné zálohování, a možnost alokovat systému určitou velikost operační paměti. Virtuální systémy lze snadno přenést a aplikovat na jiný hardware, pokud původní i cílový procesor obsahují stejné instrukční sady. Plná virtualizace je vhodná především pro běžné uživatele. Je náročná na fyzický hardware a čtení z pevných disků je pomalejší. Mezi nejznámější platformy patří VMware (Workstation, Player, Server), Virtual PC, VirtualBox, Hyper-V a KVM.




**Obrázek 7 – Schéma úplné virtualizace [7]**

### 2.2.2 Paravirtualizace

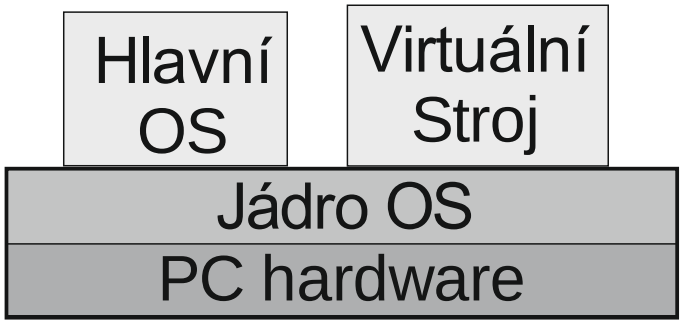
Fyzická vrstva hardwaru není zcela oddělena od virtuálního stroje kvůli neúplné virtualizaci fyzického hardwaru. Problém nastává při virtualizaci procesoru, protože ten musí běžet ve dvou režimech – privilegovaném a uživatelském. Jádro operačního systému běží v režimu privilegovaném. Úkolem tohoto režimu je kontrolovat uživatele a přístup k hardwaru, aby nemohla být narušena integritu dat či jiné běžící programy (např. kontrola přístupu na pevný disk). Uživatelský režim je poskytnut již běžícím aplikacím. Jádra všech operačních systémů jsou vzájemně propojeny, aby mohly přistupovat ke sdíleným prostředkům, které přiděluje virtuální monitor (hypervizor). Zde nastává problém, protože hypervizor nemůže běžet ve stejném režimu jako jádro operačního systému. Proto potřebujeme ještě jednu úroveň s nejvyšším stupněm ochrany, aby se hypervizor a jádra operačních systémů nemohla vzájemně ovlivňovat. Řešením je upravit jádro systému tak, aby nemohlo provést žádnou operaci, která by narušila stupeň ochrany monitoru. Procesory Intel mají čtyři stupně ochrany. V nulté a nejvíce chráněné vrstvě běžně běží jádro operačního systému a ve třetí vrstvě aplikace s nejnižším stupněm ochrany. Stupeň jedna a dva se běžně nevyužívají. Při paravirtualizaci se do nultého stupně zavede hypervizor a jádro systému se posune o jeden stupeň ochrany níže. Důsledkem je pomalejší provádění řídících instrukcí. Proto jsou jádra systému modifikována, aby operační systémy měly možnost pracovat s těmi částmi pamětí, které neovlivní zabezpečení hypervizoru. Mezi nástroje nabízející tento způsob virtualizace patří například XEN.




**Obrázek 8 – Schéma paravirtualizace**

### 2.2.3 Virtualizace na úrovni OS

Jak již název napovídá, virtualizujeme na nainstalovaném operačním systému hostitele. Takto využíváme jádro systému pro několik dalších vzájemně izolovaných virtuálních strojů a vytváříme z něj virtulizační vrstvu. Základní podmínkou pro použití této technologie je nutnost využít pro virtualizaci stejné operační systémy a to včetně verze. Výkonost takového systému je vysoká, protože využívají stejné jádro a všechny systémy mohou přímo přistupovat k hardwaru hostitelské počítače. Výkonost virtualizace a je srovnatelná s výkonem na fyzickém počítači právě kvůli nízké režii. Hlavní nevýhodou takového systému je náchylnost k havárii, protože pád jediného systému způsobí havárii všech systémů. Další nevýhodou je nemožnost instalace softwaru, který přímo zasahuje do jádra systému, jako jsou firewally nebo antispywarové či antivirové programy atd. Tento způsob virtualizace je podporován systémy, jako jsou FreeBSD Jail nebo Solaris Containers aj.




**Obrázek 9 – Virtualizace na úrovni jádra [7]**

### 2.2.4 Emulace

Koncepce této technologie spočívá v úplné virtualizaci hardwarových komponent. Jedná se o nejpomalejší způsob virtualizace, neboť instrukce z hostovaného operačního systému musí

být překládány do instrukcí, které hostitelský systém umí provést. Můžeme virtualizovat celý procesor včetně registrů i paměti. Další výhodou je možnost nasimulovat hardware, který není dostupný, např. na jednoprocesorovém systému nasimulujeme vícejádrový procesor. Nevýhodou simulace jiné platformy je nemožnost použití virtualizační technologie procesoru (Vanderpool, Pacific) ani jiný hardware počítače. Využití takto virtualizovaných systému může být především při vývoji nebo spouštění aplikací určených pro jinou platformu. Mezi typické zástupce patří DosBox, VICE, Wine nebo Bosch.

## 2.3 Serverové virtualizační platformy

### 2.3.1 WMware

Společnost WMware se zabývá vývojem komerčních produktů a na trhu patří mezi přední hráče ve virtualizaci datových center. Produkty jsou určeny jak pro desktopy tak servery a představují komplexní nástroje vhodná pro datová centra. V současné době jsou vyvíjeny produkty pro serverová řešení WMware Server a Infrastructure.

WMware Infrastructure je licencované řešení původně založené na systému WMware ESX Server. Pro zajištění optimální výkonosti systému založeného na Red Hat Linuxu je nutné vlastnit certifikovaný server. Spravování a vytváření nových virtuálních strojů lze pomocí webové konzoly. V této konzoli lze i přiřazovat jednotlivým strojů i dostupné prostředky.

WMware Server patří mezi zdarma dostupné produkty, které především slouží k vyzkoušení serverové virtualizace, než k ostrému nasazení. Mají některá omezení a nemají plnou technickou podporu, jako produkty Infrastructure.

WMware vSephere je systém založený na balíku Infrastructure. Důvodem zavedení byla lepší optimalizace a s tím i spojený nárůst výkonnosti systému. Dále možnost vytvoření vlastního interního cloudového úložiště a jeho správa a možnost snížit plánované výpadky kvůli údržbě a migracím systému. Jako první společnost přišla s možností migrace virtuálních strojů mezi hostitelskými počítači za plného chodu.

Technologie WMware je založena na koncepci emulace a plné virtualizace. Hardware hostitelského systému je plně virtualizovaný pro hostované systémy. Díky tomu jsou virtuální stroje snadno přenositelné pouhým zkopírováním virtuálního systému z jednoho počítače na druhý. Jediné omezení pro přenos spočívá v procesoru, který nesmí obsahovat různé instrukční sady, jak je to u jiných emulátorů. Společnost WMware vyvíjí své balíky tak, aby

nemusely být virtualizovány instrukční sady, jako je tomu u emulátorů, ale používá virtualizační technologie procesorů, jako u plné virtualizace.

### 2.3.2 Xen

Xen je open-sourcový projekt původně vyvinutý na univerzitě v Cambridge. První verze podporovala jen paravirtualizaci a pro správný chod takového serveru bylo potřeba modifikovat jádra systému se známým kódem, jako je třeba Linux. V dnešní době už Xen prošel značným vývojem a od verze 3.0. Obsahuje i podporu hardwarové virtualizace, a proto je možné virtualizovat i systémy Windows.

Xen se skládá ze skriptů a doplňků pro Linux nebo FreeBSD, které umožňují virtualizaci ovládat prostřednictvím svého hypervizoru. Abychom mohli virtualizaci Xenu řídit, potřebujeme příslušné ovladače. Ovladače dělíme na back-end (spravují přístup k hardwaru) a front-end (propojují virtuální stroj a hardware).

Tato platforma disponuje vysokou efektivitou a může běžet jak v úplné virtualizaci, tak i paravirtualizaci. Další kladností je nárůst rychlosti komunikace mezi virtuálními systémy, protože nejsme limitováni fyzickou vrstvou TCP/IP modelu, jako je to u fyzických strojů.

V roce 2007 došlo ke spojení XenSourcu se společností Cintrix. Původní Xen byl přejmenován na XenServer a převeden na edice Free, Advanced, Enterprise a Platinum. Verze free je snadno dostupná a obsahuje základní nástroje včetně živé migrace, zálohování a pokročilejší správu. I přes tuto skutečnost verze free stále obsahuje zveřejněné kódy.

### 2.3.3 KVM

KVM (Kernel-based Virtual Machine) je mladý projekt šířený jako svobodný software. Původní vývoj směřoval k desktopovému řešení, ale dnes nabízí í řešení pro servery, a to s operačním systémem Windows i Linux. Dnes se využívá i na poli cloud computingu, jako nízkonákladové řešení.

KVM patří mezi platformy pro úplnou virtualizaci i paravirtualizaci. Virtualizační engine je začleněn do jádra systému Linux a nezáleží, jestli jde o verzi x86 nebo x64. KVM pro svůj chod potřebuje hardwarovou podporu (AMD-V nebo Intel-VM). Zvláštností KVM je balíček xenner, který umožnuje spouštět paravirtualizované systémy vytvořené v Xenu.

Obrovskou výhodou je začleněnost do jádra systému Linux, protože jeho vývoj jde neustále kupředu a stále spadá pod licenci open source. Ovládání systému lze provádět jak z grafického rozhraní, tak ze systémové konzole. Druhý způsob je méně náročný na zatížení serveru. Grafické rozhraní potřebuje navíc další prostředky a není v systému zahrnuto - lze však doinstalovat. Grafické rozhraní je možné nainstalovat na jiný Linuxový počítač v síti a přes SSH protokol se k serveru připojit a spravovat jej.

#### 2.3.3.1 Instalace KVM

Virtualizační platforma KVM potřebuje pro svůj chod operační systém. Po nainstalování operačního systému Debian je nutno ověřit, zdali procesor podporuje virtualizaci. To ověříme výpisem informací ze souboru /proc/info například příkazem cat. Pokud v řádku flags najedeme položku vmx (procesory Intel) nebo svm (procesory AMD), je virtualizace procesorem podporována. Pokud se nám tyto parametry nezobrazí, nelze použít hardwarovou virtualizaci, ale pouze pomalejší qemu.

Po ověření hardwarové virtualizace nainstalujeme virtualizaci KVM prostřednictvím balíčkovacího systému aptitude. Instalace balíčků do systému je proces, který upravuje systém, proto je nezbytné ji provádět s právy superuživatele root. Tyto práva lze získat přepnutím na uživatele root, nebo použitím příkazu sudo. Pro virtualizaci KVM jsou nezbytné kromě qemu-kvm i další balíčky, jako jsou libvirt, virtinst a bridge-utils, které slouží ke správě virtualizace (libvirt), instalace a import virtualizovaných OS (virtinst), nebo balíčku umožňujícího vytvoření bridge na úrovni hostujícího OS (bridge-utils). Instalace se spustí po zadání příkazu *apt-get install qemu-kvm libvirt-bin virtinst bridge-utils.*

#### 2.3.3.2 Virtuální síťové rozhraní

Hlavní operační systém je připojen do sítě prostřednictvím vlastní IP adresy, kterou má přidělenou ke konkrétnímu fyzickému rozhraní (např. eth0). Při tvorbě serveru se musíme rozhodnout, jak bude vypadat síťová konektivita pro virtualizované operační systémy. Virtuální stroje mohou být založeny na třech modelech – izolovaný, směrovaný a s překladem adres (NAT). U izolovaného rozhraní nelze hostované OS připojit k internetu a je umožněna pouze komunikace mezi nimi a hostitelem. Směrované rozhraní umožňuje komunikaci se sítí, ke které je připojený i hostitel. Problém nastane, pokud je hostitelský OS připojený k veřejné síti (Internet). Každý virtuální stroj tak potřebuje vlastní veřejnou IP adresu. Poslední možností je použití překladu adres. V tomto modelu je požadována pouze jedna IP adresa pro komunikaci se sítí, ke které je hostitel připojen. Všechny hostované operační systémy se za

tuto adresu schovají. V tomto modelu není potřeba řešit zabezpečení komunikace mezi virtuálními stroji za NATem, ale zabezpečit komunikaci na úrovni fyzického rozhraní. Pomocí knihovny libvirt lze každý síťový model poměrně snadno implementovat, ale naopak má i několik omezení, pro která se vyplatí vytvořit síťové rozhraní pomocí balíku bridge-util.

#### 2.3.3.3 Instalace virtuálního bridge

Virtuální bridge se stává ze dvou částí – definice bridge a definice síťového rozhraní. Všechny konfigurace spočívají v modifikaci souboru */etc/networking/interfaces*.

```
auto virbr1                                          #vytvoření virtuální síťové karty
iface virbr1 inet manual
        pre-up /sbin/ip link add virbr1 type dummy   #po startu OS vytvoří rozhraní
        up /sbin/ip link set virbr1 address 52:54:00:7e:27:af   #při startu rozhraní nastaví
                                                                #MAC adresu

auto virbr10                                         #definice virtuálního bridge
iface virbr10 inet static                            #bridge má statické parametry
        bridge_ports virbr10-dummy                   #síťové rozhraní zařezené do bridge
        bridge_stp on                                #povolení stp protokolu – redukce smyček
        bridge_fd 2
        address 192.168.122.1                        #nastavení IP adresy bridge
        netmask 255.255.255.0                        #nastavení masky sítě
```

Do bridge lze přidat libovolné množství portů. Pokud do bridge přiřadíme i fyzické rozhraní (např eth0), musí mít IP adresu ze stejného rozsahu. Tato úprava má své výhody při použití více veřejných IP adres.

Po restartu operačního systému se zapne virtuální síťové rozhraní a virtuální bridge. Tyto změny lze aplikovat i po restartování služby spravující síťová rozhraní příkazem *service networking restart.*

#### 2.3.3.4 Dnsmasq

Pro přidělený server je k dispozici pouze jedna veřejná IP adresa, proto použijeme překlad adres a vytvoříme síť za NATem. Každá taková síť by měla obsahovat vlastní DNS a DHCP server.

Dnsmasq je snadno konfigurovatelný DNS a DHCP server pro malé sítě. Proti ostatním DNS serverům nejde o náročný software, protože využívá lokálních záznamů standardně uložených v */etc/hosts*, a pokud není schopen na dotaz odpovědět, pošle jej nadřazenému DNS serveru, který je definován v souboru */etc/resolf.conf*. Další výhodou tohoto softwaru je integrovaný DHCP server. Po instalaci DNS serveru do hlavního operačního systému příkazem *apt-get install dnsmasq* je nutné nakonfigurovat soubor */etc/dnsmasq.conf.*

Na fyzickém serveru jsou k dispozici dvě fyzická síťová rozhraní eth0 a od místní sítě odpojené rozhraní eth1. Další rozhraní je virtuální rozhraní zařazené do bridge virbr1 a samotný bridge virbr10. Konfigurační soubor upravíme následovně:

- Vyloučíme rozhraní, na kterých nemá DNS server naslouchat. Nejdůležitější je druhý příkaz, který zakazuje používání DNS serveru z veřejné sítě.
  *except-interface=lo*
  *except-interface=eth0*

- Nastavíme rozhraní, na kterých bude dnsmasq naslouchat.
  *interface=virbr10*

- Nastavíme IP adresu, na které bude server naslouchat.
  *listen-address=192.168.122.1 #IP adresa bridge*

- Nastavíme rozhraní, IP rozsah a dobu zápůjčky IP adresy DHCP serveru
  *dhcp-range=virbr10,192.168.122.2,192.168.122.254,24h*

#### 2.3.3.5 Proxy server

Proxy server je v tomto případě software, který odděluje síť LAN od Internetu. Funguje jako prostředník překládající dotazy klienta na server a posílá zpět odpověď. Bez proxy serveru na hlavním operačním systému nelze pracovat s internetovou konektivitou ve virtuálních systémech. Proxy server v Debianu nainstalujeme pomocí balíčkovacího systému příkazem *apt-get install squid3*. Po instalaci je potřeba provést konfiguraci virtualizovaných operačních systémů. V hlavním operačním systému slouží ke konfiguraci soubor /etc/squid3/squid.conf, ale po instalaci nebyl nijak modifikován. Proxy server v Debianu standardně naslouchá na portu 3128 a používá jako doménu název systému, na kterém je nainstalován. Pokud nechceme použít systémové jméno, ale název vlastní, upravíme v konfiguračním souboru řádek *visible_hostname novy_nazev*.

Ve virtualizovaných operačních systémech distribuce Debian je nutno upravit tři soubory:

- */etc/profile* – konfigurační soubor proměnného prostředí Linuxu. Na konec souboru přidáme následující řádky.

  *MY_PROXY_URL="http://name:3128/"*
  *HTTP_PROXY=$MY_PROXY_URL*
  *HTTPS_PROXY=$MY_PROXY_URL*
  *FTP_PROXY=$MY_PROXY_URL*
  *http_proxy=$MY_PROXY_URL*
  *https_proxy=$MY_PROXY_URL*
  *ftp_proxy=$MY_PROXY_URL*
  *export HTTP_PROXY HTTPS_PROXY FTP_PROXY http_proxy https_proxy ftp_proxy*

- */etc/apt/apt.conf* – konfigurační soubor balíčkovací služby aptitude. Na konec souboru opět přidáme nové řádky.

  *Acquire::http::proxy "http://name:3128/";*
  *Acquire::https::proxy "https:/name:3128/";*
  *Acquire::ftp::proxy "ftp://name:3128/";*

- */etc/wgetrc* – konfigurační soubor programu zabývajícího se stahováním souborů. Často může být použit ke stahování balíčků softwarů starších verzí, které nejsou již dostupné prostřednictvím služby aptitude.

  *http_proxy = http://name:3128/*
  *https_proxy = http://name:3128/*
  *ftp_proxy = http://name:3128/*

Hodnotu name můžeme zaměnit za jméno systému, na kterém běží squid3. Pokud jsme změnili jméno v konfiguračním souboru, musíme použít změněný parametr. Druhou možností je nastavit IP adresu rozhraní, na kterém squid3 naslouchá (virbr10 – IP 192.168.122.1). Po této konfiguraci bude proxy server přijímat od klientů požadavky, které následně přepošle na server a předá jim zpět odpovědi.

### 2.3.3.6 Virtuální stroj – OS Linux

Operační systémy Linux je možné v prostředí KVM paravirtualizovat a tím zvýšit výkon těchto strojů. Každý stroj můžeme opatřit specifickým hardwarem parametry a definovat zda jde o textový nebo grafický operační systém. Tvorbou virtuálních strojů se zabývá balík *virt-install.* Virtuální stroj lze vytvořit následujícím příkazem:

```
virt-install \
--name owncloud \                          #pojmenování virtuálního stroje
--ram 2048 \                               #velikost operační paměti v MB
--disk path=~/cesta/owncloud.img,size=20 \ #stanovení cesty, názvu a velikosti v GB disku
--vcpus 2 \                                #počet procesorů
--os-type linux \                          #specifikování typu OS
--os-variant generic \
--network bridge=virbr10 \                 #připojení virtuálního stroje do bridge
--graphics none \                          #vypnutí grafického režimu
--console pty,target_type=serial \         #nastavení konzole
--location 'url' \                         #místo url zadáme cestu k internetovému zrcadlu
--extra-args 'console=ttyS0,115200n8 serial'
```

Po vytvoření virtuálního stroje se zároveň spustí i instalace, která opět funguje na principu dotaz – odpověď. V instalaci postupujeme stejným způsobem, ale problémy s disky nebo zavaděčem nenastanou. Pokud se systém zeptá na instalaci programů, opět odznačíme desktopové prostředí. Následně vybereme SSH server a dokončíme instalaci.

Po dokončení instalace systému se virtuální stroj restartuje a zamrzne načítací obrazovka, nikoliv virtuální stroj. Problém je způsobem nesprávným nastavením parametrů konzole v zavaděči virtuálního OS, které musíme upravit. Do virtualizovaného systému se lze připojit pouze přes SSH protokol z hlavního operačního systému. Protože IP adresa hostovaného systému je dynamická, musíme ji zjistit pomocí příkazu *arp* -e. Příkaz vypíše všechny adresy IP a MAC včetně názvů rozhraní, ke kterému jsou další zařízení připojena. Po zjištění IP adresy se připojíme k virtuálnímu systému příkazem:

```
ssh uzivatel@IP_adr_virt_stroje
```

Po připojení a ověření uživatele je nutné vytvořit případně editovat soubor */etc/init/ttyS0.conf* následující konfigurací:

*start on stopped rc or RUNLEVEL=[2345]*
*stop on runlevel [!2345]*
*respawn*
*exec /sbin/getty -L 115200 ttyS0 vt102*

Editovaný soubor */etc/init/ttyS0.conf* uložíme a provedeme úpravu souboru zavaděče */etc/default/grub*. Ve zdrojvém kódu povedeme několik úprav:

- Upravíme řádek *GRUB_CMD_LINUX=““* na *GRUB_CMDLINE_LINUX="text console=tty0 console=ttyS0,115200n8"*
- Řádek *GRUB_TERMINAL=console* odkomentujeme a upravíme na hodnotu *serial*. Tato úprava má za následek vypnutí console a zapnutí sériového rozhraní, přes které budeme přistupovat k virtuálnímu systému. Pod odkomentovaný řádek ještě vložíme další parametry sériového rozhraní jako je např. rychlost

  *GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --stop=1"*

Výše provedené změny musíme aktualizovat v zavaděči systému příkazem *update-grub* a vypneme virtuální stroj. Poslední úpravu je nutno provést v samotném XML souboru virtuálního stroje spuštěného příkazem *virsh edit nazev_virtualniho_stroje*. V otevřeném XML souboru nalezneme všechny parametry virtuálního stroje, jako jsou počet a typy síťových karet, CPU a mnoho dalších. Ve struktuře XML vyhledáme definici konzole a upravíme je:

```
<serial type='pty'>
    <target port='0'/>
</serial>
<console type='pty'>
    <target type='serial' port='0'/>
</console>
```

Soubor uložíme, spustíme virtuální počítač a můžeme se přihlásit.

#### 2.3.3.7 Virtuální stroj – OS Windows

Hlavní výhodou instalace operačního systému bez grafického rozhraní je úspora hardwarových prostředků a zvýšení výkonosti virtuálního stroje. Takový stroj je nutno spravovat pomocí příkazů zadávaných do terminálu. Operační systémy založené na Linuxu nechávají na uživateli volbu instalace textového nebo grafického režimu. U systémů z rodiny Windows ale tato možnost neexistuje a grafické rozhraní je nutné. V Linuxu je z grafického prostředí správa virtuálních strojů prováděna pomocí aplikace zvané *virt-manager*, ale při použití textového režimu nelze použít a tedy ani zobrazit obrazovku virtualizovaného grafického operačního systému z hlavního operačního systému. Většina serverů je spravována vzdáleně a pro tuto potřebu byl vytvořen protokol SPICE, který umožňuje přenášet obraz prostřednictvím sítě LAN. Ze vzdáleného počítače se tak lze připojit na server prostřednictvím virt-manageru (OS Linux) nebo virt-vieweru (OS Windows)

Instalace sytému Windows se opět provádí pomocí balíku virt-install, který můžeme nainstalovat následujícím způsobem:

```
virt-install \
-n win7 \                          # nastavení jména virtuálního stroje
-r 2048 \                          #přidělení operační paměti v MB
--vcpus=2 \                        # nastavení počtu procesorů
--os-type=windows \                #specifikace operačního systému
--os-variant=win7 \                #upřesnění distribuce
--cdrom ~/cesta/win7prof.iso \
--disk path=~/cesta/virtio-win-drivers-20120712.vfd,device=floppy \
--disk ~/cesta/win7.img,size=30,device=disk,bus=virtio \
-w bridge=virbr10,model=virtio \   #nastavení síťové karty
--graphics spice,listen=0.0.0.0 \  #nastavení grafického protokolu SPICE a
                                   IP adresy, na které SPICE server naslouchá
--noautoconsole                    #vypnutí terminálového přístupu
```

Aby instalace mohla být úspěšně dokončena, je potřeba kromě obrazu instalačního disku potřeba stáhnout VIRTIO ovladače. Ovladače musí být umístěny jako obraz disketové mechaniky, protože se dvěma virtuálními CDROM mechanikami neumí virt-install pracovat. Po spuštění virtuálního stroje a tedy i instalace je dostupný SPICE server, ke kterému se vzdáleně připojíme.


**Obrázek 10 – Remote Viewer připojení k serveru SPICE**

V novém okně se otevře obrazovka vzdáleného počítače a pokračujeme v instalaci běžným způsobem. Instalace pokračuje bez problému až do místa, kdy mají být načteny připojené disky. Instalační WindowsPE ve svém základu neobsahuje ovladače pro jejich načtení. Tyto ovladače nainstalujeme z připojené virtuální disketové mechaniky a dokončíme instalaci.

Počáteční konfiguraci systému Windows provedeme po prvním startu. Nastavíme proxy server, statickou IP adresu, DNS servery a stáhneme ovladače systému. Systémy Windows bývají často napadány viry, proto nainstalujeme i základní antivirovou ochranu.

#### 2.3.3.8 Ovládání KVM

Z grafického prostředí je ovládání a správa virtuálních strojů prováděna pomocí programu zvaného *virt-manager*. Při použití terminálového režimu příkazem *virsh* otevřeme prostředí pro správu virtualizace. Mezi základní příkazy pro správu patří příkazy:

| | |
|---|---|
| *console name* | *#připojení k terminálu* |
| *start name* | *#spuštění virtuálního stroje* |
| *shutdown name* | *#vypnutí virtuálního stroje* |
| *destroy name* | *#násilné vypnutí* |
| *reboot name* | *#restartování stroje* |
| *autostart name* | *#automatické spuštění po startu PC* |
| *autostart --disable name* | *#vypnutí automatického zapínání* |

*list --all* *#výpis všech virtuálních strojů*

*exit* *#ukončení prostředí pro správu*

#### 2.3.3.9 Připojení a formátování pevného disku

Přidělený server disponuje třemi pevnými disky. Dva menší disky zapojené v RAIDU 1 o kapacitě 320 GB a třetím o kapacitě 2 TB určeného pro ukládání dat. Aby bylo možné pevný disk použít, je nutné ho naformátovat a vytvořit souborový systém. Užitečným nástrojem pro správu pevných disků v Linuxu je utilita *fdisk*. Informace o připojených discích a RAID polích vypíšeme příkazem *fdisk –l*. Vybereme disk, který chceme formátovat a poznamenáme jeho umístění např. */dev/sdc1*. Utilitu pro správu konkrétního disku spustíme příkazem *fdisk /dev/sdc*. Otevře se příkazové prostředí utility, kde pomocí písmen zadáváme příkazy. Nejdříve smažeme stávající oddíl zadáním písmena d. Následně vytvoříme nový oddíl volbou n, změny zapíšeme na disk volbou w a ukončíme utilitu příkazem q. Novému oddílu vytvoříme jmenovku *e2lalbel /dev/sdc1 DataDisk* a spustíme formátování *mkfs.ext3 /dev/sdc1*.

Na nově naformátovaném disku vytvoříme dvě složky příkazem *mkdir owncloudData* a *mkdir sambaData*. Každá složka musí obsahovat obraz virtuálního disku pro jeden virtuální stroj o velikosti 800GB. Disk vytvoříme použitím syntaxe

*qemu-img create –f raw owncloud.img 800G.*

Nově vytvořené obrazy disků připojíme k virtualizovanému operačnímu systému

*virsh attach-disk owncloud --source /media/DataHDD/owncloudData/owncloud.img *
*--target vdb –persistent*

Po připojení pevného disku se přepneme příkazem *virsh console owncloud* do virtuálního systému a opět pomocí příkazu *fdisk /dev/vdb1* vytvoříme nový oddíl, který následně naformátujeme *mkfs.ext4 /dev/vdb1*. Naformátovaný pevný disk musíme připojit do virtuálního OS. Nejdříve vytvoříme složku, do které budeme chtít disk připojit *mkdir /dev/data* a souborový oddíl připojíme ke složce *mount /dev/vdb1 /mnt/data*. Připojený disk už jen zavedeme do tabulky souborových systémů tak, že do souboru */etc/fstab* připíšeme řádek */dev/vdb1 /mnt/new-disk ext4 defaults 0 0.*

Stejným způsobem vytvoříme i pevný disk pro virtuální stroj samba.

# 3 SLUŽBY PROVOZOVANÉ NA SERVERECH

Servery jsou výkonné počítače, které tvoří infrastrukturu počítačové sítě. Hlavním důvodem proč používáme počítače je distribuce služeb ostatním uživatelům. Každý server nabízející nějaké služby musí být stabilní a odolávat výpadkům. Na každém serveru se doporučuje provozovat jednu službu a proto je využívána virtualizace.

Jako službu označujeme nějakou aplikaci dostupnou ostatním uživatelům. V případě serverů tyto aplikace distribuujeme prostřednictvím sítí. Celý internet je počítačová síť, pomocí které poskytovatelé služeb své služby nabízejí, aniž by o tom koncový uživatel věděl. Všechny služby komunikují mezi počítači prostřednictvím protokolů a portů. Na serverech můžeme provozovat jednu nebo více služeb jako jsou web, VPN, sdílení souborů, email,…

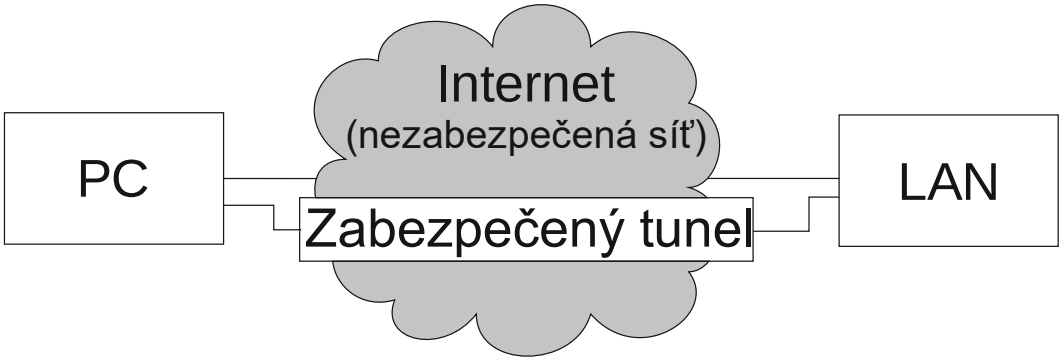
Služba web je provozována webovým serverem. Každý takový server má na svém disku uložené webové stránky, které prostřednictvím protokolu HTTP nabízí klientům. Může jít o jednoduché statické webové stránky založené pouze na jazyce HTML (HyperText Markup Language) nebo o dynamické stránky, které pro svůj chod potřebují podporu skriptovacího jazyka jako je PHP (Hypertext Preprocesor).

Nejčastěji používaným webových serverem je Apache. Server je dostupný jak pro operační systémy typu Unix, Linux tak i Windows. Konfigurace serveru se provádí pomocí konfiguračních souborů. Pro vytváření více webových rozhraní není potřeba provádět další instalace na jiné virtuální stroje, jelikož sám si umí jednotlivé virtuální instance vytvářet a spravovat. Je velmi snadno modifikovatelný a jednoduchým způsobem lze přidávat další funkce jako je podpora jazyka PHP nebo práce s databázemi. Bez dalších komplikací umí používat i protokol FTP, který je na webhostingu stále nezbytný.

PHP je skriptovací programovací jazyk, určený především pro programování internetových stránek a webových aplikací. Vznikl v roce 1995 a bývá využíván v kombinaci s MySQL nebo PostgreSQL. Skripty vytvořené v PHP jsou prováděny pomocí interpreta, který čte řádek po řádku a provádí aktuální příkazy.

Databázové servery jsou stroje pro ukládání informací. Tyto informace jsou zpřístupněny pomocí SQL serveru. Komunikace probíhá zasíláním dotazů na server, který je vyhodnocuje a odpovídá na ně. Aktuálně existuje mnoho databázových programů založených na různých principech.

Poslední zde zmíněnou službou je VPN. Pomocí této služby můžeme přes internet vytvořit zabezpečený kanál. Prostřednictvím těchto šifrovaných tunelů můžeme uživateli povolit přístup ze sítě internet do intranetu. Po úspěšné autentizaci a autorizaci klienta mu mohou být zpřístupněny sdílené firemní prostředky jako síťové disky, databázové servery či jiné služby. Sítě VPN jsou spravovány VPN servery, které s klienty komunikují prostřednictvím protokolů (PPTP, SSTP, IPsec, OpenVPN,…).




**Obrázek 11 - Princip VPN**

# 4 PŘÍSTUP K DATŮM

## 4.1 NFS

NFS neboli Network File System (síťový souborový systém) je protokol určený k přenosu a sdílení dat ze vzdáleného počítače vyvinutý firmu Sun Microsystems. Je primárně určen pro sdílení dat mezi operačními systémy typu Unix, které ho mají zakomponovaný ve svém jádře. První verze NFS byly postaveny na transportním protokolu UDP, ale od verze NFSv3 lze použít i TCP. Ověřování úspěšnosti přenosu je již zakomponováno v protokolu v kontrolních součtech NFS, proto i přenos přes UDP je spolehlivý.

Protokol NFS od jeho vydání obsahoval bezpečnostní chyby, které byly opraveny až po vydání verze NFSv4. Základní bezpečnostní mechanismy a autentifikace uživatele jsou zajištěny použitím protokolu Kerberos také od zmíněné verze. Bez použití protokolu Kerberos je přístup uživatelů ke sdílenému adresáři ověřován na úrovni počítače pomocí DNS jména nebo IP adresy počítače. Pro každého uživatele tak musíme na serveru vytvořit záznam, kam uložíme informace o jeho počítači. Data jsou standardně sdílena s oprávněním pouze pro čtení, ale podle přepínačů můžeme nastavit i oprávnění zápisu. Při použití technologie Kerberos můžeme ověřit i daného uživatele pomocí jména a hesla, ale konfigurace je složitější.

Další nevýhodou proti sambě je složitější konfigurace i na straně klienta, kterému musíme nastavit i jaká má oprávnění (zápis, čtení, může spouštět programy,…).

Výhodou NFS je možnost vytvoření diskového oddílu, který Linux může považovat za svůj root oddíl. Do takového oddílu můžeme nainstalovat celý operační systém a vytvořit tak tzv. bezdiskové stanice. Data tak můžeme centrálně spravovat na jednom počítači a projeví se v celé infrastruktuře.

## 4.2 FTP

FTP alias File Transfer Protocol je jeden z prvních protokolů v internetu. Od počátku byl určen k přenosu souborů přes internet ještě před vytvořením protokolu HTTP. Je určen ke stahování nebo odesílání souborů přes internet. Při provozování FTP máme dvě možnosti přístupů k souborům. První variantu můžeme nejčastěji najít u všech webhostingových serverů, kde pomocí uživatelského jména a hesla můžeme umisťovat a spravovat www

stránky. Druhá varianta je označována jako anonymní FTP. Typickým příkladem jsou zrcadla operačního systému Linux. Jakýkoliv uživatel daný soubor stáhnout a použít.

Proti NFS a Sambě je tento protokol jednodušší a zajišťuje netransparentní sdílení souborů v síti. Netransparentní způsob znamená, že koncový uživatel musí vědět, kde se daný soubor nachází, aby k němu získal přístup. U NFS a Samby tohle jako uživatelé řešit nemusíme, protože sdílený adresář ze souboru máme namapovaný například jako síťový disk.

Zajímavostí protokolu FTP je jeho koncept, protože pro komunikaci používá dva porty. Konkrétně jde o porty s čísly 20 a 21. Skrz port 21 server komunikuje s klientem a řídí tak komunikaci posíláním řídících zpráv. Druhý zmíněný port pak již slouží pro samotný přenos souborů.

Protokol FTP je založen na modelu klient-server a od toho jsou také odvozeny následující dva režimy navázání spojení. Při pasivním přenosu dat je spoj navázán klientem z libovolného dynamického portu na dynamický port serveru. Protože porty nemusí a nejsou blíže specifikovány, může nastat problém s komunikací, pokud se server nachází za firewallem nebo NATem. Typicky může jít o jakýkoliv veřejně dostupný FTP server. Druhý způsob přenos označovaný jako aktivní se těmto problémům vyhýbá, protože datový tok je zahájen ze serveru na portu č. 20 na libovolný port na straně klienta.

## 4.3 Samba

Samba je komplexní svobodný projekt poskytující služby sdílení souborů a tiskáren především pro systém Windows. Tyto služby je možné distribuovat i jiným systémům, avšak po doinstalování příslušných balíčku. Je nativně určená pro běh pod operačními systémy z rodiny Unix. Její implementace je jednoduchá a vyznačuje se nízkými náklady na provoz a údržbu. Pořizovací náklady jsou téměř nulové, jelikož je dostupná zdarma a nejsou potřeba zakoupit žádné licence pro klientské systémy.

Samba je nástroj implementující mnoho protokolů a služeb jako CIFS, NetBios po TCP/IP, Wins atp. Tento nástroj je složen ze dvou dílčích daemonů, kteří poskytují své zdroje. Tyto daemoni se nazývají snbd a nmbd. První poskytuje po síti služby pro procházení souborů a tiskáren a také jim poskytuje autentizaci a autorizaci. Druhý implementuje Microsoft NetBIOS Name server.

Z výše uvedených skutečností vyplývá, že samba pro svou funkci potřebuje několik dalších služeb. Každá služba komunikující přes síťové rozhraní, využívá TCP nebo UDP protokol na daném portu. Sambou jsou využívány porty 137, 138, 139 a 445.

Systém Samba je snadno konfigurovatelný pomocí souboru smb.conf. Jde o textový soubor, kde můžeme nastavit všechny potřebné parametry a údaje o sambě. Systém lze také konfigurovat pomocí webového rozhraní, které jde standardně vypnout z bezpečnostních důvodů.

Celý systém samba stojí na protokolu SMB (Server Message Block), který byl primárně určen pro aplikační rozhraní NetBIOS - vyvinuté společností IBM. Za dobu vývoje byl nápad firmy IBM převzat společností Microsoft a integrován do systému Windows. Proto dnes protokol SMB komunikuje skrze protokoly TCP/IP. Je založen na SMB žádostech klienta a SMB odpovědích serveru.

### 4.3.1 Instalace a konfigurace

Sambu nainstalujeme do Linuxu prostřednictvím balíčkovací služby přikazem *apt-get install samba*. Po dokočení instalace přejdeme ke konfiguraci a tvorbě sdílených adresářů. Sdílené adresáře vytvoříme běžným způsobem jako složky příkazem mkdir, např. *mkdir /mnt/sambaData/data*. Každé složce musíme přidělit skupinu uživatelů, která má práva se složkou pracovat. Skupinu založíme *groupadd data* a připojíme ji ke složce *chgrp data /mnt/sambaData/data*. Vytvořené složce je potřeba nastavit práva a omezit tak vstup uživatelů mimo vytvořenou skupinu *chmod 770 /mnt/sambaData/data.* Číslo 770 je číselné vyjádření pravomocí uživatelů. Práva jsou nastavována postupně pro vlastníka, skupinu a ostatní uživatele. Číslo 7 je součtem základních pravomocí – čtení (read – č. 4), zápisu (write – č. 2) a spouštění (x – č. 1). Pokud chceme vlastníkovi, skupině nebo ostatním uživatelům zamezit přístup ke složce, použijeme číslovku 0.

Aby sambu bylo možné používat, nestačí pouze vytvořit složky a nastavit oprávnění. Sambu musíme nastavit v souboru */etc/samba/smb.conf* . Samba je projekt pro sdílení mezi různými operačními systémy a proto jí musíme definovat, jaké chceme používat kódování. To definujeme přidáním řádku *unix charset = UTF-8* do globálních nastavení například pod řádek *## Browsing/Identification ###*. O několik řádků níže ověříme správné nastavení pracovní skupiny *workgroup = WORKGROUP*. Skupina workgroup je výchozí nastavení pro počítače Windows. Pokud používáme jinou

skupinu tak název změníme. Samba ve výchozím nastavení umožňuje sdílet i domovské adresáře uživatelů, které jsou uložené na systémovém disku. V této práci data ukládám na datový disk, proto domovské adresáře nechci sdílet a zakomentuji přidáním středníku na začátek řádku tyto řádky

```
[homes]
  comment = Home Directories
  browseable = no
  read only = yes
```

Na konec celého souboru už jen přidáme řádky, kterými nastavíme sdílení složky data.

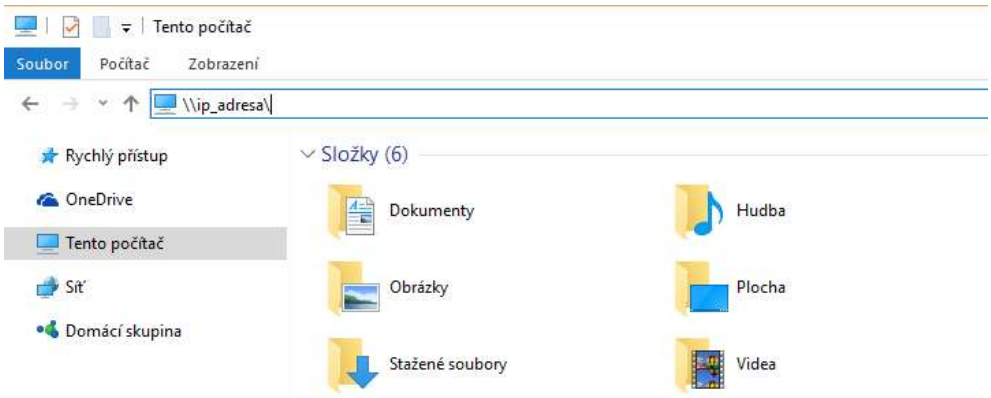
```
[Data]                          #jméno složky v síti
  path = /home/security         #cesta k složce
  writable = yes                #povolení zápisu do složky
  create mode = 0770            #oprávnění pro objekty uvnitř složky
  directory mode = 0777         #oprávnění pro složku
  guest ok = no                 # zakázání přístupu uživatelům mimo skupinu
  valid users = @data           #přiřazení skupiny, která má právo přistupovat
```

Po aktuálním nastavení, musíme do skupiny data přidat již vytvořené uživatele v operačním systému *smbpasswd -a garfield* a dále zadáme heslo, ke kterému nás systém vyzve. Vytvořeného uživatele přiřadíme ke skupině data *usermod -G data garfield* a restartujeme démona spravujícího sambu *service smbd restart.*
Pokud se snažíme nastavit heslo uživateli, který v systému Linux neexistuje, samba ohlásí při zadávání hesla chybu. Tato chyba lze odstranit vytvořením uživatele a opakováním výše uvedených příkazů

```
useradd garfield                #přidáme uživatele do systému
usermod -L garfield             #zamknutí uživatele
```

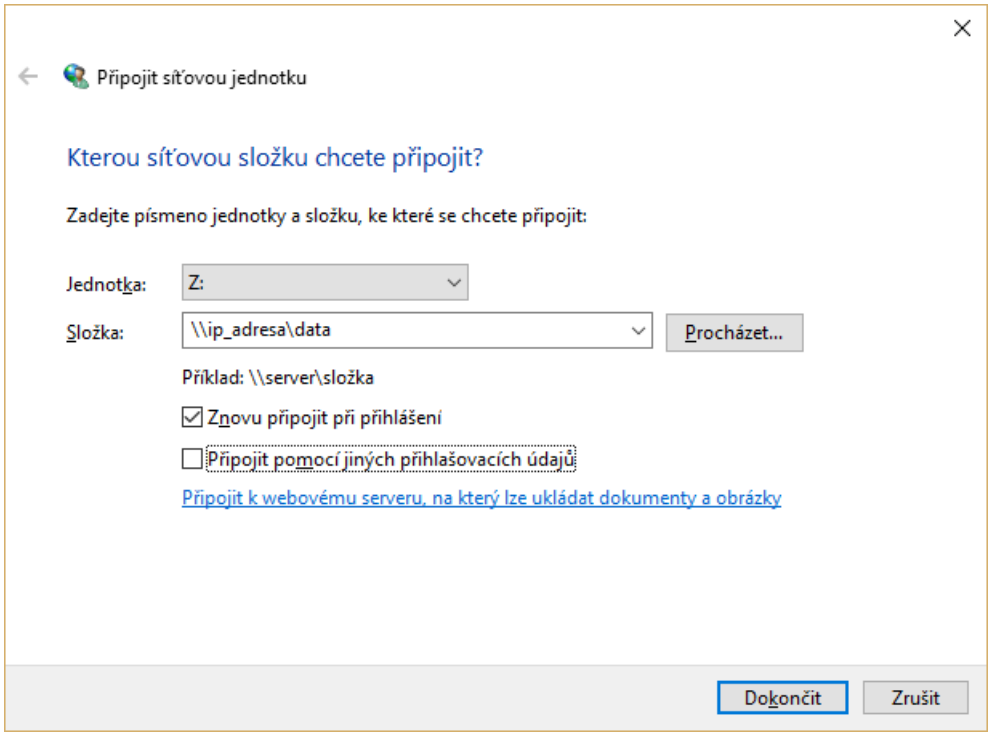
### 4.3.2 Nastavení klientského PC

Na klientských počítačích Windows klikneme na průzkumník souborů a do pole adresa zadáme *ip_adresa_serveru.*




**Obrázek 12 - Zadání IP do průzkumníka Windows**

Po pokusu o přihlášení si server vyžádá ověřovací údaje. Po jejich zadání zobrazí všechny sdílené složky. Podle nastavení Samby může uživatel vstoupit pouze do složek, ke kterým má oprávnění. Po kliknutí na vybranou složku pravým tlačítkem ji lze namapovat jako lokální disk stanice.




**Obrázek 13 - Namapování sdílené složky jako disk**

# 5 CLOUDOVÁ ÚLOŽIŠTĚ

## 5.1 Cloud computing

Termín cloud computing je dnes hojně užívaným termínem. Princip této technologie je založen na využití internetu a sdílení prostředků vzdálených serverů.

Tato technologie uživateli umožňuje přistupovat ke svým datům a pracovat s nimi odkudkoliv. Uživatel může provádět různě náročné operace (např. vyhledávat v rozsáhlých databázích), ale nestará se o to, jaké prostředky potřebuje nebo má k dispozici.

Z pohledu provozovatele cloudu je situace značně komplikovanější. Správné cloudové úložiště musí umět dobře pracovat se svými prostředky. Požadavky na sdílený výkon se dynamicky mění a cloudové centrum se jim musí umět přizpůsobit. Pokud je potřeba více prostředků, může je přidělit, nebo v případě nadbytku odebrat a přidělit jiným uživatelům. Klienti v tomto případě platí za spotřebované prostředky cloudového úložiště (poskytnutý výpočetní výkon, úložné kapacity atp.). Pokud chce klient ušetřit za využití těchto zdrojů, může si postavit vlastní výpočetní centrum se servery, které propojí do malého cloudu a propojí je s cloudem dodavatele. V takovém případě klient primárně využívá prostředky svého cloudu.

### 5.1.1 Model nasazení

Jedná se o první způsob dělení cloudu, který nám říká, jakým způsobem je systém poskytován koncovému uživateli. Dělíme jej na čtyři skupiny:

- Veřejný cloud – veřejný poskytovatel nabízí své prostředky koncovým uživatelům jako službu. Tyto služby mohou být placené nebo neplacené. Jaké prostředky jsou uživateli poskytnuty a o tom, jestli jsou poskytovány zabezpečeně nebo nezabezpečeně rozhoduje poskytovatel.
- Privátní cloud – typický pro cloud umístěný v sídle koncového uživatele (organizace). Prostředky tohoto centra jsou poskytovány ostatním uživatelům typicky po intranetu.
- Komunitní cloud – tento model je typický pro privátní cloud, jehož prostředky jsou sdílené pro skupinu lidí nebo mezi několika organizacemi. Nejčastěji může jít o nějaké uživatele se stejným zájmem nebo zázemím (sesterské organizace).

- Hybridní cloud – kombinuje prvky cloudu privátního a veřejného. Organizace si postaví svůj privátní cloud, který primárně využívá. Pokud tomuto systému nedostačují sdílené prostředky, mohou být využity i prostředky veřejného cloudu poskytovatele.

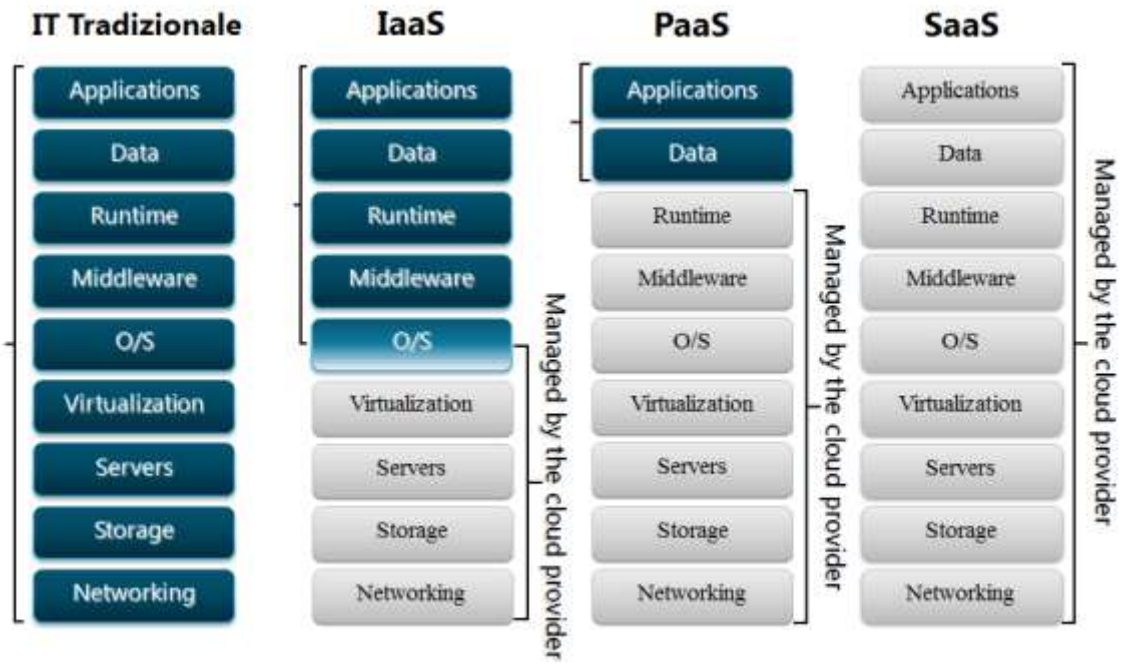
### 5.1.2 Dělení cloudu podle modelu služeb

Dělení podle modelu služeb můžeme též nazvat jako distribuční model. Distribuční model rozdělujeme na tři vrstvy:

- IaaS – infrastruktura jako služba, poskytuje pouze základní prvky cloud computingu. Poskytovatel spravuje pouze poskytovaný hardware, jako jsou servery a datová úložiště a softwarový hypervizor. Zákazník tak má neomezené možnosti, jak takové centrum využít. Také může spravovat a vytvářet vlastní virtuální stroje. Veškeré náklady na další licence si hradí zákazník. Účtování poplatků za tuto službu je závislé na přístupném hardwaru, jako je počet procesorů, datová velikost uložených dat, doba jeho používání nebo propustnost komunikační sítě. Tato služba je vhodná především pro začínající firmy, protože zákazník nemusí nakupovat vlastní zařízení a tím může výrazně ušetřit další náklady. Nejznámějšími zástupci této vrstvy jsou Amazon a GoGrid.

- PaaS - platforma jako služba poskytující zázemí pro vývoj aplikací. Uživatel neřeší to, na čem jeho aplikace běží. Poskytovatel nabízí svůj hardware, na kterém provozuje a spravuje vlastní virtuální stroje podle požadavků zákazníka. Zákazník neřeší náklady na další licence za operační systémy, ale pouze licence za svoje provozované aplikace. Tato vrstva modelu je nejmladší, a proto zatím není příliš rozšířená. PaaS je provozován společnostmi jako AppEngine či Force.com.

- SaaS – software jako služba poskytující kompletní servis pro koncové uživatele. Pod pojmem cloud si většina uživatelů představí právě tuto službu, kdy máme implementovány na serveru všechny potřebné balíky od softwaru přes kalendář, až pod email. Dodavatel tak přes internet nabízí standartní, ale i zakázkový software, který udržuje a aktualizuje. Klient platí skutečně to, co využívá. Neřeší tak žádné pořizovací náklady na licence ani hardware. Poskytovatel si ale účtuje kromě skutečně využitých služeb i pravidelný paušální poplatek. Tyto služby poskytuje například společnost Google se svými Google Apps nebo Dropbox.




**Obrázek 14 - Distribuční model cloud computingu [63]**

## 5.2 Cloudová řešení

### 5.2.1 Google Drive a OneDrive

Obě řešení jsou komerční a ke svému webovému prostoru nabízí i další služby, jako je kalendář, online správa dokumentů, sdílení dat pomocí odkazů atd. Každá společnost nabízí klienta pro platformu Windows i Linux. Instalace Microsoft klienta je trochu složitější na Linuxu a musí se řešit přes správce balíčků. Výhodou OneDrive je klient integrovaný do operačních systémů Windows 8 a Windows 10.

Abychom mohli cloud využívat, musíme se zaregistrovat. Myšlenkou vývojářů je pod jedním účtem nabízet více než cloudové úložiště. Každý z vývojářů to vzal podle svého. Microsoft pod svým účtem nabízí nejen přístup k OneDrive, ale umožňuje i sdílet nastavení uživatelových počítačů přes internet. Za ukázkový příklad lze považovat změnu tapety na jednom počítači, která se projeví i na ostatních počítačích. Prostřednictvím účtu Microsoft také můžeme přistupovat do jejich online obchodu Store a stahovat zde zajímavé aplikace pro desktopy i mobilní telefony. Společnost Google vychází ze stejné myšlenky, ale orientuje ji zejména na mobilní operační systém Android. Na podobné myšlence funguje i iCloud od firmy Apple.

### 5.2.2 Dropbox

Dropbox patří mezi největší komerční cloudová řešení. Umožňuje přistupovat k datům odkudkoliv pomocí webového rozhraní nebo klientského softwaru. Jde o nejvíce multiplatformní cloudový systém, jehož klientský systém je vyvinut pro většinu platforem, ať již mobilních nebo desktopových, jako jsou Linux, Windows, Mac OS X, Android, iOS, BlackBerry nebo Windows Phone. Neoficiální verze klienta je vyvinuta i pro mobilní systém Symbian.

Dropbox je licenčně rozdělen na čtyři modely – freemium, pro, business a enterprise. Každá verze se liší velikostí úložiště a denní limitem na přenos dat. Tento limit u verze freemium činí 10% než u verzí business a pro. Všechny verze umožňují manipulovat s daty a mají integrován MS Office 365 pro jejich úpravu. Vyšší verze nabízí plno dalších nástrojů a vylepšení jako je zálohování dat, zabezpečení odkazů pro sdílení souborů heslem, vzdálené zablokování přístupu pro dané zařízení nebo začlenění služeb třetích stran.

Synchronizační nástroj, jako u většiny cloudových řešení, je může být integrován do operačního systému a na pozadí provádět zálohování a synchronizaci dat. Data jsou ukládána do cloudu zašifrovaně stejně jako i jejich přenos je zašifrovaný. O bezpečnost dat se starají dnešní standarty jako SSL a AES-256. Tímto je docíleno zabezpečení dat proti zneužití.

### 5.2.3 Aero FS

Komerční cloudový systém určený pro nasazení na vlastních serverech. Server nabízí vysoké zabezpečení dat, neboť přenos dat mezi zařízeními je zašifrován standartními průmyslovými technologiemi jako je RSA-2048, AES-256 a SSL/TLS.

Jednou z hlavních výhod tohoto cloudu je rychlost synchronizace dat. Cloud může běžet na serveru, ale není to podmínkou. Aero FS umožňuje synchronizovat data peer-to-peer mezi koncovými zařízeními. Pokud cloud je umístěn na serveru, můžeme soubory nahrávat, vytvářet verze souborů, sdílet, spravovat a synchronizovat. Synchronizace dat probíhá skrze synchronizačního klienta. Tento synchronizační nástroj je dostupný na desktopové i mobilní platformy jako jsou Windows, Linux, Mac, Android a iOS. Aero FS podporuje snadnou integraci do infrastruktury Active Directory. S rozvojem mobilních platforem a snadnou dostupností dat kdekoliv roste i bezpečnostní riziko neoprávněného přístupu k datům skrze odcizená zařízení. Tento systém je založen na přístupových certifikátech a umožnuje administrátorům zamezit přistup k datům z odcizených zařízení jejich zablokováním.

I když jde o komerční produkt, existuje verze pro týmy, kterou můžeme stáhnout i užívat zdarma. Tato verze má mnoho omezení, mezi která patří možnost vytvořit maximálně 30 uživatelských účtů. Další omezení se týkají zejména správců, kterým není umožněno vzdáleně omezit a řídit přístupy uživatelů, není možné provádět auditování přístupů nebo cloudový systém zaintegrovat do služby Active Directory. Pokud jsou tato omezení pro uživatele podstatná, musí si připlatit za enterprise řešení. Řešení enterprise už není zdarma pro omezený počet uživatelů, ale platí se za licenci pro každého uživatele.

Instalace této technologie na server není nijak náročná. Z internetu po vyřízení všech podmínek výrobce umožní z vlastních stránek stáhnout virtuální disk, na kterém je systém předinstalovaný. Tento virtuální disk je už nezbytné jen spustit pomocí nějakého virtualizačního nástroje (např. VirtualBox). V případě serveru ho připojíme k virtuálnímu stroji, spustíme a provedeme dodatečnou konfiguraci.

### 5.2.4 Pydio

Pydio (Put Your Data In Orbit) je svobodný software šířený pod licencí GPL. Cloud pro svůj chod potřebuje být umístěn na webovém serveru s podporou PHP, JavaScriptu a MySQL. Pydio umožňuje data sdílet, spravovat, synchronizovat a spravovat pomocí integrovaného textového editoru.

Proti jiným cloudům nabízí pouze základní funkce. Jeho desktopový klient je na bázi jazyka Java a je dostupný pro Windows a Mac OS. Pro mobilní přístup slouží aplikace pro iOS a Android. Ačkoliv Pydio není příliš rozšířené, najdeme ho na všech zařízeních NAS firmy LaCie, kde zastupuju roli správce souborů. Instalace není náročná a zvládne ji i začátečník.

### 5.2.5 OwnCloud

OwnCloud je balík pro vytvoření cloudového řešení. Je velmi podobný Dropboxu, který mimo jiný připomíná i svojí funkčností. Tento cloudový systém je vytvářen jako svobodný systém pod licencí open-source. Je snadno dostupný a můžeme jej snadno nainstalovat na svůj vlastní server. Uživatel tak už není limitován přiděleným prostorem, ale prostorem fyzickým. Další výhodou tohoto řešení je snížení rizika úniku citlivých dat.

Řešení OwnCloud mimo cloudu nabízí i další služby jako kontakty, záložky, službu pro zkracování odkazů, kalendář atd. Služby, které nejsou v balíku nativně implementovány, lze snadno doinstalovat použitím přídavných modulů. OwnCloud je kompatibilní se všemi

platformami. Do klientských počítačů lze naistalovat aplikaci, která umožní uživateli svá data synchronizovat se serverem nebo vytvářet verze souborů.

OwnCloud je serverová aplikace, která pro svůj běh potřebuje webový server s podporou jazyka PHP a nainstalovanou jednu z mnoha podporovaných databází. Instalaci aplikace zahájíme nahráním staženého balíku na server a po zadání IP adresy webového serveru se nám zobrazí konfigurační dialog, pomocí kterého celý cloudový systém nastavíme a uvedeme do chodu.

#### 5.2.5.1 Instalace MySQL

Mysql server nainstalujeme známým způsobem na vlastní virtuální stroj *apt-get install mysql-server-5.5 phpmyadmin.* Instalační proces nás při instalaci vyzve k zadání hesla pro uživatele databáze root. Je-li instalace úspěšně dokončena, spustíme správce databáze *mysql –u root –p* a zkontrolujeme nastavení domén pro uživatele root dotazem

*select user,host,password from mysql.user;* *#na konci dotazu vždy ;*

Dotaz nám vypíše následující tabulku, ve které musí být záznam pro hosta gandalf-mysql

**Tabulka 1- Nastavení domén uživatele root**

| user | host | password |
|---|---|---|
| root | localhost | *13E45A2779A19D9AA45A942DABD |
| root | gandalf-mysql | *13E45A2779A19D9AA45A942DABD |
| root | 127.0.0.1 | *13E45A2779A19D9AA45A942DABD |
| root | ::1 | *13E45A2779A19D9AA45A942DABD |
| debian-sys-maint | localhost | *13E45A2779A19D9AA45A942DABD |

S databází MySQL byl nainstalováni i webový správce databáze phpMyAdmin, kterého nechceme spouštět po startu virtuálního stroje kvůli úspoře systémových prostředků. Tuto službu vypneme příkazem *update-rc.d apache2 disable.* Další změnu provedeme v souboru */etc/apache2/ports.conf*, kde provedeme změnu portu z *Listen 80* na *Listen 81*. Na portu 80 bude naslouchat webový server cloudového úložiště, který je umístěn na jiném stroji.

#### 5.2.5.2 Instalace OwnCloud

Cloudové úložiště nainstalujeme na jiný virtuální stroj, než máme databázi MySQL. Instalace cloudového úložiště je složitější neboť balíčkovací systém standardně zahrnuje staré verze. Nové verze musíme zavést sérií příkazů.

```
wget -nv \
https://download.owncloud.org/download/repositories/stable/Debian_8.0/Release.key \
-O Release.key                          #získáme klíč starší verze 8.0
apt-key add - < Release.key             #získaný klíč vložíme do balíčkovací služby
sh -c "echo 'deb http://download.owncloud.org/download/repositories/stable/Debian_8.0/ /'
>> /etc/apt/sources.list.d/owncloud.list"   #nastavíme odkaz, kde hledat aktualizace
apt-get update                          #aktualizujeme službu aptitude
apt-get install owncloud                #naistalujeme balík owncloud
```

S instalací owncloudu se nainstalují i některé nepotřebné balíky pro databázi MySQL. Naše cloudové úložiště bude ale používat jiný databázový server, proto můžeme tyto balíky odinstalovat

```
apt-get remove mysql-server
apt-get remove mysql-server-5.5
apt-get remove mysql-client-5.5
```

Výše zadané příkazy provedly instalaci systému a další konfiguraci provedeme z webového prohlížeče připojením na doménu http://ip_serveru/owncloud. Za IP adresu serveru jsem použil v lokální síti IP 192.168.0.111. Načítání stránky selže s výpisem chyby *Nelze načíst soubor index.php*. Tato chyba lze opravit zakomentováním řádku *RewriteRule .* index.php [PT,E=PATH_INFO:$1]* v souboru /var/www/owncloud/.httacces. V tomto souboru také nastavíme vynucení zabezpečeného připojení následujícím kódem

```
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R,L]
```

Po úpravě .htaccess souboru provedeme konfiguraci ve webovém prohlížeči podle obr. 15.


**Obrázek 15 – Základní konfigurace OwnCloud**

V některých případech cloud při instalaci oznámí chybu zápisu do úložiště a konfiguračních souborů. Chybu odstraníme povolením zápisu do kořenového adresáře příkazem *chown www-data:www-data -R /var/www/owncloud*. Instalaci jsme dokončili tlačítkem dokončit nastavení a otevře se nám prostřední OwnCloud.

### 5.2.5.3 Zabezpečení cloudu

Zabezpečení cloudu dále musíme provést ve virtuálním stroji. V první řadě musíme zabezpečit protokol HTTP. Do souboru */etc/apache2/ports.conf* na konec doplníme řádky

*<IfModule mod_headers.c>*

*Header always set Strict-Transport-Security "max-age=15768000; includeSubDomains; preload"*

*</IfModule>*

Pro vyšší zabezpečení je výhodnější zabezpečit komunikaci mezi klientem a serverem prostřednictvím protokolu HTTPS. Tento protokol využívá asymetrické kryptografie, které jsou založeny na bezpečnostních klíčích. Ty je nutné vygenerovat.

*sudo –i* *#přepnutí na uživatele root*

*cd /etc/ssl/private* *#otevření adresáře private*

*openssl genrsa -aes128 -out server.key 2048* *#generování klíčů*

*openssl rsa -in server.key -out server.key*

*openssl req -new -days 3650 -key server.key -out server.csr*

*openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 3650*
*chmod 400 server.* #změna práv na čtení vlastníkem*
*openssl req -new -x509 -days 365 -nodes -out /etc/apache2/ss l/owncloud.pem -keyout /etc/apache2/ssl/owncloud.key*

Aby webserver mohl používat certifikáty, v konfiguračním souboru */etc/apache2/sites-available/default-ssl.conf* musíme upravit následující řádky

*ServerAdmin mail@server*
*SSLCertificateFile /etc/apache2/ssl/owncloud.crt*
*SSLCertificateKeyFile /etc/apache2/ssl/owncloud.key*

Modifikovaný soubor uložíme, na serveru Apache spustíme SSL příkazem *a2ensite default-ssl* a restartujeme službu spravující webový server příkazem *service apache2 restart*.
Na závěr zabezpečení zbývá ještě nastavit vynucené přesměrování na zabezpečený protokol HTTPS. Editujeme soubor */var/www/owncloud/.htaccess* přídáním řádků

*RewriteCond %{HTTPS} off*
*RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R,L]*

do struktury tagů *<IfModule mod_rewrite.c> </IfModule>*.

#### 5.2.5.4 Ladění

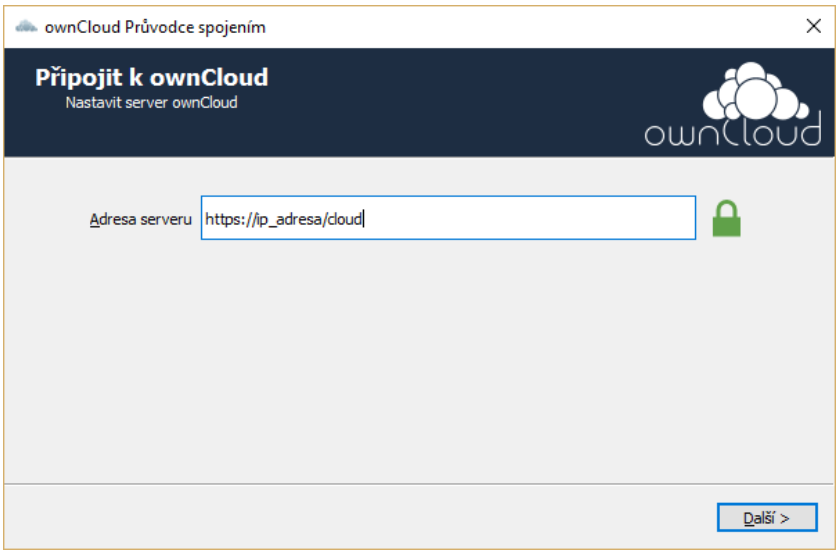
OwnCloud je dostupný z domény https://ip/owncloud. Pro uživatele je nepodstatné znát jméno úložiště, proto doménu změníme na https://ip/cloud úpravou aliasu v souboru */etc/apache2/sites-available/owncloud.conf* z /owncloud na /cloud. Dále upravíme řádek *'overwrite.cli.url' => 'http://ip/owncloud'* v souboru */var/www/owncloud/config/config.php* na hodnotu *'http://ip/cloud'*. Poslední částí je přepsat v souboru */var/www/owncloud/.htaccess* všechny slova owncloud na cloud.

Zbylá nastavení jako je navýšení maximální velikosti nahrávaného souboru na 16 GB lze provést v administraci hlavního uživatelského účtu.

#### 5.2.5.5 Instalace klienta

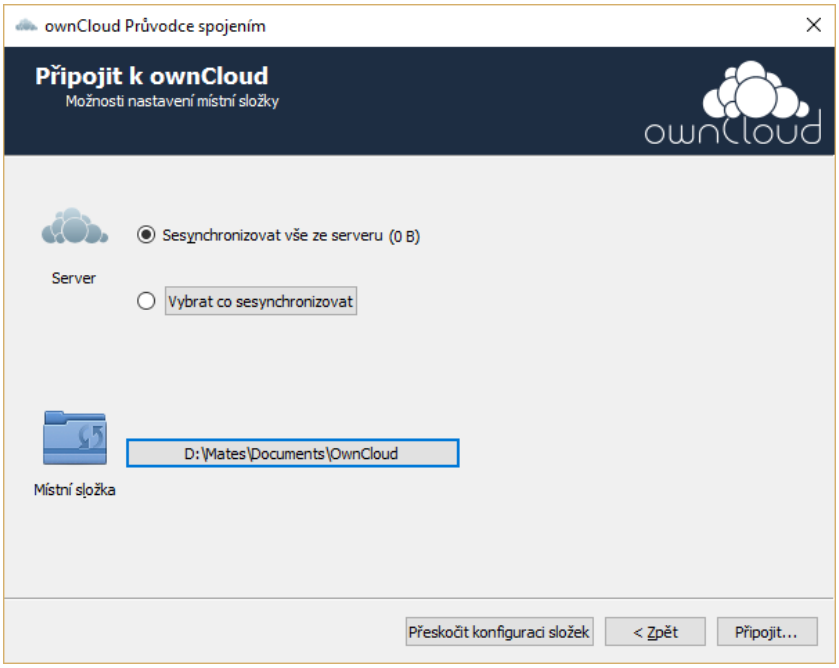
Ke cloudu se můžeme připojit po zadání adresy IP nebo doménového jména do pole adresa ve webovém prohlížeči. Druhou možností, jak pracovat s cloudem na lokálním počítači, je nainstalovat si klienta. Klienta je možné volně stáhnout na domácích stránkách Owncloudu.

Po stažení a instalaci klienta jej spustíme. Při prvním spuštění se klient zeptá na adresu, ke které se má připojit.




**Obrázek 16 - Přihlášení klienta OwnCloud**

Po stisknutí tlačítka další, se zobrazí okno s upozorněním, že certifikát webového serveru je podepsaný sám sebou a není tak důvěryhodný. Zaškrtneme volbu *Přesto tomuto certifikátu důvěřovat* a pokračujeme stiskem tlačítka OK. V následujícím kroku si server vyžádá přihlašovací údaje. Po ověření nastavíme synchronizačního klienta (viz obr. 16) a klikneme na tlačítko *připojit*. Po připojení již potvrdíme tlačítkem dokončit.




**Obrázek 17 – Nastavení synchronizačního klienta**

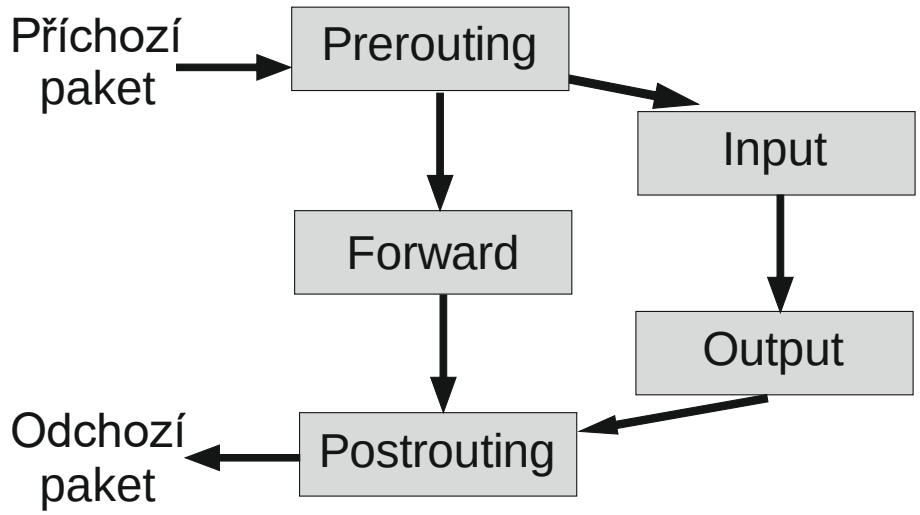
# 6 OMEZENÍ PŘÍSTUPU

Všechny servery slouží k poskytování služeb nebo dat ostatním autorizovaným uživatelům. Některé služby chceme distribuovat konkrétní skupině nebo naopak konkrétní skupině povolit z internetu přístup do intranetu.

## 6.1 Firewall

Firewall je hardware nebo software zamezující nebo omezující komunikaci mezi sítěmi typicky LAN a WAN. V Debianu je integrován paketový firewall zvaný iptables, který pakety filtruje a rozhoduje, jak s nimi bude naloženo. Hlavní nevýhodou iptables je skutečnost, že se po startu opět nastaví výchozí pravidla, která povolí všechnu komunikaci. To je pro použití serveru nevhodné, proto musíme doinstalovat balíček netfilter-persistent příkazem *apt-get install netfilter-persistent.* Iptables se skládá z pěti základních bloků, ve kterých pakety filtruje:

- Prerouting – změní cílovou adresu tzv. přesměrování portů (port forwarding)
- Forward – přesměruje pakety mimo operační systém na jiný PC
- Input – filtrování příchozích paketů pro operační systém
- Output – filtruje odchozí pakety pro operační systém
- Postrouting – zaměňuje zdrojovou adresu tzv. maškaráda (masquerade)




**Obrázek 18 – Schéma paketového firewallu**

Největší pozornosti musí být věnována při filtrování paketu do bloků input a forward. V těchto blocích musí být odhalen případný útok. Základní myšlenkou při filtrování paketů je všechno zakázat a povolovat potřebné.

### 6.1.1 Input

Nejdříve je nutné nastavit vstupní blok. První příkaz povoluje zpětnou smyčku tzv. localhost. Tato smyčka musí být povolena. Při pádu konektivity lze příkazem ping otestovat, zda dané rozhraní funguje.

*iptables -A INPUT -i lo -j ACCEPT*

Druhý příkaz dovoluje vytvořit nové spojení na základě již vytvořeného.

*iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT*

Všechny počítače v síti VUT jsou připojeny do sítě přes veřejné adresy IP, ale služby provozované na serveru nesmí být dostupné odkudkoliv z internetu. Níže uvedený příkaz zahazuje všechny pakety, které nejsou z rozsahu VUT.

*iptables -A INPUT -i eth0 -m iprange ! --src-range 147.229.1.1-147.229.255.255 -j DROP*

Po odfiltrování všech paketů, které nepatří do VUT rozsahu již povolíme icmp zprávy, DHCP a DNS server. V posledním kroku také povolíme přístup do systému přes SSH server.

*iptables -A INPUT -p icmp -j ACCEPT*
*iptables -A INPUT -s 192.168.122.0/24 -d 192.168.122.1 -j ACCEPT*
*iptables -A INPUT -i virbr10 -p tcp -m tcp --dport 53 -j ACCEPT*
*ptables -A INPUT -i virbr10 -p udp -m udp --dport 53 -j ACCEPT*
*iptables -A INPUT -i virbr10 -p tcp -m tcp --dport 67 -j ACCEPT*
*iptables -A INPUT -i virbr10 -p udp -m udp --dport 67 -j ACCEPT*
*iptables -A INPUT -i eth0 -p tcp -m tcp --dport 22 -j ACCEPT*

Paketový firewall iptables prochází příkazy postupně od shora dolů, dokud nenalezne pravidlo pro příchozí paket. Pokud žádné pravidlo nenalezne, provede s ním akci, která je stanovena výchozím pravidlem. V Debianu jsou ve výchozím nastavení pakety přijímány, proto je důležité je zakázat.

*iptables -P INPUT DROP*

### 6.1.2 Forward

V bloku forward dochází k přesměrování paketů na jinou IP adresu. Směrování paketů je nutné povolit v souboru */etc/sysctl*.conf nastavením *net.ipv4.ip_forward=1*. Změny se projeví po restartu počítače.

V chainu forward rozhodneme, co s příchozími pakety provedeme (zahodíme, přepošleme). Pokud není nalezena shoda se žádným z pravidel, je aplikováno výchozí pravidlo, které zahazuje všechny pakety.

*iptables -P FORWARD DROP*

První pravidlo opět povoluje navázání nového spojení na základě již vytvořeného.

*iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT*

Tento server také poskytuje služby prostřednictvím sítě VPN. Před zahozením všech paketů, které nejsou z VUT rozsahu musíme přesměrovat pakety pro VPN server.

*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.130 -p tcp --dport 1194 -j ACCEPT*
*iptables -A FORWARD -i eth0 -m iprange ! --src-range 147.229.1.1-147.229.255.255 -j DROP*

Na serveru jsou umístěny ve virtuálním bridgy virtuální stroje. Mezi těmito stroji musíme povolit komunikaci.

*iptables -A FORWARD -i virbr10 -s 192.168.122.0/24 -j ACCEPT*
*iptables -A FORWARD -i virbr10 -o virbr10 -j ACCEPT*

Na serveru také provozujeme další služby. Tyto služby komunikují přes některé porty, které je pro správnou funkci povolit. Jedná se o služby samba (porty 137 – 139, 445) webový server (porty 80,443).

*iptables -A FORWARD -o virbr10 -d 192.168.122.120 -p tcp --dport 443 -j ACCEPT*
*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.120 -p tcp --dport 80 -j ACCEPT*
*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.100 -p tcp --dport 445 -j ACCEPT*
*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.100 -p udp --dport 137 -j ACCEPT*
*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.100 -p tcp --dport 137 -j ACCEPT*
*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.100 -p udp --dport 138 -j ACCEPT*
*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.100 -p tcp --dport 138 -j ACCEPT*

*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.100 -p tcp --dport 139 -j ACCEPT*

Na virtuálním stroji s nainstalovanou databází je nainstalován phpMyAdmin pro její správu. Tento webový server je nakonfigurován pro komunikaci přes port 81.

*iptables -A FORWARD -i eth0 -o virbr10 -d 192.168.122.110 -p tcp --dport 81 -j ACCEPT*

Na závěr všech pravidel je ještě nutné povolit příkaz ping.
*iptables -A FORWARD -i eth0 -o virbr10 -p icmp --icmp-type echo-reply -j ACCEPT*

### 6.1.3 Postrouting

Server v této práci má k dispozici jednu veřejnou IP adresu a celá síť používá překlad adres tzv. maškarádu, která zaměňuje IP adresu zdroje za IP veřejnou IP adresu. Přijaté odpovědi opět posílá klientovi, který požadavek poslal. Maškarádu nastavíme

*sudo iptables -t nat -A POSTROUTING -o eth0 -s 192.168.122.0/24 -j MASQUERADE*

### 6.1.4 Prerouting

Prerouting je podobný postroutingu. Na základě portu rozhodne jaká služba je požadována a podle toho změní cílovou adresu. Z preroutingu je paket přesměrován do jednoho ze dvou vstupních bloků – input nebo forward.

*iptables -t nat -A PREROUTING -p tcp --dport 1194 -j DNAT --to-destination 192.168.122.130:1194*
*iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.122.120:80*
*iptables -t nat -A PREROUTING -p tcp --dport 443 -j DNAT --to-destination 192.168.122.120:443*
*iptables -t nat -A PREROUTING -p udp --dport 137 -j DNAT --to-destination 192.168.122.100:137*
*iptables -t nat -A PREROUTING -p tcp --dport 137 -j DNAT --to-destination 192.168.122.100:137*
*iptables -t nat -A PREROUTING -p udp --dport 138 -j DNAT --to-destination 192.168.122.100:138*
*iptables -t nat -A PREROUTING -p tcp --dport 138 -j DNAT --to-destination 192.168.122.100:138*

*iptables -t nat -A PREROUTING -p udp --dport 139 -j DNAT --to-destination 192.168.122.100:139*
*iptables -t nat -A PREROUTING -p tcp --dport 139 -j DNAT --to-destination 192.168.122.100:139*
*iptables -t nat -A PREROUTING -p tcp --dport 445 -j DNAT --to-destination 192.168.122.100:445*
*iptables -t nat -A PREROUTING -p tcp --dport 81 -j DNAT --to-destination 192.168.122.110:81*

### 6.1.5 Logování paketů

Sledování paketů je výhodné při ladění firewallu nebo ke sledování komunikace. Na serveru jsou zaznamenávány všechny pakety, které došli k výchozímu pravidlu.

*iptables -A INPUT -j LOG --log-prefix '[IPTABLES INPUT]: '*
*iptables -A FORWARD -j LOG --log-prefix '[IPTABLES FORWARD ]: '*

Zachycené pakety můžeme vypsat ve výpisu jádra příkazem *dmesg*, případně příkazem *dmesg –C* smazat výpis logu jádra.

### 6.1.6 Správa iptables

Po vytvoření pravidel je nezbytné konfiguraci uložit. Provedeme tak zadáním příkazu *service netfilter-persistent save.* Místo příkazu save můžeme použít např. *start, stop, restart*.

Iptables lze i exportovat příkazem *iptables-save > nazev_a_umisteni_souboru* nebo obnovit ze souboru *iptables-restore < nazev_a_umisteni_souboru.*

## 6.2 OpenVPN

OpenVPN je software pro vytvoření VPN sítě zabezpečené prostřednictvím RSA klíčů. Je založen na dvou balících openvpn a easy-rsa, které v prostření Linuxu nainstalujeme příkazem *apt-get install openvpn easy-rsa*. Klíče se generují pomocí utility easy-rsa. V kořenovém adresáři */etc/openvpn*/ vytvoříme složku *mkdir easy-rsa.* Do této složky nakopírujeme celou utilitu *cp -r /usr/share/easy-rsa/* /etc/openvpn/easy-rsa.*

### 6.2.1 Generování certifikátů

Ověření uživatele probíhá na základě RSA klíčů, které si musíme vytvořit. Pokud generujeme poprvé, upravíme si v souboru /etc/openvpn/easy-rsa/vars proměnné v řádcích za hodnoty v komentáři

```
export KEY_COUNTRY="US"                    #CZ
export KEY_PROVINCE="CA"                   #CZ
export KEY_CITY="SanFrancisco"             #Brno
export KEY_ORG="Fort-Funston"              #FEEC VUTBR
export KEY_EMAIL="me@myhost.mydomain"      #admin@utko.feec.vutbr
export KEY_OU="MyOrganizationalUnit"       #UTKO
```

Po úpravě souboru a před generováním souborů je výhodné se přepnout na uživatele root příkazem *sudo –i*. Přepnutí nám ušetří mnoho problémů s generováním certifikátů. Nyní jako uživatel root se přepneme do výše uvedeného adresáře příkazem *cd /etc/openvpn/easy-rsa/* a nahrajeme soubor s proměnnými použitím příkazu *source vars*. Pokud generujeme certifikáty poprvé je nutné spustit příkaz *./clean-all*, který vytvoří domovskou složku pro vytvořené certifikáty /keys/. Příkaz *./clean-all* spustíme pouze nyní, jelikož smaže i všechny předchozí vygenerované klíče.

Prvním vygenerovaným klíčem bude klíč certifikační autority. Tento klíč je nutný pro generování certifikátu a vygenerujeme ho pouze poprvé příkazem *./build-ca*. Ze souboru vars se načtou námi upravené proměnné, které stačí potvrdit stiskem tlačítka ENTER. Hodnoty lze i změnit pouhým zapsáním hodnoty nové a potvrzením klávesou ENTER. Na konci generování nás program dvakrát vyzve k potvrzení zápisu a klíč se vytvoří. Stejným postupem vytvoříme klíče pro Diffie-Hellmanovu výměnu klíčů, server i klienty.

```
./build-key-server server          # vygenerování klíče pro server
./build-dh                         #Diffie-Hellmanův klíč
./build-key nazev_klienta          #vygenerování klíče pro klienta.
```

Na závěr tvorby klíčů vytvoříme statický klíč, který chrání před útoky typy Man in the Middle.

```
openvpn --genkey --secret ta.key
```

Nově vytvořené klíče nalezneme ve složce */etc/openvpn/easy-rsa/keys*, do kterého se přepneme příkazem *cd*. Pro správnou funkci VPN serveru potřebujeme klíče ca.crt,

dh2048.pem, server.crt, server.key a ta.key. Ty nakopírujeme do kořenového adresáře openvpn *cp ca.crt dh2048.pem server.crt server.key tak.key /etc/opennvpn/* a příkazem *exit* se odhlásíme z uživatele root.

### 6.2.2 Konfigurace serveru

Pro konfiguraci serveru se přepneme do adresáře *cd /etc/openvpn/*. Do tohoto umístění kopírujeme konfigurační soubor

*cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz server.conf.gz*

a rozbalíme jej *gunzip server.conf.gz*. Rozbalený soubor *server.conf* otevřeme v editoru a upravíme

| | |
|---|---|
| *;proto tcp* | *#odkomentujeme (smažeme ;)* |
| *proto udp* | *#zakomentujeme (přidáme ;)* |
| *dh dh1024.pem* | *#přepíšeme na dh2048.pem* |
| *server 10.8.0.0 255.255.255.0* | *# změníme na 192.168.222.0 255.255.255.0* |
| *ifconfig-pool-persist ipp.txt* | *#zakomentujeme* |
| *;push "redirect-gateway def1 bypass-dhcp"* | *#odkomentováním můžeme povolit přesměrování #všeho provozu přes server* |
| *;push "dhcp-option DNS 208.67.222.222"* | *#odkomentujeme a změníme IP192.168.222.1* |
| *tls-auth static.key 0* | *#odkomentujeme a přepíšeme static.key na ta.key* |
| *;cipher AES-128-CBC* | *#odkomentujeme a přepíšeme na AES-256-CBC* |
| *;max-clients 100* | *#odkomentujeme a změníme max. počet na 2* |

Na závěr přidáme řádek pro ověřování zablokovaných certifikátů.

*crl-verify /etc/openvpn/easy-rsa/keys/crl.pem*

Tento soubor musíme vytvořit příkazem

*./revoke-full nazev_certifikatu.*

Pro vytvoření výše uvedeného souboru je nutné vygenerovat jeden certifikát navíc a zamítnout ho. Zamítnuté certifikáty nelze nijak obnovit a je nutné vytvořit nové.

### 6.2.3 Konfigurace klienta Windows

Ze stránek projektu OpenVPN stáhneme instalační soubor a nainstalujeme. Ve složce *C:\Program Files\OpenVPN\config* vytvoříme soubor pojmenovaný *nazev_uzivatele.ovpn* a otevřeme ho v poznámkovém bloku a vložíme do něj řádky:

```
Client                          #řekneme, že tato instalace je klient
dev tun                         #nastavíme práci openVPN na síťovou vrstvu
proto tcp                       #protokol pro komunikaci mezi server a klientem
remote 192.168.0.111 1194       # DNS nebo IP VPN serveru včetně portu
resolv-retry infinite
nobind                          #vypnutí naslouchání klientů na určitém portu
persist-key
persist-tun
ca ca.crt                       #nastavení názvu klíče certifikační autority
cert garfield.crt               #jméno certifikátu uživatele
key garfield.key                #jméno klíče uživatele
remote-cert-tls server          #vzdálený server je certifikační autoritou
tls-auth ta.key 1               #statický klíč proti útokům man in the middle
cipher AES-256-CBC              #šifrování komunikace
comp-lzo                        #nastavení komprese spojení
verb 3                          #nastavení logování
```

Do vytvořeného profilu musíme doplnit klíče, které máme uložené na serveru. Tyto klíče musíme přenést zabezpečeně např. pomocí protokolu SCP.

```
mkdir /etc/openvpn/easy-rsa/garfield/keys   #vytvoříme složku
cd /etc/openvpn/easy-rsa/keys               #přepneme se do adresáře keys
cp ta.key garfield.crt garfield.key ca.crt /garfield    #nakopírujeme klíče do složky
```

A následně přeneseme klíče do domovského hlavního operačního systému. Odtud jej distribuujeme opět bezpečným způsobem ke klientovi.

```
scp vpn@192.168.122.130:/etc/openvpn/easy-rsa/keys/garfield.* ~/
```

VPN server je dostupný za NATem, proto se nelze k němu přes SSH protokol připojit přímo a certifikáty musíme přenášet komplikovaněji. Druhým způsobem jak lze přenést certifikáty je

připojit se přes SSH ke hlavnímu operačnímu systému a vytvořit SSH tunel do virtuálního systému. Po vytvoření tunelu je možné se přihlásit k SSH serveru klienta.

#### 6.2.3.1 Přihlášení klienta

Ve Windows musíme klienta spouštět správy administrátora. Po spuštění se ve stavové liště objeví šedá ikona počítače.

**Obrázek 19 – OpenVPN ve stavové liště Windows**

Po dvojitém kliknutí na ikonu klient otevře dialogové okno a začne se přihlašovat. Po jeho zmizení a zezelenání ikony je VPN spojení aktivováno. Všechny služby jsou nyní dostupné na IP adrese 192.168.222.1.

### 6.2.4 Firewall

Všechna komunikace, VPN sítě je přesměrována na server VPN portem 1194. OpenVPN server si vytvoří vlastní rozhraní tun0, přes které komunikuje. Tohle rozhraní je dostupné až po přihlášení k VPN serveru, ale i tak může být dostupné napadeno útočníkem. Také přes rozhraní tun0 musíme nastavit přesměrování kvůli dostupnosti sítě ve VPN síti. Proto použijeme iptables s následujícími pravidly.

*iptables -P INPUT DROP*
*iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT*
*iptables -A INPUT -p udp -m udp --dport 53 -j ACCEPT*
*iptables -A INPUT -i lo -j ACCEPT*
*iptables -A INPUT -i eth0 -j ACCEPT*
*iptables -A INPUT -p icmp -j ACCEPT*
*iptables -A INPUT -j LOG --log-prefix '[IPTABLES INPUT]: '*
*iptables -P FORWARD DROP*
*iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT*

*iptables -A FORWARD -i tun0 -o eth0 -d 192.168.122.120 -p tcp --dport 80 -j ACCEPT*
*iptables -A FORWARD -i tun0 -o eth0 -d 192.168.122.120 -p tcp --dport 443 -j ACCEPT*
*iptables -A FORWARD -i tun0 -o eth0 -d 192.168.122.100 -p tcp --dport 445 -j ACCEPT*
*iptables -A FORWARD -i tun0 -o eth0 -d 192.168.122.100 -p udp --dport 137 -j ACCEPT*
*iptables -A FORWARD -i tun0 -o eth0 -d 192.168.122.100 -p tcp --dport 137 -j ACCEPT*
*iptables -A FORWARD -i tun0 -o eth0 -d 192.168.122.100 -p udp --dport 138 -j ACCEPT*
*iptables -A FORWARD -i tun0 -o eth0 -d 192.168.122.100 -p tcp --dport 138 -j ACCEPT*
*iptables -A FORWARD -i tun0 -o eth0 -d 192.168.122.100 -p tcp --dport 139 -j ACCEPT*
*iptables -A FORWARD -j LOG --log-prefix '[IPTABLES FORWARD ]: '*
*iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.122.120:80*
*iptables -t nat -A PREROUTING -p tcp --dport 443 -j DNAT --to-destination 192.168.122.120:443*
*iptables -t nat -A PREROUTING -p udp --dport 137 -j DNAT --to-destination 192.168.122.100:137*
*iptables -t nat -A PREROUTING -p tcp --dport 137 -j DNAT --to-destination 192.168.122.100:137*
*iptables -t nat -A PREROUTING -p udp --dport 138 -j DNAT --to-destination 192.168.122.100:138*
*iptables -t nat -A PREROUTING -p tcp --dport 138 -j DNAT --to-destination 192.168.122.100:138*
*iptables -t nat -A PREROUTING -p udp --dport 139 -j DNAT --to-destination 192.168.122.100:139*
*iptables -t nat -A PREROUTING -p tcp --dport 139 -j DNAT --to-destination 192.168.122.100:139*
*iptables -t nat -A PREROUTING -p tcp --dport 445 -j DNAT --to-destination 192.168.122.100:445*

Aby byla pravidla uchována i po restartování stroje s VPN serverem, musí být trvalá.

*apt-get install netfilter-persistent* *#instalace balíků pro vytvoření trvalých pravidel*
*service netfilter-persistent save* *#uložení aktuálních pravidel*

Nyní máme povolenou komunikaci z virtuální sítě k virtualizovaným strojům, ale nedostupnou cestu zpět. Virtuální systémy a ani hostitelský systém nemá uložený záznam o

tom, kam má směrovat pakety pro síť 192.168.222.0/24. Proto do hlavního operačního systému nastavíme statickou cestu, která se vždy nahraje po startu operačního systému pomocí služby cron. Crona spustíme příkazem *sudo crontab –e* a na konec souboru přidáme řádek

*@reboot /sbin/route add -net 192.168.222.0 netmask 255.255.255.0 gw 192.168.122.130*

# 7 ZÁLOHOVÁNÍ – NÁVRH ŘEŠENÍ

Jak již bylo v práci zmíněno, server disponuje třemi pozicemi pro 3,5“ pevné disky. První dva sloty jsou obsazeny dvěma shodnými pevnými disky, z nichž má každý kapacitu 320 GB. Tyto dva disky jsou zálohovány SataRAIDem, který disky zrcadlí v reálném čase (typ RAID 1). Zrcadlena jsou všechna data důležitá pro chod systému, jejichž výpadek by ohrozil stabilitu systému, přesněji hlavní operační systém a obrazy pevných disků všech virtuálních strojů. Na serveru jsou i virtuální stroje Samba a OwnCloud, pracující s uživatelskými daty. Pro tyto data jsou vytvořené dva virtuální disky o kapacitě 800 GB a jsou uložené na třetím pevném disku. V případě havárie tohoto disku jsou všechna uživatelská data nenávratně ztracena, ale není tak ovlivněn chod hlavního operačního systému. Ztráta uživatelských dat je velmi nežádoucím faktorem pro každé datové úložiště, proto musíme i poslední disk zabezpečit proti ztrátě dat jedním z následujících návrhů.

## 7.1 Úprava serveru

První možností zálohování je upravit stávající server. Základní deska serveru Intel S5000VCL umožňuje zapojit přes rozhraní SATA-II až šest pevných disků, zatímco tento server má vyvedeny hotswap rámečky pro disky tři. Při úpravě serveru a vyvedení napájení z napájecího zdroje a SATA kabelů mimo server lze tak připojit minimálně další disk. Hlavní výhodou tohoto řešení je vytvoření dalšího pole typu RAID 1 pomocí SataRAIDU a zálohovat tak data v reálném čase. Finanční náklady na úpravu zahrnují cenu nového disku a potřebou kabeláž. Toto řešení je efektivní, ale není příliš bezpečné z hlediska úrazu elektrickým proudem a proti fyzickým poškozením disku, protože kabeláž i disk musí být umístěny mimo kovové tělo serveru. Také se mírně zvýší zatížení procesoru, který bude muset spravovat dvě pole RAID.

## 7.2 Zálohování přes USB

Druhou možností jak zálohovat data je použití externího pevného disku připojeného přes sběrnici USB. Hlavní výhodou tohoto řešení je jednoduchost, protože stačí pouze připojit pevný disk a nastavit zálohování, které nelze provádět v reálném čase. Do hlavního operačního systému by musel být umístěn skript, který by byl spouštěn v době nejmenšího využití serveru, například v noci. Spouštění skriptu by bylo obstaráváno službou Cron. Další hlavní nevýhodou je pomalá rychlost sběrnice USB 2.0. Při kopírování tak velkého objemu dat by záloha probíhala velmi dlouho. Při zálohování přes USB by bylo výhodné použít disk o

celkové minimální kapacitě 3 TB, aby při zálohování bylo možné nejdříve nahrát aktuální obraz disku a poté smazat starý. Tímto způsobem lze zamezit chybě při nenadálé havárii systému například vlivem výpadku proudu.

## 7.3 Zálohování přes rozhraní ethernet

Přidělený server obsahuje dvě gigabitové síťové karty. První slouží pro hlavní konektivitu do sítě internet. Druhá síťová karta je nevyužita. Při zálohování přes gigabitovou síť nejsme limitováni rychlostí přenosu, ale rychlostí zápisu pevných disků, která je výrazně nižší.

### 7.3.1 NAS

První možností jak přes ethernet zálohovat je použít zařízení NAS. Zařízení NAS lze pohodlně spravovat pomocí webového rozhraní. Dostupný hardware umožňuje vytvořit a spravovat pole RAID 1 a RAID 0 nebo jejich kombinaci RAID 10. V případě použití této koncepce je možné interní disk ze serveru vyjmout a umístit jej přímo do zařízení, kde by se data zrcadlila na druhý pevný disk. Zařízení by se propojila s vnitřní sítí serveru a pomocí protokolu samba nebo NFS, které jsou v NAS již implementovány výrobcem. Zálohování dat by mohlo probíhat v reálném čase jako při použití SataRAIDu.

### 7.3.2 Datové úložiště

Datové úložiště je mnohonásobně levnější varianta NAS. Jde o jeden 3,5“ pevný disk zabudovaný v rámečku. Datové úložiště má stejné vlastnosti jako NAS, ale neumožňuje vytváření polí RAID. Disponuje správou přes webové rozhraní a implementovanými protokoly NFS a samba. Zařízení by opět bylo připojeno do interní sítě serveru a připojeno do hlavního operačního systému jedním z uvedených protokolů. U této varianty by v serveru zůstaly využity všechny sloty pro pevné disky. Zálohování dat by muselo probíhat stejně jako při použití pevného disku v době nejmenšího využití serveru, ale čas zálohování by byl výrazně kratší.

## 7.4 Shrnutí

Všechny základní parametry výše uvedených možností shrneme v následující tabulce.

**Tabulka 2 – Shrnutí možností zálohování**

| Varianta | Úprava server | Externí disk | NAS | Datové úložiště |
|---|---|---|---|---|
| **Díly** | Kabeláž, HDD 2 TB | Externí HDD, 3 TB | Zařízení NAS, HDD 2 TB | Zařízení datového úložiště |
| **Zálohování** | v reálném čase | v noci | v reálném čase | v noci |
| **Rozhraní** | Sata II | USB 2.0 | Ethernet | Ethernet |
| **Přibližná cena** | 2 800 Kč | 3 300 Kč | 5 200 Kč | 4 000 Kč |

Nejefektivnějším a nejbezpečnějším řešení by bylo použití zařízení NAS, které používá dva pevné disky. Jeden ze dvou pevných disků už máme umístěný v serveru a druhý by bylo nutné přikoupit. Na serveru by nevznikla žádná přídavná zátěž a data by byla zálohována v reálném čase. Úpravou serveru také dosáhneme zálohování v reálném čase, data budou zabezpečená, ale řešení bude v serverové místnosti vypadat nevzhledně a při nesprávné manipulaci hrozí i úraz elektrickým proudem nebo poškození serveru při zkratování vodičů a tím i zkratováním zdroje napájení. Možností jak této variantě předejít je vytvoření vlastního nezávislého zdroje napájení, ale tím se i zvýší cena a tím i náročnost této práce. U posledních dvou možností lépe vyhovuje zálohování přes ethernet kabel. Data jsou zabezpečená v případě krádeže celého disku a cenový rozdíl už není tak výrazný.

# ZÁVĚR

Cílem této bakalářské práce bylo navrhnout řešení pro datové úložiště a implementovat ho na fyzický server. Největší důraz byl kladen na výběr virtualizační platformy. V první fázi výběru jsem se soustředil na XenServer od firmy Cintrix. Tento hypervizor sice minimálně zatěžuje hardware počítače a vlastní grafické rozhraní, ale po hlubším zkoumání jsem narazil na problémy. Grafické prostředí je sice jednoduché, ale složitější konfigurace často musí být prováděny pomocí příkazů. Dalším problémem u XenServeru se mi jevila rozšířená podpora. Dalším důvodem proč jsem nezvolil XenSever ale KVM, je omezenost. Některé nástroje nejsou dostupné, a pokud bychom je v budoucnu potřebovali, musí se zakoupit vyšší licence. U KVM nic takového neexistuje a to co v systému chybí, lze zdarma a jednoduše doinstalovat. Další krásou KVM je příkazové prostředí Linuxu, na kterém běží. Lze tak snadno pomocí integrovaných funkcí nasadit NAT nebo firewall.

Distribuci Debian jsem zvolil kvůli její otevřenosti a snadné modifikovatelnosti. Ve srovnání s distribucí Ubuntu mi pro tuto operaci přijde více profesionální. I když obě distribuce mají mnoho prvků společných, Debian je více odladěný a Ubuntu je jen jeho nástavba.

Největší komplikace nastaly při instalaci hlavního operačního systém s podporou RAID. Debian při instalaci spouštěl démona pro instalaci softwarového RAIDu a blokoval tím SataRAID. Dalším problémem byla instalace virtuální sítě pro virtualizované stroje. Při instalaci byly použity výchozí knihovny libvirt, které umožňují snadnou instalaci virtuální sítě. Virtuální síť vytvořená tímto způsobem byla opravdu funkční, ale při nastavování firewallu vkládala výchozí pravidla před pravidla nastavená uživatelem. Takto se po každém startu serveru zablokovala všechna konektivita ke službám poskytovaných virtuálními stoji a bylo nutné firewall manuálně restartovat. Tato virtuální síť se chovala problematicky i pro proxy a DNS server, které po ní nesprávně komunikovali. Kvůli těmto uvedeným problémům síť, vytvořená pomocí knihoven libvirt, byla deaktivována a nastavena na úrovni hlavního operačního systému balíkem bridge-utils. V rámci optimalizace datového úložiště bylo provedeno testování serveru. Nyní je zařízení připojeno do sítě VUT, kde pracuje dle očekávání.

# LITERATURA


[1] RUEST, Danielle a Nelson RUEST. *Virtualizace: podrobný průvodce*. Vyd. 1. Brno: Computer Press, 2010, 408 : il. ISBN 978-80-251-2676-9.

[2] NOVOTNÝ, V. *Architektura sítí.* První vydání. Brno: FEKT VUT v Brně, 2012.

[3] Virtualizace v kostce: K čemu, jak a proč? *Systemonline*[online]. 2010 [cit. 2015-12-13]. Dostupné z: *http://www.systemonline.cz/clanky/virtualizace-v-kostce.htm*

[4] David Pašek. TŘI Z NEJSILNĚJŠÍCH – SROVNÁNÍ SERVEROVÉ VIRTUALIZACE VMWARE VS. CITRIX VS. MICROSOFT. *vmwarenews.* [online]. 2008 [cit. 2015-12-13]. Dostupné z: *http://www.vmwarenews.cz/vmw/vmwnews.nsf/0/53bb1b111be5a818c12575e5005f83a6*

[5] Eric Siebert. Choosing vSphere vs. Hyper-V vs. XenServer. TechTarget. [online]. 2006-2015 [cit. 2015-12-13]. Dostupné z: *http://searchitchannel.techtarget.com/feature/Choosing-vSphere-vs-Hyper-V-vs-XenServer*

[6] *Virtualizace v Linuxu* [online]. Praha, 2008 [cit. 2015-12-13]. Dostupné z: https://www.vse.cz/vskp/id/15116

[7] Virtualizace na úrovni jádra operačního systému. *abclinuxu.* [online]. 30. 7. 2007 [cit. 2015-12-13]. Dostupné z:*http://www.abclinuxu.cz/clanky/system/virtualizace-na-urovni-jadra-operacniho-systemu*

[8] Paul Venezia. Test: Souboj hlavních virtualizačních platforem. *Computerworld*. [online]. 09. 08. 2011 [cit. 2015-12-13]. Dostupné z: *http://computerworld.cz/testy/test-souboj-hlavnich-virtualizacnich-platforem-43672*

[9] Ondřej Suchy. Podniková virtualizace na Linuxu?. *Systemonline.* [online]. 2010 [cit. 2015-12-13]. Dostupné z:*http://www.systemonline.cz/virtualizace/podnikova-virtualizace-na-linuxu.htm*

[10] OpenVZ and XEN virtualization technology insight and comparison. *sitewalley.com.* [online]. 8. 12. 2009 [cit. 2015-12-13]. Dostupné z:*https://sitevalley.com/blog/xen-and-openvz-technology-insight-and-comparison/*

[11] Richard Kaplička. Instalace Samba serveru na Linux distro Debian. *Richard Kapička – Blog.* [online]. 9. 6. 2011 [cit. 2015-12-13]. Dostupné z:*https://rikap.cz/20110609-instalace-samba-serveru-na-linux-distro-debian*

[12] Mohammad Dosoukey. How to Deploy Multiple Virtual Machines using Network Install (HTTP, FTP and NFS) under KVM Environment – Part 2. *Tecmint.*[online]. 23.1. 2015 [cit. 2015-12-13]. Dostupné z: *http://www.tecmint.com/multiple-virtual-machine-installation-using-network-install-kvm/*

[13] Základy virtualizace – KVM. *M-Linux.* [online]. 23.1 2015 [cit. 2015-12-13]. Dostupné z: *http://m-linux.cz/2015/01/zaklady-virtualizace-kvm/*

[14] Virtualizace XEN. *Osobní stránky informatika.* [online]. [cit. 2015-12-13]. Dostupné z: *http://www.hlobil.net/24762-virtualizace-xen*

[15] David Kutálek. Samba. [online]. [2002] [cit. 2015-12-13]. Dostupné z: *http://www.fi.muni.cz/~kas/p090/referaty/2002-podzim/skupina10/samba.html*

[16] Leoš Houser. Virtualizace KVM – řešení budoucnosti. *Linux, bezpečnost, virtualizace, cloud computing.* [online]. 28. 10. 2015 [cit. 2015-12-13]. Dostupné z: *https://it-blog.cz/virtualizace-kvm/*

[17] Chapter 34. Advanced Configuration Techniques. *samba.org.* [online]. 30. 6. 2005 [cit. 2015-12-13]. Dostupné z:*https://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/cfgsmarts.html*

[18] Tadeáš Pařík. Virtualizace. *Ubuntu.cz.* [online]. 11. 9. 2013 [cit. 2015-12-13]. Dostupné z: *https://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/cfgsmarts.html*

[19] Petr Bouška. ownCloud 5.0 – internetové úložiště ve firmě. *samuraj-cz.com.* [online]. 14. 3. 2013 [cit. 2015-12-13]. Dostupné z: *http://www.samuraj-cz.com/clanek/owncloud-5-0-internetove-uloziste-ve-firme/*

[20] ownCloud. *root.cz.* [online]. 1998 – 2015 [cit. 2015-12-13]. Dostupné z: *http://www.root.cz/n/owncloud/*

[21] Adam Štrauch. OwnCloud: vlastní cloudové úložiště. *root.cz.* [online]. 1. 6. 2012 [cit. 2015-12-13]. Dostupné z: *http://www.root.cz/clanky/owncloud-vlastni-cloudove-uloziste/*

[22] Adam Štrauch. ownCloud: instalace vlastního cloudu v praxi. *root.cz.* [online]. 8. 6. 2012 [cit. 2015-12-13]. Dostupné z:*http://www.root.cz/clanky/owncloud-instalace-vlastniho-cloudu-v-praxi/*

[23] Ubuntu. *root.cz.* [online]. 1998 – 2015 [cit. 2015-12-13]. Dostupné z: *http://www.root.cz/n/ubuntu/*

[24] *Virtuální prostředí pomocí softwaru Xen* [online]. Brno, 2009 [cit. 2015-12-13]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=18407

[25] Jirka Bourek. Xen vs. KVM: Výkon CPU. *abclinuxu.* [online]. 6. 5. 2010 [cit. 2015-12-13]. Dostupné z: *Jirka Bourek. Xen vs. KVM: Výkon CPU. abclinuxu. [online]. 6. 5. 2010 [cit. 2015-12-13]. Dostupné z: http://www.abclinuxu.cz/clanky/xen-vs.-kvm-souboj-v-extremnich-podminkach?page=1#shrnuti*

[26] Chapter 1. Learning the Samba. *Samba.org.* [online]. 2002 [cit. 2015-12-13]. Dostupné z: *https://www.samba.org/samba/docs/using_samba/ch01.html*

[27] Ondřej Mizoun. LINUX + SAMBA – příklad z firemní praxe. *samba-sweb.cz.* [online]. 17. 12. 2004 [cit. 2015-12-13]. Dostupné z:*http://samba.sweb.cz/samba/*

[28] Ondřej Jakubčík. Jak se tančí Samba?. *Linuxexpres.* [online]. 16. 5. 2007 [cit. 2015-12-13]. Dostupné z: *http://samba.sweb.cz/samba/*

[29] Cloud Computing pro každého: Jasně a stručně. *BusinessIT.* [online]. srpen 2011 [cit. 2015-12-13]. Dostupné z: *http://www.businessit.cz/cz/cloud-computing-vysvetleni-jasne-a-strucne.php*

[30] Douglas K Barry. Software as a Service (SaaS). *Service Architecture.* [online]. 2000-2015 [cit. 2015-12-13]. Dostupné z: *http://www.service-architecture.com/articles/cloud-computing/software_as_a_service_saas.html*

[31] Rita Pužmanová. Jak pochopit a uchopit cloud computing. *Systemonline.* [online]. 10/2008 [cit. 2015-12-13]. Dostupné z:*http://www.systemonline.cz/sprava-it/jak-pochopit-a-uchopit-cloud-computing.htm*

[32] *Porovnání virtualizačních technologií* [online]. Jihlava, 2013 [cit. 2015-12-13]. Dostupné z: https://is.vspj.cz/bp/get-bp/student/18644/thema/3676

[33] NFS File Server. *Debian.* [online]. [cit. 2015-12-13]. Dostupné z: *https://debian-handbook.info/browse/stable/sect.nfs-file-server.html*

[34] FTP File Server. *Debian.* [online]. [cit. 2015-12-13]. Dostupné z: *https://debian-handbook.info/browse/stable/sect.ftp-file-server.html*

[35] Virtualizace desktopů. *Comparex.* [online]. [cit. 2015-12-13]. Dostupné z: *http://www.comparex-group.com/web/cz/cs/produkty-sluzby/datova-centra/virtualizace/virtualizace-desktopu/virtualizace-desktopu.htm*

[36] Red Hat Enterprise Linux. *Fedora.* [online]. 30. 12. 2014 [cit. 2015-12-13]. Dostupné z: *https://fedoraproject.org/wiki/Red_Hat_Enterprise_Linux*

[37] David Čápka. 1. díl – Úvod do operačního systému Linux. *itnetwork.* [online]. 5. 7. 2014 [cit. 2015-12-13]. Dostupné z:*http://www.itnetwork.cz/linux/zaklady/uvod-do-operacniho-systemu-linux/*

[38] Petr Kulhavý. Stavíme mailový server: instalace. *root.cz.* [online]. 20. 10. 2003 [cit. 2015-12-13]. Dostupné z: *http://www.root.cz/clanky/stavime-mailovy-server-instalace/#ic=serial-box&icc=text-title*

[39] ToMášMarný. Vydání. *Ubuntu.cz.* [online]. 23. 10. 2015 [cit. 2015-12-13]. Dostupné z: *http://wiki.ubuntu.cz/vyd%C3%A1n%C3%AD*

[40] Lenka Kosková-Třísková. Linux: Představení a porovnání. *kosek.* [online]. 3. 3. 2011 [cit. 2015-12-13]. Dostupné z:*http://www.kosek.cz/lenka/linux/swn_linux_1.html*

[41] Jak vytvořit bezdiskový stroj s operačním systémem GNU/Linux. *DCE.* [online]. 7. 7. 2015 [cit. 2015-12-13]. Dostupné z:*https://support.dce.felk.cvut.cz/mediawiki/index.php/Jak_vytvo%C5%99it_bezdiskov%C3%BD_stroj_s_opera%C4%8Dn%C3%ADm_syst%C3%A9mem_GNU/Linux*

[42] Michal Kočer . Kořenový systém souborů přes NFS. *ucw.* [online]. [cit. 2015-12-13]. Dostupné z:*http://www.ucw.cz/~hubicka/skolicky/novejadro/node3.html*

[43] *Virtualizace - porovnání dvou daných platforem* [online]. 2009, Duben 2009 [cit. 2015-12-13]. Dostupné z: https://is.bivs.cz/th/5251/bivs_m/DP_Bures_Michal_Virtualizace-porovnani_dvou_danych_platforem.pdf

[44] Petr Procházka. Rychlokurz: Serverová virtualizace a sítě. *Computerworld.* [online]. 5. 1. 2010 [cit. 2015-12-13]. Dostupné z:*http://computerworld.cz/technologie/rychlokurz-serverova-virtualizace-a-site-5422*

[45] Vít Petrjanoš. Virtuální desktop – když je na stole jen placka s drátem. *Computerworld.* [online]. 30. 3. 2013 [cit. 2015-12-13]. Dostupné z:*http://computerworld.cz/technologie/virtualni-desktop-kdyz-je-na-stole-jen-placka-s-dratem-49693*

[46] HTTP server project. *Apache.* [online]. 1997-2015 [cit. 2015-12-13]. Dostupné z: *https://httpd.apache.org/ABOUT_APACHE.html*

[47] Michal Dočekal. Správa linuxového serveru: Úvod do konfigurace Apache. *Linuxexpres.* [online]. 6. 1. 2011 [cit. 2015-12-13]. Dostupné z:*http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-uvod-do-konfigurace-apache*

[48] Distribuce. *Linuxexpres.* [online]. [2014] [cit. 2015-12-13]. Dostupné z: *http://www.linuxexpres.cz/distribuce*

[49] Databázový server. *eKaskáda.* [online]. [cit. 2015-12-13]. Dostupné z: *https://www.ekaskada.cz/helpsystem/html/K_PPDatabazovyServer_H.htm*

[50] Michal Polák. Začínáme s Linuxem: Výběr distribuce. *abclinuxu.* [online]. 9. 2. 2013 [cit. 2015-12-13]. Dostupné z:*http://www.abclinuxu.cz/clanky/zaciname-s-linuxem-vyber-distribuce*

[51] Michal Polák. Začínáme s Linuxem: Grafická prostředí. *abclinuxu.* [online]. 13. 3. 2013 [cit. 2015-12-13]. Dostupné z:*http://www.abclinuxu.cz/clanky/zaciname-slinuxem-graficka-prostredi#!/-1/*

[52] Michal Polák. 1.2. What is GNU/Linux?. *Debian.* [online]. [cit. 2015-12-13]. Dostupné z: *https://www.debian.org/releases/lenny/ia64/ch01s02.html.en*

[53] 1.2. What is GNU/Linux?. *O Debianu.* [online]. 19. 7. 2015 [cit. 2015-12-13]. Dostupné z: *https://www.debian.org/intro/about*

[54] *Debian GNU/Linux — instalační příručka* [online]. 2004-2015. [cit. 2015-12-13]. Dostupné z:*https://www.debian.org/releases/stable/amd64/install.pdf.cs*

[55] Hackerspace Pardubice. *Virtualizace PC* [online]. [cit. 2015-12-13]. Dostupné z: *http://hackerspacepardubice.cz/wiki/lib/exe/fetch.php?media=wiki:prezentace:virtualizace-pc.pdf*

[56] Jana Dvořiková. *Virtualizace pomocí Novell Open Enterprise Server 2* [online]. 2008. [cit. 2015-12-13]. Dostupné z:*https://www.isss.cz/archiv/2008/download/prezentace/dvorakova_novell.pdf*

[57] Karel Stýblo. *Cloud* [online]. 7. 4. 2014. [cit. 2015-12-13]. Dostupné z: *http://www.cs.vsb.cz/Files/osobni_stranky/michal-radecky/IT/2013/pr8-cloud.pdf*

[58] Pydio. *Wikipedia: the free encyclopedia.* [online]. 2001- [cit. 2015-12-13]. Dostupné z: *https://de.wikipedia.org/wiki/Pydio*

[59] Dmitri Popov. Pydio-tutorial. *pro-linux.de.* [online]. 12.5.2015 [cit. 2015-12-14]. Dostupné z: *http://www.pro-linux.de/artikel/2/1756/pydio-tutorial.html*

[60] O VPN. *Vše co jste chtěli vědět o VPN, ale báli jste se zeptat.* [online]. 2005 [cit. 2015-12-13]. Dostupné z: *http://home.zcu.cz/~ondrous/*

[61] Just what is SMB?. *samba.org.* [online]. 8. 11. 2002 [cit. 2015-12-13]. Dostupné z: *https://www.samba.org/cifs/docs/what-is-smb.html*

[62] Jak funguje FTP. *poradna.net.* [online]. 30. 9. 2006 [cit. 2015-12-13]. Dostupné z: *http://pc.poradna.net/a/view/307878-jak-funguje-ftp*

[63] Gallery of Cloud Service Models Iaas Saas Paas How Microsoft. *Travia.* [online]. [cit. 2015-12-13]. Dostupné z: *http://www.traviasuite.com/cloud-service-models-iaas-saas-paas-how-microsoft.html*

# SEZNAM POUŽITÝCH ZKRATEK

| | |
|---|---|
| AES | Advanced Encryption Standard |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name Server |
| FTP | File Transfer Protocol |
| GRUB | GNU gRand Unified Bootloader |
| GUI | Graphical User Interface |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol Security |
| IaaS | Infrastructure as a Service |
| IP | Internet Protocol |
| KVM | Kernel-based Virtual Machine |
| LAMP | Linux, Apache, MySQL, PHP (Perl) |
| MAC | Media Access Control |
| NAS | Network Attached Storrage |
| NAT | Network Adress Translation |
| NFS | Network File Systém |
| OS | Operační systém |
| PaaS | Platform as a service |
| PHP | Hypertext Preprocesor |
| RAID | Redundant Array of Inexpensive/Independent Disks |
| RDP | Remote Desktop |
| RHEL | Red Hat Enterprise Linux |
| SMART | Self-Monitoring, Analysis, and Reporting Technology |
| SMB | Server Message Block |
| SPICE | Simple Protocol for Independent Computing Environments |
| SSH | Secure SHell |
| SSL | Secure Sockets Layer |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| UDP | User Datagram Protocol |
| URL | Uniform Resource Locator |
| USB | Universal Seriál Bus |
| VPN | Virtual Private Network |

# SEZNAM OBRÁZKŮ



# SEZNAM TABULEK